Thesis

Reliable classification in digital and physical worlds under active adversaries and prior ambiguity

TARAN, Olga

Abstract

Counterfeiting and piracy are among the main problems for modern society. Many traditional anti-counterfeiting technologies become quickly obsolete in view of the rapid technological progress that offers a wide range of modern high-tech tools and applications to the counterfeiters. At the same time, many new approaches to anti-counterfeiting such as printable graphical codes appear thanks to the advancement of modern mobile technologies and machine learning algorithms. The security of printable codes in terms of their reproducibility by unauthorized parties remains largely unexplored. Thesis addresses a problem of anti-counterfeiting of physical objects and aims at investigating the authentication aspects and the resistances to illegal copying of the modern printable graphical codes from machine learning perspectives. A special attention is paid to a reliable authentication on the modern mobile phones. Also the robustness to adversarial examples in the digital world and training under the limited amount of labeled data are investigated.

Reference

TARAN, Olga. *Reliable classification in digital and physical worlds under active adversaries and prior ambiguity*. Thèse de doctorat : Univ. Genève, 2021, no. Sc. 5566

DOI : 10.13097/archive-ouverte/unige:152982
URN : urn:nbn:ch:unige-1529828

UNIVERSITÉ DE GENÈVE
Département d'Informatique

FACULTÉ DES SCIENCES
Professeur Svyatoslav Voloshynovskiy

Reliable classification in digital and physical worlds under active adversaries and prior ambiguity

THÈSE

présentée à la Faculté des sciences de l'Université de Genève
pour obtenir le grade de Docteur ès sciences, mention informatique

par

Olga TARAN

de

Kertch (Ukraine)

Thèse N° 5566

GENÈVE
Repro-Mail - Université de Genève
juin 2021

UNIVERSITÉ DE GENÈVE

FACULTÉ DES SCIENCES

DOCTORAT ÈS SCIENCES, MENTION INFORMATIQUE

Thèse de Madame Olga TARAN

intitulée :

«Reliable Classification in Digital and Physical Worlds under Active Adversaries and Prior Ambiguity»

La Faculté des sciences, sur le préavis de Monsieur S. VOLOSHYNOVSKIY, professeur ordinaire et directeur de thèse (Département d'informatique), Monsieur T. GOLLING, professeur associé (Département de physique nucléaire et corpusculaire), Monsieur P. BAS, docteur (Université de Lille CNRS, Lille, France), Madame I. TKACHENKO, docteure (Institut de la Communication LIRIS UMR CNRS, Université Lumière, Lyon 2, France), Monsieur T. HOLOTYAK, docteur (Département d'informatique, Université de Genève), autorise l'impression de la présente thèse, sans exprimer d'opinion sur les propositions qui y sont énoncées.

Genève, le 11. 06.2021

Thèse - 5566 -

Le Doyen

N.B. - La thèse doit porter la déclaration précédente et remplir les conditions énumérées dans les "Informations relatives aux thèses de doctorat à l'Université de Genève".

To my beloved parents,
to whom I owe my life and all my achievements.
To my dearest husband,
without whom the achievements of recent years would not have been possible.

Acknowledgements

The years of my PhD study were an unforgettable and invaluable life experience. It would not have been possible to write this Thesis without the help and support of the kind people around me, to only some of whom it is possible to give particular mention here.

First of all, I would like to express a deep appreciation to my supervisor Prof. Sviatoslav Voloshynovskiy for his enthusiasm for the project, for his support, encouragement and patience. His deep understanding and knowledge of the subject helped me in my study. His guidance has made these years of research an inspiring experience for me.

I would like to thank to Taras Holotyak as well for his ideas, advices and invaluable support.

It has also to be mentioned that this work has benefited from collaborations, discussions and suggestions of my colleagues, former and current, from Stochastic Information Processing group. In complete random order: Joakim Tutt, Denis Ullmann, Maurits Diephuis, Sohrab Ferdowsi, Dimche Kostadinov, Shideh Rezaeifar, Behrooz Razeghi, Roman Chaban. A special thank goes to Slavi Bonev for his support with samples and acquisition module and research collaboration.

I am thankful to our Ukrainian colleagues Oleksandr Makoveychuk and Vitaliy Kinakh for their support.

Special thanks to the jury committee for their helpful suggestions. I owe them my heartfelt appreciation.

I would like to thank Tobias Golling and his group for collaboration, interesting meetings and stimulating discussion. I am grateful as well to the Computer Vision and Multimedia Laboratory members.

My work at the University of Geneva was also aided by the members of CUI, including Daniel Agulleiro and Nicolas Mayencourt, whose assistance in administrative and technical questions cannot be overestimated.

At last, but not the least, I would like to thank our secretaries Maëlle Saintilan and Anne-Isabelle Giuntini for their help with many everyday administrative issues.

From the bottom of my heart I would like to say a big thank to my parents, my husband and all my family for their never-ending love, support and infinite patience.

Résumé

Aujourd'hui, la contrefaçon et le piratage font partie des principaux problèmes de la société moderne. La contrefaçon de médicaments, de produits alimentaires, de cosmétiques, de pièces mécaniques et de marchandises en général présente des risques considérables pour le bien-être et la santé du public, pour les entreprises et la réputation de la valeur des marques. De plus, de nombreuses technologies traditionnelles de lutte contre la contrefaçon deviennent rapidement obsolètes au vu de la rapidité des progrès technologiques qui offrent un large éventail d'outils et d'applications modernes de haute technologie aux contrefacteurs tels que les systèmes modernes d'apprentissage automatique, les imprimantes et scanners numériques industriels de haute qualité. D'un autre côté, de nombreuses nouvelles approches dans la lutte contre la contrefaçon apparaissent grâce aux progrès des technologies mobiles modernes et des algorithmes d'apprentissage automatique.

Dans les dernières années, les *codes graphiques imprimables* ont attiré beaucoup d'attention comme un lien entre le monde physique et le monde numérique, ce qui présente un grand intérêt pour l'internet des objets et les applications de protection des marques. La sécurité des codes imprimables en termes de reproduction par des tiers non autorisés ou de clonabilité reste largement inexplorée. À cet égard, cette Thèse aborde un problème de lutte contre la contrefaçon d'objets physiques et vise à étudier les aspects d'authentification et de résistances à la copie illégale des codes graphiques imprimables modernes du point de vue de l'apprentissage automatique. Une attention particulière est accordée à une authentification fiable sur les téléphones portables modernes qui offrent des possibilités sans précédent en matière d'imagerie, de calcul et de communication.

Puisque les codes graphiques imprimables sont déjà utilisés dans la pratique pour la protection de divers produits vitaux, nous ne pouvons pas étudier directement leur sécurité pour éviter tout dommage éventuel à la sécurité de ces produits et de nuire aux activités des entreprises prestataires de services. Pour ces raisons, nous considérons la sécurité des codes graphiques imprimables en tant que telle en utilisant le principe de conception générique et de détection des copies qui sous-tend les solutions commerciales. Toute similitude avec une technologie existante serait plus une coïncidence qu'un objectif ciblé. L'objectif principal est de démontrer une approche générale applicable à la majorité des codes graphiques imprimables conçus avec des principes de modulation identiques plutôt que d'étudier les aspects de clonabilité et d'authentification de certains codes graphiques imprimables particuliers.

Dans le même temps, la plupart des technologies modernes de détection des copies effectuent la détection des contrefaçons sur la base d'une authentification par apprentissage automatique qui est une sous-classe du problème général de classification multi-classes. Toutefois, les récentes découvertes ont révélé la vulnérabilité des techniques de classification basées sur des méthodes classiques telles que les machines à vecteurs de support et même les classificateurs d'apprentissage profond. Un adversaire actif peut créer diverses permutations de données, connues sous le nom d'*exemples adverses*, afin de tromper la fiabilité de la décision de classification. Ces exemples adverses peuvent être utilisés contre les systèmes de classification numériques et physiques.

Pour cette raison, dans cette Thèse, une attention particulière est accordée à la classification fiable dans les mondes numérique et physique dans les hypothèse d'un jeu de données restreint et en présence d'exemples adverses cherchant à tromper le système.

Pour combattre les exemples adverses, la Thèse propose un mécanisme de défense multi-canal basé sur une randomisation avec une clé dans un domaine de transformation spécial. La randomisation basée sur la transformation avec une clé partagée entre les étapes d'entraînement et d'inférence préserve les gradients dans les sous-espaces définis par la clé pour le défenseur. En même temps, il empêche la propagation inverse du gradient et la création de systèmes de contournement pour l'attaquant. Un avantage supplémentaire de la randomisation multi-canaux est l'agrégation qui fusionne les sorties soft de tous les canaux, augmentant ainsi la fiabilité du score final. Le partage d'une clé secrète crée un avantage informationnel pour le défenseur par rapport à l'attaquant, à l'instar des techniques cryptographiques. L'évaluation expérimentale démontre une robustesse accrue de la méthode proposée face à certaines des meilleures attaques connues aujourd'hui.

Le manque de données d'entraînement labellisées, considéré comme une *ambiguïté préalable*, joue aussi un rôle important. Le problème du manque de données d'entraînement labellisées est étudié du point de vue de la classification semi-supervisée basée sur le principe du goulot d'étranglement de l'information avec une décomposition variationnelle. Afin de comprendre plus profondément le rôle et l'impact des différents éléments du goulot d'étranglement de l'information variationnelle sur la précision de la classification, deux types de distributions cibles sur l'espace latent du classificateur sont étudiées, à savoir des distributions choisies à priori ou apprises pendant l'entraînement. La Thèse étudie comment les paramètres du cadre proposé influencent les performances du classificateur et démontre que les résultats obtenus avec ce système sont comparables aux meilleures résultats dans la littérature d'aujourd'hui et approchent même les performances d'une classification complètement supervisée.

Enfin, les méthodes proposées sont étudiées pour l'authentification de codes imprimés industriellement en utilisant des téléphones portables comme dispositif d'authentification dans le cadre d'attaques par clonabilité via des techniques de reproduction manuelles ainsi que via des techniques d'apprentissage automatique.

Abstract

Nowadays, counterfeiting and piracy are among the main problems for modern society. Counterfeiting of medication, food, cosmetics, mechanical parts and goods in general poses tremendous risks to public welfare and health, businesses and brand value reputation. Along with the fact that many traditional anti-counterfeiting technologies become quickly obsolete in view of the rapid technological progress that offers a wide range of modern high-tech tools and applications to the counterfeiters such as modern machine learning systems, high quality digital industrial printers and scanners. On the other hand, many new approaches to anti-counterfeiting appear thanks to the advancement of modern mobile technologies and machine learning algorithms.

In the recent years, the *printable graphical codes* attracted a lot of attention as a link between the physical and digital worlds, which is of great interest for the internet of things and brand protection applications. However, the security of printable codes in terms of their reproducibility by unauthorized parties or clonability remains largely unexplored. In this respect this Thesis addresses a problem of anti-counterfeiting of physical objects and aims at investigating the authentication aspects and the resistances to illegal copying of the modern printable graphical codes from machine learning perspectives. A special attention is paid to a reliable authentication on the modern mobile phones that offer unprecedented imaging, computation and communication possibilities, thus moving object authentication from specialized authorities closer to end consumers.

Since the printable graphical codes are in practical usage already for the protection of various life-critical products, we cannot consider their security directly to avoid any possible damage of the security of products and harm to the businesses of service providing companies. For these reasons, we consider the security of printable graphical codes as such using generic design and copy detection principle behind the commercial solutions. In this way, a similarity with some existing technology is rather a coincidence than a targeted objective. Thus, the main goal is to demonstrate a general approach applicable to the majority of printable graphical codes designed with identical modulation principles rather than to investigate the clonability and authentication aspects of some particular printable graphical codes.

At the same time, most of modern copy detection technologies perform the detection of fakes based on a machine learning authentication that is a sub-class of general multi-class classification problem. However, the recent findings revealed vulnerability of classification

techniques based on classical methods such as support vector machines and even deep learning classifiers. An active adversary can create various permutations to data known as *adversarial examples* targeting to trick the reliability of classification decision. These adversarial examples can be used against both digital and physical classification systems.

For this reason, in this Thesis, a particular attention is paid to the reliable classification in both digital and physical worlds under prior ambiguity that is related to both the lack of the labeled training data and the occurrence of the adversarial examples and their particular design and objectives behind.

To combat the adversarial examples this Thesis proposes a multi-channel defense mechanism based on a key-based randomization in a special transform domain. The transform based randomization with a key shared between the training and inference stages preserves the gradients in key-defined sub-spaces for the defender. At the same time, it prevents the gradient back propagation and the creation of various bypass systems for the attacker. An additional benefit of multi-channel randomization is the aggregation that fuses soft-outputs from all channels, thus increasing the reliability of the final score. The sharing of a secret key creates an information advantage to the defender over the attacker similar to cryptographic techniques. Experimental evaluation demonstrates an increased robustness of the proposed method to a number of known state-of-the-art attacks.

Not less important is a lack of labeled training data that can be also casted as a *prior ambiguity*. The problem of the lack of labeled training data is studied from the perspective of semi-supervised classification based on the *information bottleneck formulation* with a variational decomposition. To deeper understand the role and impact of different elements of variational information bottleneck on the classification accuracy, two types of priors on the latent space of classifier, namely, *hand-crafted* and *learnable priors*, are investigated. This Thesis studies how the parameters of the proposed framework influence the performance of classifier and demonstrates how the proposed framework compares to the state-of-the-art methods and approaches the performance of fully supervised classification.

Finally, the proposed methods are investigated on the authentication of industrially printed codes using mobile phones as an authentication device under hand-crafted and machine learning based clonability attacks.

Acronyms

AAE Adversarial autoencoder

AE Autoencoder

BIB-AE Bounded information bottleneck AE

CA Canon printer

CatGAN Categorical generative adversarial networks

CNN Convolutional neural networks

DCT Discrete cosine transform

DE Differential evolution optimization

DNN Deep neural networks

ELBO Evidence lower bound

HC Hand-crafted

HCP Hand-crafted priors

HP Hewlett-Packard printer

IB Information bottleneck

KDA Key-based diversified aggregation mechanism

KL-divergences Kullback–Leibler divergence

LP Learnable priors

LX Lexmark printer

ML Machine learning

MMD Maximum mean discrepancy

MSE Mean squared error

NN Neural network

QR Quick response codes

OC-SVM One class support-vector machine

OECD Organization for economic co-operation and development

PGC Printable graphical codes

SA Samsung printer

SGD Stochastic gradient descent

SS Semi-supervised classification

T-SNE T-distributed stochastic neighbor embedding

VAE Variational autoencoder

Table of contents

List of figures **xv**

List of tables **xxi**

1 Introduction **1**
1.1 Anti-counterfeit technologies . . . 2
1.1.1 Authentication technologies (classical) . . . 3
1.1.2 Track and trace technologies . . . 4
1.1.3 Physical unclonable functions (PUFs) . . . 5
1.2 Security of PGC . . . 6
1.3 Scope of thesis . . . 8
1.4 Thesis outline . . . 11
1.5 Contributions . . . 13

2 Adversarially robust classification **15**
2.1 Introduction . . . 15
2.2 Previous work: defenses and attacks . . . 18
2.2.1 Defense strategies . . . 19
2.2.2 Adversarial attacks . . . 23
2.2.2.1 C&W attack . . . 24
2.2.2.2 PGD attack . . . 25
2.2.2.3 One Pixel Attack . . . 25
2.3 Classification algorithm based on KDA . . . 25
2.4 Randomization using key-based sign flipping in the DCT domain . . . 29
2.5 Results and discussion . . . 31
2.5.1 Attacks' scenarios . . . 31
2.5.2 Empirical results and discussion . . . 33
2.5.2.1 Gray-box transferability: from a single-channel to a multi-channel . . . 33

2.5.2.2 Gray-box transferability: from a multi-channel to a multi-channel . . . 35
2.5.2.3 Black-box direct attack . . . 36
2.5.2.4 Adversarial distortions . . . 38
2.5.2.5 Key-based aggregation . . . 38
2.6 Conclusions . . . 39

3 Semi-Supervised Classification 41
3.1 Introduction . . . 42
3.2 Related work . . . 43
3.3 IB with hand-crafted priors . . . 46
3.3.1 Decomposition of the first term: hand-crafted regularization . . . 47
3.3.2 Decomposition of the second term . . . 48
3.3.3 Supervised and semi-supervised models with/without hand-crafted priors 48
3.4 IB with learnable priors . . . 49
3.4.1 Decomposition of latent space regularizer . . . 50
3.4.2 Decomposition of reconstruction space regularizer . . . 50
3.4.3 Semi-supervised models with learnable priors . . . 51
3.4.4 Links to state-of-the-art models . . . 52
3.4.4.1 Links to unsupervised models . . . 52
3.4.4.2 Links to semi-supervised models . . . 55
3.5 Experimental results . . . 59
3.5.1 Experimental setup . . . 59
3.5.2 Discussion MNIST . . . 60
3.5.3 Latent space of trained models . . . 62
3.5.4 Discussion SVHN . . . 64
3.6 Conclusions . . . 64

4 Copy detection patterns 67
4.1 Introduction . . . 68
4.2 PGC datasets . . . 71
4.2.1 DP1C and DP1E datasets . . . 73
4.2.2 Indigo mobile dataset . . . 75
4.2.3 Indigo scanner dataset . . . 78
4.3 Research questions . . . 79
4.4 Supervised classification with respect to the HC fakes . . . 84
4.4.1 Five class classification . . . 84
4.4.2 Two class classification . . . 86
4.4.3 Conclusions . . . 87

4.5 One-class classification with respect to the HC fakes . . . 88
4.5.1 Spatial domain data analysis . . . 88
4.5.2 One-class classification from the IB point of view . . . 96
4.5.2.1 $\mathcal{L}_1(\boldsymbol{\phi}_\mathrm{a}, \boldsymbol{\theta}_\mathrm{t}) = -\beta_\mathrm{t}\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}}$ . . . 100
4.5.2.2 $\mathcal{L}_2(\boldsymbol{\phi}_\mathrm{a}, \boldsymbol{\theta}_\mathrm{t}) = -\beta_\mathrm{t}\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}} + \beta_\mathrm{t}\mathcal{D}_\mathrm{t}$ . . . 102
4.5.2.3 $\mathcal{L}_3(\boldsymbol{\phi}_\mathrm{a}, \boldsymbol{\theta}_\mathrm{t}, \boldsymbol{\theta}_\mathrm{x}) = -\beta_\mathrm{t}\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}} - \beta_\mathrm{x}\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$ . . . 103
4.5.2.4 $\mathcal{L}_4(\boldsymbol{\phi}_\mathrm{a}, \boldsymbol{\theta}_\mathrm{t}, \boldsymbol{\theta}_\mathrm{x}) = -\beta_\mathrm{t}\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}} + \beta_\mathrm{t}\mathcal{D}_\mathrm{t} - \beta_\mathrm{x}\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}} + \beta_\mathrm{x}\mathcal{D}_\mathrm{x}$ . . . 104
4.5.3 Conclusions . . . 105
4.6 ML fakes authentication . . . 106
4.6.1 Details of the setup . . . 107
4.6.2 Fakes production . . . 109
4.6.3 One-class classification with respect to the ML fakes . . . 112
4.6.4 Supervised classification with respect to the ML fakes . . . 114
4.6.5 Conclusions . . . 116
4.7 Digital templates estimation . . . 116
4.7.1 Details of the setup . . . 117
4.7.2 Estimation results . . . 118
4.7.3 Conclusions . . . 121
4.8 Conclusions . . . 122

5 Conclusions and Future work **125**

References **129**

Appendix A List of publications and patent **141**

Appendix B Robust classifier based on the KDA **145**
B.1 Technical details of training . . . 145

Appendix C Semi-supervised training **147**
C.1 Supervised training without latent space regularization . . . 147
C.2 Semi-supervised training without latent space regularization and with class label regularizer . . . 149
C.3 Supervised training with hand-crafted latent space regularization . . . 153
C.4 Semi-supervised training with hand-crafted latent space and class label regularizations . . . 157
C.5 Semi-supervised training with learnable latent space regularization . . . 161
C.6 Semi-supervised training with learnable latent space regularization and adversarial reconstruction . . . 164

Appendix D Copy detection patterns **169**
D.1 Supervised classification with respect to the HC fakes 169
D.1.1 Technical details of training 169
D.1.2 Five class supervised classification 171
D.1.3 Two class supervised classification 173
D.2 One class classification with respect to the HC fakes 175
D.2.1 OC-SVM in the spatial domain 175
D.2.2 One class classification from the IB point of view 177
D.2.2.1 Mutual information decomposition tricks 177
D.2.2.2 Technical details of training 178
D.2.2.3 The PGC authentication in the first scenario 181
D.2.2.4 The PGC authentication in the second scenario 183
D.2.2.5 The PGC authentication in the third scenario 185
D.2.2.6 The PGC authentication in the fourth scenario 187
D.3 ML fakes authentication 189
D.3.1 ML fakes production: technical details of training 189
D.3.2 One class classification with respect to the ML fakes 192
D.3.3 Supervised classification with respect to the ML fakes 194
D.4 Digital templates estimation 195

List of figures

1.1 Examples of anti-counterfeit technologies. 3

1.2 The general scheme of PGC life cycle. The defender-verifier pair is playing against active adversary, i.e., attacker, trying to trick the classifier decision of verifier. $\mathrm{P_{d/a}}$ denotes the printing by a defender/attacker. $\mathrm{A_{a/v}}$ means an acquisition by a attacker/verifier. $\mathbf{t}$ corresponds to the original digital template. $\mathbf{x}_s$ is a corresponding printed physical original code, while $\mathbf{x}$ is a digitized original code. $\mathbf{f}_s$ and $\mathbf{f}$ denotes the printed physical and corresponding digitized fake codes respectively. $\mathbf{y}_s$ stands for a physical code from the public domain that might be either original $\mathbf{x}_s$ or fake $\mathbf{f}_s$, while $\mathbf{y}$ denotes the corresponding digitized codes. The classifier might produce either hard decision 0/1 (fake/original) or soft one ranging from 0 to 1. 7

1.3 Scope of Thesis addressing adversarial examples in digital world and fakes in physical world. The classifiers in both cases should be robust to adversarial examples and fakes. 8

2.1 The information access diagram: the defender has an access to the training data and secret shared between the training and test stages while the attacker has only access to the shared training dataset and can observe the output decision of the classifier. 16

2.2 Classifier training: a traditional classifier has an access to training data samples $\{\mathbf{x}_i, c_i\}_{i=1}^{M}$ generated from $P_{\mathcal{D}}(\mathbf{x})$. The classifier learns a set of parameters $\boldsymbol{\theta}$ to output a decision $\hat{c} \in \{1, ..., M_c\}$ or to reject an input ($\varnothing$). 17

2.3 Classifier's decision boundaries: (a) without rejection and (b) with rejection. Note the difference in the decision regions of trained classifiers. 19

2.4 The attacker-defender game in adversarial classification: (a) the attacker produces an adversarial example $\mathbf{x}^{adv}$ from a host $\mathbf{x}$ by a mapper $\mathbf{x}^{adv} = g_{\boldsymbol{\alpha}}(\mathbf{x}, \boldsymbol{\epsilon})$; (b) the defender answers by the pre-filtering $\varphi_{\boldsymbol{\beta}}(\mathbf{x}^{adv})$ to obtain an estimation $\hat{\mathbf{x}}$ on the original host class manifold; (c) an alternative defense strategy by a randomization of input adversarial image as $\hat{\mathbf{x}} = \mathbf{x}^{adv} + \boldsymbol{\epsilon}^{d'}$, the resulting sample will be outside attacker's target class with a small probability that the resulting sample will be in the original host class that requires the classifiers retraining; (d) the proposed defense strategy consists of pre-filtering by $\varphi_{\boldsymbol{\beta}}(\mathbf{x}^{adv})$ and addition of defender's randomized perturbation $\boldsymbol{\epsilon}^{d}$: $\tilde{\mathbf{x}} = \varphi_{\boldsymbol{\beta}}(\mathbf{x}^{adv}) + \boldsymbol{\epsilon}^{d}$, such that $\|\boldsymbol{\epsilon}^{d}\|_2^2 \ll \|\boldsymbol{\epsilon}^{d'}\|_2^2$. 20

2.5 Explanation of multi-channel randomization: given a training dataset, the defender introduces a random perturbation $\boldsymbol{\epsilon}^{d_l}$, $1 \leq l \leq L$, to each sample $\{\mathbf{x}_i\}_{i=1}^{M}$ and trains L classifiers. Since the perturbation is known at training, i.e., all samples obtain the stationary "bias" by $\boldsymbol{\epsilon}^{d_l}$ for the same classifier l, the randomization has a limited impact on the classifier performance. Since the attacker has no access to the defender's perturbations and all of them are equilikely, an equivalent manifold for the attacker is expanded thus leading to higher entropy and thus increasing the learning complexity. 21

2.6 Generalized diagram of the proposed multi-channel system with the KDA. . . 26

2.7 Randomized transformation $\mathbb{P}_{ji}$, $1 \leq j \leq J$, $1 \leq i \leq I$ examples: (a) randomized sampling, (b) randomized permutation, (c) randomized sign flipping in the sub-block defined in orange. All transforms are key-based. 27

2.8 Local key based sign flipping in the DCT sub-bands: (a) sub-bands, (b) original image (c) image with a sign flipping in V sub-band, (d) image with a sign flipping in H sub-band and (e) image with a sign flipping in D sub-band. . . . 29

2.9 Multi-channel classification with the local DCT sign flipping. 30

2.10 Examples of original images from each class from MNIST (top line) and Fashion-MNIST (middle line) and CIFAR-10 (bottom line) datasets. 31

2.11 Adversarial examples: (left) original image $\mathbf{x}$, (middle) adversarial example $\mathbf{x}^{adv}$, (right) absolute value of the adversarial perturbation $\boldsymbol{\epsilon}$ computed as $|\boldsymbol{\epsilon}| = |\mathbf{x} - \mathbf{x}^{advv}|$. 32

3.1 Classification with the hand-crafted latent space regularization. 44

3.2 Classification with the learnable latent space regularization. 51

3.3 Latent space $\mathbf{a}$ (of size 1024) of classifier. 63

3.4 Latent space $\mathbf{z}$ (of size 20) of auto-encoder. 64

4.1 ScanTrust secure QR code[1]. 68

4.2 Examples of the dot gain effect: (a) - (b) a black symbol surrounded by white symbols increases its size but remains well detectable; (c) - (d) a white symbol surrounded by black symbols might disappear under strong dot gain. 69
4.3 Examples of different types of PGC. 70
4.4 An example of a two-layer QR code that consists of a top and a bottom layers [1]. 71
4.5 Examples of digital templates used in DP1C and DP1E datasets. 73
4.6 The DP1C and DP1E datasets: the first raw corresponds to the scans produced by Cannon scanner and the second raw is produced by Epson scanner for all considered printers. One can note a considerable variability among different printers, while the scans produced by two different scanners visually look to be quite correlated ones. 74
4.7 The scans of empty substrate (paper) by the Cannon and Epson scanners with the same setting of parameters as for Fig. 4.6. 75
4.8 Examples of (a) a binary digital template used for printing and (b) the printed original code from the Indigo mobile dataset. 76
4.9 The schematic representation of the mobile phone acquisition setup. 76
4.10 Examples of original and fake codes with symbol size 5×5 taken by a mobile phone from the Indigo mobile dataset. 77
4.11 Examples of three types of codes with different symbol size from the Indigo scanner dataset. All codes are scanned at 1200 ppi. 78
4.12 The overview of investigated authentication and clonability aspects of the PGC. The authentication task is studied from the side of defender with respect to the HC and ML copy fakes. The clonability of PGC is investigated from the side of attacker with respect to the HC and ML approaches. 79
4.13 T-SNE of the latent space of the classification model trained on originals and all type of fakes. 85
4.14 The latent space T-SNE visualization of the supervised two class classifier trained on the originals and one type of fakes. 87
4.15 The 2D T-SNE visualisation of the original and fake codes in the spatial domain (a horizontal axis denotes T-SNE dimension 1 and the T-SNE dimension 2 is on the vertical axis): (a) presents the direct RGB images' visualisation; (b) is based on the xor difference between the corresponding digital templates and printed codes binarized via simple threshold method with an optimal threshold determited individually for each printed code via Otsu's method [2]; (c) visualizes the differences between the physical references and the corresponding printed original and fake codes. 89

4.16 The distribution of the Hamming distances between the references (digital or physical) and the corresponding printed codes (original and fakes). In case of the physical references the binarization is applied as a simple thresholding with an optimal threshold determined individually for each code via Otsu's method. 90
4.17 The distribution of the ℓ_1 and ℓ_2 distances between the references (digital or physical) and the corresponding printed codes (original and fakes). 91
4.18 The 2D PGC separability in the spatial domain with respect to the ℓ_2, ℓ_1 and Hamming distances between the references (digital or physical) and the corresponding printed codes (original and fakes) (part 1). 92
4.19 The 2D PGC separability in spatial domain with respect to the ℓ_2, ℓ_1, Hamming distances and Pearson correlation between the references (digital or physical) and the corresponding printed codes (original and fakes) (part 2). 93
4.20 The decision boundaries of OC-SVM trained with respect to the Pearson correlation and Hamming distance between the reference (digital or physical) and the corresponding original printed code. The visualisation is given for the samples from the Indigo mobile test sub-set. 95
4.21 Auto-encoding model based on the IB principle: the input $\mathbf{x}$ is "compressed" to $\mathbf{z}/\mathbf{a}$ via the parametrized mapping $q_{\phi}(\mathbf{z}|\mathbf{x})/q_{\phi_{\mathrm{a}}}(\mathbf{a}|\mathbf{x})$ leading to a bottleneck representation $\mathbf{z}/\mathbf{a}$. 96
4.22 The feature extraction for the one-class classification based on the estimation of the reference templates via $\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}}$ and $\mathcal{D}_{\mathrm{t}}$ and the printed codes via $\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$ and $\mathcal{D}_{\mathrm{x}}$ terms. 99
4.23 The one-class classification training procedure: the encoder and decoder parts of the auto-encoder model shown in Fig. 4.22 are pre-trained and fixed (as indicated by a "*"); the OC-SVM is trained on the outputs of $\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}}$ and $\mathcal{D}_{\mathrm{t}}$ terms that are the results of $I^{L}_{\boldsymbol{\theta}_{\mathrm{t}},\boldsymbol{\phi}_{\mathrm{a}}}(\mathbf{A};\mathbf{T})$ decomposition and the $\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$ and $\mathcal{D}_{\mathrm{x}}$ terms that are the results of $I^{L}_{\boldsymbol{\theta}_{\mathrm{t}},\boldsymbol{\theta}_{\mathrm{x}}}(\mathbf{T};\mathbf{X})$ decomposition. 100
4.24 The first scenario results visualization: the distribution of symbol-wise Hamming distance between the original digital templates $\mathbf{t}$ and the corresponding estimations $\hat{\mathbf{t}}$ obtained via the encoder model trained with respect to the $\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}}$ term. 101
4.25 The second scenario results visualisation: the 2D distribution of *(i)* the symbol-wise Hamming distance between the original digital templates $\mathbf{t}$ and the corresponding estimations $\hat{\mathbf{t}}$ obtained via the encoder model trained with respect to the $\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}}$ term and *(ii)* the corresponding responses of the discriminator model trained with respect to the $\mathcal{D}_{\mathrm{t}}$ term. 102

4.26 The third scenario results visualization: (a) the distribution of *(i)* the symbol-wise Hamming distance between the digital templates and its corresponding estimations via the encoder model trained with respect to the $\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}}$ term and *(ii)* the ℓ_2 distance between the printed codes and its corresponding reconstructions by the decoder model trained with respect to the $\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$ term; (b) the OC-SVM decision boundaries. 103
4.27 The fourth scenario results visualization: (a) the distribution of *(i)* the symbol-wise Hamming distance between the digital templates and its corresponding estimations via the encoder model trained with respect to the $\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}}$ term and *(ii)* the ℓ_2 distance between the printed codes and its corresponding reconstructions by the decoder model trained with respect to the $\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$ term; (b) the OC-SVM decision boundaries. 104
4.28 Hand-crafted digital templates estimation methods. 107
4.29 The general scheme of ML estimation of digital templates from the printed counterparts based on the IB principle. 108
4.30 Examples of the produced ML fakes in the DP1C and DP1E datasets. 112
4.31 The examples of the OC-SVM decision boundaries for the different printers in the DP1C dataset. All fakes are produced based on the ML-estimates. 114
4.32 The DP1C dataset examples of the printing irregularity and deviations between the original and fake codes for the CA and HP printers. 116

C.1 Baseline classifier based on $\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}}$ term. The blue shadowed regions are not used. 148
C.2 Semi-supervised classifier based on the cross-entropy term $\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}}$ and categorical class discriminator $\mathcal{D}_{\mathrm{c}}$. No latent space regularization is applied. The blue shadowed regions are not used. 149
C.3 Supervised classifier based on the cross-entropy term $\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}}$ and hand-crafted latent space regularization $\mathcal{D}_{\mathrm{a}}$. The blue shadowed parts are not used. 153
C.4 Semi-supervised classifier based on the cross-entropy term $\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}}$ and hand-crafted latent space regularization $\mathcal{D}_{\mathrm{a}}$. The blue shadowed parts are not used. 157
C.5 Semi-supervised classifier with learnable priors: the cross-entropy $\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}}$, MSE $\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$, class label $\mathcal{D}_{\mathrm{c}}$ and latent space regularization $\mathcal{D}_{\mathrm{a}}$. The blue shadowed parts are not used. 161
C.6 Semi-supervised classifier with learnable priors: the cross-entropy $\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}}$, MSE $\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$, adversarial reconstruction $\mathcal{D}_{\mathrm{x}}$, class label $\mathcal{D}_{\mathrm{c}}$ and latent space regularizer $\mathcal{D}_{\mathrm{z}}$. The blue shadowed parts are not used. 164

D.1 The OC-SVM classification error in % on the test sub-set in the spatial domain with respect to the Pearson correlation and Hamming distance between the printed codes and the corresponding references (digital or physical). 176

D.2 The general scheme of the system used in the first scenario and trained with respect the $\mathcal{D}_{t\hat{t}}$ term. The gray shadowed regions are not used. 181
D.3 The general scheme of the system used in the second scenario and trained with respect to the $\mathcal{D}_{t\hat{t}}$ and $\mathcal{D}_{t}$ terms. The gray shadowed regions are not used. . . 183
D.4 The general scheme of the system used in the third scenario and trained with respect to the $\mathcal{D}_{t\hat{t}}$ and $\mathcal{D}_{x\hat{x}}$ terms. The gray shadowed regions are not used. . 185
D.5 The general scheme of the system used in the fourth scenario and trained with respect to the $\mathcal{D}_{t\hat{t}}$, $\mathcal{D}_{t}$, $\mathcal{D}_{x\hat{x}}$ and $\mathcal{D}_{x}$ terms. The gray shadowed regions are not used. 187
D.6 The examples of the OC-SVM decision boundaries for the different printers in the DP1E dataset. 192

List of tables

2.1 Classification error (%) on the first 1000 test samples for the *gray-box* C&W transferability attacks from a single-channel model to a multi-channel model. 33

2.2 Classification error (%) on the first 1000 test samples (CIFAR-10) for the *gray-box* PGD transferability attacks from a single-channel model to a multi-channel model with randomly selected channels (the average results over 10 runs). 34

2.3 Classification error (%) on the first 1000 test samples (CIFAR-10) for the *gray-box* OnePixel transferability attacks from a multi-channel model to a multi-channel model under different keys. 35

2.4 Classification error (%) on the first 1000 CIFAR-10 test samples for the direct *black-box* OnePixel attacks. 36

2.5 Adversarial distortion. 37

2.6 Classification error (%) on the first 1000 test samples for the *gray-box* C&W transferability attacks from a single-channel model to a multi-channel model with randomly selected channels (the average results over 10 runs). 38

2.7 Classification error (%) on the first 1000 test samples (CIFAR-10) for the multi-channel system against the direct *gray-box* OnePixel attacks with randomly selected channels (the average results over 10 runs). 39

3.1 Semi-supervised classification error (%) for the optimal parameters (Appendix C.1 - C.6) defined on the MNIST dataset (D - deterministic; S - stochastic). . 60

3.2 Execution time (hours) per 100 epochs on one NVIDIA GPU. For the SVHN the models with the learnable latent space priors are trained with a learning rate 1e-4 that explains the longer time but without optimization of Lagrangians, i.e., the Lagrangians are re-used from pre-trained MNIST model. All the others models are trained with a learning rate 1e-3. 62

4.1 An overview of the datasets of PGC: *(i)* publicly available state-of-the-art and *(ii)* created and investigated in the current Thesis. The "original" and "fakes" correspond to the printed and enrolled on the corresponding equipment codes. 72

4.2 The summary of research questions with respect to the hand-crafted (HC) fakes. 82
4.3 The summary of research questions with respect to the machine learning (ML) fakes. 83
4.4 The average (over five runs) classification error in % on the test sub-set of the classification model trained in a supervised way on the originals and all type of fakes[2]. 85
4.5 The average (over five runs) classification error in % on the test sub-set of the classification model trained in a supervised way on the originals and only one type of fakes[3]. 86
4.6 The average (over five runs) OC-SVM classification error in % on the test sub-set. 94
4.7 The average (over five runs) authentication error in % on the test sub-set. . . 105
4.8 Execution time (hours) per 100 epochs on one NVIDIA GPU with a learning rate 1e-4 for the considered scenarios. 106
4.9 The average (over three runs) estimation error in % based on the symbol-wise Hamming distance between the original digital templates and the corresponding estimations in the DP1C test sub-set. As indicated in the first column, each model is trained only on the codes printed on one corresponding printer. At the test stage each model is tested on the codes printed on different printers as shown by the corresponding columns. 110
4.10 The average (over three runs) estimation error in % based on the symbol-wise Hamming distance between the original digital templates and the corresponding estimations in the DP1E test sub-set. As indicated in the first column, each model is trained only on the codes printed on one corresponding printer. At the test stage each model is tested on the codes printed on different printers as shown by the corresponding columns. 111
4.11 The average (over five runs) OC-SVM classification error in % on the test sub-set with respect to the ML fakes based on the *ConvBN* model. 113
4.12 The average (over five runs) supervised classification error in % with respect to the ML fakes based on the *ConvBN* model. 115
4.13 The estimation error in % based on the symbol-wise Hamming distance between the original digital templates with the symbol's size 5×5 and the corresponding estimations. For the ML approaches the average results over three runs are given. 119
4.14 The estimation error in % based on the symbol-wise Hamming distance between the original digital templates with the symbol's size 4×4 and the corresponding estimations. For the ML approaches the average results over three runs are given. 120

4.15 The estimation error in % based on the symbol-wise Hamming distance between the original digital templates with the symbol's size 3×3 and the corresponding estimations. For the ML approaches the average results over three runs are given. 121

C.1 The network parameters of baseline classifier trained on $\mathcal{D}_{c\hat{c}}$. The encoder is trained with and without batch normalization (BN) after Conv2D layers. . . 147
C.2 The network parameters of semi-supervised classifier trained on $\mathcal{D}_{c\hat{c}}$ and $\mathcal{D}_c$. The encoder is trained with and without batch normalization (BN) after Conv2D layers. 150
C.3 The performance (percentage error) of **deterministic** classifier based on $\mathcal{D}_{c\hat{c}} + \alpha_c \mathcal{D}_c$ for the encoder with and without batch normalization as a function of Lagrangian multiplier α_c and the number of labeled examples. 151
C.4 The performance (percentage error) of **stochastic** classifier with supervised noisy data (noise std = 0.1, # noise realization = 3) based on $\mathcal{D}_{c\hat{c}} + \alpha_c \mathcal{D}_c$ for the encoder with and without batch normalization as a function of Lagrangian multiplier α_c and the number of labeled examples. 152
C.5 The network parameters of supervised classifier trained on $\mathcal{D}_{c\hat{c}}$ and $\mathcal{D}_a$. The encoder is trained with and without batch normalization (BN) after Conv2D layers. $\mathcal{D}_a$ is trained in the adversarial way. 154
C.6 . The performance (percentage error) of **deterministic** classifier based on $\mathcal{D}_{c\hat{c}} + \alpha_a \mathcal{D}_a$ for the encoder with and without batch normalization as a function of Lagrangian multiplier. 155
C.7 The performance (percentage error) of **stochastic** classifier with supervised noisy data (noise std = 0.1, # noise realization = 3) based on $\mathcal{D}_{c\hat{c}} + \alpha_a \mathcal{D}_a$ for the encoder with and without batch normalization as a function of Lagrangian multiplier. 156
C.8 The network parameters of semi-supervised classifier trained on $\mathcal{D}_{c\hat{c}}$, $\mathcal{D}_a$ and $\mathcal{D}_c$. The encoder is trained with and without batch normalization (BN) after Conv2D layers. $\mathcal{D}_a$ and $\mathcal{D}_c$ are trained in the adversarial way. 158
C.9 The performance (percentage error) of **deterministic** classifier based on $\mathcal{D}_{c\hat{c}} + \alpha_a \mathcal{D}_a + \alpha_c \mathcal{D}_c$ for the encoder with and without batch normalization. . . 159
C.10 The performance (percentage error) of **stochastic** classifier with supervised noisy data (noise std = 0.1, # noise realization = 3) based on $\mathcal{D}_{c\hat{c}} + \alpha_a \mathcal{D}_a + \alpha_c \mathcal{D}_c$ for the encoder with and without batch normalization. 160
C.11 The encoder and decoder of semi-supervised classifier trained based on $\mathcal{D}_{c\hat{c}}$, $\mathcal{D}_c$ and $\mathcal{D}_z$. The encoder is trained with and without batch normalization (BN) after Conv2D layers. $\mathcal{D}_c$ and $\mathcal{D}_z$ are trained in the adversarial way. 162

C.12 The performance (percentage error) of **deterministic** classifier based on $\mathcal{D}_{c\hat{c}} + \mathcal{D}_c + \mathcal{D}_z + \mathcal{D}_{x\hat{x}}$ for the encoder with and without batch normalization. . 162
C.13 The performance (percentage error) of **stochastic** classifier with supervised noisy data (noise std = 0.1, # noise realization = 3) based on $\mathcal{D}_{c\hat{c}}+\mathcal{D}_c+\mathcal{D}_z+\mathcal{D}_{x\hat{x}}$ for the encoder with and without batch normalization. 163
C.14 The network parameters of semi-supervised classifier trained based on $\mathcal{D}_{c\hat{c}}$, $\mathcal{D}_c$ and $\mathcal{D}_z$. The encoder is trained with and without batch normalization (BN) after Conv2D layers. $\mathcal{D}_c$ and $\mathcal{D}_z$ are trained in the adversarial way. 165
C.15 The performance (percentage error) of **deterministic** classifier based on $\mathcal{D}_{c\hat{c}}+\mathcal{D}_c+\mathcal{D}_z+\mathcal{D}_{x\hat{x}}+\alpha_x\mathcal{D}_x$ for the encoder with and without batch normalization.166
C.16 The performance (percentage error) of **stochastic** classifier with supervised noisy data (noise std = 0.1, # noise realization = 3) based on $\mathcal{D}_{c\hat{c}} + \mathcal{D}_c + \mathcal{D}_z + \mathcal{D}_{x\hat{x}} + \alpha_x\mathcal{D}_x$ for the encoder with and without batch normalization. 167

D.1 The architecture of the model used for the supervised classification and trained with respect to the cross-entropy term $\mathcal{D}_{c\hat{c}}$, where c equals to 5 for the five class classification scenario and to 2 for the two class classification case. . . . 170
D.2 Two classes (original, fake) classification error in % on the test sub-set for a model trained in a supervised way on the printed original codes and four types of fakes. 171
D.3 Three classes (original, fake #1, fake #2) classification error in % on the test sub-set for a model trained in a supervised way on the printed original codes and four types of fake. 171
D.4 Five classes (original, fake # 1 white, fake # 1 gray, fake #2 white and fake # 2 gray) classification error in % on the test sub-set for a model trained in a supervised way on the printed original codes and four types of fakes. 172
D.5 Two classes (original, fake) classification error in % on the test sub-set for a model trained in a supervised way on the printed originals and fakes #1 white. The test is performed for all type of fakes. 173
D.6 Two classes (original, fake) classification error in % on the test sub-set for a model trained in a supervised way on the printed originals and fakes #1 gray. The test is performed for all type of fakes. 173
D.7 Two classes (original, fake) classification error in % on the test sub-set for a model trained in a supervised way on the printed originals and fakes #2 white. The test is performed for all type of fakes. 174
D.8 Two classes (original, fake) classification error in % on the test sub-set for a model trained in a supervised way on the printed originals and fakes #1 white. The test is performed for all type of fakes. 174

D.9 The architecture of the estimation and reconstruction models, where the number of input and output channels c equals to 1 in the case of the $\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}}$ and to 3 in the case of the $\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$. 179
D.10 The architecture of the discriminator based on the $\mathcal{D}_{\mathrm{t}}$ and $\mathcal{D}_{\mathrm{x}}$ terms, where the number of the input channels c equals to 1 in case of $\mathcal{D}_{\mathrm{t}}$ and to 3 in case of $\mathcal{D}_{\mathrm{x}}$. 180
D.11 The one class classification error in % on the test sub-set with respect to the first scenario optimization problem. 182
D.12 The one class classification error in % on the test sub-set with respect to the second scenario optimization problem. 184
D.13 The one class classification error in % on the test sub-set with respect to the third scenario optimization problem. 186
D.14 The one class classification error in % on the test sub-set with respect to the fourth scenario optimization problem. 188
D.15 The architecture of the *LinearBN* estimation model. 190
D.16 The architecture of the *ConvBN* estimation model. 190
D.17 The estimation error in % on the test sub-sets with respect to the symbol wise Hamming distance between the original digital templates and the corresponding estimations. 191
D.18 The OC-SVM classification error in % on the test sub-sets of DP1C and DP1E datasets with respect to the ML attacks based on the *ConvBN* model. 193
D.19 The supervised classification error in % on the DP1C and DP1E test sub-sets with respect to the ML attacks based on the *ConvBN* model. 194
D.20 The architecture of the model used for the digital templates estimation from the printed counterparts with the symbol size 5×5 and 4×4. 196
D.21 The architecture of the model used for the digital templates estimation from the printed counterparts with the symbol size 3×3. 197
D.22 The estimation error in % based on the symbol wise Hamming distance between the original digital templates with the symbol's size 5×5 and the corresponding estimations. 198
D.23 The estimation error in % based on the symbol wise Hamming distance between the original digital templates with the symbol's size 4×4 and the corresponding estimations. 199
D.24 The estimation error in % based on the symbol wise Hamming distance between the original digital templates with the symbol's size 3×3 and the corresponding estimations. 200

Chapter 1

Introduction

Counterfeiting and piracy are among the main negative factors affecting the modern economy and security. Counterfeiting is not a new phenomenon. While the counterfeiting of banknotes and valuable documents was the primary focus of counterfeiters for centuries, nowadays, the counterfeiting expands to encompass almost all production sectors. There are evidences that counterfeited products can be found everywhere from banknotes and forged documents such as IDs, diplomas, certificates, etc., food, medication to luxury and art objects. The presence of counterfeited products has a number of consequences for both legitimate brands and their consumers. While brands loose revenue and customer trust, consumers are simultaneously finding themselves out of pocket and saddled with sub-quality goods. Beyond this, there are also far wider-reaching consequences, such as health and safety issues, especially as a consequence of buying medical products, life-care and babies' products, cosmetics and electronics that are fake. This is due to the use of sub-par materials and the lack of observed safety standards during manufacturing. The pharmaceutical industry is increasingly harmed by counterfeited drugs that pose a serious threat to patient safety. The COVID-19 crisis pointed out the dangers posed by the global trade of counterfeited masks and vaccines.

In 2019, the Organisation for Economic Co-operation and Development(OECD) reported that in 2016 the volume of international trade in counterfeited and pirated products reached up to 3.3 % of world trade (about 509 billion USD) [3]. In 2013, the OECD estimated that up to 2.5 % of world trade was in counterfeit and pirated goods (about 461 billion USD). A study of the situation in the European Union (EU) shows that in 2016 the import of counterfeited and pirated products into the EU was about 121 billion EUR (134 billion USD), which represents up to 6.8 % of EU imports, against 5 % of EU imports in 2013 [3]. These results clearly show that the global counterfeiting and piracy progress rapidly, thus representing considerable and unprecedented risks and damages at all levels of states, companies and ordinary people.

Without loss of generality, it is possible to determine two main categories of fakes: *(i) forgeries* and *(ii) counterfeits*. Forgery (or tampering) is a process of manipulating

or altering of an original object or document. Counterfeiting is a process of making or creating an unauthorized imitation of a genuine article. The combating of counterfeits by an authentication of physical objects is the main research problem of the current work. Although, the anti-forgeries technologies are out of scope of this work, the considered approaches might be applied to some extent to the forgeries too.

Historically, physical objects protection against counterfeiting is based on physical features being difficult to duplicate, copy or clone. From this perspective, the authentication, as an integral part of the anti-counterfeiting process, is considered as a process of authenticity verification of overt, covert or forensic features present or added to the object that allow to verify it as genuine. In this work, we mainly focus on the features extracted from images captured from physical objects using cameras of mobile phones.

At the same time, the authentication represents a binary classification problem, which in turn belongs to a class of more general multi-class classification technologies. Due to the advancement of deep learning technologies, the modern methods of classification are mainly based on deep learning architectures. The main progress in deep learning is reported in image processing and computer vision applications that deal with digital data. However, despite very impressive results these technologies demonstrate high vulnerability in face of meticulously tampered samples known as adversarial examples that trick the decision of classifier. Therefore, the robustness of classifiers in the authentication of physical objects yet withstanding the adversarial examples in physical and digital forms represents a great theoretical interest and is of central practical importance.

That is why in this work, we consider both security features that can be used for the robust and reliable authentication based on corresponding classifiers.

1.1 Anti-counterfeit technologies

Rapid technological advancement offers the counterfeiters a wide range of tools and algorithms such as high quality printers, scanners, image processing and modern machine learning, etc. At the same time, the technological innovations are used not only by the counterfeiters. The arsenal of modern anti-counterfeiting means is also broad and includes technologies such as RFID [4], NFC tags [5], holograms [6, 7], special printing materials and techniques [8, 9], material signatures [10, 11], etc., that all together can be used to protect and authenticate genuine products [12, 13]. However, some of these technologies are based on propriatory principles and might be even obsolete.

In our study, we mainly focus on the protection and authentication of packaging as an integral part of object manufacturing, shipment, distribution and consumption.

Without loss of generality, it is possible to split the existing anti-counterfeit technologies into three main categories: *(i)* authentication technologies (classical), *(ii)* track and trace technologies and *(iii)* physical unclonable functions. In each category, there exists a large

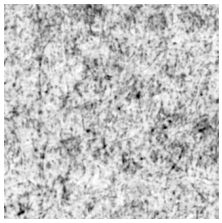
(a) Security Inks and Dyes.

(b) Holograms.

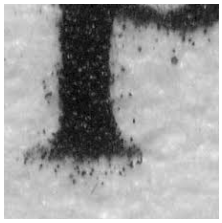
(c) Labels.

(d) RFID and chip based solutions.

(e) Serialization: 2D bar codes.



(f) Paper natural randomness.



(g) Printing process randomness [14].

(h) Copy detection patterns and 2D bas code[1].

Fig. 1.1 Examples of anti-counterfeit technologies.

variety of technologies. Not pretending to be exhaustive, here are some examples of the most well-known and widely used technologies for each category.

1.1.1 Authentication technologies (classical)

- *Security Inks and Dyes*

  Security inks and dyes are widely used in the authentication of banknotes and valuable documents. The ink based security features are popular in consumer packaging as well. Such technologies are most suitable for the packaging, where an artwork contains a lot of graphical elements. The authentication is performed under a special illumination. Secure properties of ink or dye are revealed as shown in Fig. 1.1(a). The pros and cons of security inks and dyes are:

  – **Pros:** difficult and yet expensive for copying; un-removable from an object;

[1]https://www.ledgerinsights.com/scantrust-anti-counterfeit-blockchain

  - **Cons:** considerably increase the products' cost; require user education; might require special manufacturing pipelines; special light and equipment for validation.

- *Holograms*

  Holograms create an illusion of a three-dimensional image or a "dynamic" image. When observed under different angles, a hologram might reveal different images. An example of a hologram is given in Fig. 1.1(b). Holograms can combine several-layer security features. During the last years, the technology of the hologram production became more available for the counterfeiters too. The authentication procedure is generally based on the end customer perception whether a hologram is on a product and if yes, what is the quality of this hologram. Being intuitively simple, such a procedure assumes that the verifier should be well informed about the security features of a hologram under the inspection. Given a huge number of objects and holograms, it is difficult and unfeasible to train all end customer about these features. At the same time, an automatic authentication of holograms is not a trivial task and we are not aware about any industrial solution for the automatic hologram verification on mobile phones. Finally, the cost of holograms for mass markets is still high. Thus, in summary:

  - **Pros:** difficult to copy; do not require special equipment for validation;
  - **Cons:** become relatively easy to clone for counterfeiters, require user's awareness; increase the product cost and require additional manufacturing modification; no automatic authentication.

- *Labels*

  Labels are typically used in addition to other security technologies. They can be incorporated into a packaging design or as a standalone label applied to a finished product. An example of label is illustrated in Fig. 1.1(c). In summary:

  - **Pros:** easy to manufacture and deploy; relatively cheap;
  - **Cons:** can be removed and counterfeited; change manufacturing process.

1.1.2 Track and trace technologies

- *RFID* or *Chip Based Solutions*

  RFID might be considered as the next level of evolution in anti-counterfeiting technologies allowing machine readable authentication. An example of a RFID chip is shown in Fig. 1.1(d). Potentially, the chip based solutions allow to verify a location in real time, to communicate with other containers in near proximity, to report on potential improper storage or an attempt of product damage or opening, etc. Software advances are making these solutions more attainable. RFID also allow contactless reading.

  - **Pros:** contactless reading; scanning is done by fixed readers and without human involvement, which minimizes the possibility of error; smart technologies might provide other tracking benefits;
  - **Cons:** require heavy infrastructure; do not work when goods are in transit, especially when there are multiple modes of transportation involved; in an open access, may operate in unknown or untrusted environments; exposed to attacks; expensive for massive usage.

- *Serialization*

  The use of 1D and 2D codes allows to encode and store a considerable amount of information directly on a physical object. An example of a 2D bar code is shown in Fig. 1.1(e). The modern mobile apps allow consumers or brand owners to scan and authenticate codes quickly and reliably. The serialized products can be linked to corresponding records in service providing datasets. More recently, the serialized products are also registered to blockchains. Altogether, it creates an option of object tracking and tracing from the moment of its manufacturing till the final consumption and even recycling. Being, at the same time, a very attractive solution, the serialization based on printed or engraved 1D or 2D codes is known to be vulnerable to a simple copying or even regeneration due to high robustness of 1D and 2D codes. Therefore, in summary:

  - **Pros:** carry out a lot of information about a product in a compact form; easily authenticated by manufacturer and consumer; might be integrated into blockchain solutions; cheap; can be reproduced at the packaging production time;
  - **Cons:** easy clonable even by non-experienced counterfeiters; sometimes an artwork design does not allow to use codes for various aesthetic or standardization reasons.

1.1.3 Physical unclonable functions (PUFs)

- *Natural randomness*

  The natural randomness of microstructure of physical objects or substrate is considered as a unique and non-clonable signature of substrate [15–17]. An example of paper microstructure image is shown in Fig. 1.1(f). Natural randomness is gaining more popularity thanks to the increase of imaging capacity of modern mobile phones. The pros and cons of natural randomness are:

  - **Pros:** unclonable; cheap in enrollment; does not require manufacturing pipeline change; does not require modification of physical objects;
  - **Cons:** the verification process is quite sensitive to the illumination; additional synchronization features might be needed.

- *Printing process randomness*

  The main idea behind printing process randomness is based on the absence of a mathematical model describing the interaction between the printing inks and substrate. In this respect, unique properties of printed artworks result from stochasticity of printing process [14, 18, 19] as shown in Fig. 1.1(g). The main features of printing randomness are:

  - **Pros:** unclonable; cheap in production; is created directly in a process of printing;
  - **Cons:** faces the same problems as the natural randomness: sensitivity to the acquisition conditions and the need of high resolution camera or scanner for the verification; a need of accurate synchronization.

- *Copy detection patterns*

  The copy detection patterns are based on the printing of modulated or unmodulated sets of dots that undergo a dot gain effect during printing. The verification process is based on the so-called information loss principle: each time the pattern is printed or scanned some information is lost about the original digital template. Quite often, copy detection patterns are integrated into the traditional 2D codes, like for example Quick Response (QR) codes [20] or DataMatrix codes [21] as shown in Fig. 1.1(h). Codes obtained in such a way are referred to as *Printable Graphical Codes (PGC)*. It is assumed that the distortions introduced during the printing are random and non-invertible due to the dot gain. However, recently, it was shown that the digital template could be efficiently estimated using the modern machine learning algorithms [22, 23]. In summary:

  - **Pros:** easily authenticated by the manufacturer and consumer; cheap; compatible with 2D codes;
  - **Cons:** under certain conditions might be clonable using modern machine learning algorithms.

1.2 Security of PGC

Despite a huge diversity of anti-counterfeiting methods, all of them have their own advantages and drawbacks. Taking into account the cost, scalability, omnipresence and the recent interests of security industry to PGC, we choose PGC as an attractive solution to the anti-counterfeiting problem. At the same time, the security analysis of PGC remains a little studied problem. Therefore, the current work is dedicated to the investigation of PGC security in view of its robustness to the counterfeiting of physical objects.

The general life cycle of PGC is shown in Fig. 1.2. It starts from the printing $\mathrm{P_d}$ of specially designed digital templates $\{\mathbf{t}_i\}_{i=1}^{M}$ by a manufacturer (hereinafter referred to as the

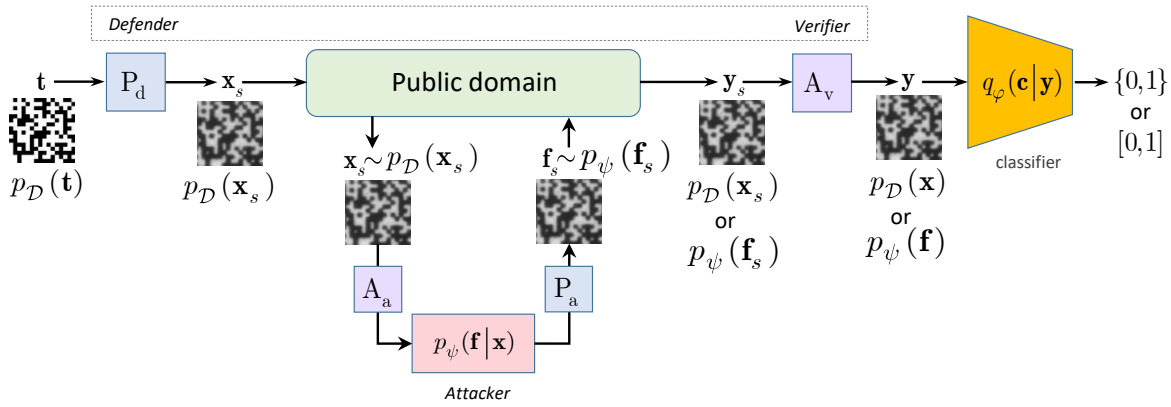
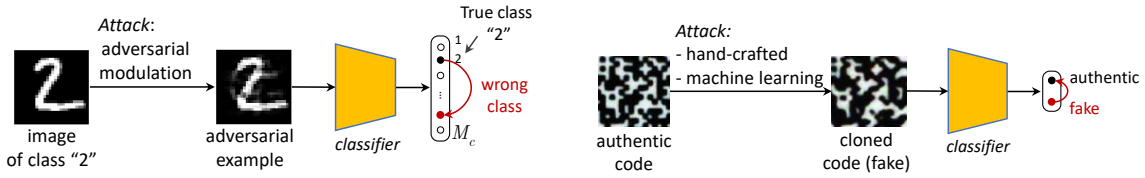


Fig. 1.2 The general scheme of PGC life cycle. The defender-verifier pair is playing against active adversary, i.e., attacker, trying to trick the classifier decision of verifier. $\mathrm{P}_{\mathrm{d/a}}$ denotes the printing by a defender/attacker. $\mathrm{A}_{\mathrm{a/v}}$ means an acquisition by a attacker/verifier. $\mathbf{t}$ corresponds to the original digital template. $\mathbf{x}_s$ is a corresponding printed physical original code, while $\mathbf{x}$ is a digitized original code. $\mathbf{f}_s$ and $\mathbf{f}$ denotes the printed physical and corresponding digitized fake codes respectively. $\mathbf{y}_s$ stands for a physical code from the public domain that might be either original $\mathbf{x}_s$ or fake $\mathbf{f}_s$, while $\mathbf{y}$ denotes the corresponding digitized codes. The classifier might produce either hard decision 0/1 (fake/original) or soft one ranging from 0 to 1.

defender), where M denotes the number of objects. The printed codes $\{\mathbf{x}_{s_i}\}_{i=1}^{M}$ are going to a public domain. The counterfeiter (hereinafter referred to as the *attacker*) having access to these publicly available codes can produce different types of falsificated codes $\{\mathbf{f}_{s_i}\}_{i=1}^{M_f}$ (hereinafter referred to as *fakes*).

The fundamental goals and used instruments of the fakes' production are very diverse. For example, some approaches aim at estimating the parameters of protection elements, like, for example, the secret key(s), parameters of copy detection patterns or embedded secret information. The others aim at direct estimation the digital templates of copy detection patterns from the printed counterparts. In [24] the authors formulate a validation task as a statistical problem and solve this problem by using flipping probabilities and morphological filtering. A blind channel estimation and symbol detection algorithm are proposed in [25]. The authors assume that the received signal is modeled as a hidden Markov chain and propose a version of the forward-backward algorithm to estimate the marginal posterior probability. They show that, when the code structure is taken into account, the estimation of the marginal posterior probability is easier and more accurate. In [26] the authors apply so-called smart attack scenario, where they try to estimate the digital template by evaluating the inverse print & scan model parameters. The approximation of the inverse model is considered as a high-pass linear filtering. In [27] the attacker tries to estimate the original digital template by assuming to have collected several printed realizations of the same original template, which can be quite often practical scenario. Not less important scenario is an estimation



(a) Digital world.

(b) Physical world.

Fig. 1.3 Scope of Thesis addressing adversarial examples in digital world and fakes in physical world. The classifiers in both cases should be robust to adversarial examples and fakes.

of used printer & scanner devices. The role of proper estimation of print & scan model is investigated, for example, in [26, 28].

Without loss of generality, it is possible to split the fake generation attacks into two main categories:

- *Hand-crafted* (HC) attacks that are based on the experience and know-how of the attacker.
- *Machine learning* (ML) attacks that provide more powerful tools and gain more popularity in recent decades due to the rapid development of deep learning models [22, 23, 29, 30]. The ML attacks are based on data $\{\mathbf{t}_j, \mathbf{x}_j\}_{j=1}^{J}$ available for training.

The produced fakes $\{\mathbf{f}_{s_i}\}_{i=1}^{M_f}$ ($M_f \leq M$) are distributed in the public domain. A verification consists in the digitization $\mathrm{A_v}$ of the codes $\mathbf{y}_s$ from the public domain (using scanner, some special reader or modern mobile phone) and their authentication through a classifier that can produce either a hard decision 0 or 1 (fake/authentic) or a soft one ranging from 0 to 1.

1.3 Scope of thesis

In the scope of this Thesis we address the problem of adversarially robust classification in the digital and physical worlds (Fig. 1.3).

Digital world

Besides a huge variability of different factors that impact the reliability and accuracy of the classification in digital world, in general, it is possible to highlight two important factors: *(i)* adversarial robustness of classifier by itself and *(ii)* the amount and quality of the training data available for both attacker and verifier.

Adversarially robust classification: In recent years, classification techniques based on deep neural networks (DNN) are widely used in many fields such as computer vision, natural language processing, self-driving cars, etc. However, the vulnerability of DNN-based classification systems to the *adversarial attacks* [31] questions their usage in many critical applications.

The development of robust DNN-based classifiers is a critical point for future deployment of these methods. Not less important issue is the understanding of the mechanisms behind this vulnerability. Moreover, it is not completely clear how to link the machine learning with cryptography to create an information advantage of the defender over the attacker. In this respect, in this Thesis, we consider a key-based diversified aggregation (KDA) mechanism as a defense strategy in a gray- and black-box scenario. The KDA assumes that the attacker (i) knows the architecture of classifier and the used defense strategy, (ii) has an access to the training dataset but (iii) does not know the secret key and does not have an access to the internal states of the system. The robustness of the system to adversarial attacks is achieved by a specially designed key-based randomization. The proposed randomization prevents the gradients' back propagation and restricts the attacker to create a "bypass" system. The randomization is performed simultaneously in several channels. Each channel introduces its own randomization in a special transform domain. The sharing of a secret key between the training and test stages creates an information advantage to the defender over the attacker. Finally, the aggregation of soft outputs from each channel stabilizes the results and increases the reliability of the final score.

Training data: The deep supervised classifiers demonstrate an impressive performance, when the amount of labeled data is sufficient. However, their performance significantly deteriorates with the decrease of the number of labeled samples. Recently, semi-supervised classifiers based on deep generative models such as VAE (M1+M2) [32], AAE [33], CatGAN [34], etc., along with several other approaches based on multi-view and contrastive metrics [35, 36] are considered as a solution to the addressed problem. Besides the remarkable reported results, the information-theoretic analysis of semi-supervised classifiers based on the generative models and the role of different priors aiming to fulfill a gap in the lack of labeled data remain little studied. In this respect, in this Thesis we study the information bottleneck (IB) framework for semi-supervised classification with several families of priors on latent space representation. A variational decomposition of mutual information terms of IB is applied and an analysis of several regularizers is performed using this variational decomposition.

Another important issue related to the training data is a question of the quality of training data, both labeled and unlabeled, that is quite often neglected in many classification tasks but that plays an important role, when it comes to combat the counterfeiting. In this respect, the issue of the training data quality is addressed in the current Thesis from the point of view of supervised classification. The impact of the different types of fakes on the authentication accuracy of PGC is investigated with respect to the several types of HC fakes produced in this Thesis.

Physical world

In the case of physical world, we investigate the security of PGC considering the entire chain from digital template generation, printing and acquisition to verification. Fig. 1.3(b) shows the setup under analysis. This setup bears certain similarity with the classification under active adversary producing the adversarial examples in the digital world as shown in Fig. 1.3(a). At the same time, the classification problem in the physical world has several particularities. First, the general multi-class classification problem considered in the digital world reduces to a binary classification, also known as an authentication. The binary classifier outputs a score of binary decision in a favor of authentic or fake sample. Accordingly, the role of active adversary, also known as a counterfeiter, is to either reproduce PGC as close as possible to the printed authentic one via the HC or ML based attacks considered in Section 1.2 or to create a code that does not repeat the printed code one-to-one but instead contains some modifications similar to the adversary examples in the digital world aiming at triggering the decision of the binary classifier.

Considering the entire life cycle of PGC as shown in Fig. 1.2, in the scope of this Thesis, we address the impact of different factors on the robustness of authentication classifier on the side of:

- **Defender:**
  - impact of the PGC design, i.e., its structure and size of modulation symbols from which the code is constructed, on the classification/authentication accuracy versus clonability under HC and ML-based attacks;
  - impact of printing technology and its parameters by considering two types of laser and two types of inkjet desktop printers and one industrial digital modern printer such as HP Indigo and resolution of printing;
  - impact of substrates on which the codes are printed that all together determinate the quality of reproduced codes.
- **Attacker:**
  - HC attacks covering two approaches:
    - HC attacks based on copy machines with integral processing enabling to reduce the dot gain effect of authentic printing;
    - HC attacks based on an estimation of original digital templates from the printed counterparts based on the image processing that does not require training data besides the general knowledge about the code design and used printing technology;

  - ML attacks implemented on the knowledge of digital and printed training examples $\{\mathbf{t}_j, \mathbf{x}_j\}_{j=1}^{J}$ for the training of corresponding deep mapper providing an estimation of digital templates for new unseen at training printed codes.

- **Verifier:**

  - impact of acquisition device and its imaging parameters such as a model of used mobile phone, the sensor optic and focusing interval or external light, imaging from fixed system or from hand, settings such as ISO, shutter time, RAW format or YUV/RGB, demosaicing and compression;
  - impact of authentication with knowledge of only digital template, physical scan of code or both of them;
  - impact of knowledge of fakes at the training of classifier by considering a supervised classifier, semi-supervised or completely unsupervised classification in a form of one-class classifier.

The above factors are considered as a game between two parties: defender-verifier versus active attacker. Thus, defender can optimize the design of code, target at selection of a appropriate printer and substrate while the verifier should be interested in ensuring an appropriate imaging conditions and proper classification. As opposed to that, the attacker tries to produce such fakes that trick the decision of classifier.

The attacker has one important advantage over the verifier in terms of the quality of imaging acquisition and conditions at the moment of code acquisition. The attacker can use high quality scanner to produce the best estimation of codes. In contrast, the quality of imaging devices of the verifier is considerably lower. The resolution of mediocre mobile phone camera corresponds to approximately 1000 ppi while the attacker can scan with 2400 ppi and higher. The light and stability of image taking are also in the advantage of the attacker.

Therefore, the defender should design such a code-printer-substrate "system" to prevent the attacker from the reproducing of the original code but yet ensuring that the verifier can still valuably distinguish the original prints from the fakes under the considered limitations of imaging equipment. In this respect, the role of adversarially robust classification operating under the lack of knowledge of possible fakes is very important for the whole system performance.

Up to our best knowledge, these aspects have not been investigated in the prior art publications and we report the first attempt of their systematic analysis in this work.

1.4 Thesis outline

Taking into account the outstanding performance and remarkable achievements of the DNN systems in many applications and the high quality of fakes that are not distinguishable from

the originals by the naked eye, the use of the DNN based classifiers represents an obvious interest. At the same time, recently it has been shown that the DNN systems are vulnerable to adversarial attacks [31]. The development of a robust DNN-based classifiers is an important challenge on a way to combat the counterfeiting. In this respect, in Chapter 2, the robust classification problem in presence of adversarial examples is considered from the perspective of general classification of natural images to avoid the decisions biased by the specificity of the used PGC. The key-based randomized diversification mechanism is proposed as a defense strategy in the multi-channel architecture with the aggregation of classifiers' scores. It is important to remark that the proposed approach is "compliant" with the cryptographic principles, when the defender has an information advantage over the attacker expressed via the knowledge of the secret key(s) shared between the training and inference stages.

Besides being vulnerable to the adversarial attacks the classical DNN based supervised classifiers have another problem, namely, they demonstrate an impressive performance when the amount of labeled data is big, however, their performance significantly deteriorates with the decrease of the number of labeled samples. In practice, despite the availability of a big amount of unlabeled data, the labeling procedure usually is quite expensive, time consuming and not always feasible. In this respect, Chapter 3 aims at investigating the possibilities of the semi-supervised classification that is analyzed from the IB point of view. To compensate the lack of labeled data in the semi-supervised setting, two models of latent space regularization via hand-crafted and learnable priors are considered. Special attention is paid to how the proposed framework compares to the state-of-the-art methods and approaches the performance of fully supervised classification.

Chapter 4 is dedicated to the study of the authentication and copy detection aspects of PGC from the perspective of HC and ML attacks. Taking into account the complexity and multifaceted nature of the raised problem, the study progresses from simple to more complex tasks. It departs from the investigation of the performance of the base-line supervised classification with respect to the typical HC copy attacks. A particular attention is given to study the impact of the quality of data available for the training on the final authentication accuracy. The supervised classification is considered as an ideal and base-line scenario. In practice, it is quite difficult to predict in advance what kind of fakes might be produced by the attacker. Moreover, it is also known that a classifier trained in a fully supervised way on original and fakes produced by some attacks, generally demonstrates very poor performance in face of new unseen fakes. In this respect, a special attention is given to the study of the one-class classification scenario, where the authentication model is trained only on the original data disregarding the potential fakes. The clonability aspects of PGC are studied from the side of the attacker under investigation of impact of the used printing and scanning equipment and the role of the size of basic elements (symbols) in copy detection patterns. The authentication accuracy with respect to the high quality ML copy fakes is considered from the point of view of fully supervised and one-class classification tasks.

Finally, Chapter 5 summarizes the contributions and accomplishments of this Thesis and discusses future research directions.

1.5 Contributions

The contributions and achievements of this Thesis can be summarized, as follows:

Chapter 2

- Introduces a new multi-channel classification architecture with the KDA mechanism as an universal defense strategy against the gradient and non-gradient based gray and black-box adversarial attacks.
- Investigates and demonstrates the high efficiency of the proposed defense strategy against the well-known gradient and non-gradient based adversarial attacks.

Chapter 3

- Proposes a new formulation of IB for the semi-supervised classification and by using a variational decomposition converts it into a practically tractable setup with learnable parameters.
- Develops the variational IB for two classes of hand-crafted and learnable priors on the latent space of classifier and shows its link to the state-of-the-art unsupervised and semi-supervised methods.
- Investigates the role of the priors and different regularizers in the classification, latent and reconstruction spaces for the same fixed architecture under the different amount of labeled data available for training.

Chapter 4

- Introduces three new datasets of PGC, two of which are made on the industrial printing equipment and one of which, in addition, is based on the mobile phone acquisition.
- Investigates the authentication aspects of PGC with respect to the typical HC copy fakes in fully supervised multi-class and one-class classification setups.
- Studies the authentication aspects of PGC with respect to the high quality ML copy fakes in fully supervised multi-class and one-class classification setups.
- Investigates the influence of different factors on the quality and accuracy of produced ML fakes, among which are the impact of the printing equipment and the role of size of the basic elements (symbols) used in the copy detection patterns.

Chapter 2

Adversarially robust classification

The robustness of classifiers is an important problem for many security applications. At the same time, this problem is also raised in many others fields like, for examples, natural language processing or self-driving cars. This problem is not less important in the field of counterfeiting addressed in the current Thesis. To avoid the decisions biased by the specificity of CDP chosen for the investigation in current work, the problem of a robust classification is considered from the perspective of general classification of natural images. In this way, the rich statistics of natural images allow to investigate the generic principles of defenses against adversarial attacks as well as to perform a fair benchmarking with other proposed solutions. The results presented in this Chapter have been published in [37].

Notations

The small bold letters $\mathbf{x}$ are used to denote a signal that can be represented in 1D, 2D or 3D format. $\phi_{\boldsymbol{\theta}}$ denotes a classifier with parameters $\boldsymbol{\theta}$, c is used to denote a class label, $\mathbf{y}$ is a soft output of the classifier $\phi_{\boldsymbol{\theta}}$, where c corresponds to the maximum value in $\mathbf{y}$, $\boldsymbol{\epsilon}$ corresponds to the adversarial perturbation, $\boldsymbol{\epsilon}^d$ means the defender's perturbation.

2.1 Introduction

The advent of deep learning techniques [38] has stimulated the deployment of machine learning in many applications. The DNNs have been applied to solve a wide range of problems in image classification [39, 40], object detection [41, 42], face recognition [43, 44] image caption [45, 46], natural language processing [47, 48], speech recognition [49, 50], drones and robotics [51, 52], malware detection [53, 54], etc., and more science and discovery related fields, such as drug composition analysis [55], brain circuit reconstruction [56], DNA mutation impact analysis [57], etc.

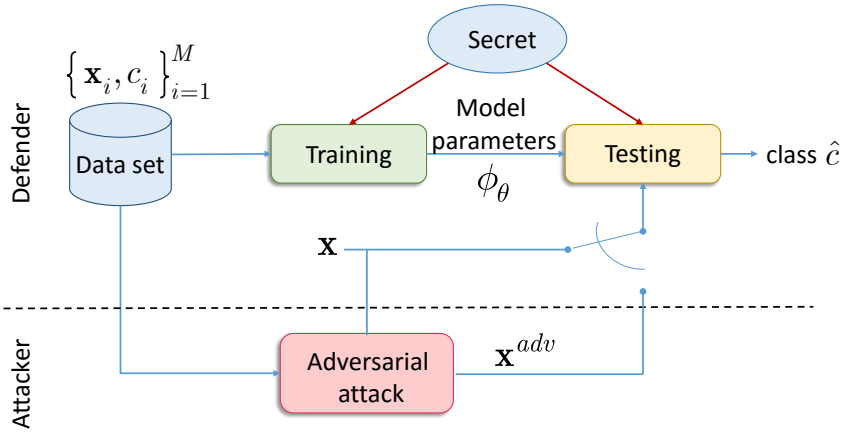


Fig. 2.1 The information access diagram: the defender has an access to the training data and secret shared between the training and test stages while the attacker has only access to the shared training dataset and can observe the output decision of the classifier.

Despite the outstanding performance and remarkable achievements, the DNN systems have recently shown to be vulnerable to *adversarial attacks* [31]. These adversarial attacks aim at tricking a decision of the DNN with high confidence during test time by introducing carefully designed perturbations to a chosen target image. These perturbations are usually quite small in magnitude and almost imperceptible to human vision system that makes them almost universal and yet very dangerous. At the same time, such attacks can cause a neural network to produce an erroneous decision about the signal or image. Even worse, the attacked models report high confidence on the produced wrong classification and it is difficult if not impossible to distinguish it from those obtained on the original data. Moreover, the same added perturbation can fool multiple network models with similar or different architectures trained for the same task [58]. Additionally, Kurakin *et al.* [59] have proven that adversarial examples also exist in physical-world scenarios. This weakness has become a major security concern and seriously questions the usage of the DNN based systems in many security- and trust-sensitive applications.

The serious implications caused by the adversarial attacks triggered a wide interest of researchers to investigate defenses for deep learning models. In recent years, various defense strategies and countermeasures to protect the DNN against adversarial attacks were proposed [60–62]. However, the growing number of defenses leads to a natural invention of new and even more universal attacks. The diversity of discovered adversarial attacks is quite broad but without loss of generality one can cluster all these attacks into three large groups [63, 64]: (1) *white-box* attacks, (2) *gray-box* attacks and (3) *black-box attacks*. The *white-box* attacks assume that the attacker has a full access to the trained model and training data. Despite a big popularity of this group of attacks, their applicability to real-life systems is questionable due to the fact that most real-world systems do not release their internal configurations

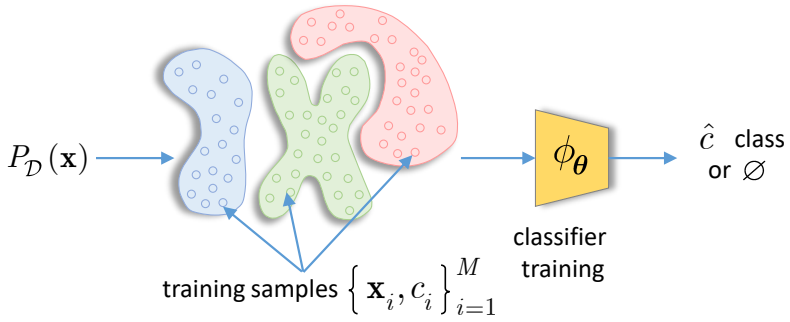


Fig. 2.2 Classifier training: a traditional classifier has an access to training data samples $\{\mathbf{x}_i, c_i\}_{i=1}^{M}$ generated from $P_{\mathcal{D}}(\mathbf{x})$. The classifier learns a set of parameters $\boldsymbol{\theta}$ to output a decision $\hat{c} \in \{1, ..., M_c\}$ or to reject an input ($\varnothing$).

and/or trained parameters. The reason behind the usage of this group of attacks is to be compliant with cryptographic principles stating that "a secure system" should assume public knowledge of the algorithm. However, this principle does not completely apply here since the defender does not use any secret key. In fact, both the defender and attacker share the same training datasets. Thus, the defender has no information advantage over the attacker.

The *gray* and *black-box* scenarios are more suited to real-life applications. The *gray-box* attacks assume that the attacker has certain knowledge about the trained model but there exist some secret unknown elements to the attacker or access to the intermediate results is limited. The *back-box* attack scenario assumes that the attacker only observes the system output to each input without any knowledge about used architecture or possibility to observe the internal states.

In this work, we consider an image classification problem that aims at investigating a new family of defense strategies inspired by the second Kerckhoffs's cryptographic principle [65] that can be applied to both gradient and non-gradient based adversarial attacks in *gray* and *black-box* scenarios. It is named as *Key based Diversified Aggregation* (KDA). The main idea behind the proposed approach is to create an information advantage of the defender over the attacker. The generalized information access diagram of the proposed system is illustrated in Fig. 2.1. The defender has an access to both training data and secret shared between the training and test stages. It is assumed that the attacker can only access the training data and can observe the decision of the classifier. The defender can combine the data and the secret in various ways like, for example, by adding secret key based random noise, by projecting the input onto the random basis vectors generated from the secrete key, via key-driven random cropping or affine transformations, etc. However, since in general case, such perturbations might lead to the classification performance drop, one can create a redundancy by applying these perturbations many times to the input thus creating multi-channel processing. In this way, the classification process is diversified in L channels possessing its own regular perturbation. Since the introduced perturbations are known to the defender, the classifier

$\phi_{\boldsymbol{\theta}_l}$ in each channel l, $1 \leq l \leq L$, is trained only for the certain defender's perturbation. To reduce a possible negative effect of perturbation that might lead to the information loss in general, the soft outputs of the classifiers in the multi-channel system are aggregated. The final decision is communicated to the output of the system in the form of class label $\hat{c} \in \{1, 2, \cdots, M_c\}$, where M_c is the number of classes. At the test stage, the defender has both a probe $\mathbf{x}$ and the secret key while the attacker has only the training set. The attacker can produce an adversarial example $\mathbf{x}^{adv}$ and observe the decision output of system $\hat{c}$ or a rejection. Since the attacker does not have a direct access to the defender's perturbation that is characterized by a sufficient entropy, the only possibility is to increase the number of adversarial tests to be performed according to the observable output. This makes the adversarial attacks less efficient against this system and more complex.

The proposed method provides the following advantages for the defender over the attacker:

- The use of the secret key creates an information advantage for the defender over the attacker.
- The multi-channel system increases the computational burden of the attacker over the defender. The attacker has to attack at least several channels simultaneously to ensure miss-classification outcome.
- The key based diversification and a limited access to the internal system states do not allow the attacker to build a "bypass" system. Unavailability of the "bypass" system makes it difficult, if not impossible, to use the gradient based *white-box* attacks, which are more efficient then the "blind" iterative *black-box* attacks.
- The right choice of aggregation operator and a possibility to choose the channels at random provide an additional degree of freedom and increase the security of the whole system.
- Finally, each channel can have an adjustable amount of randomness, that allows not only to achieve the required level of defense but it also gives a possibility to adapt to different types of attacks.

2.2 Previous work: defenses and attacks

The diagram shown in Fig. 2.2 illustrates a traditional view on the classification process. Assume that the data samples are drawn from a distribution $P_{\mathcal{D}}(\mathbf{x})$ assigned to M_c classes. The labeled training samples represent the training dataset $\{\mathbf{x}_i, c_i\}_{i=1}^{M}$ with M training samples. At the training stage, the classifier $\phi_{\boldsymbol{\theta}}$ uses the available training data to learn the parameters $\boldsymbol{\theta}$. At the test stage, given a test sample $\mathbf{x}$ the trained classifier $\phi_{\boldsymbol{\theta}}$ outputs one

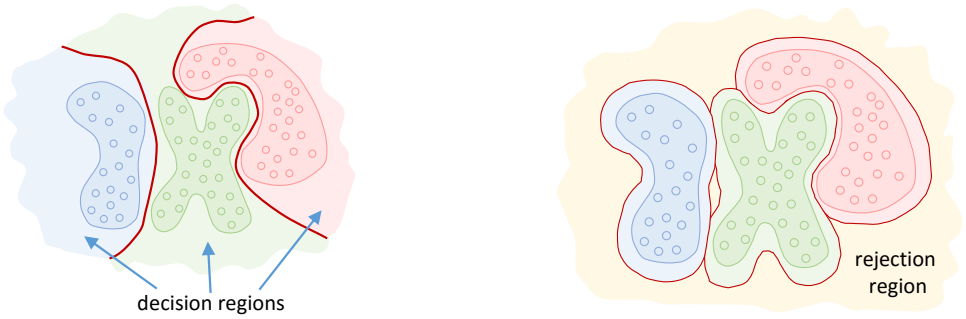


Fig. 2.3 Classifier's decision boundaries: (a) without rejection and (b) with rejection. Note the difference in the decision regions of trained classifiers.

of the classes $\hat{c} \in \{1, 2, \cdots, M_c\}$. A rejection option can be also naturally envisioned. The trained decision boundaries are schematically illustrated in Fig. 2.3.

Since Kurakin *et al.* [59] demonstrated the vulnerability of the DNN to adversarial attacks, one can observe an increasing interest to the investigation of both new attacks and development of efficient countermeasures.

First, a generic attack on the above classifier is considered. As shown in Fig. 2.4(a), the attacker produces an adversarial example $\mathbf{x}^{adv}$ from a host sample $\mathbf{x}$ of a class c by a mapper $g_{\boldsymbol{\alpha}}$: $\mathbf{x}^{adv} = g_{\boldsymbol{\alpha}}(\mathbf{x}, \boldsymbol{\epsilon})$ with some perturbation $\boldsymbol{\epsilon}$ in such a way to fool the classifier $\phi_{\boldsymbol{\theta}}$: $\phi_{\boldsymbol{\theta}}(\mathbf{x}^{adv}) = c^{adv}$, i.e., to force the classifier to produce an output $c^{adv} \neq c$. Generally, $g_{\boldsymbol{\alpha}}$ can be any non-linear mapper. However, a simple additive attack has become the most popular one: $\mathbf{x}^{adv} = \mathbf{x} + \boldsymbol{\epsilon}$. The classic approach assumes that the attacker has an access to the same training data samples as the defender. Thus, there is no information advantage of the defender over the attacker. Moreover, having general knowledge about the used classifier architecture, cost function and training algorithm, the attacker can learn with a certain degree of precision the same decision boundaries as the defender (Fig. 2.3).

2.2.1 Defense strategies

Nowadays, different types of defense strategies have been developed [62]. Without pretending to be exhaustive in the overview, only some of the well-known families of defense strategies are mentioned.

Probably, the largest family of defense strategies is based on *retraining*. Most successful works in this direction are *network distillation* proposed by Papernot *et al.* in [66] and *adversarial retraining* investigated by Goodfellow *et al.* [67], Kurakin *et al.* [59], Wu *et al.* [68], etc. The main reason for this interest is based on a belief that a well trained classifier that has access to the adversarial examples can adjust its decision boundaries and efficiently filter them out. However, the attacker has always the last word in this game and can create new unseen types of adversarial examples. At the same time to envision all possible

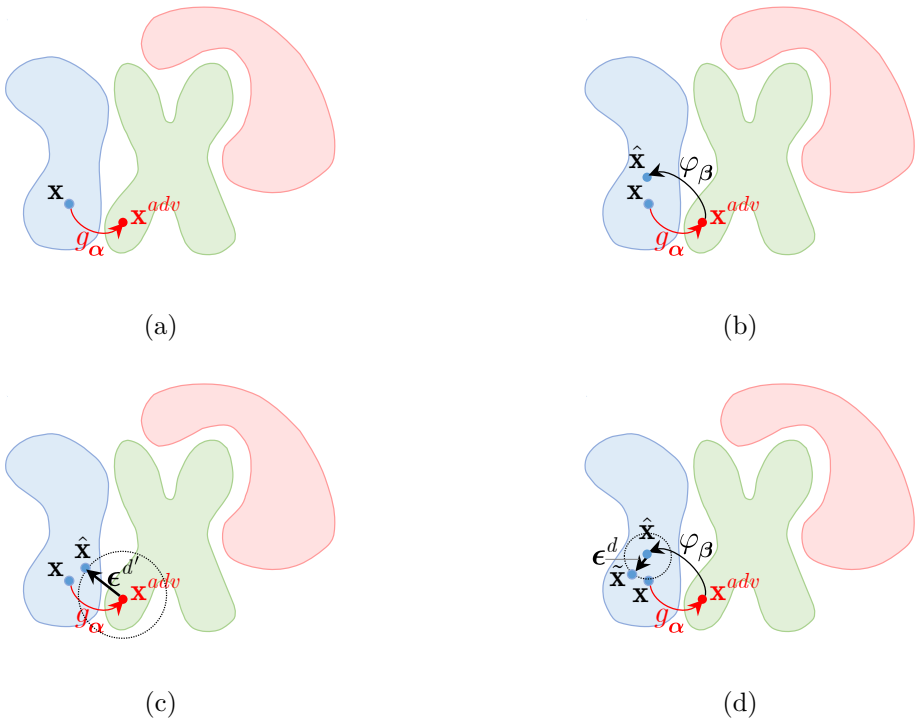


Fig. 2.4 The attacker-defender game in adversarial classification: (a) the attacker produces an adversarial example $\mathbf{x}^{adv}$ from a host $\mathbf{x}$ by a mapper $\mathbf{x}^{adv} = g_{\boldsymbol{\alpha}}(\mathbf{x}, \boldsymbol{\epsilon})$; (b) the defender answers by the pre-filtering $\varphi_{\boldsymbol{\beta}}(\mathbf{x}^{adv})$ to obtain an estimation $\hat{\mathbf{x}}$ on the original host class manifold; (c) an alternative defense strategy by a randomization of input adversarial image as $\hat{\mathbf{x}} = \mathbf{x}^{adv} + \boldsymbol{\epsilon}^{d'}$, the resulting sample will be outside attacker's target class with a small probability that the resulting sample will be in the original host class that requires the classifiers retraining; (d) the proposed defense strategy consists of pre-filtering by $\varphi_{\boldsymbol{\beta}}(\mathbf{x}^{adv})$ and addition of defender's randomized perturbation $\boldsymbol{\epsilon}^d$: $\tilde{\mathbf{x}} = \varphi_{\boldsymbol{\beta}}(\mathbf{x}^{adv}) + \boldsymbol{\epsilon}^d$, such that $\|\boldsymbol{\epsilon}^d\|_2^2 \ll \|\boldsymbol{\epsilon}^{d'}\|_2^2$.

adversarial examples on the side of the defender or to generate them it looks practically infeasible. It should be pointed out that in this setting the defender has no information advantage over the attacker.

The second large family of defense strategies is based on a *detection-rejection* approach. If one assumes that the adversarial examples are based on a modification of original data, it is natural to expect that this adversarial modification leads to the difference in statistics of original data and adversarial ones. It is worth mentioning that the adversarial example detection is similar in nature to a steganalysis problem, where the digital watermarking community has developed a rich family of methods. Summarizing this experience, one can mention that to train efficient detectors of adversarial attacks, it is needed to either know a model describing an adversarial modulation along with the statistics of original data [69] or have an access to the training datasets of original and adversarial examples. Some examples of these strategies include [70–73]. The detection of adversarial attacks might work, if the

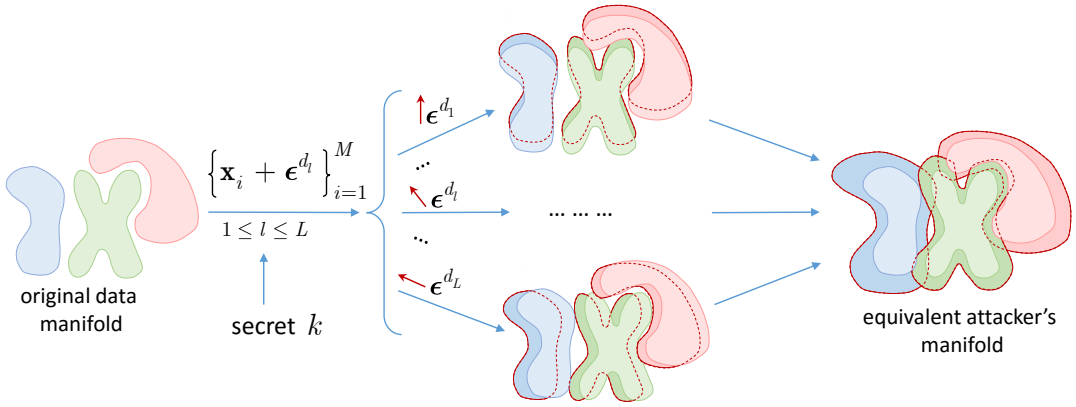


Fig. 2.5 Explanation of multi-channel randomization: given a training dataset, the defender introduces a random perturbation $\boldsymbol{\epsilon}^{d_l}$, $1 \le l \le L$, to each sample $\{\mathbf{x}_i\}_{i=1}^{M}$ and trains L classifiers. Since the perturbation is known at training, i.e., all samples obtain the stationary "bias" by $\boldsymbol{\epsilon}^{d_l}$ for the same classifier l, the randomization has a limited impact on the classifier performance. Since the attacker has no access to the defender's perturbations and all of them are equilikely, an equivalent manifold for the attacker is expanded thus leading to higher entropy and thus increasing the learning complexity.

attack statistics remain the same. Unfortunately, the detection of new attacks requires re-training and there is not guarantee that unseen examples are detectable. Finally, similarly to steganography, advanced attackers will mask the statistics of adversarial perturbations by host statistics as it is done in smooth adversarial attack [74]. This makes the tasks of defender very difficult due to low distinguishability of host and perturbation statistics.

Alternatively, one can envision an active defense strategy when the defender attempts at removing or decreasing the effect of adversarial perturbation by *pre-processing* φ_β via different types of filtering to bring the input to the original data manifold as shown in Fig. 2.4(b). Similar strategy was very efficient against robust digital watermarking, where the watermark was considered as an additive noise and the pre-filtering removed this watermark by denoising. Since the denoising is known to be very efficient in the flat regions [75] to destroy the remaining watermark completely, the additive noise was added to the regions of textures and edges. The goal of filtering can be achieved in several ways. If the model of adversarial modulation is known, the defender can develop an efficient filtering strategy using an analytically derived filter when the model of the image is assumed to be known too. Otherwise, a machine learnable image model can be used. If the model of adversarial perturbation is unknown but the training samples of original data and its adversarial perturbations are available, one can design a network mapping the adversarial input to the clean data. Finally, when only the original data are available, one can train an auto-encoder on it and then apply it to the adversarial data. The trained decoder will attempt at generating almost clean output by projecting an adversarial example onto the manifold of training data encoded into a structure

of the auto-encoder. This form of filtering will be refered to as *regeneration*. For example, Gu *et al.* in [76] propose a deep contractive autoencoder that is a variant of the classical autoencoder with an additional penalty increasing the robustness to adversarial examples. Meng *et al.* in [77] introduce *MagNet*, which combines the detector and regeneration networks. The filtering via denoising was considered in [77–79] and via compression in [80, 81]. However, since, in general case, the pre-processing φ_{β} is deterministic in nature, sooner or later, the attacker can learn and bypass it.

This leads to the need to use *randomization* as a second step [75]. Generally, the idea behind the randomization can be considered as a perturbation of adversarial image with distortion $\boldsymbol{\epsilon}^{d'}$ defined by the defender as shown in Fig. 2.4(c). The resulting sample is expected to be outside the attacker's target class. In practice the randomization is considered in various ways and might include: (a) the randomization of input, (b) the randomization of feature vectors, (c) the randomization of filters, and (d) the randomization of any decision making function parameters. For example, in [61] the authors propose to apply a random permutation to the input data as a form of randomization. Another direction is to randomize the input data by adding noise [82–84]. The input image randomization via random image resizing and padding is investigated in [85]. In [86] the authors explore the idea of stochastically combining different image transforms like, for example, discrete Fourier transform (DFT) domain perturbation, color changing, noise injection, zooming, etc., into a single barrage of randomized transformations to build a strong defense. The idea of DNN feature randomization is examined in [87–89]. Since a particular form of randomization is unknown to the attacker, the defender gains an important information advantage over the attacker. However, it can be achieved only under the condition that the applied defender's randomization tricks are properly incorporated into the classifier. Otherwise, an uninformed classifier will treat them as noise or degradation that unavoidably leads to the drop in the classification accuracy. Moreover, as it was noticed by all authors of the above papers, another problem of randomization techniques is related to the fact that after randomization the inputs might randomly swapped between the classes, which leads to the drop in the classification accuracy too.

In Fig. 2.4(d) the proposed idea of combining the pre-processing and randomization techniques is explained. First of all, it includes returning the input sample to the original data manifold through an appropriate pre-filtering and, secondly, to make the defense stochastic and to create the information advantage for the defender, the perturbing the image with a distortion $\boldsymbol{\epsilon}^{d}$, such that $\|\boldsymbol{\epsilon}^{d}\|_2^2 \ll \|\boldsymbol{\epsilon}^{d'}\|_2^2$. However, in the case of the complex geometry of classes and strong adversarial attacks, the strong perturbation $\boldsymbol{\epsilon}^{d}$ could be required too and, as a consequence, the input sample swapping between the classes could not be excluded. To overcome this shortcoming, in this work a multi-channel randomization technique as shown in Fig. 2.5 is proposed. The main idea is that in each channel l, $1 \leq l \leq L$, the defender introduces a random perturbation $\boldsymbol{\epsilon}^{d_l}$ to each sample $\{\mathbf{x}_i\}_{i=1}^{M}$ and trains l^{th} classifier. Since

the perturbation is known at training, i.e., all samples obtain a stationary "bias" by $\boldsymbol{\epsilon}^{d_l}$ for the same classifier l, the randomization has a limited impact on the classifier performance: all classes' manifolds will be just moved along the direction of perturbation on $\boldsymbol{\epsilon}^{d_l}$ and if they are separable in the original space, then they will stay separable in a new space as well. This allows to avoid the decrease of classification accuracy. Moreover, in this case, the perturbation $\boldsymbol{\epsilon}^{d_l}$ might be sufficiently big to face strong attacks. From the point of the attacker, since he has no access to the defender's perturbations $\boldsymbol{\epsilon}^{d_l}$ all of them are equilikely and an equivalent manifold for the attacker expands thus leading to higher entropy and increases the attacker's learning complexity. Moreover, the targeted attacks become more difficult since the boundaries between the classes on the expanded manifold are not clearly defined due to the random perturbations $\boldsymbol{\epsilon}^{d_l}$.

2.2.2 Adversarial attacks

Without loss of generality, one can group the state-of-the-art adversarial attacks against the DNN classifiers into two main groups [62]:

1. *Gradient* based attacks. The core idea behind this group of attacks consists in the back propagation of the targeted class label to the input layer. A function of the gradient is considered as an adversarial noise that is added to a host image. Obviously, to successfully propagate the gradient via a network it should be end-to-end differentiable.

   Without pretending to be exhaustive in overview, only some well-known attack strategies of this group are mentioned. The L-BFGS attack proposed by Szegedy et. al. in [90] is time-consuming due to the used expensive linear search and, as a consequence, is impractical for real-life applications. However, this attack served as a basis for several more successful attacks such as *Fast Gradient Sign Method* (FGSM) [67]. In contrast to L-BFGS, FGSM is fast but not all the time gives the minimal adversarial perturbation between original and targeted samples. FGSM method has several successful extensions, like FGSM with momentum [91], *One-step Target Class Method* (OTCM) [59], RAND-FGSM algorithm [92], proposed in [59] *Basic Iterative Method* (BIM), projected gradient descent (PGD) [93] the generalized version of BIM and *Iterative Least-Likely Class Method* (ILLC), etc. In addition, it should also be mentioned the *Jacobian-based Saliency Map Attack* (JSMA) [94] and the *DeepFool* approach [95] with its extension *Universal perturbation* [96]. Moreover, one should note the attack proposed by Carlini *et al.* in [97] that will be refered to as C&W attack. As it has been shown in many works, like for example in [98] and [99], this attack is among the most efficient ones against many existing defense mechanisms. Finally, Athalye et. al. in [100] propose Backward Pass Differentiable Approximation technique that aims at avoiding the gradient masking in *white-box* scenario.

2. *Non-gradient* based attacks

   The attacks of this group do not require any knowledge of the DNN gradients or the need of network differentiability. The most well-known members of this group are the *Zeroth Order Optimisation* (ZOO) [101] and the *One Pixel Attack* [102].

In the current work, the most successful representatives of each group will be considered, namely, gradient based C&W and PGD attacks and non-gradient based One Pixel attack.

In general case, for an input image $\mathbf{x} \in [0,1]^{N\times S}$ with a class label $c \in \{1, 2, \cdots, M_c\}$, the optimization problem of finding an adversarial example with the additive perturbation $\mathbf{x}^{adv} = \mathbf{x} + \boldsymbol{\epsilon}$ and target class c^{adv} can be formulated as follows:

$$\begin{aligned} \min_{\boldsymbol{\epsilon}} \quad & \mathcal{L}\Big(c^{adv}, \phi_{\boldsymbol{\theta}}(\mathbf{x}+\boldsymbol{\epsilon})\Big) + \lambda\|\boldsymbol{\epsilon}\|_p, \\ s.t. \quad & \mathbf{x}+\boldsymbol{\epsilon} \in [0,1]^{N\times S}, \end{aligned} \tag{2.1}$$

where $\mathcal{L}(.)$ is a classification loss, $\phi_{\boldsymbol{\theta}}$ is a targeted classifier, $c \neq c^{adv}$, λ is a Lagrangian multiplier and ℓ_p-norm is defined as:

$$\begin{aligned} \|\boldsymbol{\epsilon}\|_p \quad &= \left(\sum_{i=1}^{N\times S} |\epsilon_i|^p\right)^{\frac{1}{p}}, \\ \text{with} \quad & 0 \leq p \leq 2. \end{aligned}$$

2.2.2.1 C&W attack

The C&W attack proposed by Carlini and Wagner in [97] is among the most efficient attacks against many reported so far defense strategies. The authors find the formulation (2.1) difficult for solving directly due to the high non-linearity and propose an alternative definition:

$$\begin{aligned} \min_{\boldsymbol{\epsilon}} \quad & a \cdot f(\mathbf{x}+\boldsymbol{\epsilon}) + \|\boldsymbol{\epsilon}\|_p, \\ s.t. \quad & \mathbf{x}+\boldsymbol{\epsilon} \in [0,1]^{N\times S}, \end{aligned} \tag{2.2}$$

where $a > 0$ is a suitably chosen constant, $f(.)$ is a new objective function such that $\phi_{\boldsymbol{\theta}}(\mathbf{x}+\boldsymbol{\epsilon}) = c^{adv}$, if and only if $f(\mathbf{x}+\boldsymbol{\epsilon}) \leq 0$. In [97] the authors investigate several objective functions $f(.)$ and as the most efficient one they propose:

$$f(\mathbf{x}^{adv}) = \max\Big(\max_{l \neq c^{adv}} (Z(\mathbf{x}^{adv})_l) - Z(\mathbf{x}^{adv})_{c^{adv}}, -\kappa\Big), \tag{2.3}$$

where l is an index of any class while c^{adv} is an index of the adversarial class, $Z(\mathbf{x}) = \phi_{\boldsymbol{\theta}^{n-1}}(\mathbf{x})$ is the result of the network $\phi_{\boldsymbol{\theta}}$ before the last activation function that, in case of classification, usually it is a *softmax*, κ is a constant that controls the confidence of the attack.

2.2.2.2 PGD attack

Additionally to the C&W attack the PGD attack [93] that is an iterative version of FGSM attack and a generalized version of BIM attack, is considered. The PGD solves the optimization problem (1) by computing an adversarial example at the iteration $t+1$ as:

$$\mathbf{x}_{t+1}^{adv} = Proj\Big(\mathbf{x}_t^{adv} + \alpha \cdot sign\Big(\nabla \mathbf{x} \mathcal{L}(c^{adv}, \phi_{\boldsymbol{\theta}}(\mathbf{x}_t^{adv}))\Big)\Big), \tag{2.4}$$

where $Proj(.)$ keeps $\mathbf{x}_{t+1}^{adv}$ within a predefined perturbation range and valid image range and α a is the magnitude of the adversarial perturbation in each iteration.

2.2.2.3 One Pixel Attack

One Pixel attack was proposed by Su *et al.* in [102]. This attack uses a Differential Evolution (DE) optimisation algorithm [103] for the attack generation. The DE algorithm does not require the objective function to be differentiable or known but instead it observes the output of the classifier as a black-box output. The One Pixel attack aims at perturbing a limited number of pixels in the input image $\mathbf{x} \in [0,1]^{N \times S}$. The optimisation problem is formulated as:

$$\begin{array}{ll} \min_{\boldsymbol{\epsilon}} & \mathcal{L}(c^{adv}, \phi_{\boldsymbol{\theta}}(\mathbf{x} + \boldsymbol{\epsilon})), \\ s.t. & \|\boldsymbol{\epsilon}\|_0 \leq d, \end{array} \tag{2.5}$$

where d is a number of pixels to be modified in the original image $\mathbf{x}$ and $\mathcal{L}(.)$ is a classification loss.

2.3 Classification algorithm based on KDA

The generalized diagram of the proposed multi-channel system with the KDA is shown in Fig. 2.6. It consists of six main building blocks:

1. *Pre-filtering* $\varphi_{\boldsymbol{\beta}}(\mathbf{x})$ that has an optional character. The goal of this block is to return the input image $\mathbf{x}$ back to the manifold of the original class by removing high magnitude outliers introduced by the attacker, if any. One can choose a broad range of pre-filtering algorithms from a simple local mean filter to more complex algorithms such as, for example, BM3D [104] or based on DNN mappers [105].

2. *Pre-processing* of the input data in a *transform domain* via a mapping $\mathbb{W}_j$, $1 \leq j \leq J$. In general, the transform $\mathbb{W}_j$ can be any linear data independent mapper. For example it can be a random projection with the dimensionality reduction or expansion, or belong to a family of orthonormal transformations ($\mathbb{W}_j \mathbb{W}_j^T = \mathbb{I}$) like DFT (discrete Fourier transform), DCT (discrete cosines transform), DWT (discrete wavelet transform), etc. Moreover, $\mathbb{W}_j$ can also be a learnable transform. However, it should be pointed out that

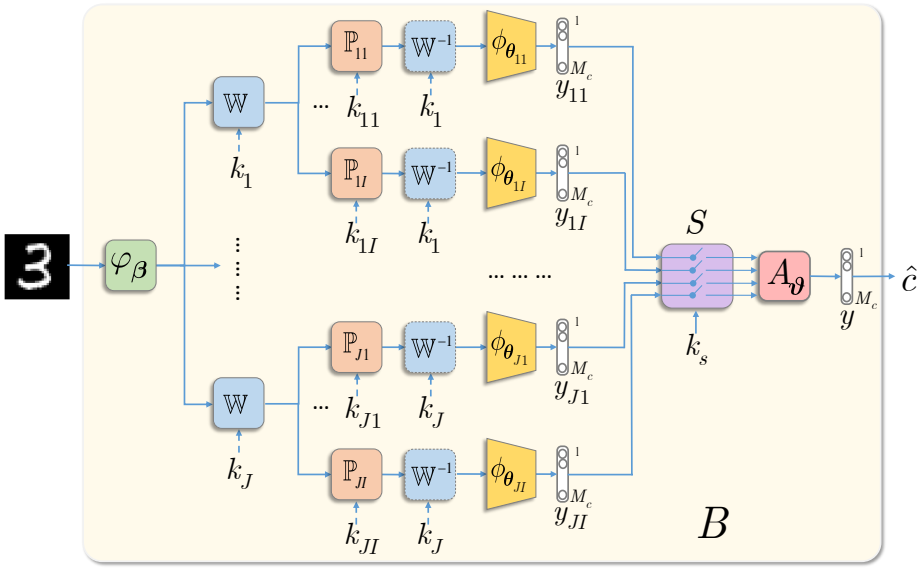


Fig. 2.6 Generalized diagram of the proposed multi-channel system with the KDA.

from the point of view of the robustness to adversarial attacks, the data independent transform $\mathbb{W}_j$ is of preference to avoid any leakage about it from the training data. Furthermore, $\mathbb{W}_j$ can be based on a secret key k_j.

3. *Data independent processing* $\mathbb{P}_{ji}$, $1 \leq i \leq I$ presents the randomization part and serves as a defense against gradient back propagation to the direct domain and the manifold expanding. One can envision several cases. As shown in Fig. 2.7(a), $\mathbb{P}_{ji} \in \{0,1\}^{l \times n}$, $l < n$, presents a lossy sampling of the input image of length n, as considered in [106]. In Fig. 2.7(b), $\mathbb{P}_{ji} \in \{0,1\}^{n \times n}$ is a lossless permutation, similar to [61]. Finally, in Fig. 2.7(c), $\mathbb{P}_{ji} \in \{-1,0,+1\}^{n \times n}$ corresponds to sub-block sign flipping. The orange color highlights the key defined region of key-based sign flipping. This operation is reversible and thus lossless for an authorized party. Moreover, to make the *data independent processing* irreversible for the attacker, it is preferable to use a $\mathbb{P}_{ji}$ based on a secret key k_{ji}.

4. *Classification block* $\phi_{\boldsymbol{\theta}_{ji}}$ can be represented by any family of classifiers. However, if the classifier is designed for the classification of data in the direct domain then it is preferable that it is preceded by $\mathbb{W}_j^{-1}$. This concerns the usage of convolutional or fully connected layers.

5. *Classifiers' selection* S with a key k_s randomly selects J_s classifiers' outputs out of JI pre-trained classifiers' outputs for a further aggregation.

$$\mathbb{P}_{ji} = \begin{matrix} \\ l \end{matrix}\overset{n}{\begin{pmatrix} 1\,0\,0\cdots\,0\,0 \\ 0\,0\,1\cdots\,0\,0 \\ \cdots\ \cdots\ \cdots\ \cdots \\ 0\,0\,0\cdots\,1\,0 \end{pmatrix}} \qquad l < n$$

(a)

$$\mathbb{P}_{ji} = \begin{matrix} \\ n \end{matrix}\overset{n}{\begin{pmatrix} 1\,0\,0\cdots\,0\,0 \\ 0\,0\,1\cdots\,0\,0 \\ \cdots\ \cdots\ \cdots\ \cdots \\ 0\,0\,0\cdots\,1\,0 \end{pmatrix}}$$

(b)

$$\mathbb{P}_{ji} = \begin{matrix} \\ n \end{matrix}\overset{n}{\begin{pmatrix} 1\,0\,0\,\ldots\,\ldots\,\ldots\,0\,0\,0 \\ 0\,1\,0\,\ldots\,\ldots\,\ldots\,0\,0\,0 \\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots \\ 0\,\ldots\,\mathbf{-1}\,0\,\ldots\,\ldots\,\ldots\,0 \\ 0\,\ldots\,0\,\mathbf{1}\,0\,\ldots\,\ldots\,\ldots\,0 \\ 0\,\ldots\,0\,0\,\mathbf{-1}\,\ldots\,\ldots\,\ldots\,0 \\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots \\ 0\,0\,0\,\ldots\,\ldots\,\ldots\,0\,1\,0 \\ 0\,0\,0\,\ldots\,\ldots\,\ldots\,0\,0\,1 \end{pmatrix}}$$

(c)

Fig. 2.7 Randomized transformation $\mathbb{P}_{ji}$, $1 \leq j \leq J$, $1 \leq i \leq I$ examples: (a) randomized sampling, (b) randomized permutation, (c) randomized sign flipping in the sub-block defined in orange. All transforms are key-based.

6. *Aggregation block* $A_{\boldsymbol{\vartheta}}$ can be represented by any operation ranging from a simple summation to learnable operators adapted to the data or to a particular adversarial attack.

As shown in Fig. 2.6, the chain of processing represented by the blocks 2, 3 and 4 can be organized in a parallel multi-channel structure that is followed by the classifiers' selector and the *aggregation block*. The final decision about the class is made based on the aggregated result. The rejection option can be also envisioned.

It should be pointed out that the access to the intermediate results inside the considered system provides the attacker a possibility to use the full system as a *white-box*. The attacker can discover the secret keys k_j and/or k_{ji}, make the system end-to-end differentiable using the Backward Pass Differentiable Approximation technique [100] or via replacing the key based blocks by the "bypass" mappers. Therefore, it is important to restrict the access of the attacker to the intermediate results within the block B (see Fig. 2.6). That satisfies the current assumption about *gray* and *black-box* attacks. Additionally, it is in the accordance with the Kerckhoffs's cryptographic principle when it is assumed that the algorithm and architecture are known to the attacker besides the used secret key that in the current case corresponds to the secret perturbations.

The training of the described classification architecture can be performed according to:

$$(\hat{\boldsymbol{\vartheta}}, \{\hat{\boldsymbol{\theta}}_{\boldsymbol{ji}}\}, \hat{\boldsymbol{\beta}}) = \underset{\boldsymbol{\vartheta}, \{\boldsymbol{\theta}_{ji}\}, \boldsymbol{\beta}}{\operatorname{argmin}} \sum_{t=1}^{T} \sum_{j=1}^{J} \sum_{i=1}^{I} \mathcal{L}(c_t, A_{\boldsymbol{\vartheta}}(\phi_{\boldsymbol{\theta}_{ji}}(\mathbb{W}_j^{-1} \mathbb{P}_{ji} \mathbb{W}_j\ \varphi_{\boldsymbol{\beta}}(\mathbf{x}_t)))), \tag{2.6}$$

where $\mathcal{L}$ is a classification loss, c_t is a class label of the sample $\mathbf{x}_t$, $A_{\boldsymbol{\vartheta}}$ corresponds to the aggregation operator with parameters $\boldsymbol{\vartheta}$, T equals to the number of training samples, J is the total number of channels and I equals to the number of classifiers per channel that, in general, can be different for each channel j. For practical implementation the I is kept equal for all channels. $\phi_{\boldsymbol{\theta}_{ji}}$ is the ith classifier of the jth channel, $\boldsymbol{\theta}$ denotes the parameters of the classifier.

In the proposed system, several practical simplifications are considered that lead to information and complexity advantages for the defender over the attacker:

- The defender training is performed per channel independently up to *selection* and *aggregation* blocks. Since J_s classifiers' outputs out of JI pre-trained classifiers are chosen for the aggregation by the defender at the test stage, the attacker should target to attack a sub-set of classifiers to trick the final decision. To guess a potentially chosen sub-set, the attacker faces $\binom{JI}{J_s}$-combinatorial problem that under properly chosen JI and J_s can represent a considerable complexity burden for the attacker. At the same time, the attacker cannot introduce a single perturbation to trick all classifiers simultaneously.

- The blocks of *data independent processing* $\mathbb{P}_{ji}$ aim at preventing gradient back propagation into the direct domain but the classifier training is adapted to a particular $\mathbb{P}_{ji}$ in each channel.

- It will be shown further by the numerical results that the usage of the multi-channel architecture with the following aggregation stabilizes the results' deviation due to the use of randomization or lossy transformations $\mathbb{P}_{ji}$, if such are used.

- The right choice of the *aggregation* operator $A_{\boldsymbol{\vartheta}}$ provides an additional degree of freedom and increases the security of the system through the possibility to adapt to specific types of attacks.

  In general case, as an aggregation operator the defender could use:

  - an additional classification network that takes as an input the soft outputs of multi-channel classifiers and outputs the final prediction. These multi-channel outputs could be, for example, aggregated into an 1D vector via summation, concatenation, etc;
  - the majority voting of the multi-channel outputs;
  - the summation of the multi-channel outputs with the maximum class selection.

  Generally speaking, since the multi-channel classifier could be trained independently from the aggregation block, the choice of aggregation operator could be defined experimentally.

(a) (b) (c) (d) (e)

Fig. 2.8 Local key based sign flipping in the DCT sub-bands: (a) sub-bands, (b) original image (c) image with a sign flipping in V sub-band, (d) image with a sign flipping in H sub-band and (e) image with a sign flipping in D sub-band.

- Moreover, the overall security level considerably increases due to the independent randomization in each channel. The main advantage of the multi-channel system consists in the fact that each channel can have an adjustable amount of randomness, that allows to obtain the required level of defense against the attacks. In a one-channel system the amount of introduced randomness can be either insufficient to prevent the attacks or too high that leads to a drop in classification accuracy. Therefore, having a channel-wise distributed randomness is more flexible and efficient for the above trade-off.

It should be pointed out that the overall complexity of training the multi-channel system is $I \times J$ higher compared to a single-channel system. At the same time, one should note that the aggregation allows relaxing a network fine-tuning complexity. Thus, there is no need in expensive parameters fine-tuning in a multi-channel system in contrast to a single-channel counterpart. The channel aggregation allows achieve an equivalent performance with "weakly" trained classifiers with lower overall complexity. For the final training the multi-channel classifiers had to have only different secret keys and different starting initialisation per channel. Moreover, in case of non-deep aggregation strategies, the defender training could be performed independently up to *selection* and *aggregation* blocks. This fact allows the defender to train several channels in parallel.

2.4 Randomization using key-based sign flipping in the DCT domain

One of the defense's core elements in the proposed multi-channel architecture shown in Fig. 2.6 is the input image randomized diversification via data independent processing $\mathbb{P}$. The simplest case of such a diversification can be considered for the direct domain with the permutation of input pixels. In fact, the algorithm proposed in [61] reflects this idea for a single channel. However, despite the reported efficiency, a single channel architecture is subject to a drop in classification accuracy, even for the original, i.e., non-adversarial, data. The performance of a permutation-based defense in a multi-channel setting has been investigated in [107]. The obtained results demonstrate a high sensitivity to the gradient

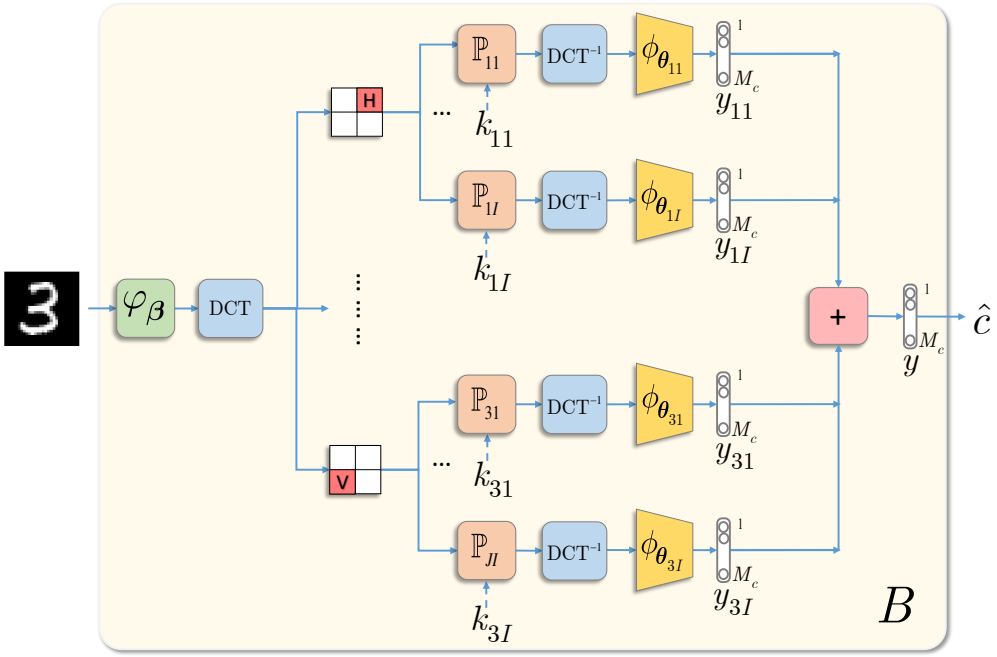


Fig. 2.9 Multi-channel classification with the local DCT sign flipping.

perturbations that degrades the performance of the classifiers. It has been shown in [107, 108] that the preservation of local correlation helps preserve the loss of the gradients and drop of classification accuracy.

In this work, the DCT as the operator $\mathbb{W}$ and the local sign flipping $\mathbb{P}_{ji} \in \{-1, 0, 1\}^{n \times n}$ based on the individual secret key k_{ji} for each classifier $\phi_{\boldsymbol{\theta}}$ are used. The term *local* means that the processing is done only in some sub-band or block of the input image. The length of the secret key k_{ji} equals the length of the corresponding sub-band, i.e., $n \times n$. In general, the image can be split into overlapping or non-overlapping sub-bands of different sizes and different positions that are kept in secret. In current experiments for the simplicity and interpretability, the image is split in the DCT domain into four non-overlapping fixed sub-bands of the same size denoted as: (L) top left that represents the low frequencies of the image, (V) vertical, (H) horizontal and (D) diagonal sub-bands as illustrated in Fig. 2.8(a). The key based sign flipping is applied independently in V, H and D sub-bands keeping all other sub-bands unchanged. The length of secret key in each sub-band corresponds to $n \times n =$ image size$/2 \times$ image size$/2$. The effects of such processing after the inverse DCT transform are perceptually almost unnoticeable and exemplified in Fig. 2.8(c) - 2.8(e).

The corresponding multi-channel architecture is illustrated in Fig. 2.9. For simplicity, as an aggregation operator $A_{\boldsymbol{\vartheta}}$ a simple summation is used and the selector S uses the outputs of all classifiers JI. For the pre-filtering φ_β there is used a custom filter based on a difference of the point of interest in the center of the window with the median value in the window

Fig. 2.10 Examples of original images from each class from MNIST (top line) and Fashion-MNIST (middle line) and CIFAR-10 (bottom line) datasets.

of size 3×3 around this point. If the magnitude of difference exceeds a specified threshold, the pixel is considered to be corrupted by the adversary and its value is replaced by a mean value computed in the window or otherwise, it is kept intact. Finally, under the introduced perturbation, each classifier $\phi_{\boldsymbol{\theta}_{ji}}$ is trained independently as:

$$(\hat{\boldsymbol{\theta}}_{ji}, \hat{\boldsymbol{\beta}}) = \underset{\boldsymbol{\theta}_{ji}, \boldsymbol{\beta}}{\operatorname{argmin}} \sum_{t=1}^{T} \mathcal{L}(c_t, \phi_{\boldsymbol{\theta}_{ji}}(\mathbb{W}^{-1}\mathbb{P}_{ji}\mathbb{W}\, \varphi_{\boldsymbol{\beta}}(\mathbf{x}_t))). \tag{2.7}$$

The soft outputs of trained classifiers are aggregated by the summation as shown in Fig. 2.9.

2.5 Results and discussion

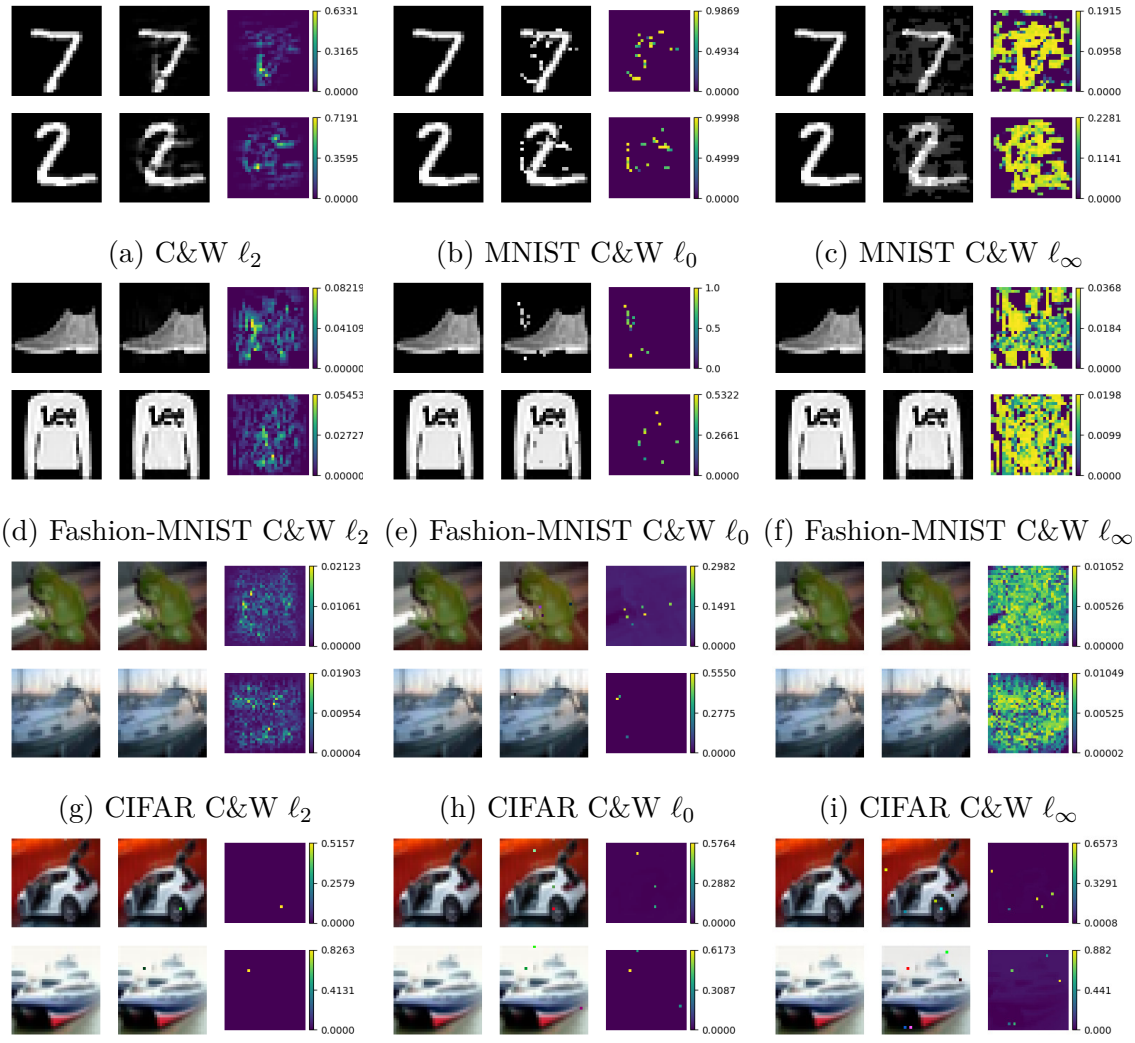
2.5.1 Attacks' scenarios

Accordingly to the central concept of the proposed defense strategy that consists in an information advantage of the defender over the attacker, the attacker has a limited access to the intermediate results and does not know the used secret keys. Therefore, the attacker is not able to attack the proposed system in a *white-box* manner and to create directly the gradient-based adversarial examples. In this respect, the efficiency of the proposed multi-channel architecture with the diversification and randomization by the key based sign flipping in the DCT domain against the adversarial attacks is tested for tree scenarios:

1. *Gray-box* transferability attacks from a single-channel model to a multi-channel model tested on (i) the C&W attack [97] with the constraints on ℓ_2, ℓ_0 and ℓ_∞ norms and (ii) the PGD attack [93].

2. *Gray-box* transferability attacks from a multi-channel model to a multi-channel model under different keys tested on the *One Pixel* attack [102] with perturbation in 1, 3 and 5 pixels.

3. *Black-box* direct attacks tested on the *One Pixel* attack [102] with perturbation in 1, 3 and 5 pixels.



(a) C&W ℓ_2 (b) MNIST C&W ℓ_0 (c) MNIST C&W ℓ_∞

(d) Fashion-MNIST C&W ℓ_2 (e) Fashion-MNIST C&W ℓ_0 (f) Fashion-MNIST C&W ℓ_∞

(g) CIFAR C&W ℓ_2 (h) CIFAR C&W ℓ_0 (i) CIFAR C&W ℓ_∞

(j) CIFAR OnePixel $p = 1$ (k) CIFAR OnePixel $p = 3$ (l) CIFAR OnePixel $p = 5$

Fig. 2.11 Adversarial examples: (left) original image $\mathbf{x}$, (middle) adversarial example $\mathbf{x}^{adv}$, (right) absolute value of the adversarial perturbation $\boldsymbol{\epsilon}$ computed as $|\boldsymbol{\epsilon}| = |\mathbf{x} - \mathbf{x}^{advv}|$.

The experiments are performed on the MNIST [109], Fashion-MNIST [110] and CIFAR-10 datasets [111]. The MNIST set of hand-written digits contains 10 classes, 60000 training and 10000 test gray-scale images of the size 28×28. The Fashion-MNIST set has 10 classes, 60000 training and 10000 test gray-scale images of the size 28×28. The CIFAR-10 consists of 60000 colour images of size 32×32 (50000 train and 10000 test) with 10 classes. Examples of images from each dataset are illustrated in Fig. 2.10. Due to the fact that the attack generation process is sufficiently slow for all considered attacks the experimental results are obtained on the first 1000 test samples. The examples of the attacked images are given in Fig. 2.11. The technical details about the practical implementation of used attacks are given in Appendix B.

Table 2.1 Classification error (%) on the first 1000 test samples for the *gray-box* C&W transferability attacks from a single-channel model to a multi-channel model.

Data type	Attacked vanilla	Transferability vanilla	Transferability KDA # channels · # classifiers		
			3	6	9
MNIST					
Original	1	0.9	0.5	0.5	0.5
C&W ℓ_2	100	6.69	4.69	4.81	4.02
C&W ℓ_0	100	14.2	7.27	7.51	6.78
C&W ℓ_∞	99.99	4.77	2.73	2.28	2.08
Fashion-MNIST					
Original	7.5	7.5	8.1	7.4	7.6
C&W ℓ_2	100	11.2	9.26	8.68	8.9
C&W ℓ_0	100	11.82	10.41	9.97	10
C&W ℓ_∞	99.9	11.59	9.19	8.52	8.79
CIFAR-10					
Original	21	20.6	21.2	19.6	19.5
C&W ℓ_2	100	25.09	22.42	21.3	21.04
C&W ℓ_0	100	30.71	24.58	23.52	23.03
C&W ℓ_∞	100	25.42	22.8	21.39	21.21

The goal of experimental validation is to confirm whether the successful adversarial attacks can trick the proposed defense mechanism.

2.5.2 Empirical results and discussion

Accordingly to the scenarios presented in Section 2.5.1, for each scenario, there is provided the detailed explanation of (i) what kind of assumptions are done, (ii) what kind of knowledge are available to the attacker and (iii) the obtained results.

2.5.2.1 Gray-box transferability: from a single-channel to a multi-channel

The results obtained for the *gray-box* transferability of the adversarial examples from a single-channel model to a multi-channel model are given in Table 2.1 for the C&W attack with the constraints on ℓ_2, ℓ_0 and ℓ_∞ norms and in Table 2.2 for the PGD attack.

The architecture for a single-channel DNN classifier (that is named vanilla) was chosen and made known to the attacker. The attacker has also an access to the same training dataset as the defender. The attacker trains his single-channel vanilla classifier and produces the adversarial examples against his system. The results of this attack are shown in the "Attacked vanilla" column of Tables 2.1 and 2.2. It is easy to see that the C&W attacks are

Table 2.2 Classification error (%) on the first 1000 test samples (CIFAR-10) for the *gray-box* PGD transferability attacks from a single-channel model to a multi-channel model with randomly selected channels (the average results over 10 runs).

Data type	Attacked vanilla	Transferability vanilla	Transferability KDA # channels · # classifiers			
			3	5	7	9
VGG 16						
Original	10.7	11.7	11.6	9.9	9.5	9
PGD	16.1	15.2	14.25	12.16	11.75	11
ResNet 18						
Original	9.5	10.6	11.7	9.3	8.8	8.1
PGD	17.9	14.9	14.7	11.29	10.67	9.7

very efficient against unprotected system. At the same time, Table 2.2 demonstrates that the PGD attack is less efficient.

The defender trains the same single-channel architecture using the same training dataset but with different initialisation of model's parameters. The results of transferability of adversarial examples to the defender's single-channel classifier are shown in the "Transferability vanilla" column of Table 2.1 and 2.2. In contrast to the claimed transferability, the current results clearly demonstrate the low efficiency of the proposed attacks even without any special defense mechanisms for the MNIST, Fashion-MNIST and CIFAR-10 datasets.

The transferabillity of the same adversarial examples to the proposed multi-channel architecture produces the results reported in the "Transfereability KDA" column of Table 2.1 and 2.2. The obtained results show that the increase of the number of channels leads to the decrease of classification error. More particularly, from Table 2.1 it is easy to see that in case of the CIFAR-10 dataset that presents a particular interest for us as a dataset with natural images, the classification error under the *C&W* ℓ_2 and *C&W* ℓ_∞ attacks is the same as in the case of the vanilla classifier on the original non-attacked data. In case of *C&W* ℓ_0 attack, there is only about 2% of attack success. The similar situation can be observed for the the PGD attack given in Table 2.2. In the case of MNIST and Fashion-MNIST datasets, the *C&W* ℓ_2 and *C&W* ℓ_∞ produce about 1-3% of successful attacks while for the *C&W* ℓ_0 this value is slightly higher and is about 2.5 - 5.5%. This is related to a high sparsity of the original images that, generally speaking, is not frequent for the natural images.

From the obtained results one can conclude that the multi-channel model demonstrates the ability to be robust to the adversarial examples generated for the single-channel model with the same architecture of DNN classifier and the ability to improve the classification accuracy on both the non-attacked original and attacked data.

Table 2.3 Classification error (%) on the first 1000 test samples (CIFAR-10) for the *gray-box* OnePixel transferability attacks from a multi-channel model to a multi-channel model under different keys.

Data type	KDA with different keys # channels · # classifiers		
	3	6	9
VGG16			
Original	12.6	11.2	10.5
OnePixel $p = 1$	13.07	10.98	10.4
OnePixel $p = 3$	12.72	11.37	10.3
OnePixel $p = 5$	12.6	11.35	10.8
ResNet18			
Original	9.95	8.4	7.7
OnePixel $p = 1$	9.75	8.17	7.8
OnePixel $p = 3$	10	8.35	7.8
OnePixel $p = 5$	10.38	8.39	8.1

2.5.2.2 Gray-box transferability: from a multi-channel to a multi-channel

The results obtained for the *gray-box* transferability of the adversarial examples from one multi-channel model to another multi-channel model under different keys are given in Table 2.3 for the *One Pixel* attack with perturbation in 1, 3 and 5 pixels.

The architecture for a multi-channel model with the proposed defense strategy was made known to the attacker. The attacker has also access to the same training dataset as the defender. The attacker only does not know the secret keys of the defender. Therefore, he/she trains his/her multi-channel classifier under a selected set of keys and produces the adversarial examples against his system. The defender, in his turn, trains the similar system under different keys and different initialization of model's parameters that remain secret. From the results reported in Table 2.3 it is easy to see that the success of attack does not exceed 0.5% compared to the classification accuracy on the original non-attacked data (rows "Original"). Moreover, as it is also observed in Tables 2.1 and 2.2, the increase of the number of channels in multi-channel model leads to the increase of classification accuracy both on the non-attacked original and attacked data.

Table 2.4 Classification error (%) on the first 1000 CIFAR-10 test samples for the direct *black-box* OnePixel attacks.

Data type	Attacked vanilla	Attacked KDA # channels · # classifiers		
		3	6	9
VGG16				
Original	10.7	11	9.2	8.9
OnePixel $p = 1$	58.04	11	9.5	8.7
OnePixel $p = 3$	72.13	10.9	8.9	8.3
OnePixel $p = 5$	79.02	12.1	9.3	9.1
ResNet18				
Original	9.5	11.1	9.1	7.8
OnePixel $p = 1$	36.96	11.5	9	7.7
OnePixel $p = 3$	49.85	11.5	9.1	7.8
OnePixel $p = 5$	59.74	11.7	9.2	7.8

2.5.2.3 Black-box direct attack

The results obtained for the direct attacks to a single-channel and a multi-channel models in the *black-box* scenario are shown in Table 2.4. The row "original" corresponds to the use of non-attacked original data.

For a single-channel and a multi-channel cases, the attacker does not have any knowledge about the classifiers' architecture, about the number of channels or about the used defense mechanisms. The attacker can only observe the predicted class for the given input. In this respect, the attacker tries to attack the classification models directly in a black-box way. The results obtained for the *One Pixel* attack with perturbation in 1, 3 and 5 pixels are shown in Table 2.4. From these results, it is easy to see that for the non-protected single-channel model ("Attacked vanilla") such kind of attacks can be sufficiently efficient: in the case of VGG16 model, the classification error is about 60-80%, in the case of ResNet18 model, it is about 35-60%. For both classifiers the increase of the number of perturbed pixels (p) leads to the increase of classification error. At the same time, the use of the proposed defense mechanism based on the KDA allows to decrease the classification error to the level of classification on the non-attacked original data or in other words it practically diminished the effect of these attacks.

To summarize the above results, it should be pointed out that as it can be seen from Tables 2.1 - 2.4, the results obtained for the non-attacked original data demonstrate that the use of the proposed multi-channel architecture, in general, allows to improve the classification accuracy of vanilla classifier. This is quite remarkable by itself since it shows that the multi-channel processing with the aggregation does not degrade the performance due to the

Table 2.5 Adversarial distortion.

Attack	Median ℓ_2-norm	Mean ℓ_2-norm
MNIST		
C&W ℓ_2	5.28e-03	5.52e-03
C&W ℓ_0	1.56e-02	1.61e-02
C&W ℓ_∞	1.24e-02	1.29e-02
Fashion-MNIST		
C&W ℓ_2	2.30e-04	5.31e-04
C&W ℓ_0	4.35e-03	4.86e-03
C&W ℓ_∞	4.43e-04	5.43e-04
CIFAR-10		
C&W ℓ_2	7.80e-05	1.19e-04
C&W ℓ_0	2.48e-03	4.55e-03
C&W ℓ_∞	1.73e-04	2.13e-04
ResNet18 (CIFAR-10)		
PGD	1.00e-04	1.37e-04
Vanilla *OnePixel* $p = 1$	5.25e-04	2.76e-03
Vanilla *OnePixel* $p = 3$	1.24e-03	3.51e-03
Vanilla *OnePixel* $p = 5$	1.86e-03	4.18e-03
Multi-channel model *OnePixel* $p = 1$	3.22e-04	2.42e-03
Multi-channel model *OnePixel* $p = 3$	1.33e-03	3.61e-03
Multi-channel model *OnePixel* $p = 5$	2.05e-03	4.33e-03
VGG16 (CIFAR-10)		
PGD	9.99e-05	1.50e-04
Vanilla *OnePixel* $p = 1$	5.86e-04	2.78e-03
Vanilla *OnePixel* $p = 3$	1.37e-03	3.69e-03
Vanilla *OnePixel* $p = 5$	2.02e-03	4.26e-03
Multi-channel model *OnePixel* $p = 1$	3.43e-04	2.25e-03
Multi-channel model *OnePixel* $p = 3$	1.27e-03	3.64e-03
Multi-channel model *OnePixel* $p = 5$	1.91e-03	4.28e-03

introduced key based diversification in contrast to many defense strategies based on the idea of randomization. Finally, the results obtained on the adversarial examples demonstrate high robustness of the proposed KDA based defense technique.

Next, the impact of several factors, such as the level of adversarial distortion and the key-based aggregation on the classification accuracy and robustness to the adversarial attacks are demonstrated.

Table 2.6 Classification error (%) on the first 1000 test samples for the *gray-box* C&W transferability attacks from a single-channel model to a multi-channel model with randomly selected channels (the average results over 10 runs).

Data type	Transferability KDA # channels · # classifiers		
	3	5	7
MNIST			
Original	0.6	0.5	0.6
C&W ℓ_2	5.06	4.77	4.44
C&W ℓ_0	7.77	7.3	7.12
C&W ℓ_∞	3.41	3.12	2.77
Fashion-MNIST			
Original	8.2	8.2	8.1
C&W ℓ_2	9.4	9.1	8.84
C&W ℓ_0	10.58	10.52	10.27
C&W ℓ_∞	9.33	9.09	8.83
CIFAR-10			
Original	21.2	20.5	19.9
C&W ℓ_2	22.92	21.22	21.1
C&W ℓ_0	27.82	25.46	24.33
C&W ℓ_∞	24.57	22.33	21.86

2.5.2.4 Adversarial distortions

The efficiency of the proposed defense strategy with increased adversarial distortions in terms of amplitude value of the adversarial noise and its behavior are investigated in the gray-box scenario presented in Table 2.1. The amplitude of the adversarial noise increases from ℓ_2 to ℓ_∞ and ℓ_0. Fig. 2.11 shows several examples of adversarial noise with indicated noise amplitude. The median and mean ℓ_2-norm of adversarial perturbation are given in Table 2.5. In all cases, the same trained model has been evaluated. The efficiency of the proposed defense strategy with the increase of adversarial distortions in terms of number of distorted pixels and its behaviour are investigated in *black-box* scenario illustrated in Tables 2.4, where $p = 1$, ... $p = 5$ indicate the increase of the number of distorted pixels. The corresponding adversarial noise amplitudes are given in Table 2.5. In all cases, the proposed KDA based defense strategy successfully resists to the adversarial distortion of different levels.

2.5.2.5 Key-based aggregation

Additionally to the multi-channel system with the fixed channels for aggregation shown in Fig. 2.9 and its results demonstrated in Tables 2.1, 2.3 and 2.4, the similar system has

Table 2.7 Classification error (%) on the first 1000 test samples (CIFAR-10) for the multi-channel system against the direct *gray-box* OnePixel attacks with randomly selected channels (the average results over 10 runs).

Data type	Attacked KDA # channels · # classifiers		
	3	5	7
VGG16			
Original	11.7	9.5	9.3
OnePixel $p = 1$	11.3	9.6	9
OnePixel $p = 3$	11.5	9.8	8.9
OnePixel $p = 5$	12	10.6	9.4
ResNet18			
Original	11.1	9.7	8.8
OnePixel $p = 1$	11.1	9.2	8.9
OnePixel $p = 3$	11.4	9.6	8.8
OnePixel $p = 5$	10.9	9.8	9.1

been investigated for the case when the channels for the aggregation were chosen based on a random key. The results averaged over 10 runs are given in Table 2.2, 2.6 and 2.7. Comparing the results for the KDA presented, for example, in Tables 2.1 and 2.6, one can notice a small degradation of performance under the random selection of channels for the aggregation in Table 2.6. This is due to the fact that the sub-bands chosen for the randomization in the setup of Table 2.1 always correspond to the three main sub-bands representing V, H and D sub-bands, whereas the sub-bands representing channels in the setup of Table 2.6 were chosen at random. This discrepancy decreases with the increase of the number of aggregated channels.

In summary, one can conclude that the obtained results indicate that the proposed KDA based defense strategy demonstrates a high robustness to the transferability attacks in the *gray-box* scenarios as well as to the direct *black-box* attacks. Moreover, it allows to improve the classification accuracy of the vanilla classifiers. Finally, it should be pointed out that, in general, the increase of the number of classification channels and *data independent processing* $\mathbb{P}_{ji}$ leads to improving the classification accuracy. However, a trade-off between the further decrease of the classification error and the increase of the complexity of the algorithm should be carefully addressed that goes beyond the scope of this work.

2.6 Conclusions

In this Chapter, a problem of DNN classifiers' protection against adversarial attacks in *gray* and *black*-box scenarios is addressed. The key-based randomized diversification mechanism

is proposed as a defense strategy in the multi-channel architecture with the aggregation of classifiers' scores. The randomized transform is a secret key-based randomization in a defined domain. The goal of this randomization is to prevent the gradient back propagation or use of "bypass" systems by the attacker. It is also important to remark that the proposed approach is "compliant" with the cryptographic principles when the defender has an information advantage over the attacker expressed via the knowledge of the secret key shared between the training and test stages. The efficiency of the proposed defense and the performance of several variations of the considered architecture on three standard datasets against a number of known state-of-the-art attacks are evaluated. The numerical results demonstrate the robustness of the proposed defense mechanism against (i) *gray-box* transferability attacks from a single-channel model to a multi-channel model under assumption that the attacker uses only the knowledge about the single-channel model architecture, (ii) *gray-box* transferability attacks from a multi-channel model to a multi-channel model trained under different keys assuming that the attacker has full knowledge about the multi-channel model architecture and used defense strategy except the defenders' secret keys, (iii) *black-box* direct attacks under assumption that the attacker has no knowledge about the model architecture or defense mechanisms. In all scenarios, as a worst case, it is assumed that the attacker uses the same dataset as the defender. Additionally, the obtained results show that using the multi-channel architecture with the following aggregation stabilizes the results and increases the classification accuracy on the attacked and non-attached original data samples.

The future work aims at investigating in details the security aspects of the proposed KDA algorithm. It looks very interesting to obtain estimates and bounds on the attacker complexity attempting at learning the introduced randomization or by-passing it by some dedicated structures. It is also important to investigate the impact of number of training examples jointly with the randomization in terms of comparison of entropy of training dataset versus needed entropy of randomization. Finally, it is important to extend the aggregation mechanism to more complex learnable strategies instead of used summation.

Chapter 3

Semi-Supervised Classification

In the context of anti-counterfeiting applications, the problem of lack of labeled data available for training is studied from the point of general semi-supervised classification. To avoid the decisions biased by the specificity of PGC chosen for the investigation in the current work, the semi-supervised classification problem is considered under the case of natural images. The semi-supervised classification is formulated as an information bottleneck (IB) framework with several families of priors on a latent space representation. In the light of IB formulation the resulting model allows to better understand the role of various previously proposed regularizers in semi-supervised classification. The developed model is also of use not only for anti-counterfeiting applications but for the general semi-supervised set of problems and applications. The results presented in this Chapter have been published in [112].

Notations

A joint generative distribution is denoted as $p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) = p_{\boldsymbol{\theta}}(\mathbf{z})p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$, whereas marginal $p_{\boldsymbol{\theta}}(\mathbf{z})$ is interpreted as a targeted distribution of latent space and marginal $p_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{z})}\left[p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})\right] = \int_{\mathbf{z}} p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})p_{\boldsymbol{\theta}}(\mathbf{z})\mathrm{d}\mathbf{z}$ as a generated data distribution with a generative model described by $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$, where $\mathbb{E}$ stands for the expected value. A joint data distribution $q_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{z}) = p_{\mathcal{D}}(\mathbf{x})q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$, where $p_{\mathcal{D}}(\mathbf{x})$ denotes an empirical data distribution and $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ is an inference or encoding model and marginal $q_{\boldsymbol{\phi}}(\mathbf{z})$ denotes a "true" or "aggregated" distribution of latent space data. The parameters of encoders are denoted as $\boldsymbol{\phi}_{\mathrm{a}}$ and $\boldsymbol{\phi}_{\mathrm{z}}$ and those of decoders as $\boldsymbol{\theta}_{\mathrm{c}}$ and $\boldsymbol{\theta}_{\mathrm{x}}$. The discriminators corresponding to Kullback–Leibler divergences are denoted as $\mathcal{D}_{\mathrm{x}}$, where the subscript indicates the space to which this discriminator is applied to. The cross-entropy metrics are denoted as $\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$, where the subscript indicates the corresponding vectors. $\mathbf{X}$ denotes random vector, while the corresponding realization is denoted as $\mathbf{x}$.

3.1 Introduction

The deep supervised classifiers demonstrate an impressive performance when the amount of labeled data is large. However, their performance significantly deteriorates with the decrease of labeled samples. Recently, semi-supervised classifiers based on deep generative models such as VAE (M1+M2) [32], AAE [33], CatGAN [34], etc., along with several other approaches based on multi-view and contrastive metrics just to mention the most recent ones [35, 36], are considered as a solution to the above problem. Besides the remarkable reported results, the information theoretic analysis of semi-supervised classifiers based on generative models and the role of different priors aiming to fulfil the gap in the lack of labeled data remain little studied. Therefore, in this work these issues are addressed using IB principle [113] and practically compare different priors on the same architecture of classifier.

Instead of considering the latent space of generative models such as VAE (M1+M2) [32] and AAE [33] trained in the unsupervised way as suitable features for the classification, the analysis departs from the IB formulation of supervised classification, where an encoder-decoder formulation of classifier is considered and the priors are imposed on its latent space. Thus, this work study an approach to semi-supervised classification based on an IB formulation with a variational decomposition of IB compression and classification mutual information terms. To deeper understand the role and impact of different elements of variational IB on the classification accuracy, two types of priors on the latent space of classifier are considered: (i) hand-crafted and (ii) learnable priors. *Hand-crafted* latent space priors impose constraints on a distribution of latent space by fitting it to some targeted distribution according to the variational decomposition of the compression term of the IB. This type of latent space priors is well known as an information dropout [114]. One can also apply the same variational decomposition to the classification term of the IB, where the distribution of labels is supposed to follow some targeted class distribution to maximize the mutual information between inferred labels and targeted ones. This type of class label space regularization reflects an adversarial classification used in AAE [33] and CatGAN [34]. In contrast, *learnable* latent space priors aim at minimizing the need in human expertise in imposing priors on the latent space. Instead, the learnable priors are learned directly from unlabeled data using auto-encoding (AE) principle. In this way, the learnable priors are supposed to compensate the lack of labeled data in the semi-supervised learning yet minimizing the need in the hand-crafted control of the latent space distribution.

There is demonstrated that several state-of-the-art models such as AAE [33], CatGAN [34], VAE (M1+M2) [32], etc., can be considered as instances of the variational IB with the learnable priors. At the same time, the role of different regularizers in the hand-crafted semi-supervised learning is generalized and linked to known frameworks such as information dropout [114].

We evaluate the proposed approach using standard datasets such as MNIST [109] and SVHN [115] on both hand-crafted and learnable features. Besides revealing the impact of different components of variational IB factorization, we demonstrate that the proposed approach outperforms prior works on these datasets.

3.2 Related work

Regularization techniques in semi-supervised learning: Semi-supervised learning tries to find a way to benefit from a large number of unlabeled samples available for training. The most common way to leverage unlabeled data is to add a special regularization term or some mechanism to better generalize to unseen data. The recent work [116] identifies three ways to construct such a regularization: (i) *entropy minimization*, (ii) *consistency regularization* and (iii) *generic regularization*. The entropy minimization [117, 118] encourages the model to output confident predictions on unlabeled data. In addition, more recent work [34] extends this concept to adversarially generated samples or fakes for which the entropy of class label distribution was suggested to be maximized. Finally, the adversarial regularization of label space was considered in [33], where the discriminator was trained to ensure the labels produced by the classifier follow a prior distribution, which was defined to be a categorical one. The consistency regularization [119, 120] encourages the model to produce the same output distribution when its inputs are perturbed. Finally, the generic regularization encourages the model to generalize well and avoid overfitting the training data. It can be achieved by imposing regularizers and corresponding priors on the model parameters or feature vectors.

In this work, we implicitly use the concepts of all three forms of considered regularization frameworks. However, instead of adding additional regularizers to the baseline classifier as suggested by the framework in [116], the corresponding counterparts are derived from a semi-supervised IB framework. In this way, their origin is justified and their impact on overall classification accuracy is investigated for the same system architecture.

Information bottleneck: In the recent years, the IB framework [113] is considered as a theoretical framework for analysis and explanation of supervised deep learning systems. However, as shown in [121], the original IB framework faces several practical issues: (i) for the deterministic deep networks, either the IB functional is infinite for network parameters, that leads to the ill-posed optimization problem, or it is piecewise constant, hence not admitting gradient-based optimization methods, and (ii) the invariance of the IB functional under bijections prevents it from capturing properties of the learned representation that are desirable for classification. In the same work, the authors demonstrate that these issues can be partly resolved for stochastic deep networks, networks that include a (hard or soft) decision rule, or by replacing the IB functional with related but more well-behaved cost functions. It is important to mention that the same authors also note that rather than trying to repair the

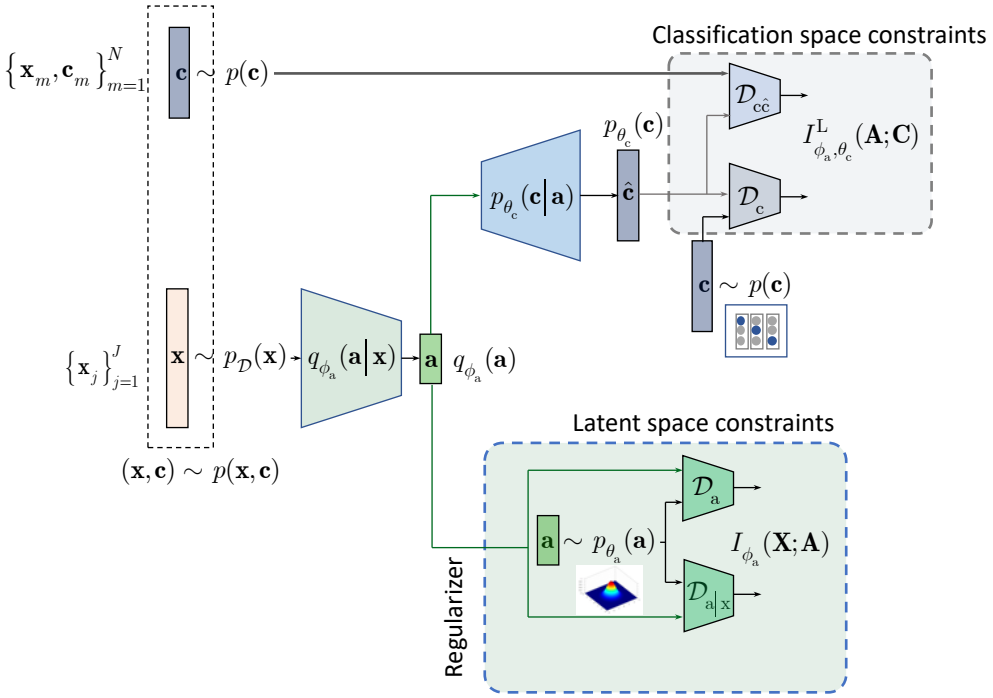


Fig. 3.1 Classification with the hand-crafted latent space regularization.

inherent problems in the IB functional, a better approach may be *to design regularizers on latent representation enforcing the desired properties directly.*

In current work, these ideas are extended using variational approximation approach suggested in [122] and that was also applied to unsupervised models in the previous work [123, 124]. More particularly, the IB framework is extend to the semi-supervised classification and as discussed above two different ways of regularization of the latent space of classifier are considered, i.e., either using traditional hand-crafted priors or suggested learnable priors. Although, the semi-supervised clustering and conditional generation are not considered in this work, the proposed findings can be extended to these problems in a way similar to prior works such as AAE [33], ADGM [125] and SeGMA [126].

The closest works: The proposed framework is closely related to several families of semi-supervised classifiers based on generative models. VAE (M1+M2) [32] combines latent-feature discriminative model M1 and generative semi-supervised model M2. A new latent representation is learned using the generative model from M1 and subsequently a generative semi-supervised model M2 is trained using embeddings from the first latent representation instead of the raw data. Semi-supervised AAE classifier [33] is based on the AE architecture, where the encoder of AE outputs two latent representations: one representing class and another style. The latent class representation is regularized by an adversarial loss forcing it to follow categorical distribution. It is claimed that it plays an essential role for the overall classification performance. The latent style representation is regularized to follow Gaussian

distribution. In both cases of VAE and AAE, the mean square error (MSE) metric is used for the reconstruction space loss. CatGAN [34] is an extension of GAN and is based on an objective function that trades-off mutual information between observed examples and their predicted categorical class distribution, against robustness of the classifier to an adversarial generative model.

In contrast to the above approaches and following the IB framework, we formulate the semi-supervised classification problem as a training of classifier that aims at compressing the input $\mathbf{x}$ to some latent data $\mathbf{a}$ via an encoding that is supposed to retain only class relevant information that is controlled by a decoder as shown in Fig. 3.1. If the amount of labeled data is sufficiently large, the supervised classifier can achieve this goal. However, when the amount of labeled examples is small such an encoder-decoder pair representing an IB-driven classifier is regularized by a latent space and adversarial label space regularizers to fill the gap in training data. The adversarial label space regularization was already used in AAE and CatGAN. The latent space regularization in the scope of IB framework was reported in [114]. In this work, there is demonstrated that both label and latent space regularizations are instances of the generalized IB formulation developed in Section 3.3. At the same time, in contrast to the hypothesis that the considered label space and latent space regularizations are the driving factors behind the success of semi-supervised classifiers, we demonstrate that the hand-crafted priors considered in these models cannot completely fulfill the lack of labeled data and lead to relatively poor performance in comparison to a fully supervised system based on a sole cross-entropy metric. For these reasons, the another mechanism of regularization of latent space based on learnable priors is analyzed as shown in Fig. 3.2 and developed in Section 3.4. Along this line, an IB formulation of AAE is provided to explain the driving mechanisms behind its success as an instance of IB with learnable priors. Finally, we present several extensions that explain the IB origin and role of adversarial regularization in the reconstruction space.

Summary: The considered methods of semi-supervised learning can be differentiated based on: (i) *the targeted tasks* (auto-encoding, clustering, generation or classification that can be accomplished depending on available labeled data); (ii) *the architecture in terms of the latent space representation* (with a single representation vector or with multiple representation vectors); (iii) *the usage of IB or other underlying frameworks* (methods derived from the IB directly or using regularization techniques); (iii) *the label space regularization* (based on available unlabeled data, augmented labeled data, synthetically generated labeled and unlabeled data, specially designed adversarial examples); (iv) *the latent space regularization* (hand-crafted regularizers and priors or learnable priors under the reconstruction and contrastive setups) and (v) *the reconstruction space regularization in case of reconstruction setup* (based on unlabeled and labeled data, augmented data under certain perturbations, synthetically generated examples).

In this work, the main focus is on the latent space regularization for the hand-crafted and learnable priors under the reconstruction setup within the IB based semi-supervised classification. We do not consider any augmentation and adversarial techniques besides a simple stochastic encoding based on the addition of data independent noise at the system input or even deterministic encoding without any form of augmentation. The regularization of the label space and reconstruction space is solely based on the terms derived from the IB framework and only includes available labeled and unlabeled data without any form of augmentation. In this way, it is important to investigate the role and impact of the latent space regularization as such in the IB based semi-supervised classification. The usage of the above mentioned techniques of augmentation should be further investigated and will likely provide an additional performance improvement.

3.3 IB with hand-crafted priors

It is assumed that a semi-supervised classifier has an access to $\{\mathbf{x}_m, \mathbf{c}_m\}_{m=1}^{N}$ training labeled samples, where $\mathbf{x}_m \in \mathbb{R}^D$ denotes m^{th} data sample and $\mathbf{c}_m$ corresponding encoded class label from the set $\{1, 2, \cdots, M_c\}$, generated from the joint distribution $p(\mathbf{c}, \mathbf{x})$, and non-labeled data samples $\{\mathbf{x}_j\}_{j=1}^{J}$ with $J \gg N$. To integrate the knowledge about the labeled and non-labeled data at training, one can formulate the IB as:

$$\mathcal{L}^{\text{HCP}}(\boldsymbol{\phi}_{\text{a}}) = I_{\boldsymbol{\phi}_{\text{a}}}(\mathbf{X}; \mathbf{A}) - \beta_{\text{c}} I_{\boldsymbol{\phi}_{\text{a}}}(\mathbf{A}; \mathbf{C}), \tag{3.1}$$

where $\mathbf{a}$ denotes the latent representation, β_{c} is a Lagrangian multiplier and the IB terms are defined as $I_{\boldsymbol{\phi}_{\text{a}}}(\mathbf{X}; \mathbf{A}) = \mathbb{E}_{q_{\boldsymbol{\phi}_{\text{a}}}(\mathbf{x},\mathbf{a})}\left[\log \frac{q_{\boldsymbol{\phi}_{\text{a}}}(\mathbf{a}|\mathbf{x})}{q_{\boldsymbol{\phi}_{\text{a}}}(\mathbf{a})}\right]$ and $I_{\boldsymbol{\phi}_{\text{a}}}(\mathbf{A}; \mathbf{C}) = \mathbb{E}_{p(\mathbf{c},\mathbf{x})}\left[\mathbb{E}_{q_{\boldsymbol{\phi}_{\text{a}}}(\mathbf{a}|\mathbf{x})}\left[\log \frac{q_{\boldsymbol{\phi}_{\text{a}}}(\mathbf{c}|\mathbf{a})}{p(\mathbf{c})}\right]\right]$, with $q_{\boldsymbol{\phi}_{\text{a}}}(\mathbf{x}, \mathbf{a})$ denoting the joint distribution, $q_{\boldsymbol{\phi}_{\text{a}}}(\mathbf{a}|\mathbf{x})$ the encoder and $q_{\boldsymbol{\phi}_{\text{a}}}(\mathbf{c}|\mathbf{a})$ decoder mappings.

According to the above IB formulation the encoder $q_{\boldsymbol{\phi}_{\text{a}}}(\mathbf{a}|\mathbf{x})$ is trained to minimize the mutual information between $\mathbf{X}$ and $\mathbf{A}$ while ensuring that the decoder $q_{\boldsymbol{\phi}_{\text{a}}}(\mathbf{c}|\mathbf{a})$ can reliably decide on labels $\mathbf{C}$ from the compressed representation $\mathbf{A}$. The trade-off between the compression and recognition terms is controlled by β_{c}. Thus, it is assumed that the information retained in the latent representation $\mathbf{A}$ represents the sufficient statistics for the class labels $\mathbf{C}$.

We assume that the encoder is parametrized by a deep network with parameters $\boldsymbol{\phi}_{\text{a}}$. However, since optimal $q_{\boldsymbol{\phi}_{\text{a}}}(\mathbf{c}|\mathbf{a})$ is unknown, the second term $I_{\boldsymbol{\phi}_{\text{a}}}(\mathbf{A}; \mathbf{C})$ is lower bounded by $I_{\boldsymbol{\phi}_{\text{a}},\boldsymbol{\theta}_{\text{c}}}(\mathbf{A}; \mathbf{C})$ using a variational approximation $p_{\boldsymbol{\theta}_{\text{c}}}(\mathbf{c}|\mathbf{a})$ to $q_{\boldsymbol{\phi}_{\text{a}}}(\mathbf{c}|\mathbf{a})$:

$$
\begin{aligned}
I_{\phi_{\mathrm{a}}}(\mathbf{A};\mathbf{C}) &\triangleq \mathbb{E}_{p(\mathbf{c},\mathbf{x})}\left[\mathbb{E}_{q_{\phi_{\mathrm{a}}}(\mathbf{a}|\mathbf{x})}\left[\log\frac{q_{\phi_{\mathrm{a}}}(\mathbf{c}|\mathbf{a})}{p(\mathbf{c})}\right]\right] = \mathbb{E}_{p(\mathbf{c},\mathbf{x})}\left[\mathbb{E}_{q_{\phi_{\mathrm{a}}}(\mathbf{a}|\mathbf{x})}\left[\log\frac{q_{\phi_{\mathrm{a}}}(\mathbf{c}|\mathbf{a})}{p(\mathbf{c})}\frac{p_{\boldsymbol{\theta}_{\mathrm{c}}}(\mathbf{c}|\mathbf{a})}{p_{\boldsymbol{\theta}_{\mathrm{c}}}(\mathbf{c}|\mathbf{a})}\right]\right] \\
&= \mathbb{E}_{p(\mathbf{c},\mathbf{x})}\left[\mathbb{E}_{q_{\phi_{\mathrm{a}}}(\mathbf{a}|\mathbf{x})}\left[\log\frac{p_{\boldsymbol{\theta}_{\mathrm{c}}}(\mathbf{c}|\mathbf{a})}{p(\mathbf{c})}\right]\right] + \mathbb{E}_{p(\mathbf{c},\mathbf{x})}\left[\mathbb{E}_{q_{\phi_{\mathrm{a}}}(\mathbf{a}|\mathbf{x})}\left[\log\frac{q_{\phi_{\mathrm{a}}}(\mathbf{c}|\mathbf{a})}{p_{\boldsymbol{\theta}_{\mathrm{c}}}(\mathbf{c}|\mathbf{a})}\right]\right] \\
&= \mathbb{E}_{p(\mathbf{c},\mathbf{x})}\left[\mathbb{E}_{q_{\phi_{\mathrm{a}}}(\mathbf{a}|\mathbf{x})}\left[\log\frac{p_{\boldsymbol{\theta}_{\mathrm{c}}}(\mathbf{c}|\mathbf{a})}{p(\mathbf{c})}\right]\right] + \mathbb{E}_{p(\mathbf{c},\mathbf{x})}\left[D_{\mathrm{KL}}(q_{\phi_{\mathrm{a}}}(\mathbf{c}|\mathbf{a})||p_{\boldsymbol{\theta}_{\mathrm{c}}}(\mathbf{c}|\mathbf{a}))\right] \\
&\geq \mathbb{E}_{p(\mathbf{c},\mathbf{x})}\left[\mathbb{E}_{q_{\phi_{\mathrm{a}}}(\mathbf{a}|\mathbf{x})}\left[\log\frac{p_{\boldsymbol{\theta}_{\mathrm{c}}}(\mathbf{c}|\mathbf{a})}{p(\mathbf{c})}\right]\right],
\end{aligned}
\tag{3.2}
$$

where $D_{\mathrm{KL}}(q_{\phi_{\mathrm{a}}}(\mathbf{c}|\mathbf{a})||p_{\boldsymbol{\theta}_{\mathrm{c}}}(\mathbf{c}|\mathbf{a})) = \mathbb{E}_{q_{\phi_{\mathrm{a}}}(\mathbf{a}|\mathbf{x})}\left[\log\frac{q_{\phi_{\mathrm{a}}}(\mathbf{c}|\mathbf{a})}{p_{\boldsymbol{\theta}_{\mathrm{c}}}(\mathbf{c}|\mathbf{a})}\right]$ and the inequality follows from the fact that $D_{\mathrm{KL}}(q_{\phi_{\mathrm{a}}}(\mathbf{c}|\mathbf{a})||p_{\boldsymbol{\theta}_{\mathrm{c}}}(\mathbf{c}|\mathbf{a})) \geq 0$. The term $I_{\phi_{\mathrm{a}},\boldsymbol{\theta}_{\mathrm{c}}}(\mathbf{A};\mathbf{C}) = \mathbb{E}_{p(\mathbf{c},\mathbf{x})}\left[\mathbb{E}_{q_{\phi_{\mathrm{a}}}(\mathbf{a}|\mathbf{x})}\left[\log\frac{p_{\boldsymbol{\theta}_{\mathrm{c}}}(\mathbf{c}|\mathbf{a})}{p(\mathbf{c})}\right]\right]$. Thus, $I_{\phi_{\mathrm{a}}}(\mathbf{A};\mathbf{C}) \geq I_{\phi_{\mathrm{a}},\boldsymbol{\theta}_{\mathrm{c}}}(\mathbf{A};\mathbf{C})$. In this case, the decoder can be implemented as a parametrized deep network with parameters $\boldsymbol{\theta}_{\mathrm{c}}$.

Thus, the IB (3.1) can be reformulated as:

$$
\mathcal{L}^{\mathrm{HCP}_L}(\boldsymbol{\phi}_{\mathrm{a}},\boldsymbol{\theta}_{\mathrm{c}}) = I_{\phi_{\mathrm{a}}}(\mathbf{X};\mathbf{A}) - \beta_{\mathrm{c}} I_{\phi_{\mathrm{a}},\boldsymbol{\theta}_{\mathrm{c}}}(\mathbf{A};\mathbf{C}). \tag{3.3}
$$

The considered IB is schematically shown in Fig. 3.1 and the detailed development of each component of the IB formulation is presented below.

3.3.1 Decomposition of the first term: hand-crafted regularization

The first mutual information term $I_{\phi_{\mathrm{a}}}(\mathbf{X};\mathbf{A})$ in (3.3) can be decomposed using a factorization by a parametric marginal distribution $p_{\boldsymbol{\theta}_{\mathrm{a}}}(\mathbf{a})$ that represents a prior on the latent representation $\mathbf{a}$:

$$
\begin{aligned}
I_{\phi_{\mathrm{a}}}(\mathbf{X};\mathbf{A}) &= \mathbb{E}_{q_{\phi_{\mathrm{a}}}(\mathbf{x},\mathbf{a})}\left[\log\frac{q_{\phi_{\mathrm{a}}}(\mathbf{x},\mathbf{a})}{q_{\phi_{\mathrm{a}}}(\mathbf{a})p_{\mathcal{D}}(\mathbf{x})}\right] = \mathbb{E}_{q_{\phi_{\mathrm{a}}}(\mathbf{x},\mathbf{a})}\left[\log\frac{q_{\phi_{\mathrm{a}}}(\mathbf{a}|\mathbf{x})}{q_{\phi_{\mathrm{a}}}(\mathbf{a})}\frac{p_{\boldsymbol{\theta}_{\mathrm{a}}}(\mathbf{a})}{p_{\boldsymbol{\theta}_{\mathrm{a}}}(\mathbf{a})}\right] \\
&= \mathbb{E}_{p_{\mathcal{D}}(\mathbf{x})}\underbrace{\left[D_{\mathrm{KL}}\left(q_{\phi_{\mathrm{a}}}(\mathbf{a}|\mathbf{X}=\mathbf{x})\|p_{\boldsymbol{\theta}_{\mathrm{a}}}(\mathbf{a})\right)\right]}_{\mathcal{D}_{\mathrm{a}|\mathrm{x}}} - \underbrace{D_{\mathrm{KL}}\left(q_{\phi_{\mathrm{a}}}(\mathbf{a})\|p_{\boldsymbol{\theta}_{\mathrm{a}}}(\mathbf{a})\right)}_{\mathcal{D}_{\mathrm{a}}},
\end{aligned}
\tag{3.4}
$$

where the first term denotes the KL-divergence $\mathcal{D}_{\mathrm{a}|\mathrm{x}} \triangleq D_{\mathrm{KL}}\left(q_{\phi_{\mathrm{a}}}(\mathbf{a}|\mathbf{X}=\mathbf{x})\|p_{\boldsymbol{\theta}_{\mathrm{a}}}(\mathbf{a})\right) = \mathbb{E}_{q_{\phi_{\mathrm{a}}}(\mathbf{a}|\mathbf{x})}\left[\log\frac{q_{\phi_{\mathrm{a}}}(\mathbf{a}|\mathbf{x})}{p_{\boldsymbol{\theta}_{\mathrm{a}}}(\mathbf{a})}\right]$ and the term denotes the KL-divergence $\mathcal{D}_{\mathrm{a}} \triangleq D_{\mathrm{KL}}\left(q_{\phi_{\mathrm{a}}}(\mathbf{a})\|p_{\boldsymbol{\theta}_{\mathrm{a}}}(\mathbf{a})\right) = \mathbb{E}_{q_{\phi_{\mathrm{a}}}(\mathbf{a})}\left[\log\frac{q_{\phi_{\mathrm{a}}}(\mathbf{a})}{p_{\boldsymbol{\theta}_{\mathrm{a}}}(\mathbf{a})}\right]$.

It should be pointed out that the encoding $q_{\phi_{\mathrm{a}}}(\mathbf{a}|\mathbf{x})$ can be both stochastic or deterministic. *Stochastic encoding* $q_{\phi_{\mathrm{a}}}(\mathbf{a}|\mathbf{x})$ can be implemented via: (a) *multiplicative encoding* applied to the input $\mathbf{x}$ as $\mathbf{a} = f_{\phi_{\mathrm{a}}}(\mathbf{x}\odot\boldsymbol{\epsilon})$ or in the latent space $\mathbf{a} = f_{\phi_{\mathrm{a}}}(\mathbf{x})\odot\boldsymbol{\epsilon}$, where $f_{\phi_{\mathrm{a}}}(\mathbf{x})$ is the output of the encoder, $\odot$ denotes the element-wise product and $\boldsymbol{\epsilon}$ follows some data independent or data dependent distribution like in information dropout [114]; (b) *additive encoding* applied

to the input $\mathbf{x}$ as $\mathbf{a} = f_{\boldsymbol{\phi}_\mathrm{a}}(\mathbf{x} + \boldsymbol{\epsilon})$ with the data independent perturbations, e.g., like in PixelGAN [127], or in the latent space with generally data-dependent perturbations of form $\mathbf{a} = f_{\boldsymbol{\phi}_\mathrm{a}}(\mathbf{x}) + \sigma_{\boldsymbol{\phi}_\mathrm{a}}(\mathbf{x}) \odot \boldsymbol{\epsilon}$, where $f_{\boldsymbol{\phi}_\mathrm{a}}(\mathbf{x})$ and $\sigma_{\boldsymbol{\phi}_\mathrm{a}}(\mathbf{x})$ are outputs of the encoder and $\boldsymbol{\epsilon}$ is assumed to be a zero mean unit variance vector like in VAE [32] or (c) *concatenative/mixing encoding* $\mathbf{a} = f_{\boldsymbol{\phi}_\mathrm{a}}([\mathbf{x}, \boldsymbol{\epsilon}])$ that is generally applied at the input of encoder. Deterministic encoding is based on the mapping $\mathbf{a} = f_{\boldsymbol{\phi}_\mathrm{a}}(\mathbf{x})$, i.e., no randomization is introduced, e.g., like one of encoding modalities of AAE [33].

3.3.2 Decomposition of the second term

The second term in (3.3) is factorized to address the semi-supervised training, i.e., to integrate the knowledge of both non-labeled and labeled data available at training:

$$\begin{aligned} I_{\boldsymbol{\phi}_\mathrm{a},\boldsymbol{\theta}_\mathrm{c}}(\mathbf{A};\mathbf{C}) &\triangleq \mathbb{E}_{p(\mathbf{c},\mathbf{x})}\left[\mathbb{E}_{q_{\boldsymbol{\phi}_\mathrm{a}}(\mathbf{a}|\mathbf{x})}\left[\log \frac{p_{\boldsymbol{\theta}_\mathrm{c}}(\mathbf{c}|\mathbf{a})}{p(\mathbf{c})}\frac{p_{\boldsymbol{\theta}_\mathrm{c}}(\mathbf{c})}{p_{\boldsymbol{\theta}_\mathrm{c}}(\mathbf{c})}\right]\right] \\ &= -\mathbb{E}_{p(\mathbf{c})}\left[\log p_{\boldsymbol{\theta}_\mathrm{c}}(\mathbf{c})\right] - \mathbb{E}_{p(\mathbf{c})}\left[\log \frac{p(\mathbf{c})}{p_{\boldsymbol{\theta}_\mathrm{c}}(\mathbf{c})}\right] \\ &\quad + \mathbb{E}_{p(\mathbf{c},\mathbf{x})}\left[\mathbb{E}_{q_{\boldsymbol{\phi}_\mathrm{a}}(\mathbf{a}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}_\mathrm{c}}(\mathbf{c}|\mathbf{a})\right]\right] \\ &= H(p(\mathbf{c}); p_{\boldsymbol{\theta}_\mathrm{c}}(\mathbf{c})) - D_{\mathrm{KL}}\left(p(\mathbf{c}) \| p_{\boldsymbol{\theta}_\mathrm{c}}(\mathbf{c})\right) - H_{\boldsymbol{\theta}_\mathrm{c},\boldsymbol{\phi}_\mathrm{a}}(\mathbf{C}|\mathbf{A}), \end{aligned} \tag{3.5}$$

with $H(p(\mathbf{c}); p_{\boldsymbol{\theta}}(\mathbf{c})) = -\mathbb{E}_{p(\mathbf{c})}\left[\log p_{\boldsymbol{\theta}_\mathrm{c}}(\mathbf{c})\right]$ denoting a cross-entropy between $p(\mathbf{c})$ and $p_{\boldsymbol{\theta}_\mathrm{c}}(\mathbf{c})$, and $\mathcal{D}_\mathrm{c} \triangleq D_{\mathrm{KL}}\left(p(\mathbf{c}) \| p_{\boldsymbol{\theta}_\mathrm{c}}(\mathbf{c})\right) = \mathbb{E}_{p(\mathbf{c})}\left[\log \frac{p(\mathbf{c})}{p_{\boldsymbol{\theta}_\mathrm{c}}(\mathbf{c})}\right]$ to be a KL-divergence between the prior class label distribution $p(\mathbf{c})$ and the estimated one $p_{\boldsymbol{\theta}_\mathrm{c}}(\mathbf{c})$. One can assume different forms of labels' $\mathbf{c}$ encoding but one of the most often used forms is *one-hot-label encoding* that leads to the categorical distribution $p(\mathbf{c}) = \mathrm{cat}(\mathbf{c})$. Finally, the conditional entropy is defined as $\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}} \triangleq H_{\boldsymbol{\theta}_\mathrm{c},\boldsymbol{\phi}_\mathrm{a}}(\mathbf{C}|\mathbf{A}) = -\mathbb{E}_{p(\mathbf{c},\mathbf{x})}\left[\mathbb{E}_{q_{\boldsymbol{\phi}_\mathrm{a}}(\mathbf{a}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}_\mathrm{c}}(\mathbf{c}|\mathbf{a})\right]\right]$.

Since $H(p(\mathbf{c}); p_{\boldsymbol{\theta}_\mathrm{c}}(\mathbf{c})) \geq 0$, one can lower bound (3.5) as $I_{\boldsymbol{\phi}_\mathrm{a},\boldsymbol{\theta}_\mathrm{c}}(\mathbf{A};\mathbf{C}) \geq I^{\mathrm{L}}_{\boldsymbol{\phi}_\mathrm{a},\boldsymbol{\theta}_\mathrm{c}}(\mathbf{A};\mathbf{C})$ where:

$$I^{\mathrm{L}}_{\boldsymbol{\phi}_\mathrm{a},\boldsymbol{\theta}_\mathrm{c}}(\mathbf{A};\mathbf{C}) \triangleq -\underbrace{D_{\mathrm{KL}}\left(p(\mathbf{c}) \| p_{\boldsymbol{\theta}_\mathrm{c}}(\mathbf{c})\right)}_{\mathcal{D}_\mathrm{c}} - \underbrace{H_{\boldsymbol{\theta}_\mathrm{c},\boldsymbol{\phi}_\mathrm{a}}(\mathbf{C}|\mathbf{A})}_{\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}}}. \tag{3.6}$$

3.3.3 Supervised and semi-supervised models with/without hand-crafted priors

Summarizing the above variational decomposition of (3.3) with the terms (3.4) and (3.6), it is possible to considered four practical scenarios:

1. *Supervised training without latent space regularization (***baseline)** based on term $\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}}$ in (3.6):

$$\mathcal{L}^{\mathrm{HCP}}_{\mathrm{S-NoReg}}(\boldsymbol{\theta}_\mathrm{c}, \boldsymbol{\phi}_\mathrm{a}) = \mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}}. \tag{3.7}$$

2. *Semi-supervised training without latent space regularization* based on terms $\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}}$ and $\mathcal{D}_{\mathrm{c}}$ in (3.6):
$$\mathcal{L}^{\mathrm{HCP}}_{\mathrm{SS-NoReg}}(\boldsymbol{\theta}_{\mathrm{c}}, \boldsymbol{\phi}_{\mathrm{a}}) = \mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}} + \mathcal{D}_{\mathrm{c}}. \tag{3.8}$$

3. *Supervised training with latent space regularization* based on term $\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}}$ in (3.6) and either term $\mathcal{D}_{\mathrm{a|x}}$ or $\mathcal{D}_{\mathrm{a}}$ or jointly $\mathcal{D}_{\mathrm{a|x}}$ and $\mathcal{D}_{\mathrm{a}}$ in (3.4):
$$\mathcal{L}^{\mathrm{HCP}}_{\mathrm{S-Reg}}(\boldsymbol{\theta}_{\mathrm{c}}, \boldsymbol{\phi}_{\mathrm{a}}) = \mathbb{E}_{p_{\mathcal{D}}(\mathbf{x})}\left[\mathcal{D}_{\mathrm{a|x}}\right] + \mathcal{D}_{\mathrm{a}} + \beta_{\mathrm{c}}\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}}. \tag{3.9}$$

4. *Semi-supervised training with latent space regularization* deploys all terms in (3.4) and (3.6):
$$\mathcal{L}^{\mathrm{HCP}}_{\mathrm{SS-Reg}}(\boldsymbol{\theta}_{\mathrm{c}}, \boldsymbol{\phi}_{\mathrm{a}}) = \mathbb{E}_{p_{\mathcal{D}}(\mathbf{x})}\left[\mathcal{D}_{\mathrm{a|x}}\right] + \mathcal{D}_{\mathrm{a}} + \beta_{\mathrm{c}}\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}} + \beta_{\mathrm{c}}\mathcal{D}_{\mathrm{c}}. \tag{3.10}$$

The empirical evaluation of these setups is given in Section 3.5. The same architecture of encoder and decoder is used to establish the impact of each term in a function of available labeled data.

3.4 IB with learnable priors

In this Section, the results obtained for the hand-crafted priors are extended to the learnable priors (LP). Instead of applying the hand-crafted regularization of the latent representation $\mathbf{a}$ as suggested by the IB (3.3) and shown in Fig. 3.1, it is assumed that the latent representation $\mathbf{a}$ is regularized by a specially designed AE as shown in Fig. 3.2. The AE based regularization has two components: (i) the latent space $\mathbf{z}$ regularization and (ii) the observation space regularization. The design and training of this latent space regularizer in a form of the AE is guided by its own IB. In the general case, all elements of AE, i.e., its encoder-decoder pair, latent and observation space regularizers are conditioned by the learned class label $\mathbf{c}$. The resulting Lagrangian with the learnable prior is[1]:

$$\mathcal{L}^{\mathrm{LP}}(\boldsymbol{\phi}_{\mathrm{a}}, \boldsymbol{\phi}_{\mathrm{z}}, \boldsymbol{\theta}_{\mathrm{c}}, \boldsymbol{\theta}_{\mathrm{x}}) = \underbrace{I_{\boldsymbol{\phi}_{\mathrm{a}}, \boldsymbol{\phi}_{\mathrm{z}}, \boldsymbol{\theta}_{\mathrm{c}}}(\mathbf{A}; \mathbf{Z}|\mathbf{C})}_{\mathrm{A}} - \beta_{\mathrm{x}} \underbrace{I_{\boldsymbol{\phi}_{\mathrm{a}}, \boldsymbol{\phi}_{\mathrm{z}}, \boldsymbol{\theta}_{\mathrm{c}}, \boldsymbol{\theta}_{\mathrm{x}}}(\mathbf{X}; \mathbf{Z}|\mathbf{C})}_{\mathrm{B}} - \beta_{\mathrm{c}} \underbrace{I^{\mathrm{L}}_{\boldsymbol{\phi}_{\mathrm{a}}, \boldsymbol{\theta}_{\mathrm{c}}}(\mathbf{A}; \mathbf{C})}_{\mathrm{C}}, \tag{3.11}$$

where β_{x} is a Lagrangian multiplier controlling the reconstruction of $\mathbf{x}$ at the decoder and β_{c} is the same as in (3.1).

The terms A and B, conditioned by the class $\mathbf{c}$, play a role of the latent space regularizer by imposing the learnable constrains on the vector $\mathbf{a}$. These two terms correspond to the hand-crafted counterpart $I_{\boldsymbol{\phi}_{\mathrm{a}}}(\mathbf{X}; \mathbf{A})$ in (3.3). The term C in the learnable IB formulation

[1]Formally one should consider $I_{\boldsymbol{\phi}_{\mathrm{a}}, \boldsymbol{\phi}_{\mathrm{z}}, \boldsymbol{\theta}_{\mathrm{c}}}(\mathbf{X}; \mathbf{Z}|\mathbf{C})$ for the term A. However, since $I_{\boldsymbol{\phi}_{\mathrm{a}}, \boldsymbol{\phi}_{\mathrm{z}}, \boldsymbol{\theta}_{\mathrm{c}}}(\mathbf{X}; \mathbf{Z}|\mathbf{C}) \leq I_{\boldsymbol{\phi}_{\mathrm{a}}, \boldsymbol{\phi}_{\mathrm{z}}, \boldsymbol{\theta}_{\mathrm{c}}}(\mathbf{A}; \mathbf{Z}|\mathbf{C})$ due to the Markovianity of considered architecture, the decomposition starts from $\mathbf{A}$ [128], Data Processing Inequality, Theorem 2.8.1.

corresponds to the classification part of hand-crafted IB in (3.3) and can be factorized along the same lines as in (3.6). Therefore, the factorization of terms A and B will be considered.

One can also consider the following IB formulation with the learnable priors with no conditioning on $\mathbf{c}$ in term A in (3.11) leading to an unconditional counterpart D below that can be viewed as an IB generalization of semi-supervised AAE [33]:

$$\mathcal{L}^{\text{LP}}_{\text{AAE}}(\boldsymbol{\phi}_{\text{a}}, \boldsymbol{\phi}_{\text{z}}, \boldsymbol{\theta}_{\text{c}}, \boldsymbol{\theta}_{\text{x}}) = \underbrace{I_{\boldsymbol{\phi}_{\text{a}},\boldsymbol{\phi}_{\text{z}}}(\mathbf{A};\mathbf{Z})}_{\text{D}} - \beta_{\text{x}} \underbrace{I_{\boldsymbol{\phi}_{\text{a}},\boldsymbol{\phi}_{\text{z}},\boldsymbol{\theta}_{\text{c}},\boldsymbol{\theta}_{\text{x}}}(\mathbf{X};\mathbf{Z}|\mathbf{C})}_{\text{B}} - \beta_{\text{c}} \underbrace{I^{\text{L}}_{\boldsymbol{\phi}_{\text{a}},\boldsymbol{\theta}_{\text{c}}}(\mathbf{A};\mathbf{C})}_{\text{C}}. \tag{3.12}$$

3.4.1 Decomposition of latent space regularizer

Considering $p_{\boldsymbol{\phi}_{\text{a}},\boldsymbol{\phi}_{\text{z}},\boldsymbol{\theta}_{\text{c}}}(\mathbf{x},\mathbf{a},\mathbf{c},\mathbf{z}) = p_{\mathcal{D}}(\mathbf{x}) q_{\boldsymbol{\phi}_{\text{a}}}(\mathbf{a}|\mathbf{x}) p_{\boldsymbol{\theta}_{\text{c}}}(\mathbf{c}|\mathbf{a}) q_{\boldsymbol{\phi}_{\text{z}}}(\mathbf{z}|\mathbf{a},\mathbf{c})$ we decompose the term A in (3.11) using variational factorization as:

$$\begin{aligned}
I_{\boldsymbol{\phi}_{\text{a}},\boldsymbol{\phi}_{\text{z}},\boldsymbol{\theta}_{\text{c}}}(\mathbf{A};\mathbf{Z}|\mathbf{C}) &= \mathbb{E}_{p_{\boldsymbol{\phi}_{\text{a}},\boldsymbol{\phi}_{\text{z}},\boldsymbol{\theta}_{\text{c}}}(\mathbf{x},\mathbf{a},\mathbf{c},\mathbf{z})} \left[\log \frac{q_{\boldsymbol{\phi}_{\text{z}}}(\mathbf{z}|\mathbf{a},\mathbf{c})}{q_{\boldsymbol{\phi}_{\text{z}}}(\mathbf{z}|\mathbf{c})} \frac{p_{\boldsymbol{\theta}_{\text{z}}}(\mathbf{z})}{p_{\boldsymbol{\theta}_{\text{z}}}(\mathbf{z})} \right] \\
&= \mathbb{E}_{p_{\mathcal{D}}(\mathbf{x})} \Big[\mathbb{E}_{q_{\boldsymbol{\phi}_{\text{a}}}(\mathbf{a}|\mathbf{x})} \Big[\mathbb{E}_{p_{\boldsymbol{\theta}_{\text{c}}}(\mathbf{c}|\mathbf{a})} \underbrace{\left[D_{\text{KL}} \left(q_{\boldsymbol{\phi}_{\text{z}}}(\mathbf{z}|\mathbf{A}=\mathbf{a},\mathbf{C}=\mathbf{c}) \| p_{\boldsymbol{\theta}_{\text{z}}}(\mathbf{z}) \right) \right]}_{\mathcal{D}_{\text{z}|\text{a,c}}} \Big] \Big] \\
&\quad - \mathbb{E}_{p_{\mathcal{D}}(\mathbf{x})} \Big[\mathbb{E}_{q_{\boldsymbol{\phi}_{\text{a}}}(\mathbf{a}|\mathbf{x})} \Big[\mathbb{E}_{p_{\boldsymbol{\theta}_{\text{c}}}(\mathbf{c}|\mathbf{a})} \underbrace{\left[D_{\text{KL}} \left(q_{\boldsymbol{\phi}_{\text{z}}}(\mathbf{z}|\mathbf{C}=\mathbf{c}) \| p_{\boldsymbol{\theta}_{\text{z}}}(\mathbf{z}) \right) \right]}_{\mathcal{D}_{\text{z}|\text{c}}} \Big] \Big],
\end{aligned} \tag{3.13}$$

where $\mathcal{D}_{\text{z}|\text{a,c}} \triangleq D_{\text{KL}} \left(q_{\boldsymbol{\phi}_{\text{z}}}(\mathbf{z}|\mathbf{a},\mathbf{c}) \| p_{\boldsymbol{\theta}_{\text{z}}}(\mathbf{z}) \right) = \mathbb{E}_{q_{\boldsymbol{\phi}_{\text{z}}}(\mathbf{z}|\mathbf{a},\mathbf{c})} \left[\log \frac{q_{\boldsymbol{\phi}_{\text{z}}}(\mathbf{z}|\mathbf{a},\mathbf{c})}{p_{\boldsymbol{\theta}_{\text{z}}}(\mathbf{z})} \right]$ and $\mathcal{D}_{\text{z}|\text{c}} \triangleq D_{\text{KL}} \left(q_{\boldsymbol{\phi}_{\text{z}}}(\mathbf{z}|\mathbf{c}) \| p_{\boldsymbol{\theta}_{\text{z}}}(\mathbf{z}) \right) = \mathbb{E}_{q_{\boldsymbol{\phi}_{\text{z}}}(\mathbf{z}|\mathbf{c})} \left[\log \frac{q_{\boldsymbol{\phi}_{\text{z}}}(\mathbf{z}|\mathbf{c})}{p_{\boldsymbol{\theta}_{\text{z}}}(\mathbf{z})} \right]$ denote the KL-divergence terms and $q_{\boldsymbol{\phi}_{\text{z}}}(\mathbf{z}|\mathbf{c}) = \mathbb{E}_{p_{\mathcal{D}}(\mathbf{x})} \left[\mathbb{E}_{q_{\boldsymbol{\phi}_{\text{a}}}(\mathbf{a}|\mathbf{x})} \left[q_{\boldsymbol{\phi}_{\text{z}}}(\mathbf{z}|\mathbf{a},\mathbf{c}) \right] \right]$.

3.4.2 Decomposition of reconstruction space regularizer

Denoting $p_{\boldsymbol{\phi}_{\text{a}},\boldsymbol{\phi}_{\text{z}},\boldsymbol{\theta}_{\text{c}},\boldsymbol{\theta}_{\text{x}}}(\mathbf{x},\mathbf{a},\mathbf{c},\mathbf{z}) = p_{\mathcal{D}}(\mathbf{x}) q_{\boldsymbol{\phi}_{\text{a}}}(\mathbf{a}|\mathbf{x}) p_{\boldsymbol{\theta}_{\text{c}}}(\mathbf{c}|\mathbf{a}) q_{\boldsymbol{\phi}_{\text{z}}}(\mathbf{z}|\mathbf{a},\mathbf{c}) p_{\boldsymbol{\theta}_{\text{x}}}(\mathbf{x}|\mathbf{z},\mathbf{c})$, the term B is decomposed in (3.11) as:

$$\begin{aligned}
I_{\boldsymbol{\phi}_{\text{a}},\boldsymbol{\phi}_{\text{z}},\boldsymbol{\theta}_{\text{c}},\boldsymbol{\theta}_{\text{x}}}(\mathbf{X};\mathbf{Z}|\mathbf{C}) &= \mathbb{E}_{p_{\boldsymbol{\phi}_{\text{a}},\boldsymbol{\phi}_{\text{z}},\boldsymbol{\theta}_{\text{c}},\boldsymbol{\theta}_{\text{c}}}(\mathbf{x},\mathbf{a},\mathbf{c},\mathbf{z})} \left[\log \frac{p_{\boldsymbol{\theta}_{\text{x}}}(\mathbf{x}|\mathbf{z},\mathbf{c})}{p_{\mathcal{D}}(\mathbf{x}|\mathbf{c})} \frac{p_{\boldsymbol{\theta}_{\text{x}}}(\mathbf{x})}{p_{\boldsymbol{\theta}_{\text{x}}}(\mathbf{x})} \right] \\
&= \mathbb{E}_{p_{\boldsymbol{\theta}_{\text{c}}}(\mathbf{c})} \left[H(p_{\mathcal{D}}(\mathbf{x}|\mathbf{c}); p_{\boldsymbol{\theta}_{\text{x}}}(\mathbf{x})) \right] \\
&\quad - \mathbb{E}_{p_{\boldsymbol{\theta}_{\text{c}}}(\mathbf{c})} \underbrace{\left[D_{\text{KL}} \left(p_{\mathcal{D}}(\mathbf{x}|\mathbf{C}=\mathbf{c}) \| p_{\boldsymbol{\theta}_{\text{x}}}(\mathbf{x}) \right) \right]}_{\mathcal{D}_{\text{x}|\text{c}}} - \underbrace{H_{\boldsymbol{\phi}_{\text{a}},\boldsymbol{\phi}_{\text{z}},\boldsymbol{\theta}_{\text{c}},\boldsymbol{\theta}_{\text{x}}}(\mathbf{X}|\mathbf{Z},\mathbf{C})}_{\mathcal{D}_{\text{x}\hat{\text{x}}}},
\end{aligned} \tag{3.14}$$

where $p_{\boldsymbol{\theta}_{\text{c}}}(\mathbf{c}) = \mathbb{E}_{p_{\mathcal{D}}(\mathbf{x})} \left[\mathbb{E}_{q_{\boldsymbol{\phi}_{\mathbf{a}}}(\mathbf{a}|\mathbf{x})} \left[p_{\boldsymbol{\theta}_{\text{c}}}(\mathbf{c}|\mathbf{a}) \right] \right]$. The terms are defined as $H(p_{\mathcal{D}}(\mathbf{x}|\mathbf{c}); p_{\boldsymbol{\theta}_{\text{x}}}(\mathbf{x})) = -\mathbb{E}_{p_{\mathcal{D}}(\mathbf{x}|\mathbf{c})} \left[\log p_{\boldsymbol{\theta}_{\text{x}}}(\mathbf{x}) \right]$, $\mathcal{D}_{\text{x}|\text{c}} \triangleq D_{\text{KL}} \left(p_{\mathcal{D}}(\mathbf{x}|\mathbf{C}=\mathbf{c}) \| p_{\boldsymbol{\theta}_{\text{x}}}(\mathbf{x}) \right) = \mathbb{E}_{p_{\mathcal{D}}(\mathbf{x}|\mathbf{c})} \left[\log \frac{p_{\mathcal{D}}(\mathbf{x}|\mathbf{c})}{p_{\boldsymbol{\theta}_{\text{x}}}(\mathbf{x})} \right]$ and $\mathcal{D}_{\text{x}\hat{\text{x}}} \triangleq H_{\boldsymbol{\phi}_{\text{a}},\boldsymbol{\phi}_{\text{z}},\boldsymbol{\theta}_{\text{c}},\boldsymbol{\theta}_{\text{x}}}(\mathbf{X}|\mathbf{Z},\mathbf{C}) = -\mathbb{E}_{p_{\mathcal{D}}(\mathbf{x})} \left[\mathbb{E}_{q_{\boldsymbol{\phi}_{\text{a}}}(\mathbf{a}|\mathbf{x})} \left[\mathbb{E}_{p_{\boldsymbol{\theta}_{\text{c}}}(\mathbf{c}|\mathbf{a})} \left[\mathbb{E}_{q_{\boldsymbol{\phi}_{\text{z}}}(\mathbf{z}|\mathbf{a},\mathbf{c})} \left[\log p_{\boldsymbol{\theta}_{\mathbf{x}}}(\mathbf{x}|\mathbf{z},\mathbf{c}) \right] \right] \right] \right]$.

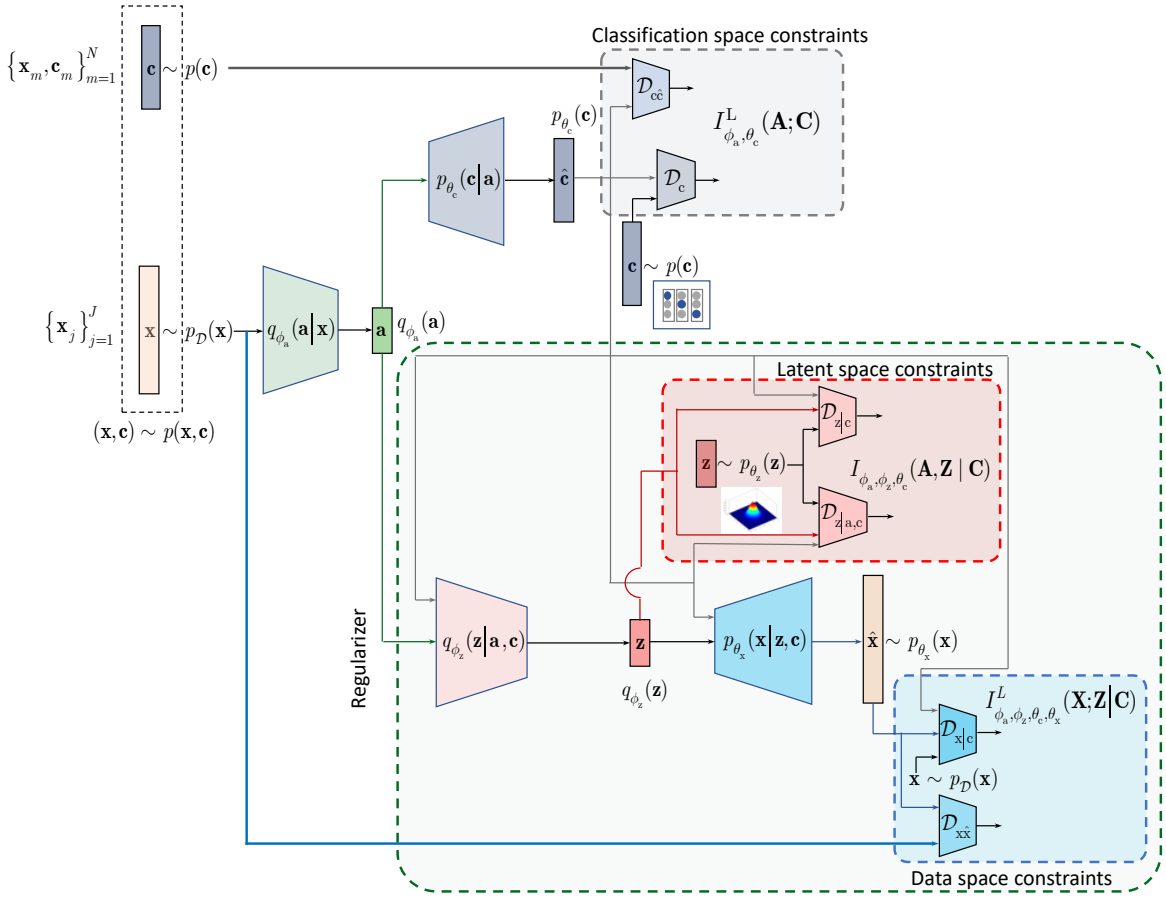


Fig. 3.2 Classification with the learnable latent space regularization.

Since $\mathbb{E}_{p_{\boldsymbol{\theta}_\mathrm{c}}(\mathbf{c})}\left[H(p_{\mathcal{D}}(\mathbf{x}|\mathbf{c}); p_{\boldsymbol{\theta}_\mathrm{x}}(\mathbf{x}))\right] \geq 0$, it can be lower bounded $I_{\boldsymbol{\phi}_\mathrm{a},\boldsymbol{\phi}_\mathrm{z},\boldsymbol{\theta}_\mathrm{c},\boldsymbol{\theta}_\mathrm{x}}(\mathbf{X};\mathbf{Z}|\mathbf{C}) \geq I^{\mathrm{L}}_{\boldsymbol{\phi}_\mathrm{a},\boldsymbol{\phi}_\mathrm{z},\boldsymbol{\theta}_\mathrm{c},\boldsymbol{\theta}_\mathrm{x}}(\mathbf{X};\mathbf{Z}|\mathbf{C}) \triangleq -\mathcal{D}_{\mathrm{x}|\mathrm{c}} - \mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$.

3.4.3 Semi-supervised models with learnable priors

Summarizing the above variational decomposition of (3.11) with the terms (3.13) and (3.14), we consider *semi-supervised training with latent space regularization* as:

$$\begin{aligned}\mathcal{L}^{\mathrm{LP}}_{\mathrm{SS-Reg}}(\boldsymbol{\theta}_\mathrm{c}, \boldsymbol{\theta}_\mathrm{x}, \boldsymbol{\phi}_\mathrm{a}, \boldsymbol{\phi}_\mathrm{z}) = \mathbb{E}_{p_{\mathcal{D}}(\mathbf{x})}\Big[\mathbb{E}_{q_{\boldsymbol{\phi}_\mathrm{a}}(\mathbf{a}|\mathbf{x})}\Big[\mathbb{E}_{p_{\boldsymbol{\theta}_\mathrm{c}}(\mathbf{c}|\mathbf{a})}\Big[\mathcal{D}_{\mathrm{z}|\mathrm{a},\mathrm{c}}\Big]\Big]\Big] \\ + \mathbb{E}_{p_{\mathcal{D}}(\mathbf{x})}\Big[\mathbb{E}_{q_{\boldsymbol{\phi}_\mathrm{a}}(\mathbf{a}|\mathbf{x})}\Big[\mathbb{E}_{p_{\boldsymbol{\theta}_\mathrm{c}}(\mathbf{c}|\mathbf{a})}\Big[\mathcal{D}_{\mathrm{z}|\mathrm{c}}\Big]\Big]\Big] \\ + \beta_\mathrm{x}\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}} + \beta_\mathrm{x}\mathbb{E}_{p_{\boldsymbol{\theta}_\mathrm{c}}(\mathbf{c})}\Big[\mathcal{D}_{\mathrm{x}|\mathrm{c}}\Big] + \beta_\mathrm{c}\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}} + \beta_\mathrm{c}\mathcal{D}_{\mathrm{c}}.\end{aligned} \tag{3.15}$$

To create a link to the semi-supervised AAE [33], we also consider (3.12), where all latent and reconstruction space regularizers are independent of $\mathbf{c}$, i.e., do not contain conditioning on $\mathbf{c}$.

Semi-supervised training with latent space regularization and MSE reconstruction based on (3.12):

$$\mathcal{L}^{\text{LP}}_{\text{SS}-\text{AAE}}(\boldsymbol{\theta}_{\text{c}}, \boldsymbol{\theta}_{\text{x}}, \boldsymbol{\phi}_{\text{a}}, \boldsymbol{\phi}_{\text{z}}) = \mathcal{D}_{\text{z}} + \beta_{\text{x}}\mathcal{D}_{\text{x}\hat{\text{x}}} + \beta_{\text{c}}\mathcal{D}_{\text{c}\hat{\text{c}}} + \beta_{\text{c}}\mathcal{D}_{\text{c}}, \tag{3.16}$$

where $\mathcal{D}_{\text{z}} \triangleq D_{\text{KL}}\left(q_{\boldsymbol{\phi}_z}(\mathbf{z})\|p_{\boldsymbol{\theta}_z}(\mathbf{z})\right) = \mathbb{E}_{q_{\boldsymbol{\phi}_z}(\mathbf{z})}\left[\log \frac{q_{\boldsymbol{\phi}_z}(\mathbf{z})}{p_{\boldsymbol{\theta}_z}(\mathbf{z})}\right]$.

Semi-supervised training with latent space regularization and with MSE and adversarial reconstruction based on (3.12) deploys all terms:

$$\mathcal{L}^{\text{LP}}_{\text{SS}-\text{AAE}_{\text{complete}}}(\boldsymbol{\theta}_{\text{c}}, \boldsymbol{\theta}_{\text{x}}, \boldsymbol{\phi}_{\text{a}}, \boldsymbol{\phi}_{\text{z}}) = \mathcal{D}_{\text{z}} + \beta_{\text{x}}\mathcal{D}_{\text{x}\hat{\text{x}}} + \beta_{\text{x}}\mathcal{D}_{\text{x}} + \beta_{\text{c}}\mathcal{D}_{\text{c}\hat{\text{c}}} + \beta_{\text{c}}\mathcal{D}_{\text{c}}, \tag{3.17}$$

where $\mathcal{D}_{\text{x}} \triangleq D_{\text{KL}}\left(p_{\mathcal{D}}(\mathbf{x})\|p_{\boldsymbol{\theta}_x}(\mathbf{x})\right) = \mathbb{E}_{p_{\mathcal{D}}(\mathbf{x})}\left[\log \frac{p_{\mathcal{D}}(\mathbf{x})}{p_{\boldsymbol{\theta}_x}(\mathbf{x})}\right]$.

3.4.4 Links to state-of-the-art models

The considered HCP and LP models can be linked with several state-of-the-art unsupervised models such VAE [129, 130], β-VAE [131], AAE [33] and BIB-AE [123] and semi-supervised models such as AAE [33], CatGAN [34], VAE (M1+M2) [32] and SeGMA [126].

3.4.4.1 Links to unsupervised models

The proposed LP model (3.11) generalizes unsupervised models without the categorical latent representation. In addition, the unsupervised models in a form of the auto-encoder are used as a latent space regularizer in the LP setup. For these reasons, there are briefly considered four models of interest, namely, VAE, β-VAE, AAE and BIB-AE.

Before we proceed with the analysis, we define an unsupervised IB for these models. We assume the fused encoders $q_{\boldsymbol{\phi}_a}(\mathbf{a}|\mathbf{x})$ and $q_{\boldsymbol{\phi}_z}(\mathbf{z}|\mathbf{a})$ without conditioning on $\mathbf{c}$ in the inference model according to Fig. 3.2. We also assume no conditionally on $\mathbf{c}$ in the generative model.

The Lagrangian of unsupervised IB is defined according to [123]:

$$\mathcal{L}^{\text{U}_L}(\boldsymbol{\theta}_{\text{x}}, \boldsymbol{\phi}_{\text{z}}) = I_{\boldsymbol{\phi}_z}(\mathbf{X}; \mathbf{Z}) - \beta_{\text{x}} I_{\boldsymbol{\phi}_z, \boldsymbol{\theta}_x}(\mathbf{Z}; \mathbf{X}), \tag{3.18}$$

where similarly to the supervised counterpart (3.4), the first term is defined as:

$$\begin{aligned} I_{\boldsymbol{\phi}_z}(\mathbf{X}; \mathbf{Z}) &= \mathbb{E}_{q_{\boldsymbol{\phi}_z}(\mathbf{x},\mathbf{z})}\left[\log \frac{q_{\boldsymbol{\phi}_z}(\mathbf{x}, \mathbf{z})}{q_{\boldsymbol{\phi}_z}(\mathbf{z})p_{\mathcal{D}}(\mathbf{x})}\right] = \mathbb{E}_{q_{\boldsymbol{\phi}_z}(\mathbf{x},\mathbf{z})}\left[\log \frac{q_{\boldsymbol{\phi}_z}(\mathbf{z}|\mathbf{x})}{q_{\boldsymbol{\phi}_z}(\mathbf{z})}\frac{p_{\boldsymbol{\theta}_z}(\mathbf{z})}{p_{\boldsymbol{\theta}_z}(\mathbf{z})}\right] \\ &= \mathbb{E}_{p_{\mathcal{D}}(\mathbf{x})} \underbrace{\left[D_{\text{KL}}\left(q_{\boldsymbol{\phi}_z}(\mathbf{z}|\mathbf{X}=\mathbf{x})\|p_{\boldsymbol{\theta}_z}(\mathbf{z})\right)\right]}_{\mathcal{D}_{\text{z}|\text{x}}} - \underbrace{D_{\text{KL}}\left(q_{\boldsymbol{\phi}_z}(\mathbf{z})\|p_{\boldsymbol{\theta}_z}(\mathbf{z})\right)}_{\mathcal{D}_{\text{z}}}, \end{aligned} \tag{3.19}$$

and similarly to (3.14) the second term is defined as:

$$
\begin{aligned}
I_{\boldsymbol{\phi}_{\mathrm{z}},\boldsymbol{\theta}_{\mathrm{x}}}(\mathbf{Z};\mathbf{X}) &= \mathbb{E}_{p_{\mathcal{D}}(\mathbf{x})}\left[\mathbb{E}_{q_{\boldsymbol{\phi}_{\mathrm{z}}}(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_{\boldsymbol{\theta}_{\mathrm{x}}}(\mathbf{x}|\mathbf{z})}{p_{\mathcal{D}}(\mathbf{x})}\frac{p_{\boldsymbol{\theta}_{\mathrm{x}}}(\mathbf{x})}{p_{\boldsymbol{\theta}_{\mathrm{x}}}(\mathbf{x})}\right]\right] \\
&= H(p_{\mathcal{D}}(\mathbf{x}|\mathbf{c}); p_{\boldsymbol{\theta}_{\mathrm{x}}}(\mathbf{x})) - \underbrace{D_{\mathrm{KL}}\left(p_{\mathcal{D}}(\mathbf{x})\|p_{\boldsymbol{\theta}_{\mathrm{x}}}(\mathbf{x})\right)}_{\mathcal{D}_{\mathrm{x}}} - \underbrace{H_{\boldsymbol{\phi}_{\mathrm{z}},\boldsymbol{\theta}_{\mathrm{x}}}(\mathbf{X}|\mathbf{Z})}_{\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}},
\end{aligned}
\tag{3.20}
$$

where the definition of all terms should follow from the above equations.

Since $H(p_{\mathcal{D}}(\mathbf{x}|\mathbf{c}); p_{\boldsymbol{\theta}_{\mathrm{x}}}(\mathbf{x})) \geq 0$, it can be lower bounded $I_{\boldsymbol{\phi}_{\mathrm{z}},\boldsymbol{\theta}_{\mathrm{x}}}(\mathbf{Z};\mathbf{X}) \geq -\mathcal{D}_{\mathrm{x}} - \mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$.

Having defined the unsupervised IB variational bounded decomposition, one can proceed with an analysis of the related state-of-the-art methods along the lines of analysis introduced in Summary part of Section 2.

VAE [129, 130] and β**-VAE** [131]:

1. *The targeted tasks*: auto-encoding and generation.

2. *The architecture in terms of the latent space representation*: the encoder outputs two vectors representing the mean and standard deviation vectors that control a new latent representation $\mathbf{z} = f_{\boldsymbol{\phi}_{\mathrm{z}}}(\mathbf{x}) + \sigma_{\boldsymbol{\phi}_{\mathrm{z}}}(\mathbf{x}) \odot \boldsymbol{\epsilon}$, where $f_{\boldsymbol{\phi}_{\mathrm{z}}}(\mathbf{x})$ and $\sigma_{\boldsymbol{\phi}_{\mathrm{z}}}(\mathbf{x})$ are outputs of the encoder and $\boldsymbol{\epsilon}$ is assumed to be a zero mean unit variance Gaussian vector.

3. *The usage of IB or other underlying frameworks*: both VAE and β-VAE use evidence lower bound (ELBO) and are not derived from the IB framework. However, it can be shown [123] that the Lagrangian (3.18) can be reformulated for VAE and $\beta-$VAE as:

$$
\mathcal{L}_{\beta-\mathrm{VAE}}(\boldsymbol{\theta}_{\mathrm{x}}, \boldsymbol{\phi}_{\mathrm{z}}) = \mathbb{E}_{p_{\mathcal{D}}(\mathbf{x})}\Big[\mathcal{D}_{\mathrm{z}|\mathrm{x}}\Big] + \beta_{\mathrm{x}}\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}, \tag{3.21}
$$

   where $\beta_{\mathrm{x}} = 1$ for VAE. It can be noted that the VAE and β-VAE are based on an upper bound on the mutual information term $I_{\boldsymbol{\phi}_{\mathrm{z}}}(\mathbf{X};\mathbf{Z}) \leq \mathbb{E}_{p_{\mathcal{D}}(\mathbf{x})}\Big[\mathcal{D}_{\mathrm{z}|\mathrm{x}}\Big]$, since $D_{\mathrm{KL}}\left(q_{\boldsymbol{\phi}_{\mathrm{z}}}(\mathbf{z})\|p_{\boldsymbol{\theta}_{\mathrm{z}}}(\mathbf{z})\right) \geq 0$. Similar considerations apply to the second term since $D_{\mathrm{KL}}\left(p_{\mathcal{D}}(\mathbf{x})\|p_{\boldsymbol{\theta}_{\mathrm{x}}}(\mathbf{x})\right) \geq 0$.

4. *The label space regularization*: does not apply here due to the unsupervised setting.

5. *The latent space regularization*: is based on the hand-crafted prior with Gaussian pdf.

6. *The reconstruction space regularization in case of reconstruction loss*: is based on the mean square error (MSE) counterpart of $\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$ that corresponds to the Gaussian likelihood assumption.

Unsupervised AAE [33]:

1. *The targeted tasks*: auto-encoding and generation.

2. *The architecture in terms of the latent space representation*: the encoder outputs one vector in stochastic or deterministic way as $\mathbf{z} = f_{\boldsymbol{\phi}_\mathrm{z}}(\mathbf{x})$.

3. *The usage of IB or other underlying frameworks*: AAE is not derived from the IB framework. As shown in [123], the AAE equivalent Lagrangian (3.18) can be linked with the IB formulation and defined as:

$$\mathcal{L}_{\mathrm{AAE}}(\boldsymbol{\theta}_\mathrm{x}, \boldsymbol{\phi}_\mathrm{z}) = \mathcal{D}_\mathrm{z} + \beta_\mathrm{x}\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}, \tag{3.22}$$

   where $\beta_\mathrm{x} = 1$ in the original AAE formulation. It should be pointed out that the IB formulation of AAE contains the term $\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$, whose origin can be explained in the same way as for the VAE. Despite of the fact that the term $\mathcal{D}_\mathrm{z}$ indeed appears in (3.22) with the opposite sign, it cannot be interpreted either as an upper bound on $I_{\boldsymbol{\phi}_\mathrm{z}}(\mathbf{X}; \mathbf{Z})$ similarly to the VAE or as a lower bound. The goal of AAE is to minimize the reconstruction loss or to maximize the log-likelihood by ensuring that the latent space marginal distribution $q_{\boldsymbol{\phi}_\mathrm{z}}(\mathbf{z})$ matches the prior $p_{\boldsymbol{\theta}_\mathrm{z}}(\mathbf{z})$. The latter corresponds to the minimization of $D_{\mathrm{KL}}\left(q_{\boldsymbol{\phi}_\mathrm{z}}(\mathbf{z}) \| p_{\boldsymbol{\theta}_\mathrm{z}}(\mathbf{z})\right)$, i.e., $\mathcal{D}_\mathrm{z}$ term.

4. *The label space regularization*: does not apply here due to the unsupervised setting.

5. *The latent space regularization*: is based on the hand-crafted prior with zero mean unit variance Gaussian pdf for each dimension.

6. *The reconstruction space regularization in case of reconstruction loss*: is based on the MSE.

BIB-AE [123]:

1. *The targeted tasks*: auto-encoding and generation.

2. *The architecture in terms of the latent space representation*: the encoder outputs one vector using any form of stochastic or deterministic encoding.

3. *The usage of IB or other underlying frameworks*: the BIB-AE is derived from the unsupervised IB (3.18) and its Lagrangian is defined as:

$$\mathcal{L}_{\mathrm{BIB-AE}}(\boldsymbol{\theta}_\mathrm{x}, \boldsymbol{\phi}_\mathrm{z}) = \mathbb{E}_{p_\mathcal{D}(\mathbf{x})}\Big[\mathcal{D}_{\mathrm{z}|\mathrm{x}}\Big] - \mathcal{D}_\mathrm{z} + \beta_\mathrm{x}\mathcal{D}_\mathrm{x} + \beta_\mathrm{x}\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}. \tag{3.23}$$

4. *The label space regularization*: does not apply here due to the unsupervised setting.

5. *The latent space regularization*: is based on the hand-crafted prior with Gaussian pdf applied to both conditional and unconditional terms. In fact, the prior for $\mathcal{D}_\mathrm{z}$ can be any but $\mathcal{D}_{\mathrm{z}|\mathrm{x}}$ requires analytical parametrization.

6. *The reconstruction space regularization in case of reconstruction loss*: is based on the MSE counterpart of $\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$ and the discriminator $\mathcal{D}_{\mathrm{x}}$. This is a distinctive feature in comparison to VAE and AAE.

In summary, BIB-AE includes VAE and AAE as two particular cases. In turns, it should be clear that the regularizer of semi-supervised model considered in this work resembles the BIB-AE model and extends it to the conditional case that will be considered below.

3.4.4.2 Links to semi-supervised models

The proposed LP model (3.11) is also related to several state-of-the-art semi-supervised models used for the classification. As pointed out in the introduction, there are only considered available labeled and unlabeled samples in the analysis. The extension to the augmented samples, i.e., permutations, synthetically generated samples, i.e., fakes, and the adversarial examples for both latent space and label space regularizations can be performed along the line of analysis but it goes beyond the scope and focus of this work.

Semi-supervised AAE [33]:

1. *The targeted tasks*: auto-encoding, clustering, (conditional) generation and classification.

2. *The architecture in terms of the latent space representation*: the encoder outputs two vectors representing the discrete class and continuous type of style. The class distribution is assumed to follow categorical distribution and style Gaussian one. Both constraints on the prior distributions are ensured using adversarial framework with two corresponding discriminators. In its original setting, AAE does not use any augmented samples or adversarial examples.

   Remark: It should be pointed out that in the proposed architecture the latent space is represented by the vector $\mathbf{a}$, which is fed to the classifier and regularizer that gives a natural consideration of IB and corresponding regularization and priors. In the case of semi-supervised AAE, the latent space is considered by the class and style representations directly. Therefore, to make it coherent with the considered case, one should assume that the class vector of semi-supervised AAE corresponds to the vector $\mathbf{c}$ and the style vector to the vector $\mathbf{z}$.

3. *The usage of IB or other underlying frameworks*: AAE is not derived from the IB framework. However, as shown in the performed analysis the semi-supervised AAE represents the learnable prior case in part of latent space regularization. The corresponding Lagrangian of semi-supervised AAE is given by (3.16) and considered in Section 3.4.3.

4. *The label space regularization*: is based on the adversarial discriminator in assumption that the class labels follow categorical distribution. This is applied to both labeled and unlabeled samples.

5. *The latent space regularization*: is based on the learnable prior with Gaussian pdf of AE.

6. *The reconstruction space regularization in case of reconstruction loss*: is only based on the MSE.

CatGAN [34]: is based on an extension of classical GAN binary discriminator designed to distinguish between the original images and fake images generated from the latent space distribution to a multi-class discriminator. The author assumes the one-hot-vector encoding of class labels. The system is considered for the unsupervised and semi-supervised modes. For both modes the one-hot-vector encoding is used to encoded class labels. For the unsupervised mode, the system has an access only to the unlabeled data and the output of the classifier is considered as a clustering to a predefined number of clusters/classes. The main idea behind the unsupervised training consists in such a training of the discriminator that any sample from the set of original images is assigned to one of the classes with high fidelity whereas any fake or adversarial sample is assigned to all classes almost equiprobably. This corresponds to the fake samples and the regularization in the label space is based on the considered and extended framework of entropy minimization based regularization. In the case of absence of fakes, this regularization coincides with the semi-supervised AAE label space regularization under the categorical distribution and adversarial discriminator that is equivalent to enforcing the minimum entropy of label space. However, the encoding of fake samples is equivalent to a sort of rejection option expressed via the activation of classes that have maximum entropy or uniform distribution over the classes. Equivalently, the above types of encoding can be considered as the maximization of mutual information between the original data and encoded class labels and minimization of mutual information between the fakes/adversarial samples and the class labels. Semi-supervised CatGAN model adds a cross-entropy term computed for the true labeled samples.

Therefore, in summary:

1. *The targeted tasks*: auto-encoding, clustering, generation and classification.

2. *The architecture in terms of the latent space representation*: there is no encoder as such and instead the system has a generator/decoder that generates samples from a random latent space $\mathbf{a}$ following some hand-crafted prior. The second element of architecture is a classifier with the min/max entropy optimization for the original and fake samples. The encoding of classes is assumed to be a one-hot-vector encoding.

3. *The usage of IB or other underlying frameworks*: CatGAN is not derived from the IB framework. However, as shown in [123], one can apply the IB formulation to the adversarial generative models as in the case of CatGAN assuming that the term $I_{\boldsymbol{\phi}_{\mathrm{a}}}(\mathbf{X};\mathbf{A}) = 0$ in (3.3) due to the absence of encoder as such. The minimization problem

(3.3) reduces to the maximization of the second term $I_{\phi_a,\theta_c}(\mathbf{A};\mathbf{C})$ expressed via its lower bound of variational decomposition (3.6). The first term $\mathcal{D}_c$ enforces that the class labels of unlabeled samples follow the defined prior distribution $p(\mathbf{c})$ with the above property of entropy minimization under one-hot-vector encoding whereas the second term $\mathcal{D}_{c\hat{c}}$ reflects the supervised part for labeled samples. In the original CatGAN formulation, the author does not use the expression for the mutual information for the decoder/generator training as it is shown above but instead uses the decomposition of mutual information via the difference of corresponding entropies (see, first two terms in (9) in [34]). As it has been pointed out, in the analysis there is not included the term corresponding to the fake samples as in original CatGAN. However, it is obvious that this form of regularization does play an important role for the semi-supervised classification. The impact of this terms requires additional studies.

4. *The label space regularization*: is based on the above assumptions for labeled samples, which are included into the cross-entropy term, unlabeled samples included into the entropy minimization term and fake samples included into the entropy maximization term in the original CatGAN method.

5. *The latent space regularization*: is based on the hand-crafted prior.

6. *The reconstruction space regularization in case of reconstruction loss*: is based on the adversarial discriminator only.

SeGMA [126]: is a semi-supervised clustering and generative system with a single latent vector representation auto-encoder similar in spirit to the unsupervised version of AAE that can be also used for the classification. The latent space of SeGMA is assumed to follow a mixture of Gaussians. Using a small labeled dataset, classes are assigned to components of this mixture of Gaussians by minimizing the cross-entropy loss induced by the class posterior distribution of a simple Gaussian classifier. The resulting mixture describes the distribution of the whole data, and representatives of individual classes are generated by sampling from its components. In the classification setup, SeGMA uses the latent space clustering scheme for the classification.

Therefore, in summary:

1. *The targeted tasks*: auto-encoding, clustering, generation and classification.

2. *The architecture in terms of the latent space representation*: a single vector representation following mixture of Gaussians distribution.

3. *The usage of IB or other underlying frameworks*: SeGMA is not derived from the IB framework but a link to the regularized ELBO an other related auto-encoders with interpretable latent space is demonstrated. However, as in previous methods it can

be linked to the considered IB interpretation of the semi-supervised methods with hand-crafted priors (3.16). An equivalent Lagrangian of SeGMA is:

$$\mathcal{L}_{\text{SeGMA}}(\boldsymbol{\theta}_{\text{c}}, \boldsymbol{\theta}_{\text{x}}, \boldsymbol{\phi}_{\text{z}}) = \mathcal{D}_{\text{z}} + \beta_{\text{x}}\mathcal{D}_{\text{x}\hat{\text{x}}} + \beta_{\text{c}}\mathcal{D}_{\text{c}\hat{\text{c}}}, \tag{3.24}$$

where the latent space discriminator $\mathcal{D}_{\text{z}}$ is assumed to be the maximum mean discrepancy (MMD) penalty that is analytically defined for the mixture of Gaussians pdf, $\mathcal{D}_{\text{x}\hat{\text{x}}}$ is represented by the MSE and $\mathcal{D}_{\text{c}\hat{\text{c}}}$ represents the cross-entropy for the labeled data defined over class labels deduced from the latent space representation.

4. *The label space regularization*: is based on the above assumptions for labeled samples, which are included into the cross-entropy term as discussed above.

5. *The latent space regularization*: is based on the hand-crafted mixture of Gaussians pdf.

6. *The reconstruction space regularization in case of reconstruction loss*: is based on the MSE.

VAE (M1+M2) [32]: is based on the combination of several models. The model M1 represents a vanilla VAE considered in section 3.4.4.1. Therefore, the model M1 is a particular case of considered unsupervised IB. The model M2 is a combination of encoder producing a continuous latent representation and following Gaussian distribution and a classifier that takes as an input original data in parallel to the model M1. The class labels are encoded using the one-hot-vector representations and follow categorical distribution with a hyper-parameter following the symmetric Dirichlet distribution. The decoder of model M2 takes as an input the continuous latent representation and output of classifier. The decoder is trained under the MSE distortion metric. It is important to point out that the classifier works with the input data directly but not with the common latent space like in the considered LP model. For this reason, it is an obvious analogy with the considered LP model (3.11) under the assumption that $\mathbf{a} = \mathbf{x}$ and all performed IB analysis directly applies to. However, as pointed by the authors, the performance of model M2 in the semi-supervised classification for the limited number of labeled samples is relatively poor. That is why the third hybrid model M1+M2 is considered when the models M1 and M2 are used in a stacked way. At the first stage, the model M1 is learned as the usual VAE. Then the latent space of model M1 is used as an input to the model M2 trained in a semi-supervised way. Such a two-stage approach closely resembles the learnable prior architecture presented in Fig. 3.2. However, the presented model is end-to-end trained with the explainable common latent space and IB origin, while the model M1+M2 is trained in two stages with the use of regularized ELBO for the derivation of model M2.

1. *The targeted tasks*: auto-encoding, clustering, (conditional) generation and classification.

2. *The architecture in terms of the latent space representation*: the stacked combination of models M1 and M2 is used as discussed above.

3. *The usage of IB or other underlying frameworks*: VAE M1+M2 is not derived from the IB framework but it is linked to the regularized ELBO with the cross-entropy for the labeled samples. The corresponding IB Lagrangian of semi-supervised VAE M1+M2 under the assumption of end-to-end training can be defined as:

$$\mathcal{L}^{\text{LP}}_{\text{SS}-\text{VAE M1+M2}}(\boldsymbol{\theta}_{\text{c}}, \boldsymbol{\theta}_{\text{x}}, \boldsymbol{\phi}_{\text{a}}, \boldsymbol{\phi}_{\text{z}}) = \mathbb{E}_{p_{\mathcal{D}}(\mathbf{x})}\Big[\mathcal{D}_{\text{z}|\text{x}}\Big] + \beta_{\text{x}}\mathcal{D}_{\text{x}\hat{\text{x}}} + \beta_{\text{c}}\mathcal{D}_{\text{c}\hat{\text{c}}} + \beta_{\text{c}}\mathcal{D}_{\text{c}}. \tag{3.25}$$

4. *The label space regularization*: is based on the assumption of categorical distribution of labels.

5. *The reconstruction space regularization in case of reconstruction loss*: is only based on the MSE.

3.5 Experimental results

3.5.1 Experimental setup

The tested system is based on (i) the deterministic encoder and decoder and (ii) the stochastic encoder of type $\mathbf{a} = f_{\boldsymbol{\phi}_{\text{a}}}(\mathbf{x} + \boldsymbol{\epsilon})$ with the data independent perturbations $\boldsymbol{\epsilon}$ and deterministic decoder for both cases. The density ratio estimator [132] is used to measure all KL-divergences. The results of semi-supervised classification on the MNIST dataset are reported in Table 3.1, where symbol D indicates the deterministic setup and symbol S corresponds to the stochastic one. To choose the optimal parameters of systems, e.g., the Lagrangian multipliers in the considered models, a 3-run cross-validation with the randomly chosen labeled examples is run as shown in Appendices C.1 - C.6. Once the model parameters are chosen, a 10 time cross-validation is run and the average results are shown in Table 3.1.

Additionally, a 10-run cross-validation is performed on the SVHN dataset [115]. The same architecture is used as for MNIST with the same encoders, decoders and discriminators. In contrast to VAE M1+M2, the normalized raw data without any pre-processing was used. Additionally, in contrast to AAE, where an extra set of 531131 unlabeled images is used for the semi-supervised training, in performed experiments only a train set of 73257 images is used for training. Moreover, the experiments are performed: (i) for the optimal parameters chosen after 3-run cross-validation for the MNIST dataset with no special adaption to SVHN dataset and (ii) under the network architectures with exactly the same number of used filters as given in Appendices C.1 - C.6 for the MNIST dataset. In summary, the main goal is to test the generalization capacity of the proposed approach but not just to achieve the best performance by fine-tuning of network parameters. The obtained results are represented in Table 3.1.

Table 3.1 Semi-supervised classification error (%) for the optimal parameters (Appendix C.1 - C.6) defined on the MNIST dataset (D - deterministic; S - stochastic).

		MNIST (100)	MNIST (1000)	MNIST (all)	SVHN (1000)
NN Baseline ($\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}}$)	$[D]$	26.31 ($\pm$0.91)	7.50 ($\pm$0.19)	0.68 ($\pm$0.05)	36.16 ($\pm$0.77)
	$[S]$	26.78 ($\pm$1.66)	7.54 ($\pm$0.25)	0.70 ($\pm$0.05)	36.28 ($\pm$0.93)
InfoMax [34]	$[S]$	33.41	21.5	15.86	-
VAE [36]	$[S]$	14.26	8.71	5.02	-
MV-InfoMax [36]	$[S]$	13.22	7.39	6.07	-
IB multiview [36]	$[S]$	3.03	2.34	2.22	-
VAE (M1 + M2) [36]	$[S]$	3.33 ($\pm$0.14)	2.40 ($\pm$0.02)	0.96	36.02 ($\pm$0.10)
CatGAN	$[S]$	1.91 ($\pm$0.10)	1.73 ($\pm$0.18)	0.91	-
AAE	$[D]$	1.90 ($\pm$0.10)	1.60 ($\pm$0.08)	0.85 ($\pm$0.02)	17.70 ($\pm$0.30)
No priors on latent space					
$\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}} + \mathcal{D}_{\mathrm{c}}$	$[D]$	20.72 ($\pm$1.58)	4.99 ($\pm$0.28)	0.69 ($\pm$0.04)	25.78 ($\pm$0.90)
	$[S]$	19.60 ($\pm$1.37)	4.49 ($\pm$0.25)	0.67 ($\pm$0.05)	26.34 ($\pm$0.80)
Hand-crafted latent space priors					
$\beta_{\mathrm{c}}\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}} + \mathcal{D}_{\mathrm{a}}$	$[D]$	27.44 ($\pm$1.40)	6.77 ($\pm$0.34)	0.91 ($\pm$0.05)	35.94 ($\pm$1.08)
	$[S]$	27.48 ($\pm$1.07)	6.91 ($\pm$0.45)	0.88 ($\pm$0.05)	35.80 ($\pm$1.21)
$\beta_{\mathrm{c}}\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}} + \mathcal{D}_{\mathrm{a}} + \beta_{\mathrm{c}}\mathcal{D}_{\mathrm{c}}$	$[D]$	12.04 ($\pm$4.46)	2.43 ($\pm$0.12)	0.81 ($\pm$0.05)	24.70 ($\pm$0.46)
	$[S]$	11.80 ($\pm$3.82)	2.40 ($\pm$0.10)	0.82 ($\pm$0.04)	24.62 ($\pm$0.54)
Learnable latent space priors					
$\beta_{\mathrm{c}}\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}} + \beta_{\mathrm{c}}\mathcal{D}_{\mathrm{c}} + \mathcal{D}_{\mathrm{z}} + \beta_{\mathrm{x}}\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$	$[D]$	1.55 ($\pm$0.21)	1.25 ($\pm$0.10)	0.74 ($\pm$0.04)	20.07 ($\pm$0.36)
	$[S]$	1.49 ($\pm$0.18)	1.43 ($\pm$0.06)	0.78 ($\pm$0.04)	20.00 ($\pm$0.31)
$\beta_{\mathrm{c}}\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}} + \beta_{\mathrm{c}}\mathcal{D}_{\mathrm{c}} + \mathcal{D}_{\mathrm{z}} + \beta_{\mathrm{x}}\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$	$[D]$	1.38 ($\pm$0.09)	1.21 ($\pm$0.10)	0.77 ($\pm$0.06)	19.75 ($\pm$0.52)
$+\beta_{\mathrm{x}}\mathcal{D}_{\mathrm{x}}$	$[S]$	1.42 ($\pm$0.10)	1.16 ($\pm$0.09)	0.79 ($\pm$0.02)	19.71 ($\pm$0.26)

The considered architectures are compared with several state-of-the-art semi-supervised methods such as AAE [33], CatGAN [34], VAE (M1+M2) [32], IB multiview [36], MV-InfoMax [36] and InfoMax [34] with 100, 1000 and 60000 training labeled samples. The expected training times for the considered models are given in Table 3.2[2].

3.5.2 Discussion MNIST

The deterministic and stochastic systems based on the learnable priors clearly demonstrate the state-of-the-art performance in comparison to the considered semi-supervised counterparts.

Baseline Neural Network (NN): the obtained results allow to conclude that, if the amount of labeled training data is large, as shown in "all" column (Table 3.1), the latent space regularization has no practically significant impact on the classification performance for both hand-crafted and learnable priors. The deep classifier is capable to learn a latent representation retaining only sufficient statistics in the latent space solely based on the cross-entropy component of IB classification term decomposition as shown in Fig. 3.3, row

[2]The source code is available at https://github.com/taranO/IB-semi-supervised-classification.

$\mathcal{D}_{c\hat{c}}$ and column "all". The classes appear to be well separable under this form of visualization. At the same time, the decrease of number of labeled samples leads to the degradation of classification accuracy as show in Table 3.1 for columns "1000" and "100". This degradation is also clearly observed in Fig. 3.3, row $\mathcal{D}_{c\hat{c}}$ and column "100", where one can observe a larger overlap between the classes compared to the column "all". The stochastic encoding via the addition of noise to the input samples does not enhance the performance with respect to the deterministic decoding for the small amount of labeled examples. One can assume that the presence of additive noise is not typical for the considered data, whereas the samples clearly differ in the geometrical appearance. Therefore, one can only assume that random geometrical permutations would be a more interesting alternative to the additive noise permutations/encoding.

No priors on latent space: to investigate the impact of unlabeled data, the adversarial regularizer $\mathcal{D}_c$ is added to the baseline classifier based on $\mathcal{D}_{c\hat{c}}$. The term $\mathcal{D}_c$ enforces the distribution of class labels for the unlabeled samples to follow the categorical distribution. At this stage, no regularization of latent space is applied. The addition of the adversarial regularizer $\mathcal{D}_c$, see "100" column (Table 3.1), allows to reduce the classification error in comparison to the baseline classifier. Moreover, the stochastic encoder slightly outperforms the deterministic one for all numbers of labeled samples. However, the achieved classification error is far away from the performance of baseline classifier trained on the whole labeled dataset. Thus, the cross-entropy and adversarial classification terms alone can hardly cope with the lack of labeled data and proper regularization of the latent space is the main mechanism capable to retain the most relevant representation.

Hand-crafted latent space priors: along this line there is investigated the impact of hand-crafted regularization in the form of the added discriminator $\mathcal{D}_a$ imposing Gaussian prior on the latent representation $\mathbf{a}$. The sole regularization of latent space with the hand-crafted prior on the Gaussianity does not reflect the complex nature of latent space of real data. As a result the performance of the regularized classifier $\beta_c\mathcal{D}_{c\hat{c}} + \mathcal{D}_a$ does not lead to a remarkable improvement in comparison to the non-regularized counterpart $\mathcal{D}_{c\hat{c}}$ for both stochastic and deterministic types of encoding. When in addition the label space regularization $\mathcal{D}_c$ is added to the final classifier $\beta_c\mathcal{D}_{c\hat{c}} + \mathcal{D}_a + \beta_c\mathcal{D}_c$, it leads to the factor of 2 classification error reduction over the cross-entropy baseline classifier but it is still far away from the fully supervised baseline classifier trained on the fully labeled dataset. At the same time, there is no significant difference between the stochastic and deterministic types of encoding.

Learnable latent space priors: along this line there is investigated the impact of learnable priors by adding the corresponding regularizations of the latent space of auto-encoder and data reconstruction. The role of reconstruction space regularization based on the MSE expressed via $\mathcal{D}_{x\hat{x}}$ and joint $\mathcal{D}_{x\hat{x}}$ and $\mathcal{D}_x$ is investigated. The addition of discriminator $\mathcal{D}_x$ slightly enhances the classification but requires almost doubled training time as shown in Table 3.2. The stochastic encoding does not show any obvious advantage over the deterministic one in

Table 3.2 Execution time (hours) per 100 epochs on one NVIDIA GPU. For the SVHN the models with the learnable latent space priors are trained with a learning rate 1e-4 that explains the longer time but without optimization of Lagrangians, i.e., the Lagrangians are re-used from pre-trained MNIST model. All the others models are trained with a learning rate 1e-3.

	MNIST	SVHN
NN Baseline ($\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}}$)	0.47 - 0.65	0.85 - 0.92
No priors on latent space		
$\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}} + \mathcal{D}_{\mathrm{c}}$	0.47 - 0.65	0.85 - 0.92
Hand-crafted latent space priors		
$\beta_{\mathrm{c}}\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}} + \mathcal{D}_{\mathrm{a}}$	0.47 - 0.65	1 - 1.05
$\beta_{\mathrm{c}}\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}} + \mathcal{D}_{\mathrm{a}} + \beta_{\mathrm{c}}\mathcal{D}_{\mathrm{c}}$	0.97 - 1.18	1.5 - 1.6
Learnable latent space priors		
$\beta_{\mathrm{c}}\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}} + \beta_{\mathrm{c}}\mathcal{D}_{\mathrm{c}} + \mathcal{D}_{\mathrm{z}} + \beta_{\mathrm{x}}\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$	1.23 - 1.6	2.25 - 2.3
$\beta_{\mathrm{c}}\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}} + \beta_{\mathrm{c}}\mathcal{D}_{\mathrm{c}} + \mathcal{D}_{\mathrm{z}} + \beta_{\mathrm{x}}\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}} + \beta_{\mathrm{x}}\mathcal{D}_{\mathrm{x}}$	1.98 - 2.42	3.5 - 3.55

this setup. The separability of classes shown in Fig. 3.3, row $\beta_{\mathrm{c}}\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}}+\beta_{\mathrm{c}}\mathcal{D}_{\mathrm{c}}+\mathcal{D}_{\mathrm{z}}+\beta_{\mathrm{x}}\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}+\beta_{\mathrm{x}}\mathcal{D}_{\mathrm{x}}$ and column "100"[3], is very close to those of column "all" and row $\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}}$, i.e., the semi-supervised system with 100 labeled examples is capable to closely approximate the fully supervised one. However, it should be pointed out that the learnable priors ensures the reconstruction of data from the compressed latent space and the learned representation is the sufficient statistics for the data reconstruction task but not for the classification one. Since the entropy of the classification task is significantly lower to those of reconstruction, such a learned representation contains more information than actually needed for the classification task. A fraction of retained information is irrelevant to the classification problem and might be a potential source of classification errors. This likely explains a gap in performance between the considered semi-supervised system and fully supervised one.

3.5.3 Latent space of trained models

To better understand the properties of classifier's latent space for both the hand-crafted and learnable priors under different amount of training samples, Fig. 3.3 and 3.4 show t-sne plots for the perplexity 30 for 100, 1000 and 60000 ("all") training labels of the MNIST dataset.

The first raw of Fig. 3.3 with the label "$\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}}$" corresponds to the classifier considered in Appendix C.1. The latent space $\mathbf{a}$ of the classifier with "all" labels demonstrates the perfect separability of classes. The classes are far away from each other and there are practically no outliers leading to the misclassification. The decrease of the number of labels in the

[3]The t-sne is shown only for this setup since it practically coincides with $\beta_{\mathrm{c}}\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}} + \beta_{\mathrm{c}}\mathcal{D}_{\mathrm{c}} + \mathcal{D}_{\mathrm{z}} + \beta_{\mathrm{x}}\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$.

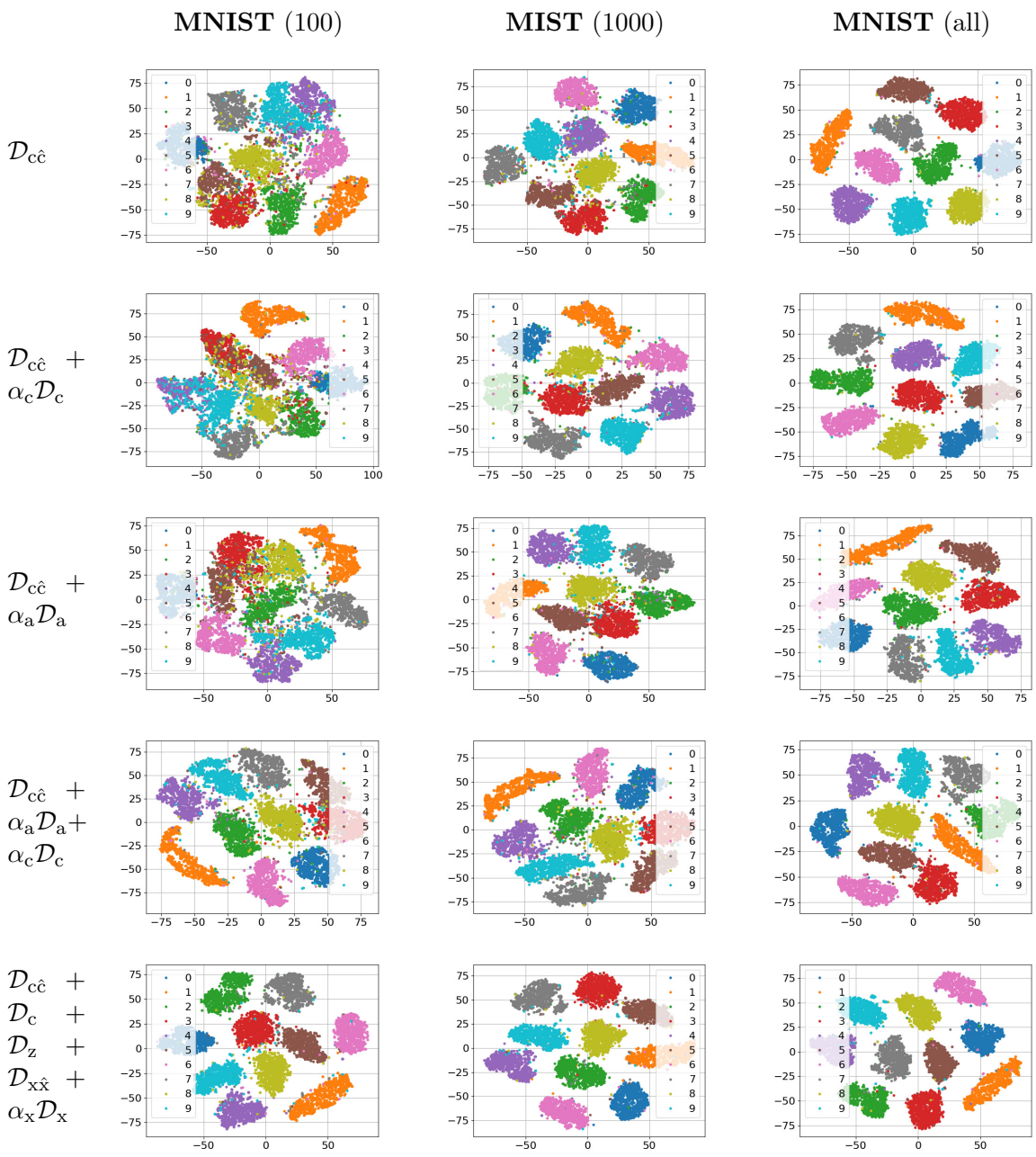


Fig. 3.3 Latent space **a** (of size 1024) of classifier.

supervised setup, see the columns 1000 and 100, leads to a visible degradation of separability between the classes.

The regularization of class label space by the regularizer $\mathcal{D}_c$ or by the hand-crafted latent space regularizer $\mathcal{D}_a$ shown in raws "$\mathcal{D}_{c\hat{c}} + \alpha_c\mathcal{D}_c$" considered in Appendix C.2 and "$\mathcal{D}_{c\hat{c}} + \alpha_a\mathcal{D}_a$" considered in Appendix C.3 for the small number of training samples equal 100 does not significantly enhance the class separability with respect to "$\mathcal{D}_{c\hat{c}}$".

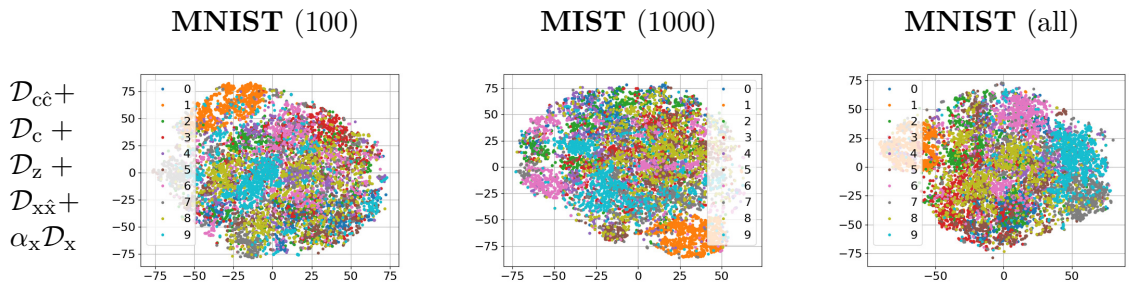


Fig. 3.4 Latent space $\mathbf{z}$ (of size 20) of auto-encoder.

At the same time, the joint usage of the above regularizers according to the model "$\mathcal{D}_{c\hat{c}} + \alpha_c\mathcal{D}_c + \alpha_a\mathcal{D}_a$" according to the model in Appendix C.4 leads to the better separability of classes for 100 labels in comparison with the previous cases. At the same time, the addition of these regularizers does not have any impact on the latent space for "all" label case.

The introduction of learnable regularization of latent space along with the class label regularization according to the model "$\mathcal{D}_{c\hat{c}} + \mathcal{D}_c + \mathcal{D}_z + \mathcal{D}_{x\hat{x}} + \alpha_x\mathcal{D}_x$" considered in Appendix C.6 enhances the class separability in the latent space of classifier for 100 label case that is also very close to the fully supervised case.

For the comparison reasons, the latent space of the auto-encoder $\mathbf{z}$ for the above model is visualized in Fig. 3.4.

3.5.4 Discussion SVHN

In the SVHN test, the Lagrangian coefficients are not optimized as it is done for MNIST. However, to compensate a potential non-optimality, the model training is performed with the reduced learning rate as indicated in Table 3.2. As a result, the training time on the SVHN dataset is longer. Therefore, 10-run validation of the proposed framework on the SVHN dataset is done with the optimal Lagrangian multipliers determined on the MNIST dataset. In this respect, one might observe a small degradation of the obtained results comparing to the state-of-the-art. Additionally, any pre-processing like PCA that is used in VAE M1+M2 is not applied and the extended unlabeled dataset is not used as it is done in case of AAE. One can clearly observe the same behavior of semi-supervised classifiers as for MNIST dataset discussed in Section 3.5.2. Therefore, it can be clearly confirmed the role of learnable priors in the overall performance observed for both datasets.

3.6 Conclusions

We introduce a novel formulation of variational information bottleneck for semi-supervised classification. To overcome the problem of original bottleneck and to compensate the lack of labeled data in the semi-supervised setting, two models of latent space regularization

via hand-crafted and learnable priors are considered. We investigate how the parameters of proposed framework influence the performance of classifier on a toy example of MNIST dataset. By end-to-end training, we demonstrate how the proposed framework compares to the state-of-the-art methods and approaches the performance of fully supervised classifier.

The envisioned future work is along the line to provide a stronger compression yet preserving only classification task relevant information since retaining more task irrelevant information does not provide distinguishable classification features, i.e., it only ensures reliable data reconstruction. In this work, we consider IB for the predictive latent space model. The contrastive multi-view IB formulation would be an interesting candidate for the regularization of latent space. Additionally, the adversarially generated examples are not used to impose the constraint on the minimization of mutual information between them and class labels or equivalently to maximize the entropy of class label distribution for these adversarial examples according to the framework of entropy minimization. This line of "adversarial" regularization seems to be a very interesting complement to the considered variational bottleneck. We consider a particular form of stochastic encoding by the addition of data independent noise to the input with the preservation of the same class labels. This also corresponds to the consistency regularization when samples can be more generally permuted including the geometrical transformations. It is also interesting to point out that the same form of generic permutations is used in the unsupervised contrastive loss based multi-view formulations for the continual latent space representation as opposed to the categorical one in the consistency regularization. Finally, the conditional generation can be an interesting line of research considering the generation from discrete labels and continuous latent space of the autoencoder.

Chapter 4

Copy detection patterns

This Chapter is dedicated to the investigation of the authentication and clonability aspects of PGC from the perspective of the hand-crafted and machine learning copy attacks. It is important to emphasise that this work does not target to investigate the clonability and authentication aspects of some particular PGC but rather to demonstrate a general approach applicable to the majority of PGC designed with identical modulation principles. In this respect the *DataMatrix* codes based on the international standard ISO/IEC 16022 [21] are used. Although the *DataMatrix* codes were initially proposed as an overt feature for personalization applications, the chosen parameters of these codes closely resemble those of the recently proposed copy detection patterns. The results presented in this Chapter have been partially published in [133].

Notations

We use the following notations. $\mathbf{t} \in \{0,1\}^{m \times m}$ denotes an original digital templates. If the authentication is performed based on the physical template, i.e., an image acquired from a printed object, we refer to it as $\mathbf{t} \in \mathbb{R}^{m \times m}$. $\mathbf{x} \in \mathbb{R}^{m \times m}$ corresponds to the original printed codes, while $\mathbf{f} \in \mathbb{R}^{m \times m}$ is used to denote fake printed codes. $\mathbf{y} \in \mathbb{R}^{m \times m}$ stands for a probe that might be either original or fake. $T_{Otsu}(.)$ denotes the thresholding method based on the threshold determined via Otsu's method [2]. $p_{\mathcal{D}}(\mathbf{t})$ and $p_{\mathcal{D}}(\mathbf{x})$ correspond to the empirical data distributions of the reference templates and original printed codes correspondingly. The discriminators corresponding to Kullback–Leibler divergences are denoted as $\mathcal{D}_{\mathrm{x}}$, where the subscript indicates the space to which this discriminator is applied to. The cross-entropy metrics are denoted as $\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$, where the subscript indicates the corresponding vectors.

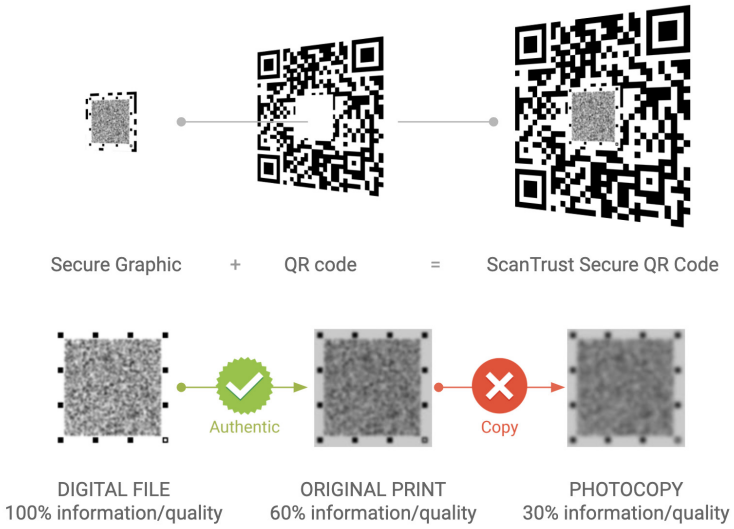


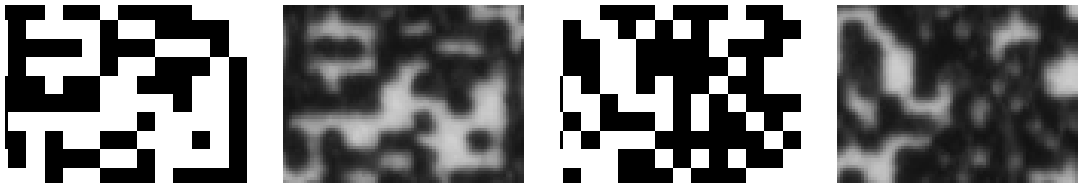
Fig. 4.1 ScanTrust secure QR code[1].

4.1 Introduction

In the modern world of globalised distributed economy it is extremely challenging to ensure a proper production, shipment, trade distribution, consumption and recycling of various products and goods of physical world. These products and goods range from everyday food to some luxury objects and art. Creation of twins of these objects with appropriate track and trace infrastructures complemented by graphical tools like blockchain represents an attractive option. However, it is very important to provide a robust, secure and unclonable link between a physical object and its digital representation in centralized or distributed databases. This link might be implemented via overt channels, like personalized codes reproduced on products either directly or in a form of coded symbologies like 1D and 2D codes or covert channels, like invisible digital watermarks embedded in images or text or printed by special invisible inks. However, many codes of this group are easily copied or can be regenerated. Thus, there is a great need in unclonable modalities that can be easily integrated with the codes. This necessity triggered the appearance and growing popularity of PGC. The PGC belongs to a family of *copy detection patterns* often "injected" into traditional 2D codes. During the last decade, the PGC attracted many industrial players and governmental organizations.

The copy detection patterns used in PGC are based on a so-called information loss principle: each time the code is printed or scanned some information about the original digital template is inevitably lost. As an example, in [134] the author proposes to use a maximum entropy image based on a secret key as a digital signature. This idea formed a basis for a so-named ScanTrust 2D bar codes as illustrated in Fig. 4.1. That codes consist

[1]https://www.weforum.org/agenda/2020/06/counterfeiters-pandemic-how-to-stop-them



(a) Digital. (b) Printed. (c) Digital. (d) Printed.

Fig. 4.2 Examples of the dot gain effect: (a) - (b) a black symbol surrounded by white symbols increases its size but remains well detectable; (c) - (d) a white symbol surrounded by black symbols might disappear under strong dot gain.

of the primary information that can be read by any 2D bar code reader and the secondary information that is made difficult to reproduce without alteration in a visible pattern [135].

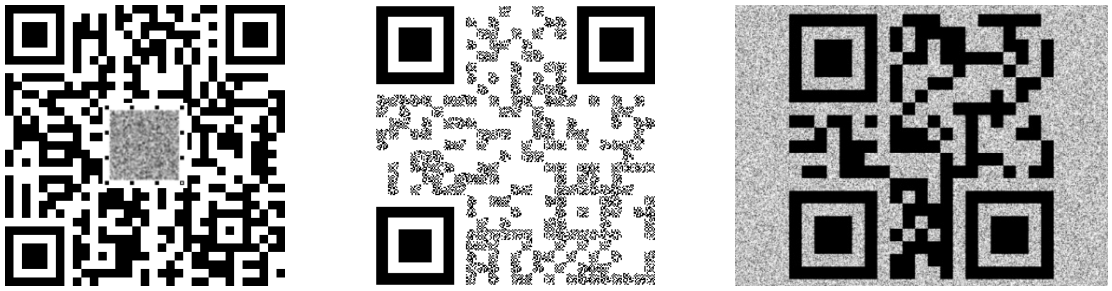
In the case of printable codes, the information loss principle is based on physical phenomena of random interaction between the ink or toner with a substrate. As a result any black dot undergoes a complex unpredictable modification and changes its shape accordingly to a dot gain effect. Generally, the black dot increases in size. Thus, the origin of its name. A white whole on a black background accordingly decreases its area due to the dot gain of nearest black dot surround as shown in Fig. 4.2.

In the case of image acquisition the information loss principle refers to a loss of image quality due to various factors that include variability of illumination, finite and discrete nature of sampling in CCD/CMOS sensors, non-linearity in sensor sensitivity, sensor noise and various sensor defects, etc. All together, the enrolled image is characterized by some variability that degrades the quality of image in terms of its correspondence to the original digital template $\mathbf{t}$ from which the code was printed.

Nowadays, there exist a big variety of different approaches aiming to combine the copy detection patterns and widely used traditional 2D codes. Without pretending to be exhaustive in the presented overview, some of the most representative approaches are mentioned below.

In general, it is possible to distinguish the standard one-level PGC and more advanced multi-level PGC. Examples of these codes are given in Fig. 4.3. Originally the multi-level PGC aimed at increasing the storage capacity of the regular PGC [136]. Recently, the multi-level PGC are considered as a tool to increase the security of standard PGC. Without loss of generality, it is possible to identify the multi-level PGC with a modulation of the main black symbols as shown in Fig. 4.3(b) and a background modulation as illustrated in Fig. 4.3(c).

The most well known multi-level PGC of the first type are so-called two level QR (2LQR) codes proposed in [140, 138], where the standard black modules are substituted by special modulated patterns. The general principles of modulation of multi-level codes were initially considered and theoretically analysed in [141]. The public level of this code is read as normal standard QR code. The texture patterns are chosen to be sensitive to the print & scan process.



(a) One-level PGC [137]. (b) Two level 2LQR code [138]. (c) Two level W-QR code [139].

Fig. 4.3 Examples of different types of PGC.

At the same time, the modulation pattern can carry out private message. Furthermore, the idea of 2LQR was extended in [142] by the use of different encrypting strategies. The anti-counterfeiting performance of these codes was mainly tested based on desktop printers and scanners. Thus, there is a great interest in validation of these codes under the industrial printing and mobile phone authentication.

The second type of multi-level PGC is so-called W-QR codes proposed in [139], where the authors substitute the background of a standard QR code by a specific random texture. The embedded texture does not affect the readability of the standard code but it should be sensitive to the print & scan process in such a way to give a possibility to authenticate the original code from the counterpart. The authors propose a particular random textured pattern, which has a stable statistical behavior. Thus, the attacker targets to estimate the parameters of the used textured pattern.

An interesting extension of the concept of multi-level PGC is represented in [1], where the authors propose a two-layer QR code, in which two messages are stored and can be retrieved separately from two different viewing directions as shown in Fig. 4.4. The proposed codes aim at increasing the storage capacity but, for the moment, do not have any protections against illegal copying.

Despite the differences in ways how the traditional QR codes and copy detection patterns are combined, in general case, the authentication of digital artwork based on the copy detection patterns is done by comparing the reference template with the printed version scanned using a scanner or camera of mobile phone. As a reference template there can be used either a digital template or printed version of the same artwork. The comparison can be done in different ways either in the spatial or frequency domain using a correlation, distance metrics or a combined score of different features, etc., [134, 26]. Alternatively, one can also envision an authentication in a transform domain using latent space of pretrained classifiers or auto-encoders.

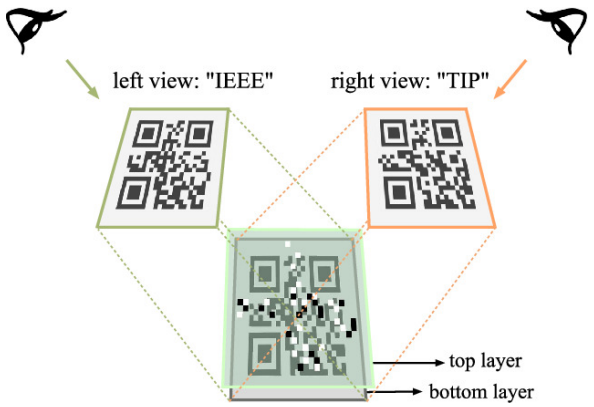


Fig. 4.4 An example of a two-layer QR code that consists of a top and a bottom layers [1].

4.2 PGC datasets

The majority of the research experiments in the domain of PGC are performed either on synthetic data or on small private datasets. The production of datasets of real PGC is a costly and timely process. It requires the printing and acquisition of the original PGC and the production and acquisition of the counterfeits preferably on the equipment close to the industrial.

Up to our best knowledge, there are only few public datasets created to investigate the clonability aspects of copy detection patterns:

- *DP0E*

  In [22] the authors reported the DP0E[2] dataset of real and counterfeited PGC based on *DataMatrix* modulation produced with four printers and enrolled by a high resolution scanner. As it is indicated in Table 4.1, this dataset contains 3456 codes among which 384 digital templates with symbol size 6×6, 1536 printed original codes and 1536 fake codes printed on the same printers as original codes.

- *CSGC*

  The CSGC dataset[3] announced in [23] consists of 3800 codes among which 950 digital templates with symbol size 1×1 and 2850 original codes printed on one printer and scanned by one scanner but under three different resolutions as indicated in Table 4.1. In their work, the authors say that the produced fakes are reprinted and are taken into account in the authentication tests. However, the produced fakes are publicly unavailable.

In current work to investigate the authentication and clonability aspect of the PGC the three datasets are created:

[2]http://sip.unige.ch/projects/snf-it-dis/datasets/dp0e
[3]https://www.univ-st-etienne.fr/graphical-code-estimation/index.html

Table 4.1 An overview of the datasets of PGC: *(i)* publicly available state-of-the-art and *(ii)* created and investigated in the current Thesis. The "original" and "fakes" correspond to the printed and enrolled on the corresponding equipment codes.

Name	Digital templates	Printing	Acquisition	# of codes
Publicly available state-of-the-art datasets				
DP0E [22]	size: 384×384 symbol size: 6×6	*Laser*, at 1200 dpi: · Samsung Xpress 430 · Lexmark CS310 *Inkjet*, at 1200 dpi: · Canon PIXMA iP7200 · HP OfficeJet Pro 8210	*Scanner*: · Epson V850 Pro at 1200 ppi	digital: 384 original: 1536 fakes: 1536 total: 3456
CSGC [23]	size: 100×100 symbol size: 1×1	*Laser*, at 600 dpi: · Xerox Phaser 6500	*Scanner*: · Epson V850 Pro at 2400 ppi at 4800 ppi at 9600 ppi	digital: 950 original: 2850 fakes: 0 total: 3800
Datasets created in this Thesis				
DP1C & DP1E	size: 384×384 symbol size: 6×6	*Laser*, at 1200 dpi: · Samsung Xpress 430 · Lexmark CS310 *Inkjet*, at 1200 dpi: · Canon PIXMA iP7200 · HP OfficeJet Pro 8210	*Scanner*: · Canon 9000F at 1200 ppi · Epson V850 Pro at 1200 ppi	digital: 384 original: 3072 fakes: 3072 total: 6528
Indigo mobile	size: 330×330 symbol size: 5×5	*Industrial*, at 812 dpi: · HP Indigo 5500 DS	*Mobile phone*: · iPhone XS auto settings	digital: 300 original: 300 fakes: 1200 total: 1800
Indigo scanner	size: 330×330 symbol size: · 5×5 · 4×4 · 3×3	*Industrial*, at 812 dpi: · HP Indigo 5500 DS	*Scanner*: · Epson Perfection 4999 at 1200 ppi	digital: 300 original: 900 fakes: 0 total: 1200

Fig. 4.5 Examples of digital templates used in DP1C and DP1E datasets.

1. Two extensions of the originally published DP0E dataset [22]:

   - **DP1C**[4]
   - **DP1E**[5]

   Both datasets are publicly available for the academic usage and consist of codes printed on two inkjet and two laser office printers as indicated in Table 4.1. The main objective of these datasets is to investigate influence of the printing and scanning equipment on the clonability aspects of copy detection patterns used in the PGC from the side of the attacker. In this regard, the printed codes are scanned at 1200 ppi resolution.

2. **Indigo mobile** dataset

   A dataset of codes printed on the industrial printer HP Indigo 5500 DS (Table 4.1). The main objective of this dataset is to investigate the general authentication capabilities of copy detection patterns from the side of the defender. In this regard, to make the authentication conditions closer to the real life environment, instead of high quality scanners the printed codes are enrolled by using a mobile phone *iPhone XS* under regular room light.

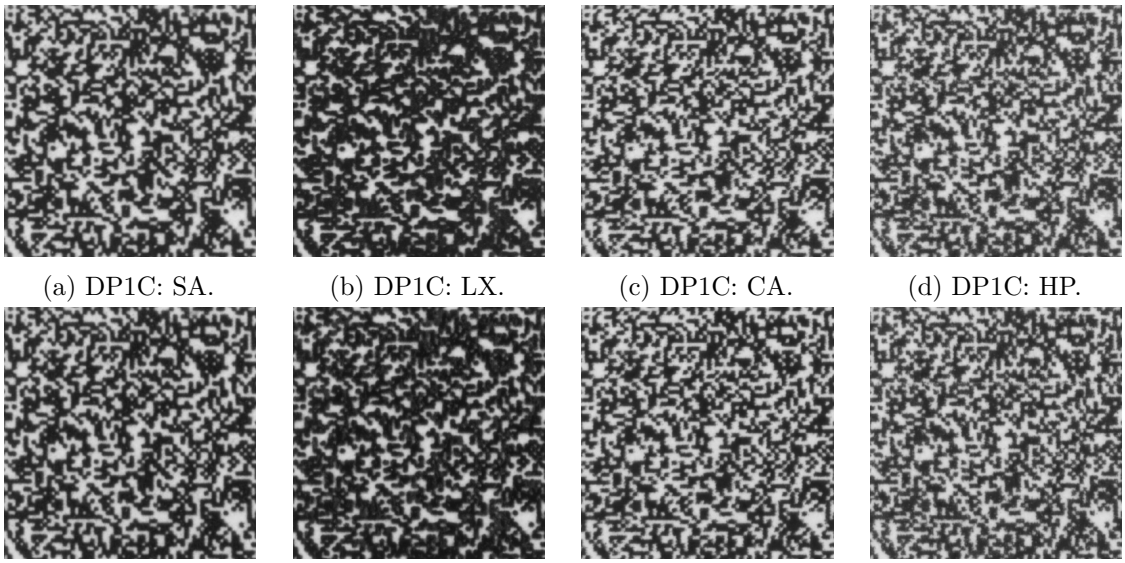
3. **Indigo scanner** dataset

   A dataset of codes with a symbols of three different sizes as indicated in Table 4.1 printed on the industrial printer HP Indigo 5500 DS and enrolled by using Epson perfection 4990 scanner. This dataset is used to investigate the capabilities of the attacker to accurately estimate the original digital templates from the printed counterparts with respect to the different sizes of copy detection pattern symbols used in PGC.

4.2.1 DP1C and DP1E datasets

The ability to perform an accurate estimation of the digital templates from the printed counterparts depends on many factors, among which the used printing and scanning equipment play the most fundamental role. While the properly selected printing equipment along with

[4]http://sip.unige.ch/projects/snf-it-dis/datasets/dp1c/
[5]http://sip.unige.ch/projects/snf-it-dis/datasets/dp1e/



(a) DP1C: SA. (b) DP1C: LX. (c) DP1C: CA. (d) DP1C: HP.

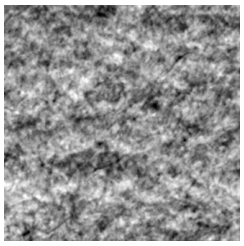
(e) DP1E: SA. (f) DP1E: LX. (g) DP1E: CA. (h) DP1E: HP.

Fig. 4.6 The DP1C and DP1E datasets: the first raw corresponds to the scans produced by Cannon scanner and the second raw is produced by Epson scanner for all considered printers. One can note a considerable variability among different printers, while the scans produced by two different scanners visually look to be quite correlated ones.

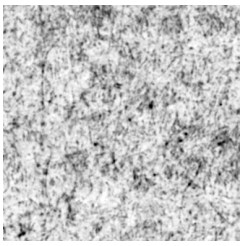
a chosen substrate plays an important role for the defender as an additional barrier for copy attacks, the attacker is interested in using the acquisition equipment of highest quality. It should be pointed out that the defender is somehow limited with his/her selection of printing equipment and substrate in view of various economical or even ecological factors. The printing is performed on a massive scale while the attacker can use the best available scanning equipment as many times as needed to succeed with his/her goal.

To investigate the influence of printing equipment on the clonability aspects of PGC from the side of attacker, we generate 384 distinct digital *DataMatrix* templates $\mathbf{t} \in \{0,1\}^{384\times384}$ with the symbols of size 6×6. Several examples of the digital templates are shown in Table 4.5. The generated codes are printed on two inkjet and two laser printers at the resolution 1200 dpi:

- Laser printers:
  - Samsung Xpress 430 (hereinafter referred to as *SA*);
  - Lexmark CS310 (hereinafter referred to as *LX*).
- Inkjet printers:
  - Canon PIXMA iP7200 (hereinafter referred to as *CA*);
  - HP OfficeJet Pro 8210 (hereinafter referred to as *HP*).



(a) Canon CanoScan 9000F.



(b) Epson Perfection V850.

Fig. 4.7 The scans of empty substrate (paper) by the Cannon and Epson scanners with the same setting of parameters as for Fig. 4.6.

To investigate the influence of scanning equipment the printed codes are scanned on two scanners at the resolution 1200 ppi:

- Canon CanoScan 9000F - the DP1C dataset;
- Epson Perfection V850 - the DP1E dataset.

Taking into account the equal printing and scanning resolutions, the enrolled codes consist of symbols of size 6×6 points. Examples of enrolled codes are shown in Fig. 4.6. It is easy to see that the SA printer provides the most accurate printing with a small dot gain. In the case of the LX printer, the dot gain is the highest one but the symbols' shape is quite accurate. In the cases of the CA and HP printers, the symbols' edges are a little torn. On the scanned codes the visual difference between the two scanners is not evident. Nevertheless, the difference in the illumination between Epson and Cannon scanners can be seen in Fig. 4.7, where the result of scanning of empty substrate (paper) is illustrated.

In total, the dataset contains 6528 codes:

- 384 distinct digital templates;
- 3072 original printed and scanned codes: 384 templates $\times$ 4 printers $\times$ 2 scanners;
- 3072 fake printed and scanned codes on the same equipment as original codes.

4.2.2 Indigo mobile dataset

An accurate authentication of the copy detection patterns is not less important than the resistance to the copy attacks. Moreover, the authentication in the industrial settings, when an industrial printer and a mobile phone are used, is of great practical importance.

To investigate the question of PGC authentication from the side of the defender, we generate 300 distinct digital *DataMatrix* templates $\mathbf{t} \in \{0,1\}^{330\times330}$ with the symbols of size 5×5. An example of the digital template is given in Fig. 4.8(a). The digital templates consist of the central copy detection pattern and four synchro-markers that allow to make an accurate synchronization and cropping of the code of interest. To simulate the real life scenario, the generated digital templates are printed on the industrial printer *HP Indigo 5500*

(a) Binary digital template. (b) Printed original code.

Fig. 4.8 Examples of (a) a binary digital template used for printing and (b) the printed original code from the Indigo mobile dataset.

DS at the resolution 812 dpi. The acquisition of the printed codes is done under regular room light using mobile phone *iPhone XS* under the automatic photo shooting settings in Lightroom application[6]. The mobile phone is held parallel to the printed code at height 11 cm as schematically shown in Fig 4.9. One photo is taken for each printed code. The

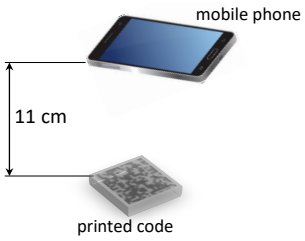


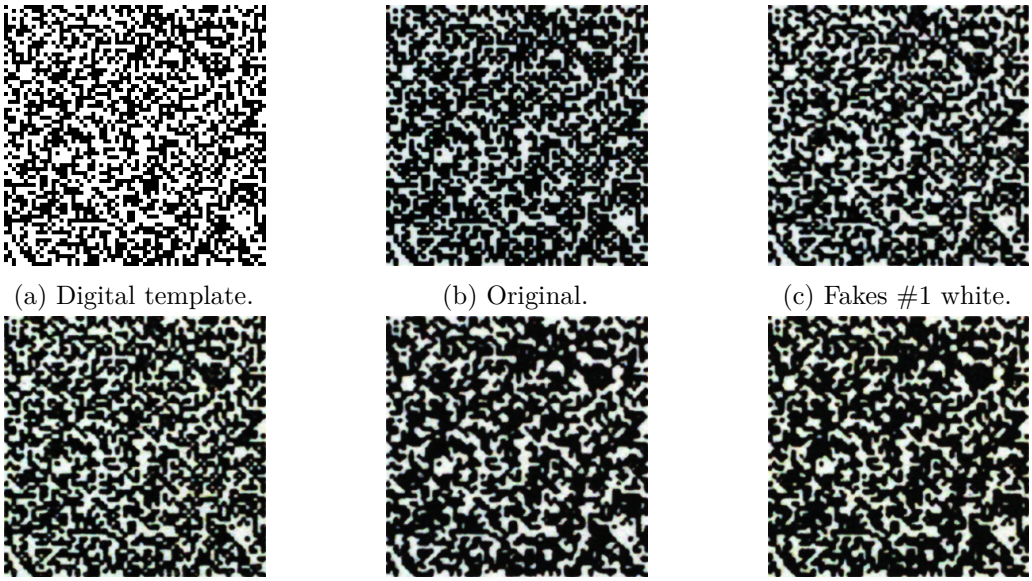
Fig. 4.9 The schematic representation of the mobile phone acquisition setup.

photos are taken in DNG format to avoid built-in mobile phone image post-processing. An example of obtained photo is shown in Fig. 4.8(b). The following cropping of the code is performed in an automatic way by applying a geometrical synchronization with four squared synchro-markers. Finally, the cropped codes are converted to the RGB format. The obtained codes are $\mathbf{x} \in \mathbb{R}^{330\times330}$ with symbols' size 5×5 points. An example of the obtained code is shown in Fig. 4.10(b).

As the most typical scenario for an unexperienced counterfeiter simulation we produce HC copies based on standard copy machines. The two different copy machines are used and the fakes are produced on two types of paper:

- Copy machines:
  1. RICOH MP C307
  2. Samsung CLX-6220FX
- Copy regime: text

[6]https://apps.apple.com/us/app/adobe-lightroom-photo-editor/id878783582



(a) Digital template. (b) Original. (c) Fakes #1 white.

(d) Fakes #1 gray. (e) Fakes #2 white. (f) Fakes #2 gray.

Fig. 4.10 Examples of original and fake codes with symbol size 5×5 taken by a mobile phone from the Indigo mobile dataset.

- Papers:
  - white paper 80 g/m^2
  - gray paper 80 g/m^2

Thus, for each original printed code we produced four fake codes:

1. *Fakes #1 white*: made by the first copy machine on the white paper.
2. *Fakes #1 gray*: made by the first copy machine on the gray paper.
3. *Fakes #2 white*: made by the second copy machine on the white paper.
4. *Fakes #2 gray*: made by the second copy machine on the gray paper.

To be coherent with the enrolled original printed codes, the acquisition of the produced fakes is done in the same way on the same mobile phone under the same photo and light settings.

In total, the Indigo mobile dataset contains 1800 codes:

- 300 distinct digital templates;
- 300 enrolled original printed codes;
- 1200 enrolled fake printed codes: 300 originals $\times$ 4 type of fakes.

Examples of the obtained digital, original and fake codes are shown in Fig. 4.10. Due to a smart morphological processing built-in the Ricoh copy machine the fakes #1 are more accurate with a dot gain close to the original codes. In the case of the fakes #2 the dot

(a) Symbol size: 5×5. (b) Symbol size: 4×4. (c) Symbol size: 3×3.

Fig. 4.11 Examples of three types of codes with different symbol size from the Indigo scanner dataset. All codes are scanned at 1200 ppi.

gain is much higher and, as a result, the symbols are less accurate. Visually the difference between the two types of used paper is not evident.

The main research question is to investigate whether the originals and produced fakes are reliably distinguishable under the mobile phone verification.

4.2.3 Indigo scanner dataset

Without a doubt the printing equipment plays an important role in the estimation of original digital templates from the printed counterparts. Along with that, the size of the used copy detection pattern elements (symbols) plays not less important role.

To explore the influence of size of used copy detection elements on the accuracy of digital templates estimation, we use 300 distinct digital *DataMatrix* templates with the mapping matrix of 66×66 symbols of three sizes: *(i)* 5×5, *(ii)* 4×4 and *(iii)* 3×3. All codes are printed on the same printer HP Indigo 5500 DS at resolution 812 dpi. The enrollment of the printed codes is performed using Epson Perfection 4990 scanner at the resolution 1200 ppi. The example of the printed codes are shown in Fig. 4.11. Taking into account the difference between the printing and scanning resolutions, the original symbol size increases in about 1.48 times ($1200/812 = 1.48$). Thus, the original codes of size 5×5 become of size 7×7, the codes of size 4×4 increase to about 5×5 and the codes 3×3 become of size 4×4.

In total, the dataset contains 1200 codes:

- 300 distinct digital templates;
- 900 original printed codes:
  - 300 codes of original size 5×5
  - 300 codes of original size 4×4
  - 300 codes of original size 3×3

This dataset is used to verify how accurately the attacker can estimate the original digital templates from the printed counterparts with different symbol size using different estimation strategies.

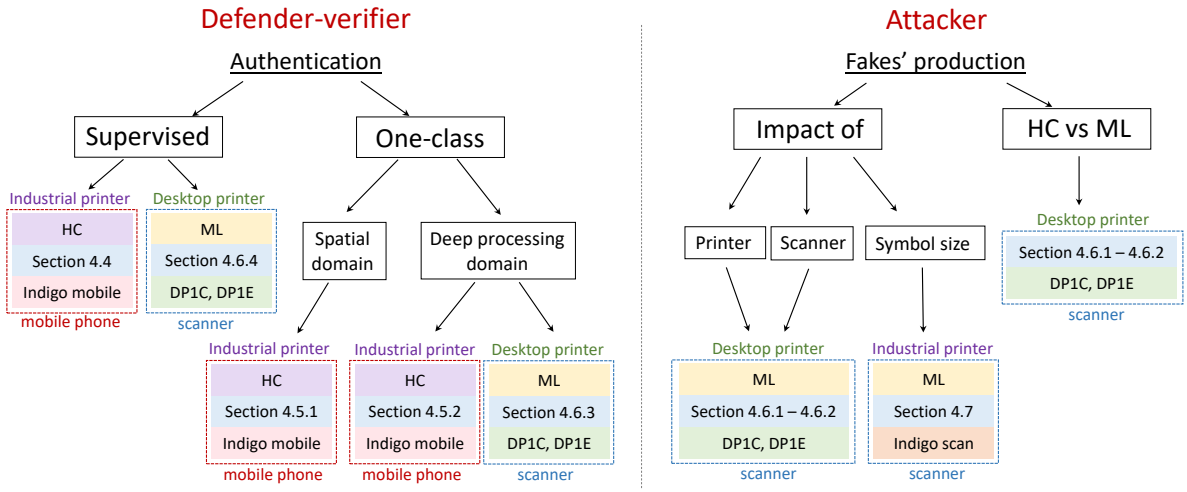


Fig. 4.12 The overview of investigated authentication and clonability aspects of the PGC. The authentication task is studied from the side of defender with respect to the HC and ML copy fakes. The clonability of PGC is investigated from the side of attacker with respect to the HC and ML approaches.

4.3 Research questions

In this work, we focus on PGC considering defender-verifier and attacker strategies. This Section aims at introducing the main research questions related to the investigation of authentication and clonability aspects of PGC. Fig. 4.12 represents overview of the problems under investigation that leads to the following research questions that are also summarized in Tables 4.2 - 4.3:

- **Hand-crafted (HC) copy fakes**

  Q.1 **(Defender-verifier):** *What is the achievable accuracy of authentication, when the classifier knows all fakes at training time, i.e., the supervised classification?*

  Due to the great achievements of the mobile phone imaging technologies, the PGC authentication via modern mobile phones under conditions close to the real life is nowadays a hot topic. In this regard, the base-line supervised authentication of the PGC with respect to the typical HC fakes is performed on the Indigo mobile dataset that includes the originals and four type of typical HC copy fakes printed on the industrial HP Indigo printer and enrolled via mobile phone under regular light conditions[7]. The question of supervised authentication of the PGC is investigated in Section 4.4. The experiments should also give an answer to the question how the quality of the used fakes impacts the authentication accuracy on

[7]The more details about the Indigo mobile dataset are given in Section 4.2.2.

the unseen types HC copy fakes. Furthermore, we are interested to demonstrate the importance of fake sample selection for the worst case fakes-based classification.

Q.2 **(Defender-verifier):** *Can the typical HC copy fakes be efficiently authenticated via one-class classification in spatial domain?*

The base-line supervised classification is an important question that provides a kind of upper bound on the expected efficiency of authentication, when all types of fakes are available for training. At the same time, it is obvious, that in practice it is quite difficult to predict in advance what kind of fakes will appear and will be presented for the authentication. In this regard, Section 4.5 investigates the efficiency of one-class classification trained in a "blind" way without observing the potential fakes at all. More particularly, Section 4.5.1 studies the one-class classification of the HC fakes in the spatial domain, i.e., in the domain of the direct observation without transforming data to some transform or latent space. The investigation is performed on the Indigo mobile dataset under conditions close to the real life environment.

Q.3 **(Defender-verifier):** *Is the HC copy fakes' authentication based on the one-class classification in a deep processing domain more efficient than in the spatial domain?*

One might expect that the one-class classification in the spatial domain might be a complex problem due to the no always trivial choice of the proper features and metrics for investigation. In addition, the class of original and fakes might be on very complex data manifolds requiring a selection of complex kernels. At the same time, the same task might be more intuitive and easy solvable in the deep processing domain or transform domain based on available data. In this regard Section 4.5.2 aims at investigating the one class classification of the PGC in the deep processing domain. Similarly to the investigation in the spatial domain, the study is performed on the Indigo mobile dataset.

- **Machine learning (ML) copy fakes**

Q.4 **(Attacker):** *What is the impact of the printing and scanning equipment on the clonability aspects of copy detection patterns?*

Besides the typical HC copy fakes produced on the standard copy machines, the current work also aims at investigating the other types of fakes, namely, the fakes based on an estimation of the original digital templates from the corresponding printed counterparts. Different factors might impact the quality of fakes produced in such a way. The investigation of the printing and acquisition equipment that are among the most important fakes, is considered in Sections 4.6.1 - 4.6.2 and is performed on the DP1C and DP1E datasets that include four desktop printers.

Since the attacker aims at producing the fakes as close as possible to the original codes, the use of the mobile phone is unreasonable and two high quality scanners are used for this study.

Q.5 **(Attacker):** *Is accuracy of ML-based fakes higher than the accuracy of HC attacks?*

From one side the ML fakes require less know-how from the attacker. From the other side they require a certain amount of training data that might be quite expensive and difficult to obtain in real life applications. Taking this into account Sections 4.6.1 - 4.6.2 compare the efficiency of machine learning and hand-crafted based approaches in terms of accuracy of estimation of the original digital templates from the corresponding printed counterparts. The evaluation is performed on the DP1C and DP1E datasets with respect to the different printing and scanning equipment.

Q.6 **(Defender-verifier):** *Can the produced ML fakes be efficiently authenticated via one-class classification in a deep processing domain?*

Similarly to the one-class classification of the HC fakes, no less important is to investigate the same task with respect to the ML fakes. The investigation in Section 4.6.3 is performed on the DP1C and DP1E datasets with respect to the high quality ML fakes produced on the different printing equipment that additionally shows how different printing quality might be beneficiary to the defender or to the attacker. To investigate the impact of ML fakes printed on the same equipment, while eliminating the impact of mobile phone imaging, the verification is performed on scanned samples.

Q.7 **(Defender-verifier):** *What is the efficiency of supervised classification of the produced ML fakes?*

The base-line supervised classification with respect to the high quality ML fakes is performed in Section 4.6.4. Similarly to the one-class classification scenario, the evaluation is done on the DP1C and DP1E datasets on the same printing and acquisition equipment. The performed experiments additionally answer to a question of impact of the printing quality as an advantage for the defender or attacker.

Q.8 **(Attacker):** *What is the impact of the symbols' size on the clonability aspects of copy detection patterns?*

The research question Q.4 aims at investigating the impact of printing and scanning equipment on the quality of the produced fakes. The size of symbols used in the copy detection patterns is also of great importance for both the defender and attacker. This factor is investigated in Section 4.7 on a specially designed Indigo scanner dataset that is printed at the high resolution on the industrial printer HP

Indigo. Moreover, to avoid a potential bias due to the quality of the acquisition equipment, a high quality scanner is used for codes' enrollment.

Table 4.2 The summary of research questions with respect to the hand-crafted (HC) fakes.

	#	Research question	Section	Dataset	Conclusions
Defender-verifier	Q.1	What is the achievable accuracy of authentication, when the classifier knows all fakes at training time, i.e., the supervised classification?	4.4	Indigo mobile	The DNN model trained in a supervised way is capable to authenticate the original PGC from the typical HC copy fakes with a high accuracy. The quality of fakes used for the training plays a very important role. The use of fakes close to the original codes at the training allows to authenticate the lower quality fakes even when the model does not see them during the training. The opposite is not true.
Defender-verifier	Q.2	Can the typical HC copy fakes be efficiently authenticated via one-class classification in a spatial domain?	4.5.1	Indigo mobile	The authentication in the spatial domain is difficult with respect to the finding right metrics and is not efficient enough in view of the great similarity between the original and fake codes.
Defender-verifier	Q.3	Is the HC copy fakes' authentication based on the one-class classification in a deep processing domain more efficient than in the spatial domain?	4.5.2	Indigo mobile	The one-class classification in the deep processing domain is more efficient than the authentication in the spatial domain. Although, the DNN based models have higher training complexity, at the inference stage, the trained models are equivalent in complexity to the authentication in the spatial domain.

Table 4.3 The summary of research questions with respect to the machine learning (ML) fakes.

	#	Research question	Section	Dataset	Conclusions
Attacker	Q.4	What is the impact of the printing and scanning equipment on the clonability aspects of copy detection patterns?	4.6.1 4.6.2	DP1C & DP1E	The four printers and two scanners are investigated. An accurate printing leads to easier estimation of the digital templates from the printed counterparts. The scanner illumination might have an important influence on the production of high quality ML fakes, especially for the printers with a big dot gain.
Attacker	Q.5	Is accuracy of ML-based fakes higher than the accuracy of HC attacks?	4.6.1 4.6.2	DP1C & DP1E	The machine learning methods demonstrate considerably higher accuracy of the estimation of digital templates than the hand-crafted approaches.
Defender-verifier	Q.6	Can the produced ML fakes be efficiently authenticated via one-class classification in a deep processing domain?	4.6.3	DP1C & DP1E	Due to the high degree of similarity between the originals and ML fakes, it is very difficult to train an efficient authentication model in a "blind" way without observing the fakes. Thus, the one-class classifier is not capable to distinguish the ML fakes for the considered code design.
Defender-verifier	Q.7	What is the efficiency of supervised classification of the produced ML fakes?	4.6.4	DP1C & DP1E	In contrast to the case of HC copy fakes, the supervised classification is not efficient enough against the ML fakes. The printing deviations might play very important role for the authentication of high quality ML fakes.
Attacker	Q.8	What is the impact of the symbols' size on the clonability aspects of copy detection patterns?	4.7	Indigo scanner	Reducing the original symbols' sizes leads to a deterioration in the accuracy of ML-based estimation. Since we are limited in the access to the symbol size 1×1 prints, the question of clonability of these codes based on the ML-attacks remains open.

4.4 Supervised classification with respect to the HC fakes

Addressing the research question Q.1 from Section 4.3, the supervised classification is chosen as a base-line to validate the authentication efficiency of the PGC. The complete availability of fakes for the classification gives the defender an information advantage over the attacker. Such a scenario is an ideal case for the defender and the worst case for the attacker. This scenario assumes that, besides the original digital templates $\{\mathbf{t}_i\}_{i=1}^{M}$ and the corresponding printed codes $\{\mathbf{x}_i\}_{i=1}^{M}$, the defender has an access to the fake codes $\{\mathbf{f}_i\}_{i=1}^{M_f}$, $M_f \leq M$.

The problem formulation and theoretical analysis of the supervised classification are given in Section 3.3. The supervised classification corresponds to the considered *supervised training without latent space regularization* described by the equation (3.7):

$$\mathcal{L}^{\text{HCP}}_{\text{S}-\text{NoReg}}(\boldsymbol{\theta}_{\text{c}}, \boldsymbol{\phi}_{\text{a}}) = \mathcal{D}_{\text{c}\hat{\text{c}}}.$$

The empirical experiments are performed on the Indigo mobile dataset that includes four types of HC copy fakes. The fakes are used in the same amount as the original printed codes, i.e., $M_f = M$. Moreover, the physics of the Indigo mobile dataset construction involves pairwise matching between the digital templates, printed original and fake codes.

The supervised classification is performed in two scenarios:

- Five class classification:
  1. original
  2. fake #1 white
  3. fake #1 gray
  4. fake #2 white
  5. fake #2 gray
- Two class classification:
  1. original
  2. fake

The detailed information about the training procedure, model architecture and the used parameters is given in Appendix D.1.1.

At the inference stage, the query sample $\mathbf{y}$, which might be either original $\mathbf{x}$ or one of the fakes $\mathbf{f}^k$, $k = 1, ..., 4$, is passed through the classifier $g_{\boldsymbol{\theta}}$ trained with respect to the cross-entropy term $\mathcal{D}_{\text{c}\hat{\text{c}}}$.

4.4.1 Five class classification

The five class supervised classification aims at investigating the performance of the base-line supervised classification scenario, where the model is trained on all classes of the data. At the inference stage three scenarios are evaluated:

[8] The detailed information about each run classification error is given in Appendix D.1.2.

Table 4.4 The average (over five runs) classification error in % on the test sub-set of the classification model trained in a supervised way on the originals and all type of fakes[8].

Type of classification	Original	Fake #1 white	Fake #1 gray	Fake #2 white	Fake # 2 gray
2 classes	0.00	0.28 (±0.3)			
3 classes	0.00	0.78 (±0.68)		0.35 (±0.5)	
5 classes	0.00	23.26 (±7.55)	21.56 (±0.81)	16.88 (±6.62)	11.35 (±4.89)

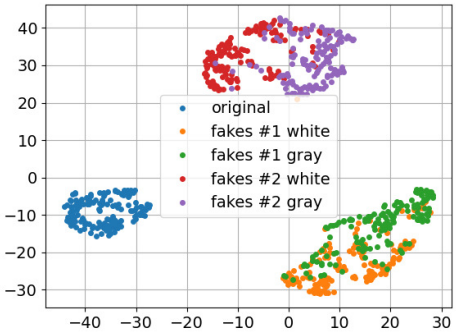


Fig. 4.13 T-SNE of the latent space of the classification model trained on originals and all type of fakes.

(a) 2 classes classification: the ability of model to distinguish the originals from the fakes without caring about the fakes type.

(b) 3 classes classification: the ability of model to distinguish the originals, fakes from the first (fakes #1) and the second (fakes #2) groups.

(c) 5 classes classification: the ability of model to distinguish all classes of the data, i.e., originals and each type of fake.

Due to the relatively small amount of the codes in the Indigo mobile dataset and to avoid the bias in the selection of data for training and testing, the classification model is trained five times on the randomly chosen data. The average classification error is given in Table 4.4. It is easy to see that the investigated model is capable to authenticate the original codes in all scenarios with the $P_{miss} = 0$. The probability of false acceptance P_{fa} about 0.28% in the two classes validation setup (2 classes type of classification) indicating that despite the visual similarity the classifier is capable to distinguish original and fakes with high enough accuracy. From the three classes validation scenario (3 classes type of classification), one can notice that the model confuses more the fakes #1 and fakes #2. We will explain the reasons for that below. The last validation scenario (5 classes type of classification) shows that for both groups of fakes the most difficult is to distinguish between the white and gray paper

Table 4.5 The average (over five runs) classification error in % on the test sub-set of the classification model trained in a supervised way on the originals and only one type of fakes[9].

Train on	Original P_{miss}	Fake #1 white P_{fa}	Fake #1 gray P_{fa}	Fake #2 white P_{fa}	Fake # 2 gray P_{fa}
Fakes #1 white	0	0	0.14 (±0.32)	0	0
Fakes #1 gray	0	0	0	0	0
Fakes #2 white	0	99.43 (±0.32)	100	0	0
Fakes # 2 gray	0	99.29 (±0.5)	99.86 (±0.32)	0	0

type fakes. In addition, in Fig. 4.13 the T-SNE visualization [143] of the latent space of the classifier trained in five class classification scenario is illustrated. From that visualization one can easily see the same phenomena: three main classes (originals, fakes #1 and fakes #2) are well separated while the samples printed on the white and gray papers are overlap. This indicates that the substrate identification is a difficult problem even for the supervised classifier.

4.4.2 Two class classification

The two class classification aims at investigating the influence of the fakes' type used for the training on the model efficiency at the inference stage. In this respect, the training is performed separately on each type of fakes. The classification accuracy is evaluated with respect to the probability of miss P_{miss} and the probability of false acceptance P_{fa}:

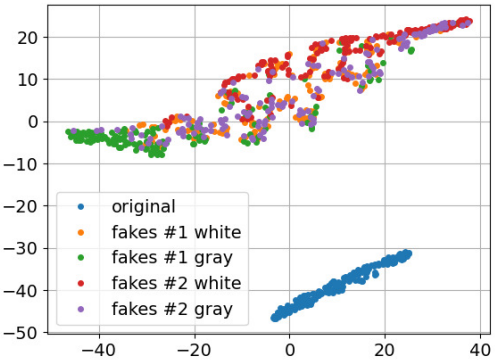
$$\begin{cases} P_{miss} &= \Pr\{g_{\boldsymbol{\theta}}(\mathbf{Y}) \neq 1 \mid \mathcal{H}_0\}, \\ P_{fa} &= \Pr\{g_{\boldsymbol{\theta}}(\mathbf{Y}) = 1 \mid \overline{\mathcal{H}}_0\}, \end{cases} \tag{4.1}$$

where $\mathcal{H}_0$ corresponds to the hypothesis that the query $\mathbf{y}$ is an original code and $\overline{\mathcal{H}}_0$ is the hypothesis that the query $\mathbf{y}$ is a fake code.

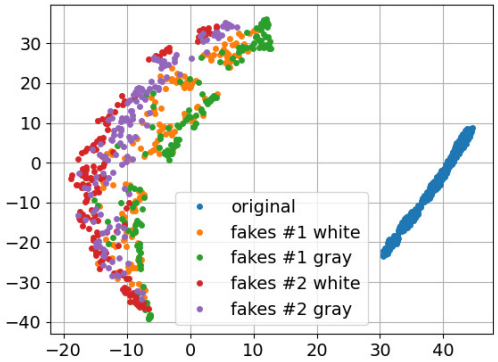
Similarly to the five class classification scenario, in each case, the model is trained five times on the randomly chosen data to avoid the bias in the training data selection. The average results are represented in Table 4.5.

The models trained on the originals and fakes #1 provide high classification accuracy on the all type of data, including the fakes #2. That is expected and can be explained by the fact that, as it is discussed in Section 4.2.2, the fakes #1 are closer to the originals, while the fakes #2 are the coarser copies of the original codes. In this regard, when the training is performed on the fakes #2, the model is not capable to distinguish the originals from the fakes #1. That is confirmed by the probability of false acceptance close to 100%. Nevertheless, the model is capable to distinguish the originals from the fakes #2 with 100% accuracy. The T-SNE visualization of the latent space of each model illustrated in Fig. 4.14

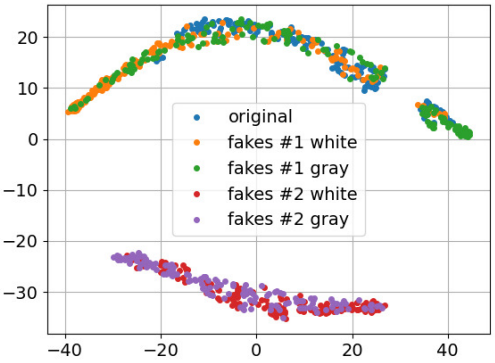
[9]The detailed information about each run classification error is given in Appendix D.1.3



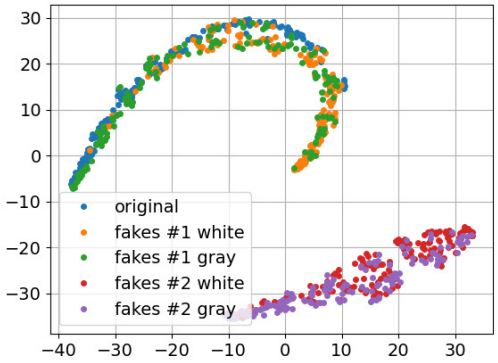
(a) Training with respect to the fakes #1 white.



(b) Training with respect to the fakes #1 gray.



(c) Training with respect to the fakes #2 white.



(d) Training with respect to the fakes #2 gray.

Fig. 4.14 The latent space T-SNE visualization of the supervised two class classifier trained on the originals and one type of fakes.

confirms these observations. From Fig. 4.14(a) and 4.14(b) that present the latent space of model trained on the originals and the fakes #1, one can see the good separability between the originals and fakes while all classes of fakes are overlap. The latent space visualization of model trained on the originals and fakes #2 illustrated in Fig. 4.14(c) and 4.14(d) shows the overlapping between the originals and the fakes #1 preserving the fakes #2 in well separable cluster.

4.4.3 Conclusions

The performed analysis of the base-line supervised classification of PGC reveals two important observations:

- In the general case, the model trained in a supervised way is capable to authenticate with a high accuracy the original PGC from the fakes produced on modern copy

machines even with build-in smart morphological processing enhancing image quality and reducing the dot gain for further reproduction.

- The quality of the fakes used for the training plays a very important role. The superior quality fakes close to the original codes are of preference for the training and allow the model to authenticate the inferior quality fakes, even when the model does not see them during the training. In contrast, if the classifier is trained on the inferior quality fakes, such a classifier is not capable to authenticate the superior quality fakes.

4.5 One-class classification with respect to the HC fakes

The supervised classification scenario considered in Section 4.4 is an ideal case for the informed defender. However, in practice, collecting fakes is time consuming and might be expensive enough process. Moreover, due to the permanent improvement of technologies available to the attacker, it is quite difficult to predict in advance what kind of fakes might be produced by the attacker next time. In this respect, the one-class classification scenario, where the authentication model is trained only on the original data disregarding the potential fakes is of great practical importance. Up to our best knowledge, such a problem was not considered in the state-of-the-art works related to the PGC authentication.

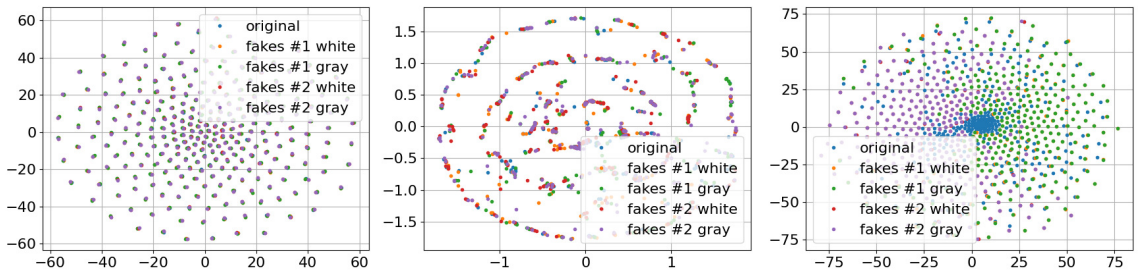
One-class classification [144] also known as an *anomaly detection* and *out-of-class detection* [145, 146] is an important class of classification that has been investigated within diverse research areas and application domains. One-class classification aims at detecting the patterns in data that do not conform to expected behavior.

A straightforward one-class classification approach consists in defining a region representing the known normal class of the data and declaring any observations in the data that does not belong to this normal region as an outliers. In regard to the PGC authentication, this task might be very challenging due to the fact that the attackers often adapt themselves to make the fake observations appear normal, thereby making the task of defining original patterns representing inliers more difficult.

4.5.1 Spatial domain data analysis

In Section 4.4 it is shown that according to results obtained for the Indigo mobile dataset the original and fake codes are well separable in the latent space of the supervised classifier. The research question Q.2 from Section 4.3 aims at investigating how these data behave in the direct image domain (hereinafter also referred to as a *spatial* domain). To answer this question, the 2D T-SNE visualizations of the data in the spatial domain are shown in Fig. 4.15.

Fig. 4.15(a) shows the direct visualisation of the RGB images. It is easy to note that the data do not form any clusters corresponding to originals or fakes. Instead, the data are



(a) (b) (c)

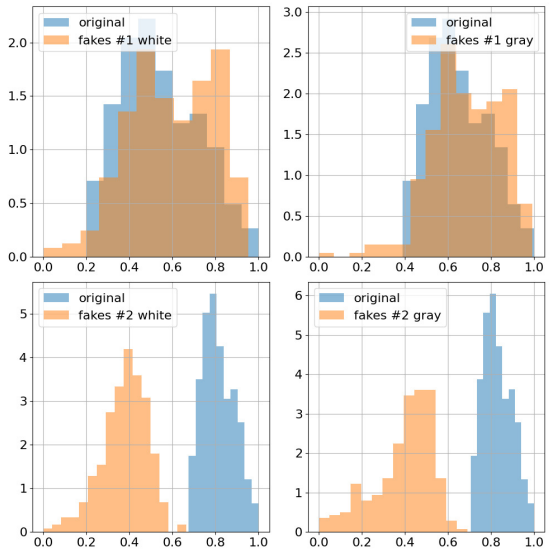
Fig. 4.15 The 2D T-SNE visualisation of the original and fake codes in the spatial domain (a horizontal axis denotes T-SNE dimension 1 and the T-SNE dimension 2 is on the vertical axis): (a) presents the direct RGB images' visualisation; (b) is based on the xor difference between the corresponding digital templates and printed codes binarized via simple threshold method with an optimal threshold determited individually for each printed code via Otsu's method [2]; (c) visualizes the differences between the physical references and the corresponding printed original and fake codes.

allocated into small groups that are formed by the original and fakes corresponding to the same digital template. Such a behavior is expectable and is explainable by the data nature.

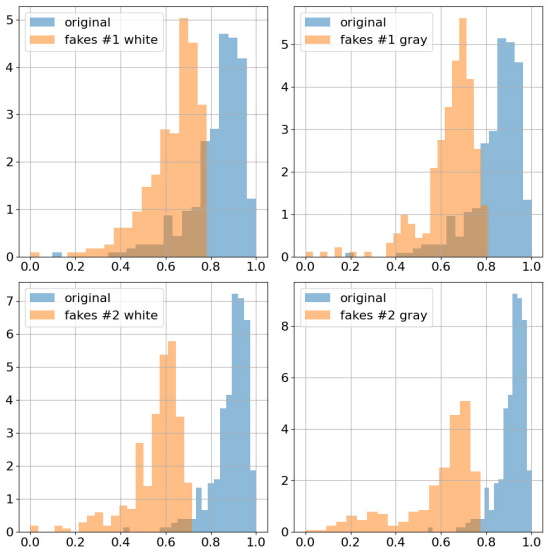
Fig. 4.15(b) demonstrates a visualization based on the xor difference between the digital templates and the corresponding printed codes binarized via a simple threshold method with an optimal threshold determited individually for each printed code via Otsu's method [2]. In general, one can observe a kind of rings that consist of the original and fakes but no clusters specific to the data types. These rings are explainable by the fact that both originals and fakes can have bigger or smaller difference with the digital template due to the dot gain in the different group of black and white symbols: a white symbol surrounded by the black symbols results in a bigger binarization error, while the black symbol surrounded by the white symbols is more likely to survive after binarization.

To better understand the role of the digital templates as a references, the Indigo mobile dataset was specially extended by the printed references (hereinafter also referred to as *physical references*[10]). From Fig. 4.15(c) that illustrates the T-SNE of the differences between the physical reference and the corresponding printed original and fake codes, it is easy to note the central dense cluster formed by the original codes (in blue) and two surrounding clusters from the fakes #1 (mostly on the right-hand side) and fakes #2 (mostly on the left-hand side). Despite this, the overall mixing of individual samples from the different classes is quite significant. This indicates that the reliable direct spatial authentication might be infeasible.

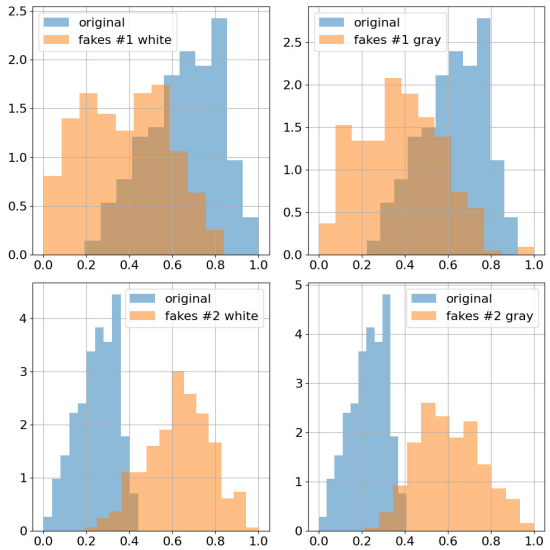
[10]The physical references correspond to the original codes' acquired for the second time on the same equipment as the first case scenario. It assumes the probable presence of small geometrical (rotation) and illumination deviations between the original codes and corresponding physical references.



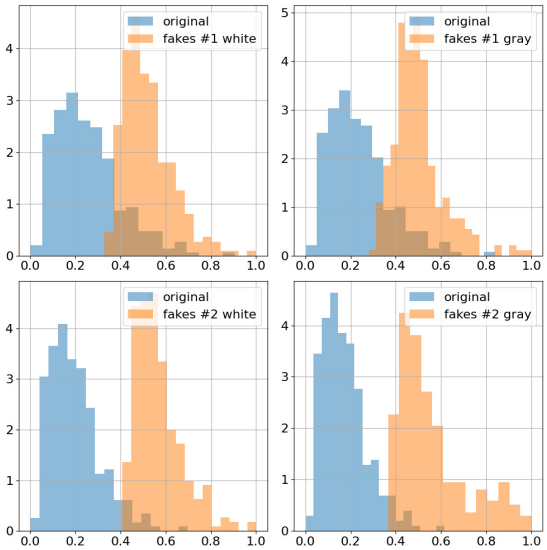
(a) Pearson correlation with respect to the digital templates/references.



(b) Pearson correlation with respect to the physical references.

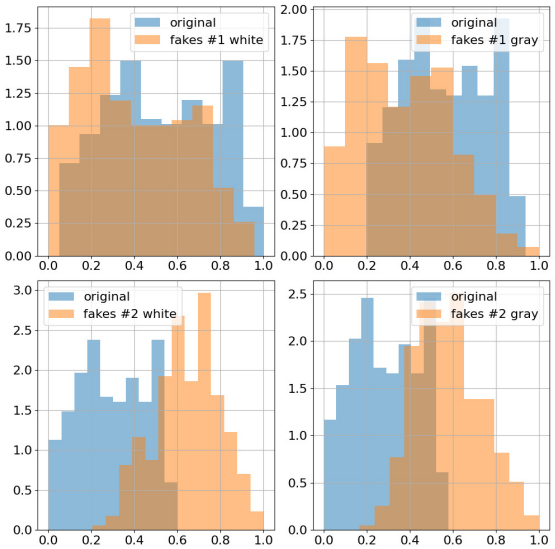


(c) Hamming distance with respect to the digital templates/references.

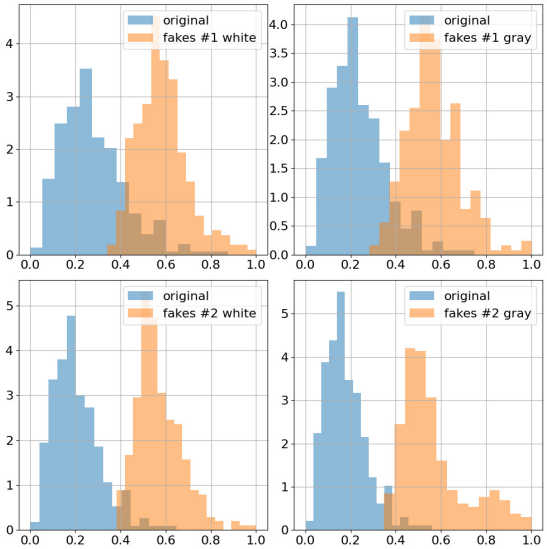


(d) Hamming distance with respect to the binarized physical references.

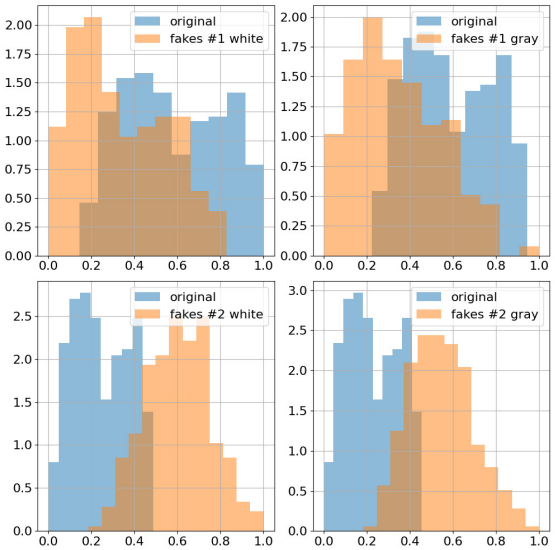
Fig. 4.16 The distribution of the Hamming distances between the references (digital or physical) and the corresponding printed codes (original and fakes). In case of the physical references the binarization is applied as a simple thresholding with an optimal threshold determined individually for each code via Otsu's method.



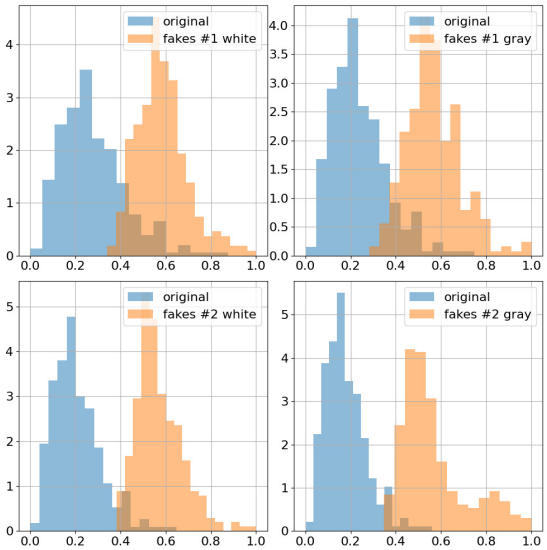
(a) ℓ_1 distance with respect to the digital templates/references.



(b) ℓ_1 distance with respect to the physical reference.

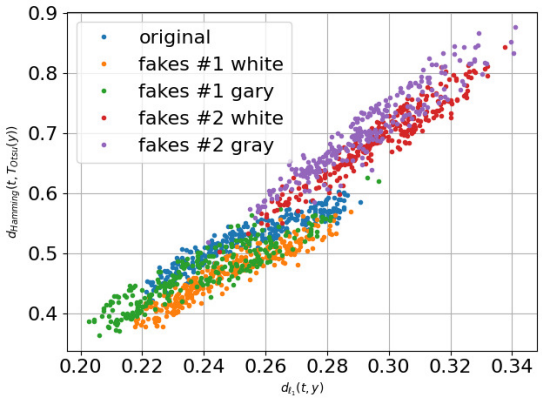


(c) ℓ_2 distance with respect to the digital templates/references.

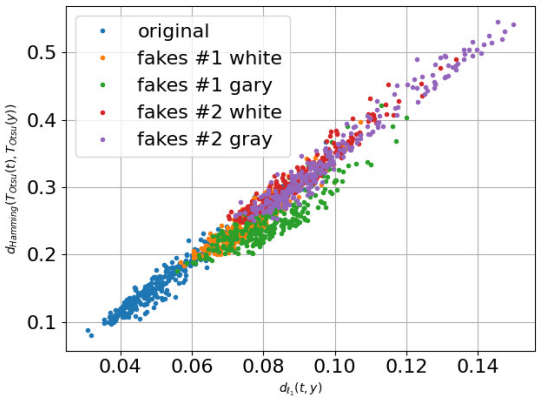


(d) ℓ_2 distance with respect to the physical reference.

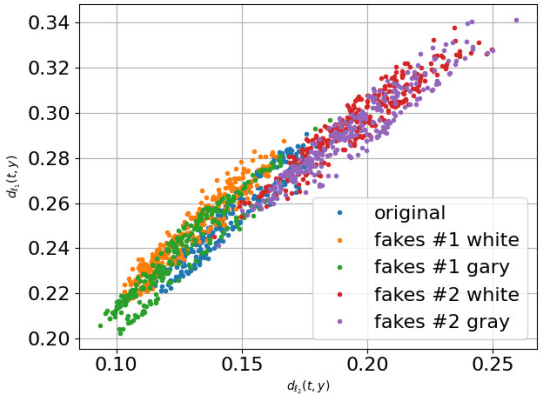
Fig. 4.17 The distribution of the ℓ_1 and ℓ_2 distances between the references (digital or physical) and the corresponding printed codes (original and fakes).



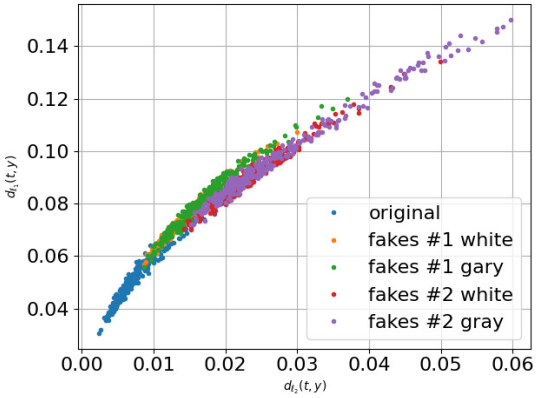
(a) ℓ_1 vs Hamming distance with respect to the digital templates/references.



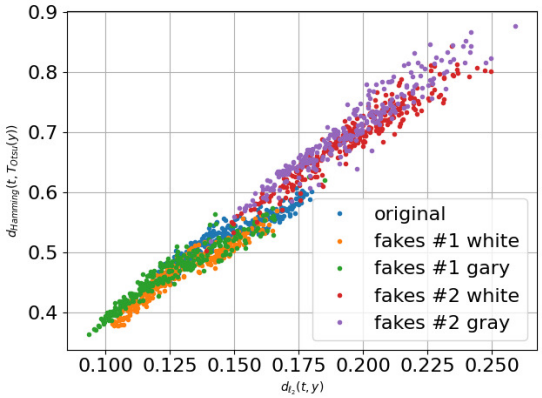
(b) ℓ_1 vs Hamming distance with respect to the physical references.



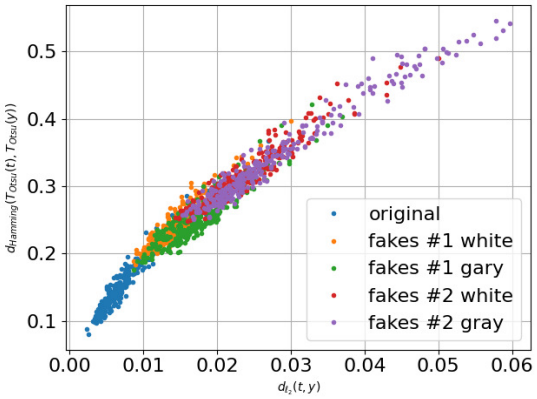
(c) ℓ_2 vs ℓ_1 distances with respect to the digital templates/references.



(d) ℓ_2 vs ℓ_1 distances with respect to the physical references.

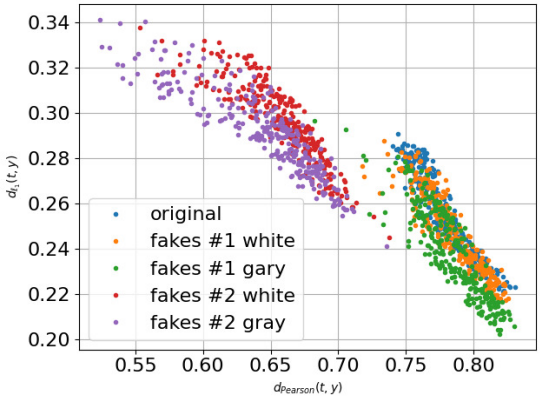


(e) ℓ_2 vs Hamming distance with respect to the digital templates/references.

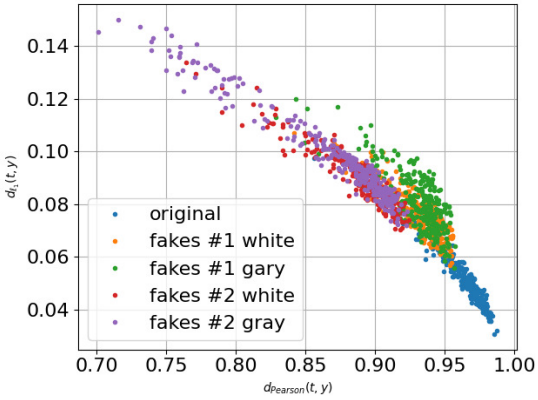


(f) ℓ_2 vs Hamming distance with respect to the physical references.

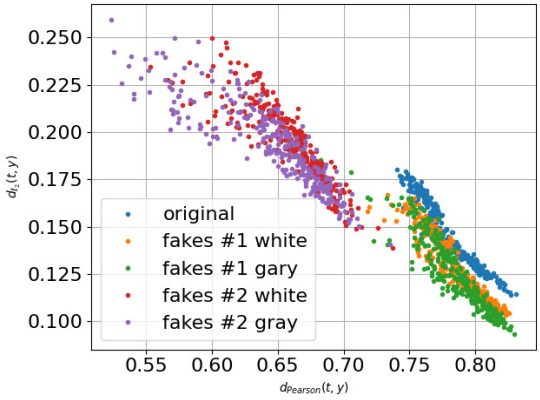
Fig. 4.18 The 2D PGC separability in the spatial domain with respect to the ℓ_2, ℓ_1 and Hamming distances between the references (digital or physical) and the corresponding printed codes (original and fakes) (part 1).



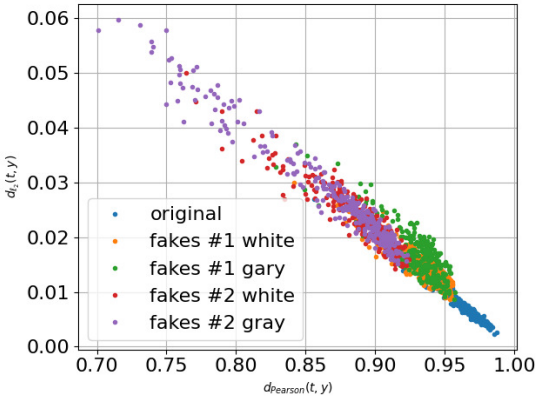
(a) Pearson correlation vs ℓ_1 distance with respect to the digital templates/references.



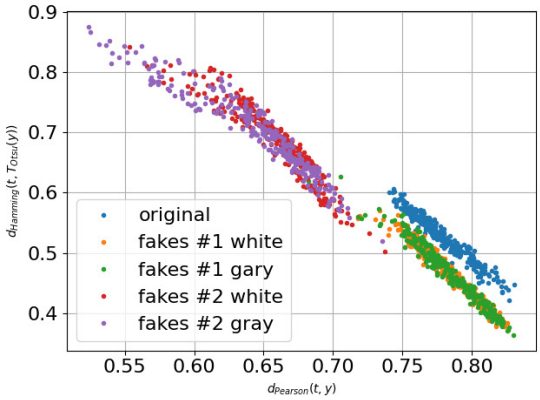
(b) Pearson correlation vs ℓ_1 distance with respect to the physical references.



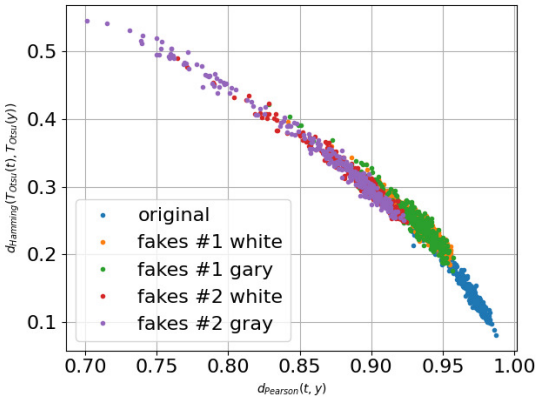
(c) Pearson correlation vs ℓ_2 distance with respect to the digital templates/references.



(d) Pearson correlation vs ℓ_2 distance with respect to the physical references.



(e) Pearson correlation vs Hamming distance with respect to the digital templates/references.



(f) Pearson correlation vs Hamming distance with respect to the physical references.

Fig. 4.19 The 2D PGC separability in spatial domain with respect to the ℓ_2, ℓ_1, Hamming distances and Pearson correlation between the references (digital or physical) and the corresponding printed codes (original and fakes) (part 2).

Table 4.6 The average (over five runs) OC-SVM classification error in % on the test sub-set.

Train on	Original P_{miss}	Fake #1 white P_{fa}	Fake #1 gray P_{fa}	Fake #2 white P_{fa}	Fake #2 gray P_{fa}
With respect to the digital templates:					
- grayscale $\mathbf{x}$	3.1 (±0.83)	2.54 (±1.93)	3.82 (±1.22)	0	0
- RGB $\mathbf{x}$	2.82 (±1.14)	2.1 (±0.86)	1.4 (±1.4)	0	0
With respect to the physical references:					
- grayscale $\mathbf{x}$	11.44 (±4.14)	35.86 (±7.38)	40.58 (±4.86)	1.72 (±2.07)	1.12 (±0.8)
- RGB $\mathbf{x}$	11.16 (±3.64)	31.84 (±6.3)	39.54 (±6.11)	1.44 (±1.69)	0.98 (±0.63)

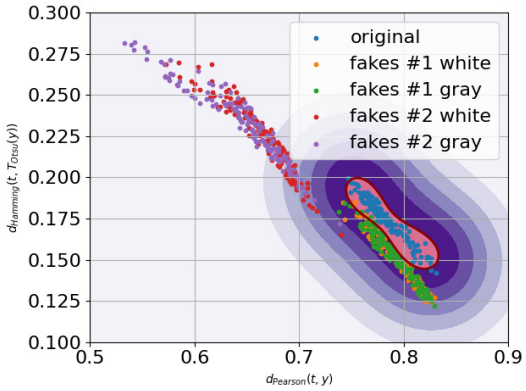
To confirm this observation, Fig. 4.16 represents the distribution of distances between the references (digital or physical) and the corresponding printed codes (original and fakes). The comparison is done with respect to the Pearson correlation and Hamming distance. For the Hamming distance the binarization is applied as a simple thresholding with an optimal threshold determined individually for each code via Otsu's method. Additionally, the results with respect to the ℓ_1 and ℓ_2 metrics are given in Fig. 4.17. In general, one can conclude that, besides some rare exceptions (the fakes # 2 in Fig. 4.16(a)), it is impossible to separate the original and fakes codes neither with respect to the digital template nor with respect to the physical reference based only on one metric. At the same time, as shown in Fig. 4.18 - 4.19 the separability with respect to the two metrics is much better.

The obtained results encourage to apply the one-class classification to the case of the Pearson correlation and Hamming distance between the printed codes and the corresponding digital or physical references shown in Fig. 4.19(e) - 4.19(f). In this regard, the one-class support vector machines (OC-SVM) [147] is chosen as a base-line approach. The detailed information about the OC-SVM training parameters is given in Appendix D.2.1.

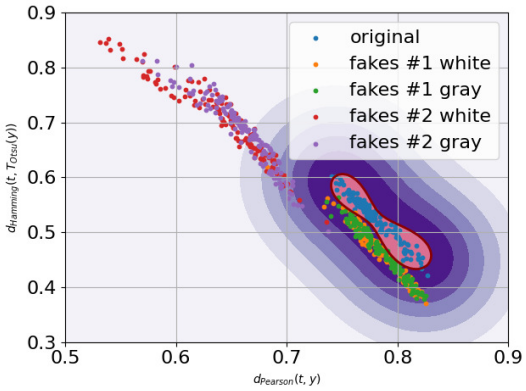
To better understand the role of used reference and the influence of color information during the acquisition of black and white codes as opposed their convertion to only grayscale images, the OC-SVM is applied with respect to four types of training data:

- With respect to the digital templates on:
  - the grayscale original codes $\mathbf{x}$;
  - the RGB original codes $\mathbf{x}$.
- With respect to the physical references on:
  - the grayscale original codes $\mathbf{x}$;
  - the RGB original codes $\mathbf{x}$.

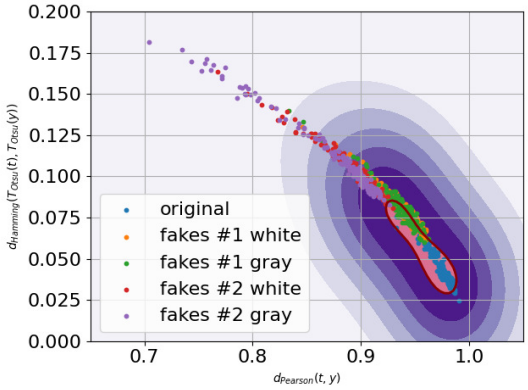
To avoid the bias in the training data selection, the OC-SVM is trained five times on randomly chosen original printed samples $\mathbf{x}$ and either digital templates or physical references. The OC-SVM is trained to minimize the P_{miss} on the validation sub-set. The obtained average classification error is represented in Table 4.6. The detailed information about



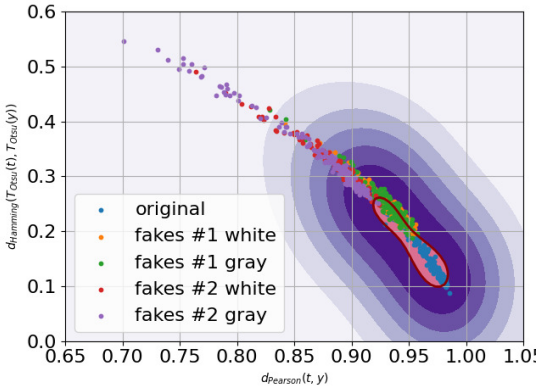
(a) With respect to the *digital* templates: grayscale data.



(b) With respect to the *digital* templates: RGB data.



(c) With respect to the *physical* references: grayscale data.



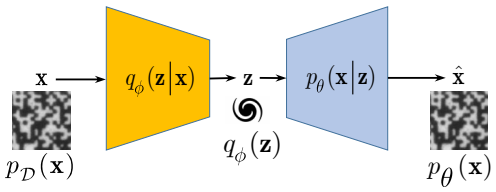
(d) With respect to the *physical* references: RGB data.

Fig. 4.20 The decision boundaries of OC-SVM trained with respect to the Pearson correlation and Hamming distance between the reference (digital or physical) and the corresponding original printed code. The visualisation is given for the samples from the Indigo mobile test sub-set.

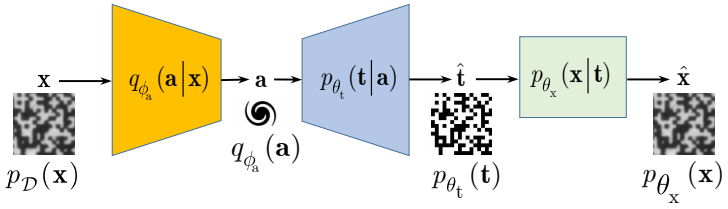
the training parameters and each run classification error is given in Appendix D.2.1. The visualisation of the OC-SVM decision boundaries is illustrated in Fig. 4.20.

Analyzing the obtained results, at first, it should be pointed out that the OC-SVM classification error based on the P_{miss} and P_{fa} is quite high and unacceptable for all types of used training data. At the same time two important conclusions can be done:

- With respect to the chosen metrics the use of the digital templates is preferable than the printed references.
- Despite the monochrome nature of the printed codes, the color information related to the enrollment via mobile phone might be useful for training.



(a) The general scheme of auto-encoding model.



(b) A particular scheme considered in Thesis with respect to the given reference codes $\mathbf{t}$.

Fig. 4.21 Auto-encoding model based on the IB principle: the input $\mathbf{x}$ is "compressed" to $\mathbf{z}/\mathbf{a}$ via the parametrized mapping $q_\phi(\mathbf{z}|\mathbf{x})/q_{\phi_a}(\mathbf{a}|\mathbf{x})$ leading to a bottleneck representation $\mathbf{z}/\mathbf{a}$.

4.5.2 One-class classification from the IB point of view

Aiming at answering the research question Q.3 from Section 4.3 about the efficiency of one-class classification in a deep processing domain, we consider a one-class classification based on the features extracted via DNN processing. More particularly, we consider a DNN auto-encoding. The general scheme of auto-encoding model is shown in Fig. 4.21(a). In a particular case of the PGC authentication, where the reference templates $\mathbf{t}$ are given, the auto-encoding can take more complex form as illustrated in Fig. 4.21(b) and can be considered as a "compression" of $\mathbf{x}$ to $\mathbf{a}$ via the parametrized mapping $q_{\phi_a}(\mathbf{a}|\mathbf{x})$ leading to a bottleneck representation $\mathbf{a}$ yet preserving a certain level of information I_t about $\mathbf{t}$ in the latent representation $\mathbf{a}$ and the information I_x about $\mathbf{x}$ in $\mathbf{t}$. Accordingly, it can be formulated as:

$$\min_{\phi_a: \begin{cases} I(\mathbf{A};\mathbf{T}) \geq I_t \\ I(\mathbf{T};\mathbf{X}) \geq I_x \end{cases}} I_{\phi_a}(\mathbf{X};\mathbf{A}), \tag{4.2}$$

and in the Lagrangian formulation as a minimization of:

$$\mathcal{L}(\phi_a) = I_{\phi_a}(\mathbf{X};\mathbf{A}) - \beta_t I(\mathbf{A};\mathbf{T}) - \beta_x I(\mathbf{T};\mathbf{X}), \tag{4.3}$$

where β_t and β_x are the regularization parameters corresponding to I_t and I_x respectively.

Decomposition of the first term

The first mutual information term $I_{\phi_a}(\mathbf{X};\mathbf{A})$ in (4.3) is defined as:

$$
\begin{aligned}
I_{\phi_{\mathrm{a}}}(\mathbf{X};\mathbf{A}) &= \mathbb{E}_{q_{\phi_{\mathrm{a}}}(\mathbf{a},\mathbf{x})}\left[\log \frac{q_{\phi_{\mathrm{a}}}(\mathbf{a},\mathbf{x})}{q_{\phi_{\mathrm{a}}}(\mathbf{a})p_{\mathcal{D}}(\mathbf{x})}\right] = \mathbb{E}_{q_{\phi_{\mathrm{a}}}(\mathbf{a},\mathbf{x})}\left[\log \frac{q_{\phi_{\mathrm{a}}}(\mathbf{a}|\mathbf{x})}{q_{\phi_{\mathrm{a}}}(\mathbf{a})}\right] \\
&= \underbrace{H_{\phi_{\mathrm{a}}}(\mathbf{A})}_{=\text{ constant}} - \underbrace{H_{\phi_{\mathrm{a}}}(\mathbf{A}|\mathbf{X})}_{=0},
\end{aligned}
\tag{4.4}
$$

where $p_{\mathcal{D}}(\mathbf{x})$ denotes the data distribution of printed codes and $q_{\phi_{\mathrm{a}}}(\mathbf{a},\mathbf{x})$ is a joint distribution, $H_{\phi_{\mathrm{a}}}(\mathbf{A}|\mathbf{X}) = -\mathbb{E}_{q_{\phi_{\mathrm{a}}}(\mathbf{a},\mathbf{x})}\left[\log q_{\phi_{\mathrm{a}}}(\mathbf{a}|\mathbf{x})\right]$ denotes the conditional entropy defined by $q_{\phi_{\mathrm{a}}}(\mathbf{a}|\mathbf{x})$. Due to the deterministic encoding $q_{\phi_{\mathrm{a}}}(\mathbf{a}|\mathbf{x}) = \delta(\mathbf{a} - f_{\phi_{\mathrm{a}}}(\mathbf{x}))$, it equals to zero. $H_{\phi_{\mathrm{a}}}(\mathbf{A}) = -\mathbb{E}_{q_{\phi_{\mathrm{a}}}(\mathbf{a})}\left[\log q_{\phi_{\mathrm{a}}}(\mathbf{a})\right]$ denotes the entropy of distribution $q_{\phi_{\mathrm{a}}}(\mathbf{a})$ and it is constant since, in the considered setup, any particular constraints on the latent space $\mathbf{a}$ are not applied. Therefore, for a given architecture of the encoder and defined latent space $\mathbf{a}$, the minimization of the term $I_{\phi_{\mathrm{a}}}(\mathbf{X};\mathbf{A})$ does not play any role.

Decomposition of the second term

The second mutual information term $I(\mathbf{A};\mathbf{T})$ in (4.3) is defined as:

$$
I(\mathbf{A};\mathbf{T}) = \mathbb{E}_{p(\mathbf{a},\mathbf{t})}\left[\log \frac{p(\mathbf{a},\mathbf{t})}{p_{\mathcal{D}}(\mathbf{t})p(\mathbf{a})}\right] = \mathbb{E}_{p(\mathbf{a},\mathbf{t})}\left[\log \frac{p(\mathbf{t}|\mathbf{a})}{p_{\mathcal{D}}(\mathbf{t})}\right], \tag{4.5}
$$

where $p(\mathbf{a},\mathbf{t})$ is a joint distribution, $p(\mathbf{a})$ and $p_{\mathcal{D}}(\mathbf{t})$ are marginals for the latent space and digital templates. It is important to highlight that $p(\mathbf{t}|\mathbf{a})$ is not defined and we proceed with its parametrization via a network $p_{\boldsymbol{\theta}_{\mathrm{t}}}(\mathbf{t}|\mathbf{a})$.

As it is shown in Appendix D.2.2.1, the mutual information (4.5) can be lower bounded by $I(\mathbf{A};\mathbf{T}) \geq I_{\boldsymbol{\theta}_{\mathrm{t}},\phi_{\mathrm{a}}}(\mathbf{A};\mathbf{T})$, where:

$$
\begin{aligned}
I_{\boldsymbol{\theta}_{\mathrm{t}},\phi_{\mathrm{a}}}(\mathbf{A};\mathbf{T}) &\triangleq -\mathbb{E}_{p_{\mathcal{D}}(\mathbf{t})}\left[\log p_{\mathcal{D}}(\mathbf{t})\right] + \mathbb{E}_{p(\mathbf{t},\mathbf{x})}\left[\mathbb{E}_{q_{\phi_{\mathrm{a}}}(\mathbf{a}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}_{\mathrm{t}}}(\mathbf{t}|\mathbf{a})\right]\right] \\
&= H_{\mathcal{D}}(\mathbf{T}) - H_{\boldsymbol{\theta}_{\mathrm{t}},\phi_{\mathrm{a}}}(\mathbf{T}|\mathbf{A}),
\end{aligned}
\tag{4.6}
$$

where $H_{\mathcal{D}}(\mathbf{T}) = -\mathbb{E}_{p_{\mathcal{D}}(\mathbf{t}))}\left[\log p_{\mathcal{D}}(\mathbf{t})\right]$ and $H_{\boldsymbol{\theta}_{\mathrm{t}},\phi_{\mathrm{a}}}(\mathbf{T}|\mathbf{A}) = -\mathbb{E}_{p(\mathbf{t},\mathbf{x})}\left[\mathbb{E}_{q_{\phi_{\mathrm{a}}}(\mathbf{a}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}_{\mathrm{t}}}(\mathbf{t}|\mathbf{a})\right]\right]$.

It is important to ensure that the estimated templates closely follow the statistics of original digital templates. That is why one can also consider the decomposition of (4.6) as:

$$
\begin{aligned}
I_{\boldsymbol{\theta}_{\mathrm{t}},\phi_{\mathrm{a}}}(\mathbf{A};\mathbf{T}) &= \mathbb{E}_{p(\mathbf{t},\mathbf{x})}\left[\mathbb{E}_{q_{\phi_{\mathrm{a}}}(\mathbf{a}|\mathbf{x})}\left[\log \frac{p_{\boldsymbol{\theta}_{\mathrm{t}}}(\mathbf{t}|\mathbf{a})}{p_{\mathcal{D}}(\mathbf{t})}\right]\right] \\
&= \mathbb{E}_{p(\mathbf{t},\mathbf{x})}\left[\mathbb{E}_{q_{\phi_{\mathrm{a}}}(\mathbf{a}|\mathbf{x})}\left[\log \frac{p_{\boldsymbol{\theta}_{\mathrm{t}}}(\mathbf{t}|\mathbf{a})}{p_{\mathcal{D}}(\mathbf{t})}\frac{p_{\boldsymbol{\theta}_{\mathrm{t}}}(\mathbf{t})}{p_{\boldsymbol{\theta}_{\mathrm{t}}}(\mathbf{t})}\right]\right] \\
&= -\mathbb{E}_{p_{\mathcal{D}}(\mathbf{t})}\left[\log p_{\boldsymbol{\theta}_{\mathrm{t}}}(\mathbf{t})\right] - \mathbb{E}_{p_{\mathcal{D}}(\mathbf{t})}\left[\log \frac{p_{\mathcal{D}}(\mathbf{t})}{p_{\boldsymbol{\theta}_{\mathrm{t}}}(\mathbf{t})}\right] + \mathbb{E}_{p(\mathbf{t},\mathbf{x})}\left[\mathbb{E}_{q_{\phi_{\mathrm{a}}}(\mathbf{a}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}_{\mathrm{t}}}(\mathbf{t}|\mathbf{a})\right]\right] \\
&= H(p_{\mathcal{D}}(\mathbf{t});p_{\boldsymbol{\theta}_{\mathrm{t}}}(\mathbf{t})) - D_{\mathrm{KL}}\left(p_{\mathcal{D}}(\mathbf{t})\|p_{\boldsymbol{\theta}_{\mathrm{t}}}(\mathbf{t})\right) + \mathbb{E}_{p(\mathbf{t},\mathbf{x})}\left[\mathbb{E}_{q_{\phi_{\mathrm{a}}}(\mathbf{a}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}_{\mathrm{t}}}(\mathbf{t}|\mathbf{a})\right]\right],
\end{aligned}
\tag{4.7}
$$

where $H(p_{\mathcal{D}}(\mathbf{t}); p_{\boldsymbol{\theta}_\mathrm{t}}(\mathbf{t})) = -\mathbb{E}_{p_{\mathcal{D}}(\mathbf{t})}\left[\log p_{\boldsymbol{\theta}_\mathrm{t}}(\mathbf{t})\right]$ denotes a cross-entropy. Since $H(p_{\mathcal{D}}(\mathbf{t}); p_{\boldsymbol{\theta}_\mathrm{t}}(\mathbf{t})) \geq 0$, one can lower bound (4.7) as $I_{\boldsymbol{\theta}_\mathrm{t},\boldsymbol{\phi}_\mathrm{a}}(\mathbf{A};\mathbf{T}) \geq I^L_{\boldsymbol{\theta}_\mathrm{t},\boldsymbol{\phi}_\mathrm{a}}(\mathbf{A};\mathbf{T})$, where[11]:

$$I^L_{\boldsymbol{\theta}_\mathrm{t},\boldsymbol{\phi}_\mathrm{a}}(\mathbf{A};\mathbf{T}) \triangleq \underbrace{\mathbb{E}_{p(\mathbf{t},\mathbf{x})}\left[\mathbb{E}_{q_{\boldsymbol{\phi}_\mathrm{a}}(\mathbf{a}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}_\mathrm{t}}(\mathbf{t}|\mathbf{a})\right]\right]}_{\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}}} - \underbrace{D_{\mathrm{KL}}\left(p_{\mathcal{D}}(\mathbf{t}) \| p_{\boldsymbol{\theta}_\mathrm{t}}(\mathbf{t})\right)}_{\mathcal{D}_\mathrm{t}}. \tag{4.8}$$

Remark: The term $\mathcal{D}_\mathrm{t}$ in (4.8) can be implemented based on the density ratio estimation [148]. The term $\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}}$ can be defined explicitly using Gaussian or Laplacian priors. In the Gaussian case, one can define $p_{\boldsymbol{\theta}_\mathrm{t}}(\mathbf{t}|\mathbf{a}) \propto \exp(-\lambda\|\mathbf{t} - g_{\boldsymbol{\theta}_\mathrm{t}}(\mathbf{a})\|_2)$ with a scale parameter λ, which leads to ℓ_2-norm, and $g_{\boldsymbol{\theta}}(\mathbf{a})$ denotes the decoder. It also corresponds to the model $\mathbf{t} = g_{\boldsymbol{\theta}_\mathrm{t}}(\mathbf{a}) + \mathbf{e}_\mathrm{a}$, where $\mathbf{e}_\mathrm{a}$ is a reconstruction error vector following the Gaussian pdf. Therefore, (4.8) reduces to:

$$I^L_{\boldsymbol{\theta}_\mathrm{t},\boldsymbol{\phi}_\mathrm{a}}(\mathbf{A};\mathbf{T}) = \underbrace{-\lambda\mathbb{E}_{p(\mathbf{t},\mathbf{x})}\left[\mathbb{E}_{q_{\boldsymbol{\phi}_\mathrm{a}}(\mathbf{a}|\mathbf{x})}\left[\|\mathbf{t} - g_{\boldsymbol{\theta}_\mathrm{t}}(\mathbf{a})\|_2\right]\right]}_{\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}}} - \underbrace{D_{\mathrm{KL}}\left(p_{\mathcal{D}}(\mathbf{t}) \| p_{\boldsymbol{\theta}_\mathrm{t}}(\mathbf{t})\right)}_{\mathcal{D}_\mathrm{t}}. \tag{4.9}$$

Decomposition of the third term

The third mutual information $I(\mathbf{T};\mathbf{X})$ in (4.3) can be decomposed in a way similar to the second term:

$$I(\mathbf{T};\mathbf{X}) = \mathbb{E}_{p(\mathbf{t},\mathbf{x})}\left[\log \frac{p(\mathbf{t},\mathbf{x})}{p_{\mathcal{D}}(\mathbf{x})p_{\mathcal{D}}(\mathbf{t})}\right] = \mathbb{E}_{p(\mathbf{t},\mathbf{x})}\left[\log \frac{p(\mathbf{x}|\mathbf{t})}{p_{\mathcal{D}}(\mathbf{x})}\right]. \tag{4.10}$$

Accordingly, it can be lower bounded by $I(\mathbf{T};\mathbf{X}) \geq I^L_{\boldsymbol{\theta}_\mathrm{t},\boldsymbol{\theta}_\mathrm{x}}(\mathbf{T};\mathbf{X})$, where:

$$I^L_{\boldsymbol{\theta}_\mathrm{t},\boldsymbol{\theta}_\mathrm{x}}(\mathbf{T};\mathbf{X}) \triangleq \underbrace{\mathbb{E}_{p(\mathbf{t},\mathbf{x})}\left[\mathbb{E}_{p_{\boldsymbol{\theta}_\mathrm{t}}(\mathbf{t}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}_\mathrm{x}}(\mathbf{x}|\mathbf{t})\right]\right]}_{\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}} - \underbrace{D_{\mathrm{KL}}\left(p_{\mathcal{D}}(\mathbf{x}) \| p_{\boldsymbol{\theta}_\mathrm{x}}(\mathbf{x})\right)}_{\mathcal{D}_\mathrm{x}}. \tag{4.11}$$

Summary

The final minimization problem schematically shown in Fig. 4.22 is:

$$\begin{aligned}(\hat{\boldsymbol{\phi}}_\mathrm{a}, \hat{\boldsymbol{\theta}}_\mathrm{t}, \hat{\boldsymbol{\theta}}_\mathrm{x}) &= \underset{\boldsymbol{\phi}_\mathrm{a},\boldsymbol{\theta}_\mathrm{t},\boldsymbol{\theta}_\mathrm{x}}{\operatorname{argmin}} \mathcal{L}^L(\boldsymbol{\phi}_\mathrm{a}, \boldsymbol{\theta}_\mathrm{t}, \boldsymbol{\theta}_\mathrm{x}) \\ &= \underset{\boldsymbol{\phi}_\mathrm{a},\boldsymbol{\theta}_\mathrm{t},\boldsymbol{\theta}_\mathrm{x}}{\operatorname{argmin}} -\beta_\mathrm{t}(\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}} - \mathcal{D}_\mathrm{t}) - \beta_\mathrm{x}(\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}} - \mathcal{D}_\mathrm{x}).\end{aligned} \tag{4.12}$$

[11] The cross-entropy computation requires knowledge of model $p_{\boldsymbol{\theta}_\mathrm{t}}(\mathbf{t})$, whereas the KL-divergence is based on the ratio of two distributions and can be computed without an explicit knowledge of distributions but only from the training samples.

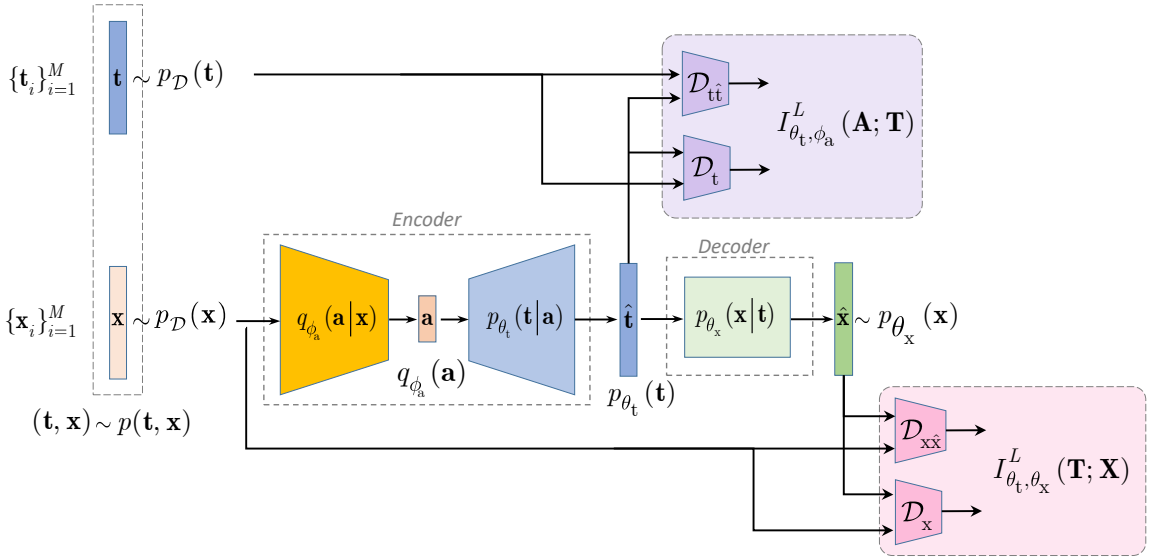


Fig. 4.22 The feature extraction for the one-class classification based on the estimation of the reference templates via $\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}}$ and $\mathcal{D}_{\mathrm{t}}$ and the printed codes via $\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$ and $\mathcal{D}_{\mathrm{x}}$ terms.

where

$$\begin{aligned}
\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}} &\triangleq \mathbb{E}_{p(\mathbf{t},\mathbf{x})}\left[\mathbb{E}_{q_{\boldsymbol{\phi}_{\mathrm{a}}}(\mathbf{a}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}_{\mathrm{t}}}(\mathbf{t}|\mathbf{a})\right]\right], \\
\mathcal{D}_{\mathrm{t}} &\triangleq D_{\mathrm{KL}}\left(p_{\mathcal{D}}(\mathbf{t}) \| p_{\boldsymbol{\theta}_{\mathrm{t}}}(\mathbf{t})\right), \\
\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}} &\triangleq \mathbb{E}_{p(\mathbf{t},\mathbf{x})}\left[\mathbb{E}_{p_{\boldsymbol{\theta}_{\mathrm{t}}}(\mathbf{t}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}_{\mathrm{x}}}(\mathbf{x}|\mathbf{t})\right]\right], \\
\mathcal{D}_{\mathrm{x}} &\triangleq D_{\mathrm{KL}}\left(p_{\mathcal{D}}(\mathbf{x}) \| p_{\boldsymbol{\theta}_{\mathrm{x}}}(\mathbf{x})\right).
\end{aligned} \tag{4.13}$$

In practice four basic scenarios of features extractors for the one-class classification are considered:

1. The reference templates estimation based on the term $\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}}$:

$$\mathcal{L}_1(\boldsymbol{\phi}_{\mathrm{a}}, \boldsymbol{\theta}_{\mathrm{t}}) = -\beta_{\mathrm{t}}\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}}. \tag{4.14}$$

2. The reference templates estimation based on the terms $\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}}$ and $\mathcal{D}_{\mathrm{t}}$:

$$\mathcal{L}_2(\boldsymbol{\phi}_{\mathrm{a}}, \boldsymbol{\theta}_{\mathrm{t}}) = -\beta_{\mathrm{t}}\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}} + \beta_{\mathrm{t}}\mathcal{D}_{\mathrm{t}}. \tag{4.15}$$

3. The estimation of the reference templates and the printed codes based on terms $\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}}$ and $\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$:

$$\mathcal{L}_3(\boldsymbol{\phi}_{\mathrm{a}}, \boldsymbol{\theta}_{\mathrm{t}}, \boldsymbol{\theta}_{\mathrm{x}}) = -\beta_{\mathrm{t}}\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}} - \beta_{\mathrm{x}}\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}. \tag{4.16}$$

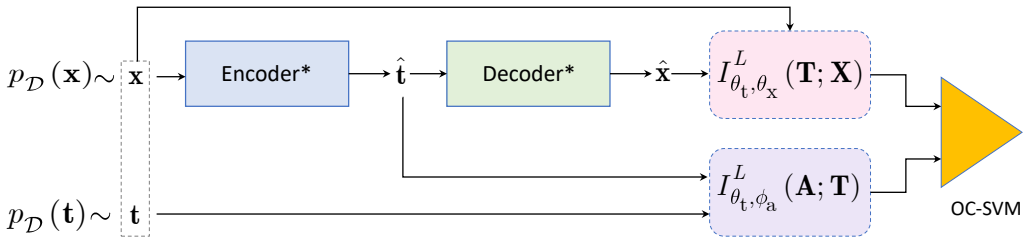


Fig. 4.23 The one-class classification training procedure: the encoder and decoder parts of the auto-encoder model shown in Fig. 4.22 are pre-trained and fixed (as indicated by a "*"); the OC-SVM is trained on the outputs of $\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}}$ and $\mathcal{D}_{\mathrm{t}}$ terms that are the results of $I^L_{\boldsymbol{\theta}_\mathrm{t},\boldsymbol{\phi}_\mathrm{a}}(\mathbf{A};\mathbf{T})$ decomposition and the $\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$ and $\mathcal{D}_{\mathrm{x}}$ terms that are the results of $I^L_{\boldsymbol{\theta}_\mathrm{t},\boldsymbol{\theta}_\mathrm{x}}(\mathbf{T};\mathbf{X})$ decomposition.

4. The estimation of the reference templates and the printed codes based on terms $\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}}$, $\mathcal{D}_{\mathrm{t}}$, $\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$ and $\mathcal{D}_{\mathrm{x}}$:

$$\mathcal{L}_4(\boldsymbol{\phi}_\mathrm{a},\boldsymbol{\theta}_\mathrm{t},\boldsymbol{\theta}_\mathrm{x}) = -\beta_\mathrm{t}\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}} + \beta_\mathrm{t}\mathcal{D}_\mathrm{t} - \beta_\mathrm{x}\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}} + \beta_\mathrm{x}\mathcal{D}_\mathrm{x}. \tag{4.17}$$

In general case, to be comparable with the one-class classification in the spatial domain discussed in Section 4.5.1, a one-class classification model based on the OC-SVM is used.

The OC-SVM training procedure shown in Fig. 4.23 uses the pre-trained and fixed encoder and decoder parts of the auto-encoder model that serves as a features extractor. As an input the OC-SVM might take different combinations of outputs of four main terms: $\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}}$, $\mathcal{D}_{\mathrm{t}}$, $\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$ and $\mathcal{D}_{\mathrm{x}}$. The exact scenarios are discussed below in Sections 4.5.2.1 - 4.5.2.4.

The empirical evaluation is performed on the Indigo mobile dataset that is split into three sub-sets: *training* with 40% of data, *validation* with 10% of data and 50% of data is used for the *test*. To avoid the bias in the choice of training and test data, each scenario's model is trained five times under randomly chosen data. Moreover, it should be pointed out that, in each scenario, the training is performed only on the training sub-set of the digital templates $\mathbf{t}$ and the corresponding original printed codes $\mathbf{x}$. The test sub-set of the fake codes are used only at the inference stage.

4.5.2.1 $\mathcal{L}_1(\boldsymbol{\phi}_\mathrm{a},\boldsymbol{\theta}_\mathrm{t}) = -\beta_\mathrm{t}\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}}$

The optimization problem $\mathcal{L}_1(\boldsymbol{\phi}_\mathrm{a},\boldsymbol{\theta}_\mathrm{t}) = -\beta_\mathrm{t}\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}}$ aims for each input printed original code $\mathbf{x}$ to output an accurate estimation $\hat{\mathbf{t}}$ of the corresponding binary digital template $\mathbf{t}$. Taking into account that, due to the nature of the used trained model, the output estimation is real valued but not binary, at the inference stage, to measure the Hamming distance the final estimation $\hat{\mathbf{t}}$ is obtained after thresholding with a threshold 0.5. The detailed information about the model architecture and the training parameters is given in Appendices D.2.2.2 and D.2.2.3.

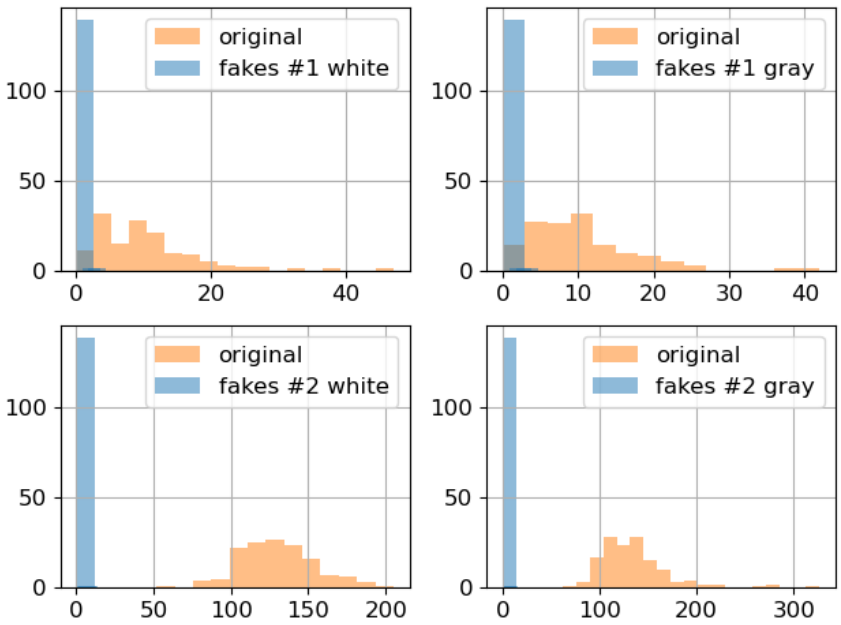


Fig. 4.24 The first scenario results visualization: the distribution of symbol-wise Hamming distance between the original digital templates $\mathbf{t}$ and the corresponding estimations $\hat{\mathbf{t}}$ obtained via the encoder model trained with respect to the $\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}}$ term.

Fig. 4.24 illustrates the distributions of the symbol-wise Hamming distance between the original digital templates $\mathbf{t}$ and the corresponding estimations $\hat{\mathbf{t}}$ obtained from the printed original and fake codes. Taking into account that the extracted feature vector consists only of one value, the OC-SVM is not used and the classification is performed based on the decision rule:

$$\begin{cases} P_{miss} &= \Pr\{d_{\text{Hamming}}(\mathbf{t}, \hat{\mathbf{t}}) > \gamma_1 \mid \mathcal{H}_0\}, \\ P_{fa} &= \Pr\{d_{\text{Hamming}}(\mathbf{t}, \hat{\mathbf{t}}) \leq \gamma_1 \mid \overline{\mathcal{H}}_0\}, \end{cases} \tag{4.18}$$

where P_{miss} is a probability of miss and P_{fa} is probability of false acceptance. The $\mathcal{H}_0$ corresponds to the hypothesis that the input code is original and the $\overline{\mathcal{H}}_0$ corresponds to the hypothesis that the input code is fake. Aiming to have $P_{miss} = 0$, the decision constant γ_1 is determined on the validation sub-set to be equal to 2 symbols. The average (over five runs) classification error is given in Table 4.7. Each run detailed classification results are given in Table D.11 in Appendix D.2.2.3.

From the obtained results it is easy to see that the one-class classification based on the encoder model trained with respect to the $\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}}$ term as shown in Fig. 4.22 allows to distinguish the originals and the fakes #2 with 100% accuracy. The obtained P_{miss} and P_{fa} are confirmed by the distribution of the Hamming distance shown in Fig. 4.24. In case of the fakes #1 the corresponding distributions overlap and the P_{fa} is about 6 - 8%.

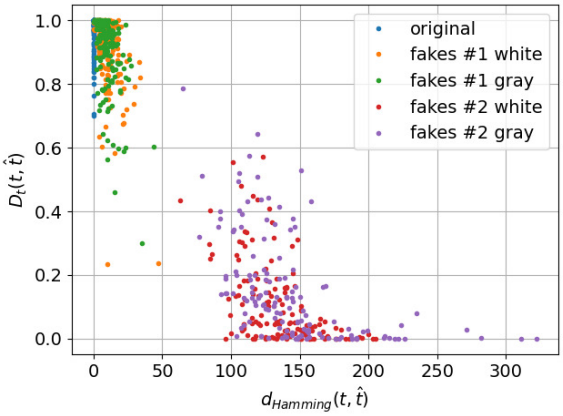


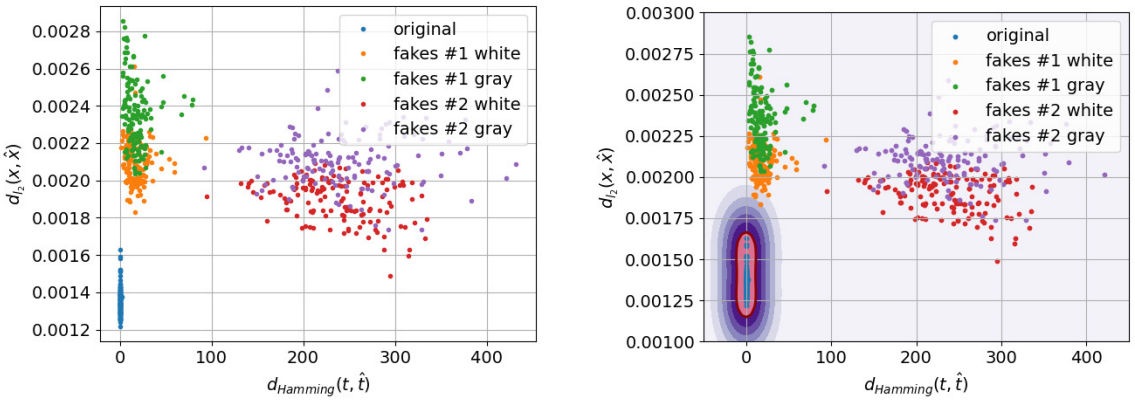
Fig. 4.25 The second scenario results visualisation: the 2D distribution of *(i)* the symbol-wise Hamming distance between the original digital templates $\mathbf{t}$ and the corresponding estimations $\hat{\mathbf{t}}$ obtained via the encoder model trained with respect to the $\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}}$ term and *(ii)* the corresponding responses of the discriminator model trained with respect to the $\mathcal{D}_{\mathrm{t}}$ term.

4.5.2.2 $\mathcal{L}_2(\boldsymbol{\phi}_{\mathrm{a}}, \boldsymbol{\theta}_{\mathrm{t}}) = -\beta_{\mathrm{t}}\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}} + \beta_{\mathrm{t}}\mathcal{D}_{\mathrm{t}}$

The optimization problem $\mathcal{L}_2(\boldsymbol{\phi}_{\mathrm{a}}, \boldsymbol{\theta}_{\mathrm{t}}) = -\beta_{\mathrm{t}}\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}} + \beta_{\mathrm{t}}\mathcal{D}_{\mathrm{t}}$ is an extension of the scenario 4.5.2.1 by the discriminator part $\mathcal{D}_{\mathrm{t}}$ that aims to distinguish between the original digital templates and the corresponding estimations. The detailed information about the training procedure and architecture of the feature extraction model are given in Appendices D.2.2.2 and D.2.2.4.

Fig. 4.25 presents the 2D distribution of *(i)* the symbol-wise Hamming distance between the original digital templates $\mathbf{t}$ and the corresponding estimations $\hat{\mathbf{t}}$ obtained based on the encoder model trained with respect to the $\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}}$ term and *(ii)* the corresponding responses of the discriminator trained with respect to the $\mathcal{D}_{\mathrm{t}}$ term as shown in Fig. 4.22. It is easy to see that, with respect to the Hamming distance, the obtained results are very close to those in Fig. 4.24, namely, the results for the original codes are close to zero and overlap with the fakes #1, while the fakes #2 are well separable. With respect to the $\mathcal{D}_{\mathrm{t}}$ discriminator decision the situation is similar too, namely, the fakes #2 are well recognizable by the decision ratio smaller then 0.5 - 0.6. At the same time, for the the fakes #1 the decision ratio is mostly bigger than 0.7 - 0.8 as well as for the originals.

The average (over five runs) authentication error based on the P_{miss} and P_{fa} calculated with respect to the decision rule (4.18) and given in Table 4.7 shows that the regularization via the discriminator $\mathcal{D}_{\mathrm{t}}$ does not have big influence and does not allow to improve the authentication error. Each run classification results are given in Table D.12 in Appendix D.2.2.4.



(a) (b)

Fig. 4.26 The third scenario results visualization: (a) the distribution of *(i)* the symbol-wise Hamming distance between the digital templates and its corresponding estimations via the encoder model trained with respect to the $\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}}$ term and *(ii)* the ℓ_2 distance between the printed codes and its corresponding reconstructions by the decoder model trained with respect to the $\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$ term; (b) the OC-SVM decision boundaries.

4.5.2.3 $\mathcal{L}_3(\boldsymbol{\phi}_{\mathrm{a}}, \boldsymbol{\theta}_{\mathrm{t}}, \boldsymbol{\theta}_{\mathrm{x}}) = -\beta_{\mathrm{t}}\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}} - \beta_{\mathrm{x}}\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$

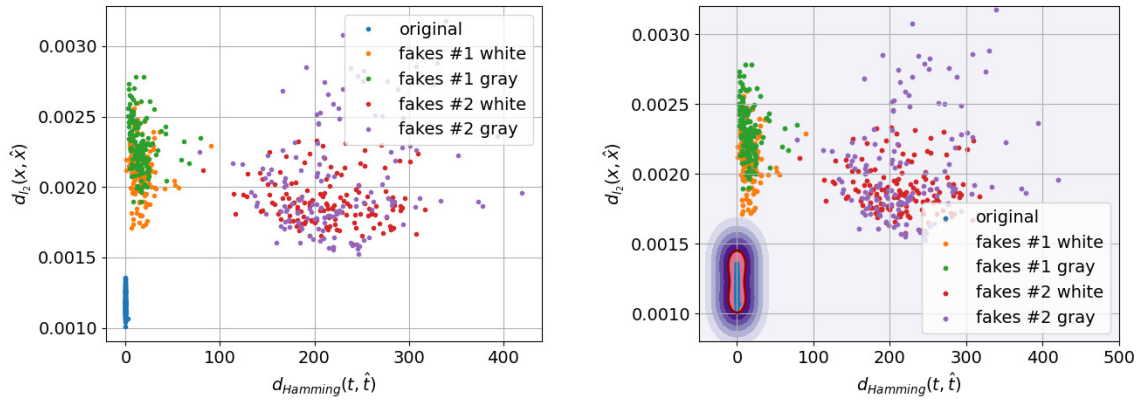
In the third scenario the term $\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$ is in charge of the printed codes reconstruction and plays a role of a learnable regularization similarly to those discussed in Section 3.4. The detailed information about the training procedure and architecture of the trained models are given in Appendices D.2.2.2 and D.2.2.5.

Fig. 4.26(a) demonstrates the obtained distribution of two metrics: the first one is the symbol-wise Hamming distance introduced in the Section 4.5.2.1 and the second one is a ℓ_2 error between the printed codes and the corresponding reconstructions obtained as an output of the decoder model trained with respect to the $\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$ term as shown in Fig. 4.22 without any additional post-processing.

The average (over five runs) authentication results based on the decision rule (4.18) is given in Table 4.7. It is easy to see that the learnable regularization via $\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$ term preserves the P_{miss} and P_{fa} on the fakes #2 to be zero, similar to the previous scenarios. At the same time, it allows to decrease the P_{fa} on the fakes #1 from 7% till 1-1.6%. Additionally, in Table 4.7 there is given the average (over five runs) authentication results based on the two metrics decision rule:

$$\begin{cases} P_{miss} &= \Pr\{d_{\mathrm{Hamming}}(\mathbf{t}, \hat{\mathbf{t}}) > \gamma_1 \;\&\; d_{\ell_2}(\mathbf{x}, \hat{\mathbf{x}}) > \gamma_2 \mid \mathcal{H}_0\} \\ P_{fa} &= \Pr\{d_{\mathrm{Hamming}}(\mathbf{t}, \hat{\mathbf{t}}) \leq \gamma_1 \;\&\; d_{\ell_2}(\mathbf{x}, \hat{\mathbf{x}}) \leq \gamma_2 \mid \overline{\mathcal{H}}_0\}, \end{cases} \tag{4.19}$$

that allows to significantly reduce the P_{fa} for the fakes #1 to about 0.28%. Aiming to have the $P_{miss} = 0$, the decision constant γ_2 is determined on the validation sub-set to be equal



(a) (b)

Fig. 4.27 The fourth scenario results visualization: (a) the distribution of *(i)* the symbol-wise Hamming distance between the digital templates and its corresponding estimations via the encoder model trained with respect to the $\mathcal{D}_{t\hat{t}}$ term and *(ii)* the ℓ_2 distance between the printed codes and its corresponding reconstructions by the decoder model trained with respect to the $\mathcal{D}_{x\hat{x}}$ term; (b) the OC-SVM decision boundaries.

0.0017 and the γ_1 equals to 2 symbols. Each run detailed results are given in Table D.13 in Appendix D.2.2.5.

In addition, Table 4.7 includes the results of OC-SVM trained with respect to the metrics under investigation (the symbol-wise Hamming distance between the digital templates and its corresponding estimations via the encoder model trained with respect to the $\mathcal{D}_{t\hat{t}}$ term and the ℓ_2 distance between the printed codes and its corresponding reconstructions by the decoder model trained with respect to the $\mathcal{D}_{x\hat{x}}$ term). The OC-SVM is trained only on the train sub-set of the original printed codes $\mathbf{x}$ and its corresponding templates $\mathbf{t}$. The example of OC-SVM decision boundaries is illustrated in Fig. 4.26(b). The OC-SVM reduces the P_{fa} to 0% for all types of fakes. However, the P_{miss} increases to about 0.28% in contrast to the previously obtained results with $P_{miss} = 0\%$.

4.5.2.4 $\mathcal{L}_4(\boldsymbol{\phi}_a, \boldsymbol{\theta}_t, \boldsymbol{\theta}_x) = -\beta_t \mathcal{D}_{t\hat{t}} + \beta_t \mathcal{D}_t - \beta_x \mathcal{D}_{x\hat{x}} + \beta_x \mathcal{D}_x$

The last considered scenario includes four terms: the main term $\mathcal{D}_{t\hat{t}}$, the discriminator $\mathcal{D}_t$ on the digital templates estimation space, the printed codes reconstruction space regularization $\mathcal{D}_{x\hat{x}}$ and the discriminator $\mathcal{D}_x$. The detailed information about the training procedure and the models architectures are given in Appendices D.2.2.2 and D.2.2.6. Similarly to the third scenario, the OC-SVM is trained with respect to the two features: *(i)* the symbol-wise Hamming distance between the original digital templates and its estimations and *(ii)* the ℓ_2 distance between the printed codes and its reconstructions. A visual representation of the mutual distribution of these metrics is shown in Fig. 4.27(a). Table 4.7 includes the average (over five runs) one-class classification error based on three criteria: the decision rules (4.18)

Table 4.7 The average (over five runs) authentication error in % on the test sub-set.

Model	Original	Fake #1 white	Fake #1 gray	Fake #2 white	Fake #2 gray
	P_{miss}	P_{fa}	P_{fa}	P_{fa}	P_{fa}
Based on the equation (4.18)					
$-\beta_{\mathrm{t}}\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}}$	0	6.38 (±2.4)	8.23 (±2.95)	0	0
$-\beta_{\mathrm{t}}\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}} + \beta_{\mathrm{t}}\mathcal{D}_{\mathrm{t}}$	0	6.81 (±1.63)	7.09 (±2.4)	0	0
$-\beta_{\mathrm{t}}\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}} - \beta_{\mathrm{x}}\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$	0	1.56 (±0.32)	0.99 (±0.81)	0	0
$-\beta_{\mathrm{t}}\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}} + \beta_{\mathrm{t}}\mathcal{D}_{\mathrm{t}} - \beta_{\mathrm{x}}\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}} + \beta_{\mathrm{x}}\mathcal{D}_{\mathrm{x}}$	0	2.41 (±1.38)	2.13 (±1.59)	0	0
Based on the equation (4.19)					
$-\beta_{\mathrm{t}}\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}} - \beta_{\mathrm{x}}\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$	0	0.28 (±0.64)	0	0	0
$-\beta_{\mathrm{t}}\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}} + \beta_{\mathrm{t}}\mathcal{D}_{\mathrm{t}} - \beta_{\mathrm{x}}\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}} + \beta_{\mathrm{x}}\mathcal{D}_{\mathrm{x}}$	0.57 (±1.27)	0	0.14 (±0.32)	0	0
Based on the OC-SVM					
$-\beta_{\mathrm{t}}\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}} - \beta_{\mathrm{x}}\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$	0.28 (±0.39)	0	0	0	0
$-\beta_{\mathrm{t}}\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}} + \beta_{\mathrm{t}}\mathcal{D}_{\mathrm{t}} - \beta_{\mathrm{x}}\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}} + \beta_{\mathrm{x}}\mathcal{D}_{\mathrm{x}}$	0.14 (±0.32)	0	0	0	0

and (4.19) and the OC-SVM. The example of OC-SVM decision boundaries is illustrated in Fig. 4.27(b). Each run classification results are given in Table D.14 in Appendix D.2.2.6.

From the obtained results, one can note that in terms of decision rule (4.18), the regularization via $\mathcal{D}_{\mathrm{t}}$ and $\mathcal{D}_{\mathrm{x}}$ discriminators is counter-productive and makes the classification error bigger in comparison with the third scenario. In case of the decision rule (4.19) the regularization leads to a significant increase of P_{miss}. At the same time, the OC-SVM allows to decrease P_{miss} in two times, from 0.28% to 0.14% preserving P_{fa} equals to zero for all types of fakes.

In summary, it should be pointed out that, despite the great performance of the fourth scenario's model, its complexity is times higher compared with the other considered scenarios. The execution time complexity in hours per 100 training epochs is given in Table 4.8 for each scenario.

4.5.3 Conclusions

The performed analysis of PGC authentication based on the one-class classification using Indigo mobile dataset shows that:

- In view of the great similarity between the original and fake codes the authentication in the spatial domain *(i)* is difficult with respect to the finding of right metrics and *(ii)* is not reliable enough due to the high overlapping between the classes.
- The authentication with respect to the digital templates is more efficient compared to the authentication with respect to the physical references.
- Despite the visually grayscale nature of the PGC the authentication based on codes taken by the mobile phone in color mode is more efficient compared to the grayscale

Table 4.8 Execution time (hours) per 100 epochs on one NVIDIA GPU with a learning rate 1e-4 for the considered scenarios.

Model	Execution time
$-\beta_{\mathrm{t}}\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}}$	2.78 - 3.05
$-\beta_{\mathrm{t}}\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}} + \beta_{\mathrm{t}}\mathcal{D}_{\mathrm{t}}$	5.12 - 5.25
$-\beta_{\mathrm{t}}\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}} - \beta_{\mathrm{x}}\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$	5.56 - 5.83
$-\beta_{\mathrm{t}}\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}} + \beta_{\mathrm{t}}\mathcal{D}_{\mathrm{t}} - \beta_{\mathrm{x}}\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}} + \beta_{\mathrm{x}}\mathcal{D}_{\mathrm{x}}$	11.11 - 11.39

mode due to the fact that the different color channels have different sensitivity and due to the information loss while converting a three-channels color image into a single-channel grayscale one.

- The authentication with respect to the DNN estimation of the digital templates and printed codes reconstruction is more efficient than the direct authentication with respect to the digital and printed codes in spatial domain.
- The main disadvantage of the DNN based models is its high training complexity compared to the direct authentication in spatial domain.
- At the same time, at the inference stage, the trained models are equivalent in terms of authentication complexity to the authentication in spatial domain.

Besides the impressive performance of the one-class classification on real samples and mobile phone verification, it should be pointed out that the above analysis is done with respect to the typical HC copy attacks. In view of the widespread use of the ML technologies, the question about the robustness to the ML attacks is an important problem nowadays. That is why the next section addresses this problem in details.

4.6 ML fakes authentication

The results obtained in Sections 4.4 and 4.5 demonstrate a high accuracy of the authentication of PGC with respect to the typical HC copy fakes. At the same time, during the last several years the ML based attacks attracted a lot of attention and are of the great interest due to their domain adaptation, especially in view of the recent advents in the theory and practice of machine learning tools.

The clonability aspects of copy detection patterns used in the PGC are defined by many factors, such as, for example, the printing and scanning equipment, the size and the individual characteristics of the used copy detection elements (symbols), etc.

Thus, the main goal of this Section is to investigate:

(a) Full area estimation. (b) Half area estimation.

Fig. 4.28 Hand-crafted digital templates estimation methods.

- the influence of the printing and scanning equipment on the clonability aspects of PGC from the side of the attacker (the research questions Q.4 and Q.5 from Section 4.3);
- the PGC authentication accuracy against the ML fakes from the side of the defender (the research questions Q.6 and Q.7 from Section 4.3).

In this respect the experiments are performed on the specially designed DP1C and DP1E datasets represented in Section 4.2.1. The two HC attacks are taken as a baseline for the comparison.

4.6.1 Details of the setup

Hand-crafted (HC) setup

In contrast to the simple HC copy fakes produced by the use of copy machines considered in Sections 4.4 and 4.5, the setup considered in this Section is focused on the fakes' production based on the hand-crafted estimation of the original digital templates from the corresponding printed counterparts with the following re-printing of the estimated templates. In such an estimation setup, the attacker uses the prior knowledge about the quantity, shape and size of the copy detection pattern symbols that might be estimated from the printed codes. Using this knowledge the two HC setups are investigated:

- The simplest method to estimate the digital template from the printed counterpart is to calculate the mean of symbol's intensity from the full symbol area as shown in Fig. 4.28(a).

- The second method uses the mean intensity of the central part of symbol as shown in Fig. 4.28(b). This is expected to compensate to a certain extent the dot gain and the sampling jitter.

Finally, the hard decision threshold 0.5 is used. However, in general case, the other priors, like for example the median value of each symbol intensities, can be easily adopted into the estimation.

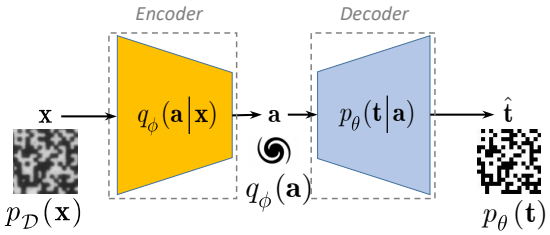


Fig. 4.29 The general scheme of ML estimation of digital templates from the printed counterparts based on the IB principle.

Machine learning (ML) setup

The ML fakes considered in the current Section are based on the same idea of digital templates estimation from the printed counterparts similar to the discussed HC setup. However, in contrast to the HC approaches, the machine learning setup assumes that any prior information about the amount and size of symbols is not taking into account in a direct way to perform some pre-processing. At the same time, this prior information is taken into account in an indirect way: it is assumed that, besides the publicly available printed codes $\{\mathbf{x}_i\}_{i=1}^M$, the attacker has an access to the corresponding original digital templates $\{\mathbf{t}_i\}_{i=1}^M$. There are various ways to obtain these training pairs. As one possible scenario, one can assume that the attacker can print the digital templates $\{\mathbf{t}_i\}_{i=1}^M$ on the same printer as the defender and scan them as $\{\mathbf{x}_i\}_{i=1}^M$. Such a setup allows to fully explore the power of training on the side of attacker.

In the general case, for the given pairs of digital and printed codes $\{\mathbf{t}_i, \mathbf{x}_i\}_{i=1}^M$[12] the templates' estimation task can be considered as an optimization problem based on the IB principle as illustrated in Fig. 4.29 that aims at "compressing" $\mathbf{x}$ into $\mathbf{a}$ via the parametrized mapping $q_{\boldsymbol{\phi}}(\mathbf{a}|\mathbf{x})$ leading to a bottleneck representation $\mathbf{a}$ yet preserving a certain level of information I_{t} about $\mathbf{t}$ in the latent representation $\mathbf{a}$. Accordingly, it can be formulated as:

$$\min_{\boldsymbol{\phi}: I(\mathbf{A};\mathbf{T}) \geq I_{\mathrm{t}}} I_{\boldsymbol{\phi}}(\mathbf{X};\mathbf{A}), \tag{4.20}$$

and in the Lagrangian formulation as a minimization of:

$$\mathcal{L}(\boldsymbol{\phi}) = I_{\boldsymbol{\phi}}(\mathbf{X};\mathbf{A}) - \beta I(\mathbf{A};\mathbf{T}), \tag{4.21}$$

where β is the regularization parameters corresponding to I_{t}.

[12] M equals to the amount of pairs of original digital and printed codes. For simplicity it is assumed that there is given the same amount of pairs for all printers.

Following the decompositions discussed in Sections 4.5.2 it is easy to show that:

- The first mutual information can be decomposed as:

$$I_{\boldsymbol{\phi}}(\mathbf{X};\mathbf{A}) = \underbrace{H_{\boldsymbol{\phi}}(\mathbf{A})}_{=\text{ constant}} - \underbrace{H_{\boldsymbol{\phi}}(\mathbf{A}|\mathbf{X})}_{=0}, \tag{4.22}$$

  where $H_{\boldsymbol{\phi}}(\mathbf{A}|\mathbf{X}) = -\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{a},\mathbf{x})}\left[\log q_{\boldsymbol{\phi}}(\mathbf{a}|\mathbf{x})\right]$ denotes the conditional entropy defined by $q_{\boldsymbol{\phi}}(\mathbf{a}|\mathbf{x})$. It equals to zero due to the deterministic encoding. $H_{\boldsymbol{\phi}}(\mathbf{A}) = -\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{a})}\left[\log q_{\boldsymbol{\phi}}(\mathbf{a})\right]$ denotes the entropy of distribution $q_{\boldsymbol{\phi}}(\mathbf{a})$. For simplicity in the considered setup no particular constraints on the latent space $\mathbf{a}$ are not applied. In this respect, the $H_{\boldsymbol{\phi}}(\mathbf{A})$ is constant and has no interest for the optimization.

- The second mutual information can be lower bounded in the same way as in Sections 4.5.2 by $I(\mathbf{A};\mathbf{T}) \geq I^L_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{A};\mathbf{T})$, where:

$$I^L_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{A};\mathbf{T}) \triangleq \underbrace{\mathbb{E}_{p(\mathbf{t},\mathbf{x})}\left[\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{a}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{t}|\mathbf{a})\right]\right]}_{\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}}} - \underbrace{D_{\mathrm{KL}}\left(p_{\mathcal{D}}(\mathbf{t})\|p_{\boldsymbol{\theta}}(\mathbf{t})\right)}_{\mathcal{D}_{\mathrm{t}}}. \tag{4.23}$$

Thus, the final minimization problem is:

$$(\hat{\boldsymbol{\phi}},\hat{\boldsymbol{\theta}}) = \underset{\boldsymbol{\phi},\boldsymbol{\theta}}{\operatorname{argmin}}\, \mathcal{L}^L(\boldsymbol{\phi},\boldsymbol{\theta}) = \underset{\boldsymbol{\phi},\boldsymbol{\theta}}{\operatorname{argmin}} -\beta\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}} + \beta\mathcal{D}_{\mathrm{t}}. \tag{4.24}$$

It was shown in Section 4.5.2.2 that, in general, the term $\mathcal{D}_{\mathrm{t}}$ does not affect much the templates estimation. Therefore, the empirical investigation is performed with respect to the scenario:

$$(\hat{\boldsymbol{\phi}},\hat{\boldsymbol{\theta}}) = \underset{\boldsymbol{\phi},\boldsymbol{\theta}}{\operatorname{argmin}} -\beta\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}}. \tag{4.25}$$

Assuming that $p_{\boldsymbol{\theta}}(\mathbf{t}|\mathbf{a}) \propto \exp(-\lambda_{\mathrm{t}}\|\mathbf{t} - g_{\boldsymbol{\theta}}(\mathbf{a})\|_2)$ similarly to the remark (4.9) in Section 4.5.2, (4.25) yields to:

$$(\hat{\boldsymbol{\phi}},\hat{\boldsymbol{\theta}}) = \underset{\boldsymbol{\phi},\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{i=1}^{M} \|\mathbf{t}_i - g_{\boldsymbol{\theta}}(\mathbf{a}_i)\|_2, \tag{4.26}$$

where $\mathbf{a}_i = f_{\boldsymbol{\phi}}(\mathbf{x}_i)$ and $f_{\boldsymbol{\phi}}$ denotes the encoder.

4.6.2 Fakes production

HC fakes

The hand-crafted estimation is performed on the DP1C and DP1E datasets. The obtained estimation error in % is measured as a symbol-wise Hamming distance between the original digital templates $\mathbf{t}$ and the corresponding estimations $\hat{\mathbf{t}}$ is given in Tables 4.9 and 4.10 for the

Table 4.9 The average (over three runs) estimation error in % based on the symbol-wise Hamming distance between the original digital templates and the corresponding estimations in the DP1C test sub-set. As indicated in the first column, each model is trained only on the codes printed on one corresponding printer. At the test stage each model is tested on the codes printed on different printers as shown by the corresponding columns.

	DP1C			
Train on \Test on	SA	LX	CA	HP
		Hand-crafted		
Full area	3.49	5.98	1.39	1.71
Half area	1.00	2.22	0.66	0.76
		LinearBN		
SA	**0.15** (±0.01)	1.13 (±0.15)	0.13 (±0.01)	0.38 (±0.06)
LX	0.28 (±0.03)	**0.34** (±0.03)	0.20 (±0.01)	0.38 (±0.01)
CA	0.22 (±0.03)	1.18 (±0.27)	**0.13** (±0.01)	0.36 (±0.03)
HP	0.30 (±0.05)	1.31 (±0.32)	0.14 (±0.03)	**0.31** (±0.08)
		ConvBN		
SA	**0.12** (±0.02)	0.77 (±0.22)	0.21 (±0.04)	0.43 (±0.10)
LX	0.64 (±0.18)	**0.23** (±0.08)	0.60 (±0.20)	0.97 (±0.26)
CA	0.29 (±0.04)	1.14 (±0.34)	**0.12** (±0.02)	0.36 (±0.06)
HP	0.23 (±0.04)	1.11 (±0.52)	0.17 (±0.07)	**0.22** (±0.04)

DP1C and DP1E correspondingly. As expected, the estimation approach based on the half symbol's area provides the best performance for all printers under investigation. Moreover, for all considered printers, except the LX, the obtained estimation error could be considered as sufficiently small, about 0.65 - 1% of the total number of symbols in the code. In case of the LX printer, in contrast to the other used printers, the error is several times higher due to the high printing dot gain specific to this printer that can be visually verified in Fig. 4.6 in Section 4.2.1.

ML fakes

For the ML estimation two types of DNN architectures based on a bottleneck principle [149] were chosen: *(i)* linear model *LinearBN* and *(ii)* convolutional model *ConvBN*. The detailed information about the used architectures and training parameters are given in Appendix D.3.1. Similarly to the HC case, the ML estimation is performed on the DP1C and DP1E datasets. To better understand the role and influence of printing equipment, the cross-printer evaluation (test) is performed, while the training is performed individually for each printer. The obtained average (over three runs) results in % are given in Tables 4.9 and 4.10.

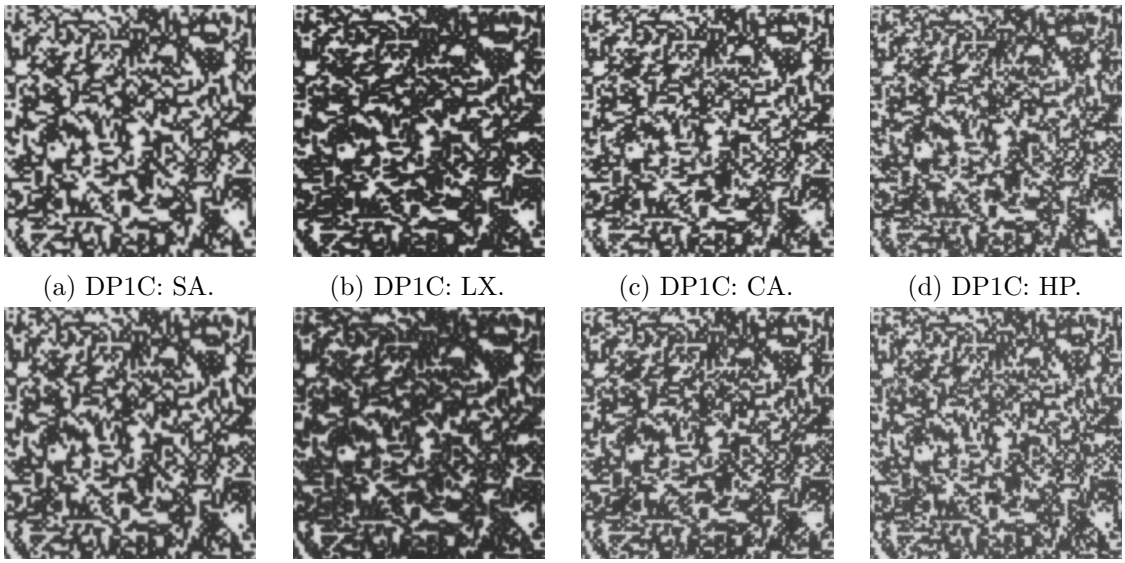
From the obtained results it can be seen that:

Table 4.10 The average (over three runs) estimation error in % based on the symbol-wise Hamming distance between the original digital templates and the corresponding estimations in the DP1E test sub-set. As indicated in the first column, each model is trained only on the codes printed on one corresponding printer. At the test stage each model is tested on the codes printed on different printers as shown by the corresponding columns.

	DP1E			
Train on \Test on	SA	LX	CA	HP
		Hand-crafted		
Full area	2.88	5.21	1.48	1.93
Half area	0.85	1.76	0.66	0.88
		LinearBN		
SA	**0.16** (±0.02)	1.48 (±0.04)	0.21 (±0.01)	0.61 (±0.06)
LX	0.40 (±0.05)	**0.71** (±0.05)	0.36 (±0.04)	0.71 (±0.02)
CA	0.16 (±0.01)	1.32 (±0.10)	**0.13** (±0.003)	0.47 (±0.03)
HP	0.22 (±0.06)	1.23 (±0.04)	0.15 (±0.002)	**0.56** (±0.12)
		ConvBN		
SA	**0.15** (±0.08)	1.35 (±0.28)	0.30 (±0.09)	0.79 (±0.21)
LX	0.43 (±0.12)	**0.42** (±0.17)	0.58 (±0.25)	1.16 (±0.25)
CA	0.21 (±0.05)	0.93 (±0.12)	**0.13** (±0.04)	0.50 (±0.09)
HP	0.11 (±0.01)	0.75 (±0.05)	0.08 (±0.01)	**0.18** (±0.02)

- For both datasets, the *ConvNN* produces more accurate estimations that results in a smaller estimation error.
- Moreover, the cross-printer tests show that, in general, the training and testing for the same printers is more efficient than the performing training and testing on different printers, especially if the printers have different printing artifacts and dot gain, as for example, the SA and HP or the SA and LX.
- Additionally, it should be pointed out that it might seem that the difference between the results on the DP1C and DP1E datasets is not obvious. Under the detailed investigation, one can notice that the DP1C estimation error on average is smaller than in case of the DP1E dataset. This is related to the difference in illumination between Epson and Cannon scanners that can be seen in Fig. 4.7, where the results of scanning of empty substrate (paper) are illustrated. Thus, one can conclude that the scanner illumination might have an important influence on the production of high quality ML estimations of the original digital templates, especially for the printers with a big dot gain, like in case of the LX printer.

In summary, it should be pointed out that, as expected, the ML estimation is more accurate compared to the results of HC estimation.



(a) DP1C: SA. (b) DP1C: LX. (c) DP1C: CA. (d) DP1C: HP.

(e) DP1E: SA. (f) DP1E: LX. (g) DP1E: CA. (h) DP1E: HP.

Fig. 4.30 Examples of the produced ML fakes in the DP1C and DP1E datasets.

Taking into account that the re-printing and acquisition process are time consuming, the investigation of PGC authentication with respect to the produced fakes is performed only for the ML estimations based on the *ConvBN* model as the most accurate and, therefore, the most difficult case. The estimations of the first run are printed and used for the further investigation. Several examples of the produced fakes are visualized in Fig. 4.30.

4.6.3 One-class classification with respect to the ML fakes

Based on the performed analysis and results obtained in Section 4.5 the one-class classification scenario $\mathcal{L}_4(\boldsymbol{\phi}_\mathrm{t}, \boldsymbol{\theta}_\mathrm{t}, \boldsymbol{\theta}_\mathrm{x}) = -\beta_\mathrm{t}\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}} + \beta_\mathrm{t}\mathcal{D}_\mathrm{t} - \beta_\mathrm{x}\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}} + \beta_\mathrm{x}\mathcal{D}_\mathrm{x}$ is considered as the most accurate and is used for the investigation of PGC authentication with respect to the produced ML fakes. Similarly to the training procedure described in Section 4.5.2.4, for each printer, at first, the feature extraction based on the auto-encoding model is performed and, secondly, the OC-SVM is trained based on the two metrics: *(i)* the symbol-wise Hamming distance between the original digital templates and its estimations and *(ii)* the ℓ_2-distance between the printed codes and its reconstructions. For both stages, the training is performed only on the train sub-set of original data. The average (over five runs) OC-SVM classification error obtained on the test sub-set is given in Table 4.11. The detailed information about each run classification results is given in Table D.18 in Appendix D.3.2. The examples of the OC-SVM decision boundaries on the DP1C dataset are illustrated in Fig. 4.31. The examples for the OC-SVM decision boundaries on the DP1E dataset are given in Fig. D.6 in Appendix D.3.2.

Table 4.11 The average (over five runs) OC-SVM classification error in % on the test sub-set with respect to the ML fakes based on the *ConvBN* model.

	DP1C		**DP1E**	
	P_{miss}	P_{fa}	P_{miss}	P_{fa}
SA	7.54 (±3.13)	87.42 (±5.5)	8.34 (±1.11)	78.22 (±6.38)
LX	21.14 (±14.19)	57.82 (±22.39)	14.48 (±6.06)	60.38 (±10.40)
CA	10.94 (±5.87)	35.56 (±9.44)	13.44 (±10.48)	10.94 (±6.75)
HP	7.14 (±2.33)	79.80 (±12.86)	6.94 (±1.03)	80.98 (±4.18)

First of all, it should be pointed out that, despite the relatively small error rate in the ML digital templates estimation for all considered printers, the classification results of OC-SVM are much more different.

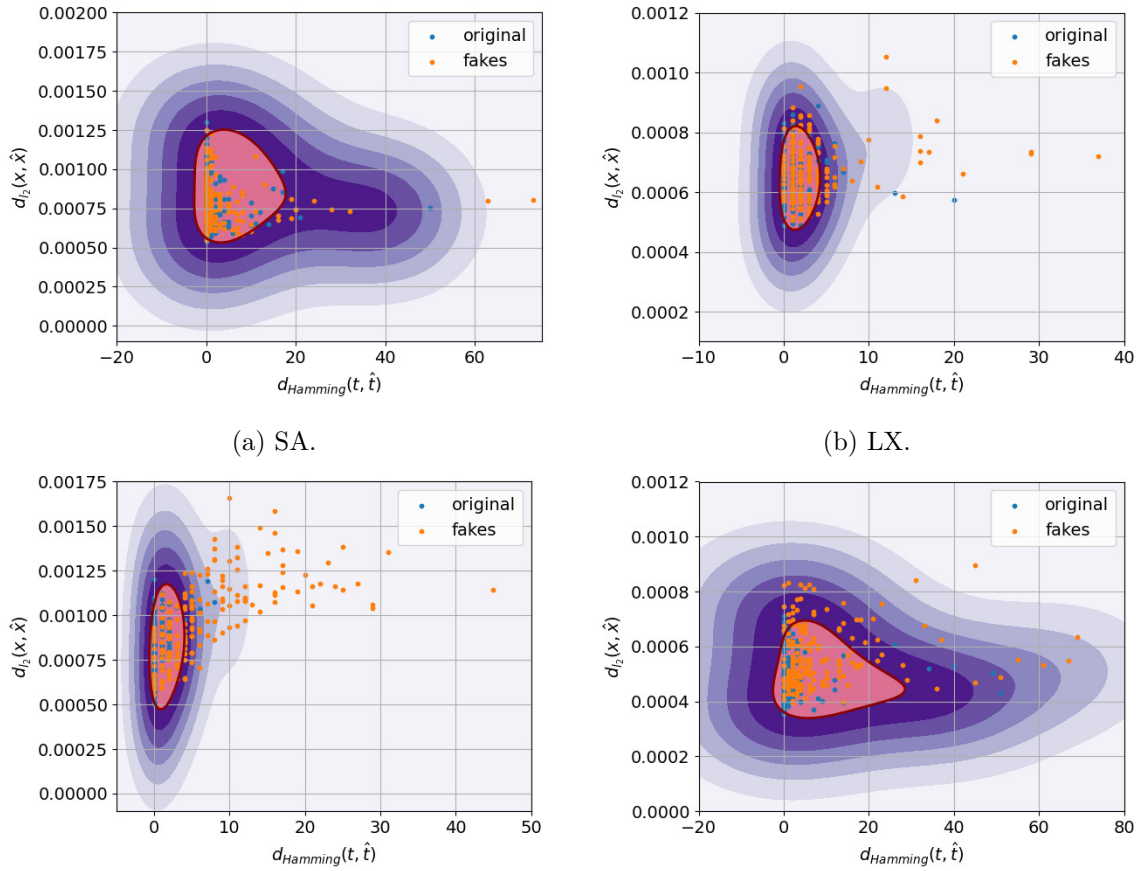
In case of the SA printer, in principle, the obtained classification error is quite expectable: relatively small P_{miss} along with a huge P_{fa}. It is explained by the fact that, due to the good print quality and small estimation error, the originals and fakes are very close.

In case of the LX printer, one can note the increasing of P_{miss} and decreasing of P_{fa} compared to the SA printer. Additionally, a higher deviation from the average value is observable. That indicates a bigger deviation in the results of the regeneration model and, as a consequence, the more difficult evaluation of the area of originals by the OC-SVM.

In case of the CA printer, it is possible to see the smallest P_{fa} compared to the other printers along with a middle level of the P_{miss}. In Fig. 4.31(c) on can see the biggest deviation in the ℓ_2-error in comparison with the other printers. Taking into account that, in contrast to the ML estimation discussed in Section 4.6.2, where the models are optimized only with respect to an accurate digital template estimation, in case of the one-class classification the regeneration model is additional optimized with respect to an accurate reconstruction of the printed codes. In this regard, in addition to the bigger deviation in the ℓ_2 error one can observe a bigger deviation in the $d_{Hamming}$ error for the fake codes. That leads to the smaller P_{fa}.

The one-class classification results for the HP printer are similar to those of the SA printer.

In summary, it should be pointed out that for all printers the training conditions are the same. Any fine-tuning of the training parameters for each particular printer has not been performed. In this respect, the obtained results might be not optimal. However, in general, it does not change the fact that, in case of the ML fakes, it is very difficult to train an accurate "blind" model (without observing the fakes) due to the high degree of similarity between the originals and fakes.



(a) SA.

(b) LX.

(c) CA.

(d) HP.

Fig. 4.31 The examples of the OC-SVM decision boundaries for the different printers in the DP1C dataset. All fakes are produced based on the ML-estimates.

4.6.4 Supervised classification with respect to the ML fakes

The supervised classification with respect to the ML fakes produced by the *ConvBN* model is performed in the same way as described in Section 4.4 with the only difference that for each printer there exists only one type of fake. The obtained average (over five runs) classification error is given in Table 4.12. The detailed information about each run classification results is given in Table D.19 in Appendix D.3.3.

It is easy to see that in case of the SA printer the supervised classification is quite unstable as evidenced by the large deviation from the average value and is inaccurate as illustrated by the high classification error. Such a result is expectable and can be explained by the small differences between the printed original and fake codes due to the very accurate printing and small ML digital templates' estimation error.

The situation with the LX printer in DP1C dataset is similar to the SA. At the same time, one can see that in case of the DP1E dataset the classification error is in about three times

Table 4.12 The average (over five runs) supervised classification error in % with respect to the ML fakes based on the *ConvBN* model.

	DP1C		**DP1E**	
	P_{miss}	P_{fa}	P_{miss}	P_{fa}
SA	34.17 (±11.12)	28.96 (±11.79)	24.16 (±7.42)	31.87 (±5.19)
LX	30.42 (±4.34)	32.60 (±12.02)	10.73 (±4.11)	10.21 (±6.73)
CA	2.08 (±1.33)	8.85 (±4.20)	1.46 (±1.62)	6.35 (±3.38)
HP	8.54 (±3.84)	20.62 (±8.45)	8.12 (±4.46)	15.31 (±7.85)

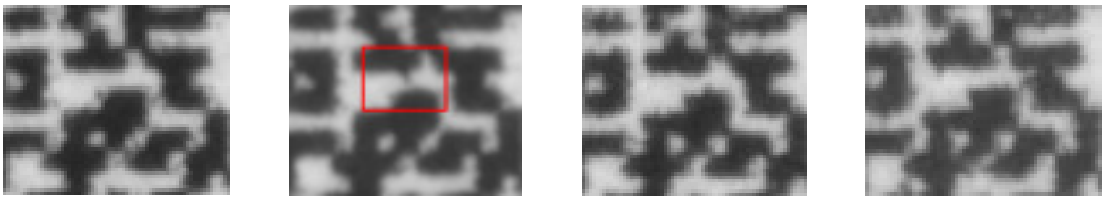
smaller. This can be explained by the fact that the ML digital templates estimation error in this case is in about two times bigger as shown in Table 4.10. Moreover, the estimation errors are mostly related to the big dot gain. As the result, the produced fakes have specific deviations from the originals, which the classifier takes into account for the authentication.

In the case of the CA printer, the classification error is the smallest one compared to the other used printers. At first glance, such a result is very strange since the amount of the estimation error in the digital templates is approximately the same as in case of the SA printer. On the other hand, in contrast to the SA, the CA printing quality is much more different: the printed symbol's borders are very uneven and ragged as can be seen in Fig. 4.6. The same printing quality can be observed for the HP printer too.

To better understand the role of the printing irregularity, it is important to pointed out that the original and fake codes are printed with a time interval of several weeks. In this respect, the changes in settings of the printers are not excluded since these printers have been used for other job printing. The investigation of the influence of printing regimes and its deviations is an important topic that deserves a special attention and study but is out of scope of the current work.

From Fig. 4.32 one can observe printing irregularity in case of the CA and HP printers. Moreover, for the CA printer it is possible to see the specific deviations marked in red in Fig. 4.32. The frequency with which these deviations occur can be noted under close examination of the CA printed codes. As a result, these irregularities are the reason of the bigger deviation in the ℓ_2 error between the printed fake codes and its regenerations for the CA printer shown in Fig. 4.31 (almost in two times bigger than in case of other printers). At the same time, these deviations are very important for the supervised classifier deviations but yet not typical for the HP printer. In this respect, the classification error for the HP printer is higher.

Finally, it should also be noted that for all printers the supervised classifiers are trained under the same conditions without any specific parameters' tuning. The detailed information about the used architecture and trained parameters is provided in Appendix D.3.3. This might be a cause of a large deviation from the average value in the obtained results.



(a) CA original. (b) CA fake. (c) HP original. (d) HP fake.

Fig. 4.32 The DP1C dataset examples of the printing irregularity and deviations between the original and fake codes for the CA and HP printers.

4.6.5 Conclusions

This Section aimed at investigating the aspects of the digital templates estimation from the corresponding printed counterparts based on the modern machine learning approaches. The two basic hand-crafted methods are considered for the comparison reasons. The four printers and two scanners are investigated.

As expected, the machine learning methods demonstrate more accurate estimation than the hand-crafted approaches. With respect to the choice of printing equipment, as it is shown by the obtained results, the accurate printing leads to more accurate estimation of the digital templates. At the same time, it can create a fake illusion for the defender that the use of less accurate printing equipment would provide the better defense against the copy attacks.

To dispel this illusion, the one-class classification and supervised classification are performed with respect to the produced ML fakes.

The one-class classification results show that, in contrast to the typical HC copy fakes investigated in Section 4.5, it is very difficult to obtain an efficient authentication model trained in a "blind" way without observing the fakes due to the high degree of similarity between the originals and ML fakes.

The considered supervised scenario does not add much optimism. The classical supervised classification, efficiently applicable against the typical HC copy fakes, is not anymore so accurate against the ML fakes. However, as the obtained results show, the printing deviations might play very important role for the authentication of high quality ML fakes. The investigation of the impact of printing deviations is related to the *printing process randomness* category and is an interesting direction for the future work.

4.7 Digital templates estimation

The clonability aspects of copy detection patterns used in the PGC are defined by many factors. The influence of printing and scanning equipment is investigated in Section 4.6. The role of size of base elements (symbols) is yet another important factor (see the research questions Q.8 from Section 4.3).

To explore the influence of symbols' size factor on the clonability aspects of copy detection patterns we proceed with a specially designed Indigo scanner dataset presented in Section 4.2.3. It is important to emphasize that all codes are printed on the same industrial printer HP Indigo 5500 DS[13] at the resolution 812 dpi. The enrollment is performed by using Epson perfection 4990 scanner at the resolution 1200 ppi.

4.7.1 Details of the setup

As it is discussed in Section 4.2.3 the digital templates are generated with three different symbols' sizes. Taking into account the difference in printing (812 dpi) and scanning (1200 ppi) resolutions, the symbols' size in the enrolled codes increases in about 1.48 times (1200 / 812 = 1.48) compared to the symbol size of digital templates used for printing. In this respect, the original symbols of size 5×5 become 7.4×7.4, the 4×4 become 5.9×5.9 and the symbols of size 3×3 become 4.4×4.4. Taking into consideration that the increasing factor is not integer, all codes are rescaled with respect to the integer number of pixels. Moreover, aiming to compensate the resulting discrepancy, all codes are additionally rescaled to the size of corresponding digital templates. Thus, we have two enrolled sets for each symbol size:

- for the digital templates with symbols of size 5×5 the enrolled codes are of size:
  - 7×7
  - 5×5
- for the digital templates with symbols of size 4×4 the enrolled codes are of size:
  - 5×5
  - 4×4
- for the digital templates with symbols of size 3×3 the enrolled codes are of size:
  - 4×4
  - 3×3

Similarly to Section 4.6.2 the fakes' production is based on the estimation of original digital templates from the corresponding printed counterparts. We assume that for each original digital template the attacker has only one printed copy. The fakes' production is investigated in four setups:

- *HC full area*

  Similarly to Section 4.6.1, the full area HC setup assumes that the attacker performs the printed codes' symbol-wise thresholding based on the mean intensity in the full symbol's area.

- *HC half area*

  The half area setup differs from the full area only by using for the analysis the reduced central symbols' area.

[13]The standard printing resolution of this printer is 812.8 dpi.

- *ML trained on the HC estimations*

  This setup combines the HC and ML approaches. The ML approach consists in the training of DNN model on the pairs of printed codes and corresponding digital templates. In the considered setup, the used digital templates are obtained via the HC estimation.

- *ML trained on the original digital templates*

  This setup assumes the usage of original digital templates for the training of DNN model and could be considered as an ideal case for the attacker.

To be coherent with the previously obtained results, the ML setups are based on the IB principle and practical scenario $\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}) = -\beta \mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}}$ discussed in Section 4.6.1. The detailed information about the used training parameters and models' architecture are given in Appendix D.4.

At the inference stage the estimation accuracy is measured with respect to the average percentage of error symbols in one code based on the symbol-wise Hamming distance between the original digital templates and the corresponding binary estimations. In addition to this, we calculate the percentage of codes estimated *(i)* without any error, *(ii)* with the 1 or 2 error symbols and *(iii)* with the amount of error symbols bigger than 2. The reported results are given with respect to the test sub-set.

4.7.2 Estimation results

Estimation of 5×5 codes. The digital templates estimation results for the codes with the original symbol size 5×5 are given in Table 4.13. Similarly to Section 4.2.3, the *HC full area* setup demonstrates the worst performance with the average percentage of error symbols about 6.5 - 7%. At the same time the HC estimation based only on the half symbol area allows to reduce the estimation error to about 1%. With respect to the used *DataMatrix* mapping matrix of 66×66 symbols as explained in Section 4.2.3 it is about 43 symbols. Which is quite significant. In addition, it should be noted that, in principle, the increasing symbols' size due to the difference in printing and scanning resolutions does not give any significant improvement with respect to the original size 5×5.

At the same time, from Table 4.13 one can observe the remarkable improvement of the estimation accuracy for the ML based approaches. The detailed information about each run results is given in Table D.22 in Appendix D.4. The ML estimation trained on the HC half area estimation as a ground truth allows to reduce the estimation error in about 1.8 times for the scans with 7×7 symbol's size. For the scans with the original 5×5 symbol's size the improvement is even more significant: from about 1% to 0.014% and the amount of codes without any estimation error from about 4.5 % to 72%.

In case of the ML model trained on the original digital templates, the obtained results are the most accurate. In case of the scans with 5×5 symbol's size, 94% of codes are estimated

Table 4.13 The estimation error in % based on the symbol-wise Hamming distance between the original digital templates with the symbol's size 5×5 and the corresponding estimations. For the ML approaches the average results over three runs are given.

Scanned codes symbol size	Average % of error symbols	% of codes with error symbols = 0	% of codes with error symbols ≤ 2	% of codes with error symbols > 2
HC full area				
7×7	6.42	0.00	0.00	100.00
5×5	6.83	0.00	0.00	100.00
HC half area				
7×7	1.07	4.67	9.33	86.00
5×5	1.07	4.67	9.33	86.00
ML trained on the HC half area estimations				
7×7	0.57 ($\pm$0.16)	18.29 ($\pm$18.82)	19.48 ($\pm$5.17)	62.23 ($\pm$23.87)
5×5	0.014 ($\pm$0.01)	72.30 ($\pm$4.43)	22.30 ($\pm$0.68)	5.4 ($\pm$3.7)
ML trained on the original digital template				
7×7	0.025 ($\pm$0.01)	74.7 ($\pm$2.37)	17.57 ($\pm$2.34)	7.66 ($\pm$1.03)
5×5	0.0015 ($\pm$0.0002)	94.14 ($\pm$1.03)	5.86 ($\pm$1.03)	0.00

without any mistake and the rest 6% of codes have less than 3 error symbols. The results obtained for the scans with symbol's size bigger than the original one are a little worse. This can be explained by the fact that the increase of the symbols' size is not regular, meaning that $1200/812 \cdot 5 = 7.4$, that results in situation, when some symbols might have bigger or smaller influence on the neighbors. It is especially critical for the white symbol surrounded by the black ones. The reduction of symbols' size close to the original might reduce this influence, especially for the ML approaches that are more sensitive to invisible at first glance deviations.

Estimation of 4×4 codes. The results obtained for the estimation of digital templates with the original symbol size equaled to 4×4 are given in Table 4.14.

In case of the HC half area estimation one can observe the increase of average estimation error from 6 - 7% for 5×5 (Table 4.13) to 10 - 11% for 4×4 (Table 4.14). Moreover, there is not any estimated code with the error less than 3 symbols in contrast to the similar method for 5×5 in Table 4.13.

The ML estimators[14] trained on the HC half area estimations as a ground truth do not demonstrate any significant improvement in contrast to the similar case demonstrated in Table 4.13. However, in comparison to the 5×5 case, one can see that the scans with increased symbol's size show a little bit better performance. In general case, the obtained phenomenon might be related to the fact that 4×4 times the scaling factor 1.48 gives 5.9×5.9 which is almost 6×6 pixels in contrast to the previous case with the 7.4×7.4 pixels, where the

[14]The each run detailed results are given in Table D.23 in Appendix D.4

Table 4.14 The estimation error in % based on the symbol-wise Hamming distance between the original digital templates with the symbol's size 4×4 and the corresponding estimations. For the ML approaches the average results over three runs are given.

Scanned codes symbol size	Average % of error symbols	% of codes with error symbols = 0	% of codes with error symbols ≤ 2	% of codes with error symbols > 2
HC full area				
5×5	10.63	0.00	0.00	100.00
4×4	10.08	0.00	0.00	100.00
HC half area				
5×5	2.35	0.00	0.00	100.00
4×4	3.30	0.00	0.00	100.00
ML trained on the HC half area estimations				
5×5	2.34 ($\pm$0.04)	0.00	0.22 ($\pm$0.38)	99.78 ($\pm$0.38)
4×4	3.13 ($\pm$0.12)	0.00	0.00	100
ML trained on the original digital template				
5×5	0.0016 ($\pm$0.0003)	95.11 ($\pm$1.54)	4.22 ($\pm$1.54)	0.67 ($\pm$0)
4×4	0.0016 ($\pm$0.0002)	94.89 ($\pm$1.02)	4.44 ($\pm$1.02)	0.67 ($\pm$0)

discrepancy in less than half of symbol might have negative effect. The detailed investigation of these phenomena is an interesting question for the future work.

In case of the ML estimators trained on the original digital templates, the obtained estimation results are the best. In general, the average estimation error is close to the previous case. However, the amount of the codes estimated with the small number of the error symbols is slightly higher: 0.67% instead of 0% demonstrated in Table 4.13.

Estimation of 3×3 codes. The estimation results obtained for the original digital templates with symbol's size 3×3 are given in Table 4.15. In general case, one can observe the same tendency as in two previous cases, namely, the HC approach based on the analysis of the full symbol's area shows the worst performance. The HC analysis of the half symbol's area allows to improve the estimation. Similarly to the case of the estimation of the codes with the original symbol's size 4×4, the ML estimators[15] trained on the HC half area estimations as a ground truth do not allow to improve the estimation accuracy due to the significant level of errors in that estimations. The best results are obtained by the ML estimators trained on the original digital templates. However, one can notice that the further reduction of symbols' size in the original digital templates makes the ML estimation more harder: the average percentage of error symbols increases in about 7 times from 0.0015 - 0.0016 to 0.011. The amount of codes without any error is reduced in about 1.3 times from about 95% to approximately 70%.

[15]The each run detailed results are given in Table D.24 in Appendix D.4

Table 4.15 The estimation error in % based on the symbol-wise Hamming distance between the original digital templates with the symbol's size 3×3 and the corresponding estimations. For the ML approaches the average results over three runs are given.

Scanned codes symbol size	Average % of error symbols	% of codes with error symbols = 0	% of codes with error symbols ≤ 2	% of codes with error symbols > 2
HC full area				
4×4	16.73	0.00	0.00	100.00
3×3	17.13	0.00	0.00	100.00
HC half area				
4×4	10.44	0.00	0.00	100.00
3×3	8.54	0.00	0.00	100.00
ML trained on the HC half area estimations				
4×4	10.25 ($\pm$0.2)	0.00	0.00	100
3×3	8.53 ($\pm$ 0.22)	0.00	0.00	100
ML trained on the original digital template				
4×4	0.018 ($\pm$0.02)	74.00 ($\pm$2.40)	23.13 ($\pm$1.68)	2.89 ($\pm$0.77)
3×3	0.011 ($\pm$0.001)	69.33 ($\pm$0.67)	27.33 ($\pm$0.67)	3.33 ($\pm$1.33)

4.7.3 Conclusions

The main goal of this Section is to investigate the robustness of PGC to the clonability attacks based on the estimation of original digital templates from the corresponding printed counterparts.

The four setups are investigated and the obtained results show the benefit of the ML approaches over the HC techniques, especially in case, when the attacker has either access to the original digital templates or when the good quality HC estimation are available.

As expected, the reduction of the original symbols' sizes leads to a deterioration of the accuracy of estimation. However, the accuracy of estimation stay relatively high: even in the case of the codes with the 3×3 symbols the amount of codes with the estimation without any error symbol is about 70%.

Finally, it can be concluded that it was not a purpose of this section to explore the authentication aspects of PGC with respect to the newly produced high quality ML fakes. That question is investigated in details in Section 4.6. Taking into account the level of errors in the estimation of original digital templates, it is obvious that the authentication of such fakes is even more difficult for the defender than in case of the ML fakes produced and investigated in Section 4.6.

4.8 Conclusions

This Chapter aims at investigating the authentication and copy detection aspects of PGC from the perspective of HC and ML copy attacks.

Due to the lack of the publicly available datasets besides DP0E and CSGC printed on the desktop printers, we extended the DP0E dataset by DP1C and DP1C datasets and created two new datasets of codes printed on the industrial printer HP Indigo (Indigo mobile and Indigo scanner datasets) to investigate the clonability of PGC behavior. The influence of the used printing equipment is investigated on the DP1C and DP1E datasets. However, due to the restricted access to a big number of different industrial printers, the desktop office printers are used for the investigation. Moreover, to make the authentication conditions close to the real life environment, in the Indigo mobile dataset the enrollment is performed by using a modern mobile phone iPhone XS under regular room light.

The investigation of the authentication aspects of PGC with respect to the typical HC copy fakes shows that the supervised classification of the HC fakes is quite accurate. Although, it is difficult to predict in advance what kind of fakes will be used at the inference stage, the properly chosen training data might be sufficient to guarantee an accurate authentication of the certain classes of fakes.

In the case of the one-class classification, the authentication accuracy depends to a large extent on a selection of features, metrics and reference templates (physical or digital), which might be a difficult task for the spatial domain. At the same time, the similar analysis in the domain of DNN processing might be more intuitive, easily controllable and more accurate. Despite the possible difficulties, in general case, the one-class classification of the typical HC copy fakes might be as accurate as the supervised authentication.

At the same time, the one-class classification as well as the supervised authentication of the high quality ML copy fakes appears to be a more difficult problem, especially in the case of an accurate printing leading to an accurate estimation of the original digital templates from the printed counterparts by the attacker.

The investigation of influence of the symbols' size of the copy detection patterns shows that, definitely, the symbols' size has an important impact on the accuracy of estimation of digital templates from the corresponding printed counterparts. The smaller size of the symbols results in the higher amount of errors in the estimations. However, when the original digital templates are available for the ML training, the obtained fakes might turn out to be accurate enough even despite the small size of symbols. In such a situation the defender-verifier might not be capable to authenticate the ML fakes with a high confidence. In the general case, there is no need for the attacker to have an access to the pairs of digital templates and corresponding printed codes used by the defender. Theoretically, the attacker can estimate the used printer from the printed counterparts and can produce the necessary amount of corresponding pairs for the training of the ML model by himself.

Finally, as a general conclusion, it should be pointed out that the clonability of PGC are determined by many factors. The continuous improvement and development of the ML technologies make the PGC quite "fragile" with respect to the potential appearances of high quality ML fakes.

Therefore, the future work should target to investigate an impact of further reduction of symbol size in the copy detection patterns. For the defender it is important to establish a dependence between the potential deterioration of the authentication accuracy and the decrease of symbols' size. From the attacker's side it is important to ensure an accurate ML fakes' production by using potentially more complex DNN based models.

Another important direction for the future research is to study of authentication and copy detection aspect of PGC based on the color copy detection patterns and their comparative analysis with the studied black and white codes.

In view of constantly improving optical characteristics of cameras of modern mobile phones, the investigation of the PGC authentication based on the unique fingerprint produced by the printing process is *(i)* a promising direction for future improvement of PGC authentication even with respect to the high quality fakes and *(ii)* an important research question for the future work.

Chapter 5

Conclusions and Future work

In this work, we address a problem of reliable classification under adversarial examples produced in digital and physical worlds. In both cases we target a "blind" classification, when the adversarial examples and fakes are not available for classifier training and compare our results with fully supervised counterparts whenever possible.

Considering the problem of reliable classification in the digital world representing a considerable research interest for both theory and practice, we mainly focus on the DNN-based models. Thesis aims at investigating the vulnerability of the DNN-based classification systems to adversarial attacks that questions the usage of DNN models for many security- and trust-sensitive domains. Based on the performed analysis of the state-of-the-art defense and attacks approaches, Thesis proposes a new defense based on key-based diversified aggregation (KDA) mechanism in a gray- and black-box scenarios. The white-box scenario attacks are out of scope of the current Thesis due to the fact that such attacks are more suitable for the synthetic laboratory conditions and rarely applicable to real world conditions. The KDA based on a multi-channel architecture provides an information advantage for the defender over the attacker due to the knowledge of the secret keys in the key-based diversification and a limited access to the internal states of the system. At the same time, the attacker knows the architecture of classifier and the used defense strategy. Moreover, he/she has no limitation in the access to the used datasets. The robustness of the whole system is achieved by a specially designed key-based randomization in a transform domain in several channels. It prevents the gradients' back propagation and the aggregation of soft outputs from each channel stabilizes the results and increases the reliability of the classification score. The evaluation of the efficiency of the proposed defense and the performance of the whole classification system is done on three standard well-known and widely used datasets against a number of known state-of-the-art attacks. The numerical results demonstrate the robustness of proposed defense mechanism against these state-of-the-art adversarial attacks and show that using the multi-channel architecture with following aggregation stabilizes the results and increases the classification accuracy.

Based on these encouraging results, we can also suggest a number of possible future extensions: *(i)* to investigate in more details the security aspects of proposed KDA algorithm, *(ii)* to obtain the estimates and bounds on the attacker complexity attempting at learning the introduced randomization or bypassing it by some dedicated structures, *(iii)* to investigate from the theoretical and practical sides the impact of number of the used training examples jointly with the randomization in terms of comparison of entropy of training dataset versus needed entropy of randomization.

The second important question addressed in Thesis and related to the reliable classification in the digital world is an impact of the amount of labeled data available for the training on the classification accuracy. Taking into account that usually we have an access to a lot of unlabeled data in view of global digitalization of today's world, the semi-supervised classification is proposed as a reliable alternative to the fully supervised classification models. To fulfill a lack of training data, we propose to consider a role and impact of priors on the latent space of classifier. In Thesis, we performed the theoretical analysis of the several families of priors on latent space representation in semi-supervised classification based on the IB principle. We practically demonstrated an impact of different regularizers on the classification accuracy. A new formulation of semi-supervised IB with the hand-crafted and learnable priors is proposed. Additionally, we showed a link of the proposed framework to the several state-of-the-art unsupervised and semi-supervised methods.

As a future work it would be interesting to investigate an "adversarial" regularization by the adversarially generated examples to impose the constraints on the minimization of mutual information between them and class labels or equivalently to maximize the entropy of class label distribution for these adversarial examples according to the framework of entropy minimization. The contrastive multi-view IB formulation would be an interesting candidate for the regularization of latent space.

Along the direction of the reliable classification in the physical world, Thesis studies the authentication and clonability aspects of modern PGC. Taking into account ethical and non-competition aspects of this problem with respect to several competitive technologies on the market, the investigation is performed on the copy detection patterns generated based on an open international standard. The main goal is to demonstrate a general approach applicable to the majority PGC designed with identical modulation principles rather than to investigate the clonability and authentication aspects of some particular PGC.

The PGC are investigated from both the defender-verifier and attacker sides. From the side of defender-verifier the aspects of the reliable classification of PGC are studied with respect to the HC and ML fakes in different setups. The base-line supervised approach is investigated with respect to the different training and test data. At the same time, taking into account that in practice the supervised scenario is rare, an one-class classification is investigated in the different domains, namely in the direct spatial domain and in the domain of DNN-based processing. It should be pointed out that the one-class classification is closer

to real life conditions, where the defender-verifier pair is not capable to predict in advance what kind of fakes the system will face. We showed that both types of the classification work well with respect to the different types of the HC fakes. At the same time, for the one-class classification in the direct spatial domain the choice of the appropriate metrics for training is a non-trivial task, while in the DNN-based processing domain, the choice is more intuitive and justified by the used models.

Simultaneously, we considered mechanisms of production of high quality ML fakes based on the achievements of modern DNN models. The different aspects impacting the fakes' production process are investigated and namely, the impact of the used printing and scanning equipment along with the impact of size of symbols used in the copy detection patterns. It is shown that, although, the both factors have a great impact, the modern DNN based approaches have a high potential for the generation of high quality fakes based on the estimation of original digital templates from the corresponding printed counterparts. It is shown that the produced ML fakes are more difficult for the authentication than the HC fakes. Therefore, the ML fakes represent a great risks for the considered code designs. At the same time, we envision several countermeasures against the ML-based fakes.

As a future work it is important to investigate the impact of further decreasing the size of symbols used in the copy detection patterns of the size 2×2 and even 1×1. The main technological challenge is a possibility of modern printing technologies to reliably reproduce 1×1 symbols at high printing resolution. Once printed, the dot gain effect will lead to considerable degradation of codes thus preventing the attacker from the reliable ML-based estimation of digital templates. At the same time, an important and interesting direction is an investigation of PGC based on the color copy detection patterns, their reliability for the robust authentication and the analysis of their basic characteristics compared to the black and white patterns. Moreover, an important question is to investigate the authentication of the PGC based on the printing process randomness (printing fingerprint). From the side of the attacker it is important to study the clonability aspects of color PGC and the abilities of attacker to estimate not only the original symbolic template but also the corresponding used color palette.

Finally, an important point is to continue the initiated work with respect to the creation of the datasets of PGC publicly available for the academic usage to stimulate the cooperation and reproducible research.

References

[1] Tailing Yuan, Yili Wang, Kun Xu, Ralph R Martin, and Shi-Min Hu. Two-layer qr codes. *IEEE Transactions on Image Processing*, 28(9):4413–4428, 2019.

[2] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.

[3] OECD and European Union Intellectual Property Office. *Trends in Trade in Counterfeit and Pirated Goods*. 2019.

[4] Ari Juels. Rfid security and privacy: A research survey. *IEEE journal on selected areas in communications*, 24(2):381–394, 2006.

[5] Ernst Haselsteiner and Klemens Breitfuß. Security in near field communication (nfc). In *Workshop on RFID security*, pages 12–14. sn, 2006.

[6] WJ Dallas. Computer-generated holograms. *The computer in optical research*, 41:291–366, 1980.

[7] Pascal Picart. *New techniques in digital holography*. John Wiley & Sons, 2015.

[8] Hsi-Chun Wang, Ya-Wen Cheng, Wan-Chi Huang, Chia-Long Chang, and Shih-Yun Lu. Using modified digital halftoning technique to design invisible 2d barcode by infrared detection. In *Intelligent Technologies and Engineering Systems*, pages 179–186. Springer, 2013.

[9] Xavier Marguerettaz, Frédéric Gremaud, Aurélien Commeureuc, Vickie Aboutanos, Thomas Tiller, and Olivier Rozumek. Identification and authentication using liquid crystal material markings, June 3 2014. US Patent 8,740,088.

[10] Sviatoslav Voloshynovskiy, Maurits Diephuis, Fokko Beekhof, Oleksiy Koval, and Bruno Keel. Towards reproducible results in authentication based on physical non-cloneable functions: The forensic authentication microstructure optical set (famos). In *2012 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 43–48. IEEE, 2012.

[11] Chau-Wai Wong and Min Wu. A study on puf characteristics for counterfeit detection. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 1643–1647. IEEE, 2015.

[12] Gianmarco Baldini, Igor Nai Fovino, Riccardo Satta, Aris Tsois, and Enrico Checchi. Survey of techniques for the fight against counterfeit goods and intellectual property rights (ipr) infringement. *Publ Off Eur Union*, pages 1–130, 2015.

[13] Sviatoslav Voloshynovskiy, Oleksiy Koval, and Thierry Pun. Secure item identification and authentication system based on unclonable features. Patent, March 20 2009. WO. Patent Application No WO 2009/115611.

[14] Baoshi Zhu, Jiankang Wu, and Mohan S Kankanhalli. Print signatures for document authentication. In *Proceedings of the 10th ACM conference on Computer and communications security*, pages 145–154, 2003.

[15] Sviatoslav Voloshynovskiy, Patrick Bas, and Taras Holotyak. Physical object authentication: detection-theoretic comparison of natural and artificial randomness. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, Shanghai, China, March,20-25 2016.

[16] Chau-Wai Wong and Min Wu. Counterfeit detection based on unclonable feature of paper using mobile camera. *IEEE Transactions on Information Forensics and Security*, 12(8):1885–1899, 2017.

[17] Rudolf Schraml, Luca Debiasi, and Andreas Uhl. Real or fake: Mobile device drug packaging authentication. In *Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security*, pages 121–126, 2018.

[18] Guy Adams, Stephen Pollard, and Steven Simske. A study of the interaction of paper substrates on printed forensic imaging. In *Proceedings of the 11th ACM symposium on Document engineering*, pages 263–266, 2011.

[19] Stephen B Pollard, Steven J Simske, and Guy B Adams. Model based print signature profile extraction for forensic analysis of individual text glyphs. In *2010 IEEE International Workshop on Information Forensics and Security*, pages 1–6. IEEE, 2010.

[20] *ISO/IEC 18004: Information Technology - Automatic identification and data capture techniques-Bar Code Symbology-QR Code. 2000.* 2000.

[21] *ISO/IEC 16022: Information technology - Automatic identification and data capture techniques - Data Matrix bar code symbology specification.* 2006.

[22] Olga Taran, Slavi Bonev, and Slava Voloshynovskiy. Clonability of anti-counterfeiting printable graphical codes: a machine learning approach. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, United Kingdom, May 2019.

[23] Rohit Yadav, Iuliia Tkachenko, Alain Trémeau, and Thierry Fournel. Estimation of copy-sensitive codes using a neural approach. In *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security*, pages 77–82, 2019.

[24] Tapas Kanungo, Robert M. Haralick, Henry S. Baird, Werner Stuezle, and David Madigan. A statistical, nonparametric methodology for document degradation model validation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1209–1223, 2000.

[25] Noura Dridi, Yves Delignon, Wadih Sawaya, and François Septier. Blind detection of severely blurred 1d barcode. In *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*, pages 1–5. IEEE, 2010.

[26] Ahmet Emir Dirik and Bertrand Haas. Copy detection pattern-based document protection for variable media. *IET Image Processing*, 6(8):1102–1113, 2012.

[27] Cléo Baras and François Cayre. 2d bar-codes for authentication: A security approach. In *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, pages 1760–1766. IEEE, 2012.

[28] Anh Thu Phan Ho, Bao An Mai Hoang, Wadih Sawaya, and Patrick Bas. Document authentication using graphical codes: Reliable performance analysis and channel optimization. *EURASIP Journal on Information Security*, 2014(1):9, 2014.

[29] Mouhamadou L Diong, Patrick Bas, Chloé Pelle, and Wadih Sawaya. Document authentication using 2d codes: Maximizing the decoding performance using statistical inference. In *IFIP International Conference on Communications and Multimedia Security*, pages 39–54. Springer, 2012.

[30] Jorge Calvo-Zaragoza and Antonio-Javier Gallego. A selectional auto-encoder approach for document image binarization. *Pattern Recognition*, 86:37–47, 2019.

[31] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

[32] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*, pages 3581–3589, 2014.

[33] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[34] Jost Tobias Springenberg. Unsupervised and semi-supervised learning with categorical generative adversarial networks. *arXiv preprint arXiv:1511.06390*, 2015.

[35] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[36] Marco Federici, Anjan Dutta, Patrick Forré, Nate Kushman, and Zeynep Akata. Learning robust representations via multi-view information bottleneck. *arXiv preprint arXiv:2002.07017*, 2020.

[37] Olga Taran, Shideh Rezaeifar, Taras Holotyak, and Slava Voloshynovskiy. Machine learning through cryptographic glasses: combating adversarial attacks by key based diversified aggregation. In *EURASIP Journal on Information Security*, number 10, June 2020.

[38] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[39] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[40] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[41] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[42] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[43] Changxing Ding and Dacheng Tao. Robust face recognition via multimodal deep face representation. *IEEE Transactions on Multimedia*, 17(11):2049–2058, 2015.

[44] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1528–1540. ACM, 2016.

[45] Ryan Kiros, Ruslan Salakhutdinov, and Rich Zemel. Multimodal neural language models. In *International Conference on Machine Learning*, pages 595–603, 2014.

[46] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015.

[47] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

[48] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing. *ieee Computational intelligenCe magazine*, 13(3):55–75, 2018.

[49] Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Brian Kingsbury, et al. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal processing magazine*, 29, 2012.

[50] Wayne Xiong, Lingfeng Wu, Fil Alleva, Jasha Droppo, Xuedong Huang, and Andreas Stolcke. The microsoft 2017 conversational speech recognition system. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5934–5938. IEEE, 2018.

[51] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[52] Marco Melis, Ambra Demontis, Battista Biggio, Gavin Brown, Giorgio Fumera, and Fabio Roli. Is deep learning safe for robot vision? adversarial examples against the icub humanoid. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 751–759, 2017.

[53] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 387–402. Springer, 2013.

[54] Joshua Saxe and Konstantin Berlin. Deep neural network based malware detection using two dimensional binary program features. In *2015 10th International Conference on Malicious and Unwanted Software (MALWARE)*, pages 11–20. IEEE, 2015.

[55] Junshui Ma, Robert P. Sheridan, Andy Liaw, George E. Dahl, and Vladimir Svetnik. Deep neural nets as a method for quantitative structure–activity relationships. *Journal of Chemical Information and Modeling*, 55(2):263–274, 2015. PMID: 25635324.

[56] Moritz Helmstaedter, Kevin L Briggman, Srinivas C Turaga, Viren Jain, H Sebastian Seung, and Winfried Denk. Connectomic reconstruction of the inner plexiform layer in the mouse retina. *Nature*, 500(7461):168, 2013.

[57] Hui Y Xiong, Babak Alipanahi, Leo J Lee, Hannes Bretschneider, Daniele Merico, Ryan KC Yuen, Yimin Hua, Serge Gueroussov, Hamed S Najafabadi, Timothy R Hughes, et al. The human splicing code reveals new insights into the genetic determinants of disease. *Science*, 347(6218):1254806, 2015.

[58] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016.

[59] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

[60] Xiaoyong Yuan, Pan He, Qile Zhu, Rajendra Rana Bhat, and Xiaolin Li. Adversarial examples: Attacks and defenses for deep learning. *arXiv preprint arXiv:1712.07107*, 2017.

[61] Olga Taran, Shideh Rezaeifar, and Slava Voloshynovskiy. Bridging machine learning and cryptography in defence against adversarial attacks. In *Workshop on Objectionable Content and Misinformation (WOCM), ECCV2018*, Munich, Germany, September 2018.

[62] Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. Adversarial examples: Attacks and defenses for deep learning. *IEEE transactions on neural networks and learning systems*, 2019.

[63] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Siwei Li, Li Chen, Michael E Kounavis, and Duen Horng Chau. Shield: Fast, practical defense and vaccination for deep learning using jpeg compression. *arXiv preprint arXiv:1802.06816*, 2018.

[64] Naveed Akhtar and Ajmal Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *arXiv preprint arXiv:1801.00553*, 2018.

[65] James L Massey. Cryptography: Fundamentals and applications. In *Copies of transparencies, Advanced Technology Seminars*, volume 109, page 119, 1993.

[66] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *Security and Privacy (SP), 2016 IEEE Symposium on*, pages 582–597. IEEE, 2016.

[67] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[68] Yi Wu, David Bamman, and Stuart Russell. Adversarial training for relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1778–1783, 2017.

[69] Pierre Moulin and Amish Goel. Locally optimal detection of adversarial inputs to image classifiers. In *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 459–464. IEEE, 2017.

[70] Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. On detecting adversarial perturbations. *arXiv preprint arXiv:1702.04267*, 2017.

[71] Dan Hendrycks and Kevin Gimpel. Early methods for detecting adversarial images. *arXiv preprint arXiv:1608.00530*, 2016.

[72] Xin Li and Fuxin Li. Adversarial examples detection in deep networks with convolutional filter statistics. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5764–5772, 2017.

[73] Reuben Feinman, Ryan R Curtin, Saurabh Shintre, and Andrew B Gardner. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*, 2017.

[74] Hanwei Zhang, Yannis Avrithis, Teddy Furon, and Laurent Amsaleg. Smooth adversarial examples. *arXiv preprint arXiv:1903.11862*, 2019.

[75] Sviatoslav Voloshynovskiy, Shelby Pereira, Alexander Herrigel, Nazanin Baumgärtner, and Thierry Pun. Generalized watermark attack based on watermark estimation and perceptual remodulation. In Ping Wah Wong and Edward J. Delp, editors, *IS&T/SPIE's 12th Annual Symposium, Electronic Imaging 2000: Security and Watermarking of Multimedia Content II*, volume 3971 of *SPIE Proceedings*, San Jose, California USA, 23–28jan 2000. (Paper EI 3971-34) - slides.

[76] Shixiang Gu and Luca Rigazio. Towards deep neural network architectures robust to adversarial examples. *arXiv preprint arXiv:1412.5068*, 2014.

[77] Dongyu Meng and Hao Chen. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 135–147. ACM, 2017.

[78] Sungyoon Lee and Jaewook Lee. Defensive denoising methods against adversarial attack. 2018.

[79] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 501–509, 2019.

[80] Xiaojun Jia, Xingxing Wei, Xiaochun Cao, and Hassan Foroosh. Comdefend: An efficient image compression model to defend adversarial examples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6084–6092, 2019.

[81] Zihao Liu, Qi Liu, Tao Liu, Yanzhi Wang, and Wujie Wen. Feature distillation: Dnn-oriented jpeg compression against adversarial examples. *arXiv preprint arXiv:1803.05787*, 2018.

[82] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

[83] Valentina Zantedeschi, Maria-Irina Nicolae, and Ambrish Rawat. Efficient defenses against adversarial attacks. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 39–49. ACM, 2017.

[84] Nic Ford, Justin Gilmer, Nicolas Carlini, and Dogus Cubuk. Adversarial examples are a natural consequence of test error in noise. *arXiv preprint arXiv:1901.10513*, 2019.

[85] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*, 2017.

[86] Edward Raff, Jared Sylvester, Steven Forsyth, and Mark McLean. Barrage of random transforms for adversarially robust defense. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6528–6537, 2019.

[87] Xuanqing Liu, Minhao Cheng, Huan Zhang, and Cho-Jui Hsieh. Towards robust neural networks via random self-ensemble. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 369–385, 2018.

[88] Zhezhi He, Adnan Siraj Rakin, and Deliang Fan. Parametric noise injection: Trainable randomness to improve deep neural network robustness against adversarial attack. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 588–597, 2019.

[89] Zhonghui You, Jinmian Ye, Kunming Li, Zenglin Xu, and Ping Wang. Adversarial noise layer: Regularize neural network by adding noise. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 909–913. IEEE, 2019.

[90] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[91] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018.

[92] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

[93] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[94] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*, pages 372–387. IEEE, 2016.

[95] Seyed Mohsen Moosavi Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, number EPFL-CONF-218057, 2016.

[96] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1765–1773, 2017.

[97] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017.

[98] Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 3–14. ACM, 2017.

[99] Warren He, James Wei, Xinyun Chen, Nicholas Carlini, and Dawn Song. Adversarial example defenses: Ensembles of weak defenses are not strong. *arXiv preprint arXiv:1706.04701*, 2017.

[100] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 274–283, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[101] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 15–26. ACM, 2017.

[102] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 2019.

[103] Rainer Storn and Kenneth Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.

[104] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095, 2007.

[105] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(Dec):3371–3408, 2010.

[106] Zhipeng Chen, Benedetta Tondi, Xiaolong Li, Rongrong Ni, Yao Zhao, and Mauro Barni. Secure detection of image manipulation by means of random feature selection. *CoRR*, abs/1802.00573, 2018.

[107] Olga Taran, Shideh Rezaeifar, Taras Holotyak, and Slava Voloshynovskiy. Defending against adversarial attacks by randomized diversification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, USA, June 2019.

[108] Olga Taran, Shideh Rezaeifar, Taras Holotyak, and Slava Voloshynovskiy. Robustification of deep net classifiers by key based diversified aggregation with pre-filtering. In *IEEE International Conference on Image Processing (ICIP)*, Taipei, Taiwan, September 2019.

[109] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2, 2010.

[110] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[111] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. The cifar-10 dataset. *online: http://www. cs. toronto. edu/kriz/cifar. html*, 2014.

[112] Slava Voloshynovskiy, Olga Taran, Mouad Kondah, Taras Holotyak, and Danilo Rezende. Variational information bottleneck for semi-supervised classification. In *Entropy Journal special issue "Information Bottleneck: Theory and Applications in Deep Learning"*, volume 22, August 2020.

[113] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5. IEEE, 2015.

[114] Alessandro Achille and Stefano Soatto. Information dropout: Learning optimal representations through noisy computation. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2897–2905, 2018.

[115] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[116] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 5049–5059, 2019.

[117] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pages 529–536, 2005.

[118] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, 2013.

[119] Dan Claudiu Cireşan, Ueli Meier, Luca Maria Gambardella, and Jürgen Schmidhuber. Deep, big, simple neural nets for handwritten digit recognition. *Neural computation*, 22(12):3207–3220, 2010.

[120] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

[121] Rana Ali Amjad and Bernhard Claus Geiger. Learning representations for neural network-based classification using the information bottleneck principle. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[122] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

[123] Slava Voloshynovskiy, Mouad Kondah, Shideh Rezaeifar, Olga Taran, Taras Hotolyak, and Danilo Rezende. Information bottleneck through variational glasses. In *NeurIPS Workshop on Bayesian Deep Learning*, Vancouver, Canada, December 2019.

[124] Yigit Ugur and Abdellatif Zaidi. Variational information bottleneck for unsupervised clustering: Deep gaussian mixture embedding. *Entropy*, 22(2):213, 2020.

[125] Lars Maaløe, Casper Kaae Sønderby, Søren Kaae Sønderby, and Ole Winther. Auxiliary deep generative models. *arXiv preprint arXiv:1602.05473*, 2016.

[126] Marek Śmieja, Maciej Wołczyk, Jacek Tabor, and Bernhard C Geiger. Segma: Semi-supervised gaussian mixture auto-encoder. *arXiv preprint arXiv:1906.09333*, 2019.

[127] Alireza Makhzani and Brendan J Frey. Pixelgan autoencoders. In *Advances in Neural Information Processing Systems*, pages 1975–1985, 2017.

[128] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[129] D.P. Kingma and Max Welling. Auto-encoding variational bayes. *arXiv:1312.6114*, 2014.

[130] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic back-propagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

[131] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR*, 2(5):6, 2017.

[132] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[133] Olga Taran, Slavi Bonev, Taras Holotyak, and Slava Voloshynovskiy. Adversarial detection of counterfeited printable graphical codes: towards "adversarial games" in physical world. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[134] Justin Picard. Digital authentication with copy-detection patterns. In *Optical Security and Counterfeit Deterrence Techniques V*, volume 5310, pages 176–183. International Society for Optics and Photonics, 2004.

[135] Justin Picard and Paul Landry. Two dimensional barcode and method of authentication of such barcode, March 14 2017. US Patent 9,594,993.

[136] Renato Villn, Sviatoslav Voloshynovskiy, Oleksiy Koval, and Thierry Pun. Multilevel 2-d bar codes: Toward high-capacity storage modules for multimedia security and management. *IEEE Transactions on Information Forensics and Security*, 1(4):405–420, 2006.

[137] Scantrust's anti-counterfeit solution isn't just about blockchain. https://www.ledgerinsights.com/scantrust-anti-counterfeit-blockchain. Accessed: 2018-08-17.

[138] Iuliia Tkachenko, William Puech, Olivier Strauss, Christophe Destruel, and J-M Gaudin. Printed document authentication using two level or code. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2149–2153. IEEE, 2016.

[139] H Phuong Nguyen, Agnès Delahaies, Florent Retraint, D Huy Nguyen, Marc Pic, and Frédéric Morain-Nicolier. A watermarking technique to secure printed qr codes using a statistical test. In *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 288–292. IEEE, 2017.

[140] Iuliia Tkachenko, William Puech, Christophe Destruel, Olivier Strauss, Jean-Marc Gaudin, and Christian Guichard. Two-level qr code for private message sharing and document authentication. *IEEE Transactions on Information Forensics and Security*, 11(3):571–583, 2015.

[141] Renato Villán, Sviatoslav Voloshynovskiy, Oleksiy Koval, and Thierry Pun. Multilevel 2d bar codes: Towards high capacity storage modules for multimedia security and management. *IEEE Transactions on Information Forensics and Security*, 1(4):405–420, December 2006.

[142] Yuqiao Cheng, Zhengxin Fu, Bin Yu, and Gang Shen. A new two-level qr code with visual cryptography scheme. *Multimedia Tools and Applications*, 77(16):20629–20649, 2018.

[143] Geoffrey Hinton and Sam T Roweis. Stochastic neighbor embedding. In *NIPS*, volume 15, pages 833–840. Citeseer, 2002.

[144] Mary M Moya, Mark W Koch, and Larry D Hostetler. One-class classifier networks for target recognition applications. *NASA STI/Recon Technical Report N*, 93:24043, 1993.

[145] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

[146] Charu C Aggarwal. Outlier analysis. In *Data mining*, pages 237–263. Springer, 2015.

[147] Yunqiang Chen, Xiang Sean Zhou, and Thomas S Huang. One-class svm for learning in image retrieval. In *Proceedings 2001 International Conference on Image Processing (Cat. No. 01CH37205)*, volume 1, pages 34–37. IEEE, 2001.

[148] I. Goodfellow et al. Generative adversarial nets. *arXiv:1406.2661*, 2014.

[149] Frantisek Grézl, Martin Karafiát, Stanislav Kontár, and Jan Cernocky. Probabilistic and bottle-neck features for lvcsr of meetings. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 4, pages IV–757. IEEE, 2007.

[150] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Appendix A

List of publications and patent

Patent

S. Voloshynovskiy, **O. Taran**, T. Holotyak, M. Ferrari, and K. Eguiazarian, "Signal sampling with joint training of learnable priors for sampling operator and decoder", US Patent App. 16/547,373, 2020.

Journal papers

1. **O. Taran**, S. Rezaeifar, T. Holotyak, and S. Voloshynovskiy, "Machine learning through cryptographic glasses: combating adversarial attacks by key based diversified aggregation," in *Proc. EURASIP Journal on Information Security*, number 10, June, 2020.

2. S. Voloshynovskiy, **O. Taran**, M. Kondah, T. Holotyak, and D. Rezende, "Variational Information Bottleneck for Semi-Supervised Classification," in *Entropy Journal special issue "Information Bottleneck: Theory and Applications in Deep Learning"*, volume 22, August, 2020.

3. D. Ullmann, S. Rezaeifar, **O. Taran**, T. Holotyak, B. Panos, and S. Voloshynovskiy, "Information Bottleneck Classification in Extremely Distributed Systems," in *Proc. Information-Theoretic Methods for Deep Learning Based Data Acquisition, Analysis and Security*, volume 22, October, 2020.

International conferences

1. S. Rezaeifar, M. Diephuis, B. Razeghi, **O. Taran**, D. Ullmann, and S. Voloshynovskiy, "Distributed Semi-Private Image Classification Based on Information-Bottleneck Principle," in *Proc. 28th European Signal Processing Conference (EUSIPCO)*, Amsterdam, Netherlands, 2020.

2. **O. Taran**, S. Bonev, T. Holotyak, and S. Voloshynovskiy, "Adversarial detection of counterfeited printable graphical codes: towards "adversarial games" in physical world," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

3. S. Voloshynovskiy, M. Kondah, S. Rezaeifar, **O. Taran**, T. Hotolyak, and D. J. Rezende, "Information bottleneck through variational glasses," in *Proc. NeurIPS Workshop on Bayesian Deep Learning*, Vancouver, Canada, 2019.

4. **O. Taran**, S. Rezaeifar, T. Holotyak, and S. Voloshynovskiy, "Robustification of deep net classifiers by key based diversified aggregation with pre-filtering," in *Proc. IEEE International Conference on Image Processing (ICIP)*, Taipei, Taiwan, 2019.

5. S. Rezaeifar, B. Razeghi, **O. Taran**, T. Holotyak, and S. Voloshynovskiy, "Reconstruction of privacy-sensitive data from protected templates," in *Proc. IEEE International Conference on Image Processing (ICIP*), Taipei, Taiwan, 2019.

6. **O. Taran**, S. Rezaeifar, T. Holotyak, and S. Voloshynovskiy, "Defending against adversarial attacks by randomized diversification," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, USA, 2019.

7. **O. Taran**, S. Bonev, and S. Voloshynovskiy, "Clonability of anti-counterfeiting printable graphical codes: a machine learning approach," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, United Kingdom, 2019.

8. **O. Taran**, S. Rezaeifar, and S. Voloshynovskiy, "Bridging machine learning and cryptography in defence against adversarial attacks," in *Proc. Workshop on Objectionable Content and Misinformation (WOCM), ECCV2018*, Munich, Germany, 2018.

9. S. Rezaeifar, **O. Taran**, and S. Voloshynovskiy, "Classification by Re-generation: Towards Classification Based on Variational Inference," in *Proc. 26th European Signal Processing Conference (EUSIPCO)*, Rome, Italy, 2018.

10. M. Ferrari, **O. Taran**, T. Holotyak, K. Egiazarian, and S. Voloshynovskiy, "Injecting Image Priors into Learnable Compressive Subsampling," in *Proc. 26th European Signal Processing Conference (EUSIPCO)*, Rome, Italy, 2018.

11. B. Razeghi, S. Voloshynovskiy, D. Kostadinov, and **O. Taran**, "Privacy Preserving Identification Using Sparse Approximation with Ambiguization," in *Proc. IEEE International Workshop on Information Forensics and Security (WIFS)*, Rennes, France, 2017, pp. 1-6.

12. **O. Taran**, S. Rezaeifar, O. Dabrowski, J. Schlechten, T. Holotyak, and S. Voloshynovskiy, "PharmaPack: mobile fine-grained recognition of pharma packages," in *Proc. European Signal Processing Conference (EUSIPCO)*, Kos, Grece, 2017[1].

[1]O. Taran and S. Rezaeifar have the equal contribution.

Appendix B

Robust classifier based on the KDA

B.1 Technical details of training

To ensure a reproducible research the complete code is available at https://github.com/taranO/multi-channel-KDA.

C&W attack

For the fair comparison, the gradient based C&W attack is tested on the classifiers with the architecture identical to those tested in [97]. The implementation is done in TensorFlow. For the training the SGD is used with a learning rate 1e-2 and weight decay 1e-6. The "attacked vanilla" and "transferability vanilla" models are trained during 50 epochs (after 50th epoch the saturation is observed) with a batch size equals to 128. In the multi-channel model, each classifier is trained during 100 epochs using Adam optimiser with a learning rate 1e-3, weight decay 1e-6 and batch size 64. For the ℓ_2 attack the learning rate is 1e-2, confidence 0, maximum number of iterations 1000 with early stopping if the gradient descent gets stuck, the minimum and maximum pixel values equal to -0.5 and 0.5 correspondingly. For the ℓ_0 and ℓ_∞ attacks the constant factor is 2, the rest of the used parameters are the same as in case of ℓ_2 attack.

OnePixel attack

The VGG16 [150] and ResNet18 [40] vanilla models are trained during 100 epochs with learning rate 1e-3, weight decay 5e-4 and batch size 128. For the VGG16 the SGD is used. In case the ResNet18 the Adam was used. In multi-channel system for each classifier the same corresponding parameters are used and the implementation is done in Pytorch.

PGD attack

The PGD attack is used to attack the VGG16 and ResNet18 models. The Pytorh implementation of PGD attack from the FoolBox library[1] is used with the next parameters: α equals to 0.5, step size 0.01 and 100 iterations.

[1]https://foolbox.readthedocs.io/en/stable/index.html

Appendix C

Semi-supervised training

To ensure a reproducible research the complete code is available at https://github.com/taranO/IB-semi-supervised-classification.

C.1 Supervised training without latent space regularization

The baseline supervised model's architecture is based on the cross-entropy term $\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}}$ (3.7) introduced in Section 3.3 and shown in Fig. C.1:

$$\mathcal{L}^{\mathrm{HCP}}_{\mathrm{S-NoReg}}(\boldsymbol{\theta}_{\mathrm{c}}, \boldsymbol{\phi}_{\mathrm{a}}) = \mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}}. \tag{C.1}$$

The parameters of encoder and decoder are shown in Table C.1. The performance of baseline supervised classifier with and without batch normalization corresponds to the parameter $\alpha_{\mathrm{c}} = 0$ in Table C.3 (deterministic scenario) and Table C.4 (stochastic scenario).

Table C.1 The network parameters of baseline classifier trained on $\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}}$. The encoder is trained with and without batch normalization (BN) after Conv2D layers.

Encoder

Size	Layer
$28 \times 28 \times 1$	Input
$14 \times 14 \times 32$	Conv2D, (BN) LeakyReLU
$7 \times 7 \times 64$	Conv2D, (BN) LeakyReLU
$4 \times 4 \times 128$	Conv2D, (BN) LeakyReLU
2048	Flatten
1024	FC, ReLU

Decoder

Size	Layer
1024	Input
500	FC, ReLU
10	FC, Softmax

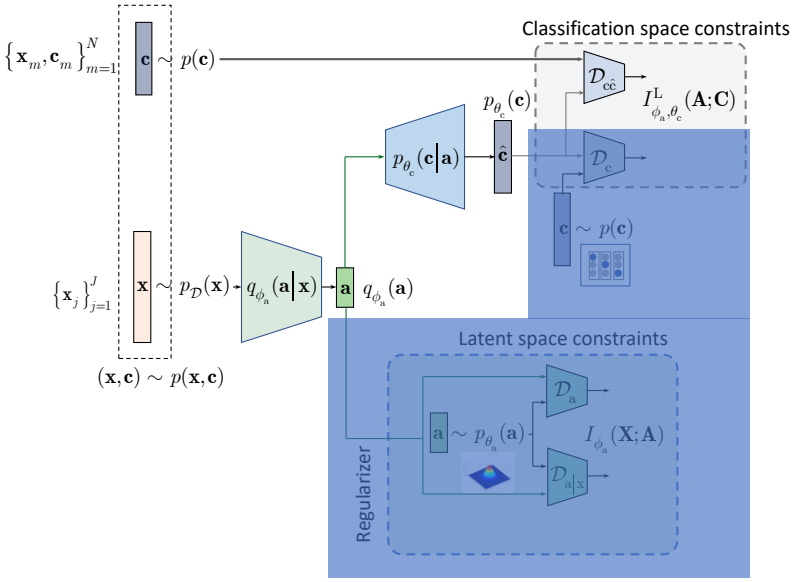


Fig. C.1 Baseline classifier based on $\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}}$ term. The blue shadowed regions are not used.

C.2 Semi-supervised training without latent space regularization and with class label regularizer

The semi-supervised model under investigation is based on terms $\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}}$ and $\mathcal{D}_{\mathrm{c}}$ in (3.8) introduced in Section 3.3 and schematically shown in Fig. C.2:

$$\mathcal{L}^{\mathrm{HCP}}_{\mathrm{SS-NoReg}}(\boldsymbol{\theta}_{\mathrm{c}}, \boldsymbol{\phi}_{\mathrm{a}}) = \mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}} + \alpha_{\mathrm{c}}\mathcal{D}_{\mathrm{c}}. \tag{C.2}$$

The parameters of encoder, decoder and discriminator are shown in Table C.2. The KL-divergence term $\mathcal{D}_{\mathrm{c}}$ is implemented in a form of density ratio estimator (DRE) [132]. In the considered practical implementation, the parameter α_{c} controls the trade-off between the cross-entropy and class discriminator terms. The discriminator $\mathcal{D}_{\mathrm{c}}$ is trained in an adversarial way based on samples generated by the decoder and from targeted distribution.

The performance of semi-supervised classifier with and without batch normalization is shown in Table C.3 (deterministic scenario) and Table C.4 (stochastic scenario).

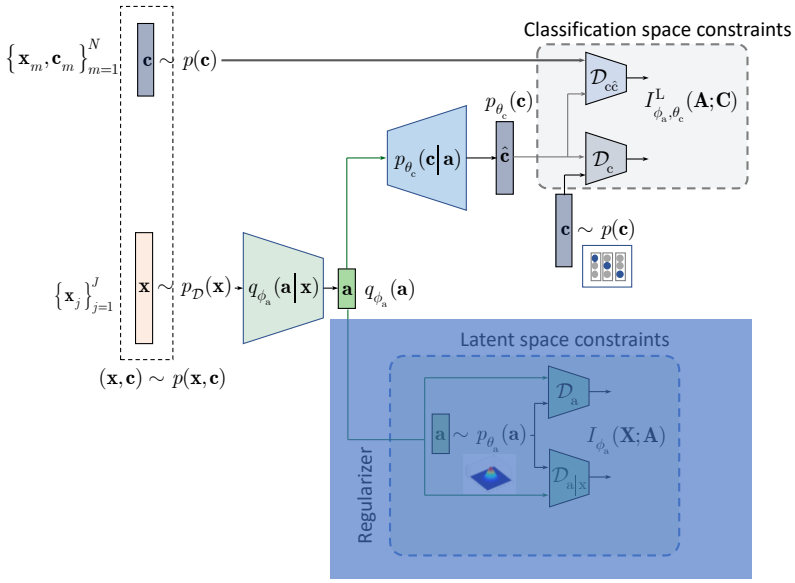


Fig. C.2 Semi-supervised classifier based on the cross-entropy term $\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}}$ and categorical class discriminator $\mathcal{D}_{\mathrm{c}}$. No latent space regularization is applied. The blue shadowed regions are not used.

Table C.2 The network parameters of semi-supervised classifier trained on $\mathcal{D}_{c\hat{c}}$ and $\mathcal{D}_c$. The encoder is trained with and without batch normalization (BN) after Conv2D layers.

Encoder

Size	Layer
$28 \times 28 \times 1$	Input
$14 \times 14 \times 32$	Conv2D, (BN), LeakyReLU
$7 \times 7 \times 64$	Conv2D, (BN), LeakyReLU
$4 \times 4 \times 128$	Conv2D, (BN), LeakyReLU
2048	Flatten
1024	FC, ReLU

Decoder

Size	Layer
1024	Input
500	FC, ReLU
10	FC, Softmax

$\mathcal{D}_c$

Size	Layer
10	Input
500	FC, ReLU
500	FC, ReLU
1	FC, Sigmoid

Table C.3 The performance (percentage error) of **deterministic** classifier based on $\mathcal{D}_{c\hat{c}}+\alpha_c\mathcal{D}_c$ for the encoder with and without batch normalization as a function of Lagrangian multiplier α_c and the number of labeled examples.

Encoder model	α_c	Runs			mean	std
		1	2	3		
MNIST 100						
without BN	0	26.56	26.24	28.04	26.95	0.96
	0.005	20.44	21.93	18.98	20.45	1.48
	0.0005	18.55	20.43	20.59	**19.86**	1.14
	1	19.23	22.42	20.57	20.74	1.60
with BN	0	29.37	29.27	30.62	29.75	0.75
	0.005	27.97	28.02	26.27	27.42	1.00
	0.0005	25.99	23.70	24.47	**24.72**	1.17
	1	27.78	31.98	35.88	31.88	4.05
MNIST 1000						
without BN	0	7.74	6.99	6.97	7.23	0.44
	0.005	5.62	6.06	5.60	**5.76**	0.26
	0.0005	6.30	6.12	6.02	6.15	0.14
	1	5.99	6.27	6.28	6.18	0.16
with BN	0	7.45	6.95	7.52	7.31	0.31
	0.005	5.57	5.08	5.22	**5.29**	0.25
	0.0005	5.60	6.05	6.22	5.96	0.32
	1	6.05	6.41	5.82	6.09	0.30
MNIST all						
without BN	0	0.83	0.83	0.74	**0.80**	0.05
	0.005	0.83	0.82	0.88	0.84	0.03
	0.0005	0.86	0.92	0.82	0.87	0.05
	1	0.72	0.85	0.87	0.81	0.08
with BN	0	0.73	0.67	0.79	0.73	0.06
	0.005	0.72	0.73	0.70	0.72	0.02
	0.0005	0.75	0.77	0.72	0.75	0.03
	1	0.67	0.68	0.73	**0.69**	0.03

Table C.4 The performance (percentage error) of **stochastic** classifier with supervised noisy data (noise std = 0.1, # noise realization = 3) based on $\mathcal{D}_{c\hat{c}} + \alpha_c \mathcal{D}_c$ for the encoder with and without batch normalization as a function of Lagrangian multiplier α_c and the number of labeled examples.

Encoder model	α_c	Runs			mean	std
		1	2	3		
MNIST 100						
without BN	0	25.75	26.61	26.59	26.32	0.49
	0.005	23.34	21.38	24.37	23.03	1.52
	0.0005	19.92	15.83	16.03	**17.26**	2.31
	1	22.51	20.48	21.28	21.42	1.02
with BN	0	30.26	31.24	29.3	30.27	0.97
	0.005	21.17	24.41	24.75	23.44	1.98
	0.0005	22.97	26.38	24.44	24.60	1.71
	1	26.62	30.43	28.44	28.50	1.91
MNIST 1000						
without BN	0	7.68	7.30	7.23	7.4	0.24
	0.005	5.59	5.16	5.80	5.52	0.33
	0.0005	5.59	6	5.84	5.81	0.21
	1	6.66	6.8	7.62	7.03	0.52
with BN	0	6.97	7.06	7.66	7.23	0.38
	0.005	4.42	4.54	4.08	**4.35**	0.24
	0.0005	5.28	5.56	5.14	5.33	0.21
	1	5.77	5.88	5.72	5.79	0.08
MNIST all						
without BN	0	0.8	0.91	0.87	0.86	0.06
	0.005	0.77	0.82	0.88	0.82	0.06
	0.0005	0.86	0.81	0.87	0.85	0.03
	1	0.93	0.85	0.92	0.90	0.04
with BN	0	0.65	0.67	0.71	0.68	0.03
	0.005	0.69	0.77	0.68	0.71	0.05
	0.0005	0.78	0.71	0.74	0.74	0.04
	1	0.71	0.64	0.62	**0.66**	0.05

C.3 Supervised training with hand-crafted latent space regularization

The supervised model under investigation is based on the cross-entropy term $\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}}$ and either term $\mathcal{D}_{\mathrm{a}|\mathrm{x}}$ or $\mathcal{D}_{\mathrm{a}}$ or jointly $\mathcal{D}_{\mathrm{a}|\mathrm{x}}$ and $\mathcal{D}_{\mathrm{a}}$ as defined by (3.9) in Section 3.3. In practical implementation, the regularization based on the adversarial term $\mathcal{D}_{\mathrm{a}}$ is considered similar to AAE due to the flexibility of imposing different priors on the latent space distribution. The implemented system shown in Fig. C.3 is based on:

$$\mathcal{L}^{\mathrm{HCP}}_{\mathrm{S-Reg}}(\boldsymbol{\theta}_{\mathrm{c}}, \boldsymbol{\phi}_{\mathrm{a}}) = \mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}} + \alpha_{\mathrm{a}}\mathcal{D}_{\mathrm{a}}, \tag{C.3}$$

where α_{a} is a regularization parameter controlling a trade-off between the cross-entropy term and latent space regularization term. The Lagrangians above are replaced with respect to (3.9) in Section 3.3 and used it in front of $\mathcal{D}_{\mathrm{a}}$ in contrast to the original formulation (3.9). It is done to keep the term $\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}}$ without a multiplier as the reference to the baseline classifier.

The parameters of encoder, decoder and discriminator are summarized in Table C.5. The performance of this classifier without and with batch normalization is shown in Table C.6 (deterministic scenario) and Table C.7 (stochastic scenario).

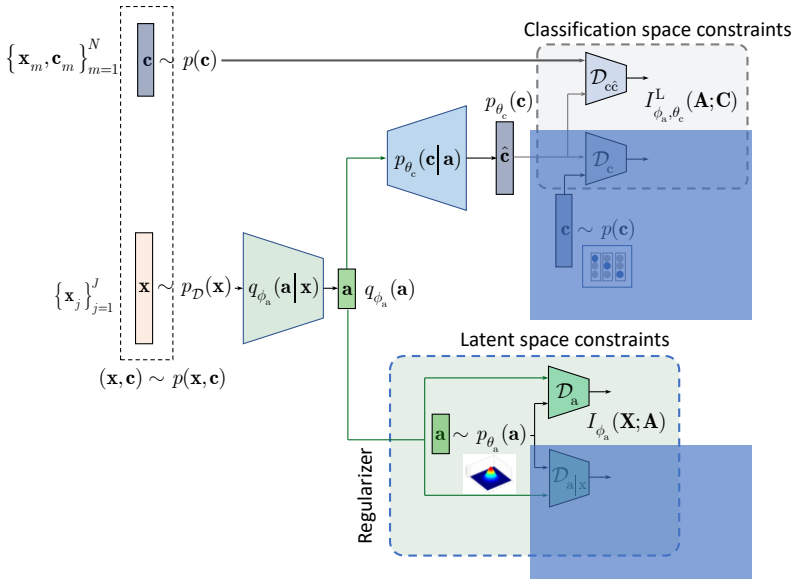


Fig. C.3 Supervised classifier based on the cross-entropy term $\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}}$ and hand-crafted latent space regularization $\mathcal{D}_{\mathrm{a}}$. The blue shadowed parts are not used.

Table C.5 The network parameters of supervised classifier trained on $\mathcal{D}_{c\hat{c}}$ and $\mathcal{D}_a$. The encoder is trained with and without batch normalization (BN) after Conv2D layers. $\mathcal{D}_a$ is trained in the adversarial way.

Encoder

Size	Layer
$28 \times 28 \times 1$	Input
$14 \times 14 \times 32$	Conv2D, (BN), LeakyReLU
$7 \times 7 \times 64$	Conv2D, (BN), LeakyReLU
$4 \times 4 \times 128$	Conv2D, (BN), LeakyReLU
2048	Flatten
1024	FC

Decoder

Size	Layer
1024	Input
500	FC, ReLU
10	FC, Softmax

$\mathcal{D}_a$

Size	Layer
1024	Input
500	FC, ReLU
500	FC, ReLU
1	FC, Sigmoid

Table C.6 . The performance (percentage error) of **deterministic** classifier based on $\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}} + \alpha_{\mathrm{a}}\mathcal{D}_{\mathrm{a}}$ for the encoder with and without batch normalization as a function of Lagrangian multiplier.

Encoder model	α_{a}	Runs			mean	std
		1	2	3		
MNIST 100						
without BN	0	26.79	27.26	27.39	27.15	0.32
	0.005	28.05	25.95	30.72	28.24	2.39
	0.0005	26.67	27.69	28.46	**27.61**	0.89
	1	33.42	33.05	34.81	33.76	0.92
with BN	0	30.37	29.32	29.82	29.83	0.52
	0.005	28.02	31.49	30.80	**30.11**	1.84
	0.0005	34.54	31.92	29.82	31.09	2.36
	1	34.43	44.35	44.25	41.01	5.70
MNIST 1000						
without BN	0	7.16	8.12	7.55	7.61	0.48
	0.005	7.02	6.34	6.59	6.65	0.34
	0.0005	6.73	6.34	6.82	**6.63**	0.26
	1	9.49	9.93	10.56	9.99	0.54
with BN	0	7.39	7.83	7.92	**7.72**	0.28
	0.005	7.94	7.15	8.53	7.88	0.69
	0.0005	8.00	9.62	9.51	9.05	0.91
	1	15.79	14.88	13.71	14.79	1.04
MNIST all						
without BN	0	0.76	0.70	0.81	**0.76**	0.06
	0.005	1.07	1.03	1.13	1.08	0.05
	0.0005	0.84	0.78	0.89	0.84	0.06
	1	4.78	7.24	4.71	5.58	1.44
with BN	0	0.68	0.68	0.69	**0.68**	0.01
	0.005	0.90	0.81	1.12	0.94	0.16
	0.0005	0.87	0.80	0.89	0.85	0.05
	1	2.37	3.61	4.35	3.44	1.00

Table C.7 The performance (percentage error) of **stochastic** classifier with supervised noisy data (noise std = 0.1, # noise realization = 3) based on $\mathcal{D}_{c\hat{c}} + \alpha_a \mathcal{D}_a$ for the encoder with and without batch normalization as a function of Lagrangian multiplier.

Encoder model	α_a	Runs			mean	std
		1	2	3		
MNIST 100						
without BN	0.005	28.13	25.16	29.9	**27.73**	2.40
	0.0005	28.05	30.03	28.11	28.73	1.13
	1	32.33	34.09	33.73	33.38	0.93
with BN	0.005	32.25	33.47	26.01	30.58	4.00
	0.0005	33.37	36.15	35.65	35.06	1.48
	1	33.37	42.37	32.46	36.07	5.48
MNIST 1000						
without BN	0.005	7.37	7.17	6.65	7.06	0.37
	0.0005	7.48	6.68	6.67	**6.94**	0.46
	1	9.48	9.94	11.61	10.34	1.12
with BN	0.005	7.82	7.97	7.81	7.87	0.09
	0.0005	9.5	8.68	9.37	9.18	0.44
	1	12.99	10.52	9.98	11.16	1.60
MNIST all						
without BN	0.005	1.19	1.09	1.06	1.11	0.07
	0.0005	0.79	0.88	0.82	0.83	0.05
	1	6.22	4.81	5	5.34	0.77
with BN	0.005	0.94	1.07	1.04	1.02	0.07
	0.0005	0.78	0.81	0.78	**0.79**	0.02
	1	4.49	3.35	2.18	3.34	1.16

C.4 Semi-supervised training with hand-crafted latent space and class label regularizations

The semi-supervised model under investigation is based on the cross-entropy term $\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}}$ and either term $\mathcal{D}_{\mathrm{a}|\mathrm{x}}$ or $\mathcal{D}_{\mathrm{a}}$ or jointly $\mathcal{D}_{\mathrm{a}|\mathrm{x}}$ and $\mathcal{D}_{\mathrm{a}}$ and the label class regularizer $\mathcal{D}_{\mathrm{c}}$ as defined by (3.10) in Section 3.3. In current implementation, the regularization based on the adversarial term $\mathcal{D}_{\mathrm{a}}$ only is considered as shown in Fig. C.4. The training is based on:

$$\mathcal{L}^{\mathrm{HCP}}_{\mathrm{S-Reg}}(\boldsymbol{\theta}_{\mathrm{c}}, \boldsymbol{\phi}_{\mathrm{a}}) = \mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}} + \alpha_{\mathrm{c}}\mathcal{D}_{\mathrm{c}} + \alpha_{\mathrm{a}}\mathcal{D}_{\mathrm{a}}. \tag{C.4}$$

The parameters of encoder, decoder and both discriminators are shown in Table C.8. The performance of this classifier without and with batch normalization is shown in Table C.9 (deterministic scenario) and Table C.10 (stochastic scenario).

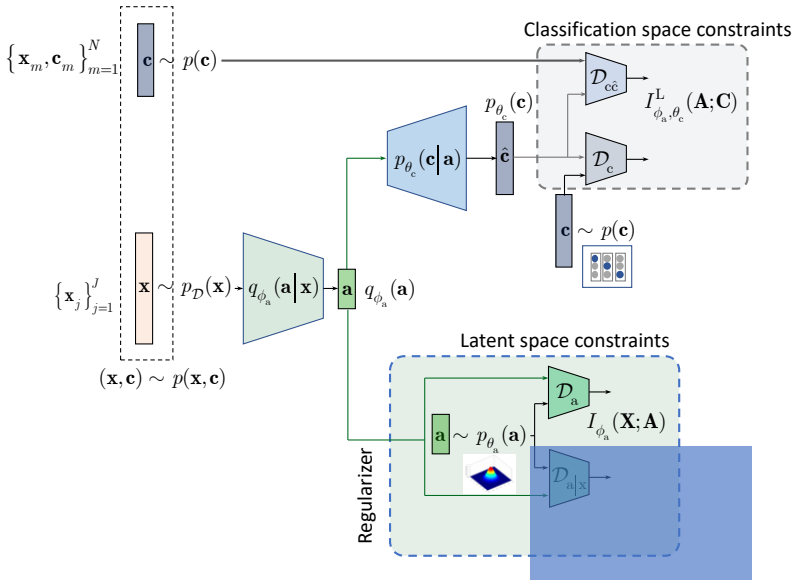


Fig. C.4 Semi-supervised classifier based on the cross-entropy term $\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}}$ and hand-crafted latent space regularization $\mathcal{D}_{\mathrm{a}}$. The blue shadowed parts are not used.

Table C.8 The network parameters of semi-supervised classifier trained on $\mathcal{D}_{c\hat{c}}$, $\mathcal{D}_a$ and $\mathcal{D}_c$. The encoder is trained with and without batch normalization (BN) after Conv2D layers. $\mathcal{D}_a$ and $\mathcal{D}_c$ are trained in the adversarial way.

Encoder

Size	Layer
$28 \times 28 \times 1$	Input
$14 \times 14 \times 32$	Conv2D, (BN), LeakyReLU
$7 \times 7 \times 64$	Conv2D, (BN), LeakyReLU
$4 \times 4 \times 128$	Conv2D, (BN), LeakyReLU
2048	Flatten
1024	FC

Decoder

Size	Layer
1024	Input
500	FC, ReLU
10	FC, Softmax

$\mathcal{D}_c$

Size	Layer
10	Input
500	FC, ReLU
500	FC, ReLU
1	FC, Sigmoid

$\mathcal{D}_a$

Size	Layer
1024	Input
500	FC, ReLU
500	FC, ReLU
1	FC, Sigmoid

Table C.9 The performance (percentage error) of **deterministic** classifier based on $\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}} + \alpha_{\mathrm{a}}\mathcal{D}_{\mathrm{a}} + \alpha_{\mathrm{c}}\mathcal{D}_{\mathrm{c}}$ for the encoder with and without batch normalization.

Encoder model	α_{a}	α_{c}	Runs			mean	std
			1	2	3		
MNIST 100							
without BN	0.005	0.005	21.39	18.12	18.34	19.28	1.83
	0.0005	0.0005	15.33	22.36	13.80	17.16	4.56
	0.005	0.0005	25.66	26.25	28.81	26.91	1.67
	0.0005	0.005	9.82	13.44	13.06	**12.11**	1.99
with BN	0.005	0.005	23.45	21.19	28.87	24.50	3.94
	0.0005	0.0005	28.57	19.06	26.37	24.67	4.98
	0.005	0.0005	26.18	26.18	25.49	25.95	0.40
	0.0005	0.005	8.96	13.82	14.76	12.52	3.11
MNIST 1000							
without BN	0.005	0.005	3.91	4.21	3.70	3.94	0.26
	0.0005	0.0005	3.54	3.72	3.54	3.60	0.10
	0.005	0.0005	6.19	5.80	7.31	6.43	0.78
	0.0005	0.005	2.80	2.82	2.83	2.82	0.02
with BN	0.005	0.005	3.30	2.94	2.93	3.06	0.21
	0.0005	0.0005	2.80	2.53	2.50	2.61	0.17
	0.005	0.0005	3.51	3.75	4.12	3.79	0.31
	0.0005	0.005	2.58	2.27	2.24	**2.37**	0.19
MNIST all							
without BN	0.005	0.005	1.04	1.07	1.07	1.06	0.02
	0.0005	0.0005	0.86	0.90	0.88	0.88	0.02
	0.005	0.0005	1.08	0.92	1.09	1.03	0.10
	0.0005	0.005	0.85	0.93	0.93	0.90	0.05
with BN	0.005	0.005	1.10	1.01	0.93	1.01	0.09
	0.0005	0.0005	0.84	0.88	0.83	0.85	0.03
	0.005	0.0005	1.10	1.12	0.93	1.05	0.10
	0.0005	0.005	0.76	0.82	0.79	**0.79**	0.03

Table C.10 The performance (percentage error) of **stochastic** classifier with supervised noisy data (noise std = 0.1, # noise realization = 3) based on $\mathcal{D}_{c\hat{c}} + \alpha_a \mathcal{D}_a + \alpha_c \mathcal{D}_c$ for the encoder with and without batch normalization.

Encoder model	α_a	α_c	Runs			mean	std
			1	2	3		
MNIST 100							
without BN	0.005	0.005	12.4	18.05	16.73	15.73	2.96
	0.0005	0.0005	15.01	11.16	14.74	13.64	2.15
	0.005	0.0005	23.31	26.61	25.41	25.11	1.67
	0.0005	0.005	9.21	9.02	10.12	**9.45**	0.59
with BN	0.005	0.005	13.55	22.48	14.72	16.92	4.85
	0.0005	0.0005	8.37	15.01	26.92	16.77	9.40
	0.005	0.0005	32.12	30.27	31.44	31.28	0.94
	0.0005	0.005	5.46	17	11.54	11.33	5.77
MNIST 1000							
without BN	0.005	0.005	3.9	4.25	4.02	4.06	0.18
	0.0005	0.0005	3.64	3.82	4.11	3.86	0.24
	0.005	0.0005	6.68	5.34	6.36	6.13	0.70
	0.0005	0.005	3.03	2.88	2.66	2.86	0.19
with BN	0.005	0.005	2.96	3.37	2.98	3.10	0.23
	0.0005	0.0005	2.87	3.10	2.73	2.90	0.19
	0.005	0.0005	3.72	3.8	4.14	3.89	0.22
	0.0005	0.005	2.57	2.39	2.28	**2.41**	0.15
MNIST all							
without BN	0.005	0.005	1.05	1.09	1.1	1.08	0.33
	0.0005	0.0005	0.94	0.96	0.9	0.93	0.03
	0.005	0.0005	1.16	1.14	1.13	1.14	0.02
	0.0005	0.005	0.88	0.92	0.91	0.90	0.02
with BN	0.005	0.005	0.98	0.84	0.94	0.92	0.07
	0.0005	0.0005	0.79	0.96	0.82	0.86	0.09
	0.005	0.0005	1.04	1.05	1.03	1.04	0.01
	0.0005	0.005	0.74	0.78	0.84	**0.79**	0.05

C.5 Semi-supervised training with learnable latent space regularization

The semi-supervised model under investigation is based on the cross-entropy term $\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}}$, the MSE term representing $\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$, the label class regularizer $\mathcal{D}_{\mathrm{c}}$ and either term $\mathcal{D}_{\mathrm{z}|\mathrm{x}}$ or $\mathcal{D}_{\mathrm{z}}$ or jointly $\mathcal{D}_{\mathrm{z}|\mathrm{x}}$ and $\mathcal{D}_{\mathrm{z}}$ as defined by (3.16) in Section 3.4. In practical implementation, the regularization of the latent space based on the adversarial term $\mathcal{D}_{\mathrm{z}}$ only is considered to compare it with the vanilla AAE as shown in Fig. C.5. The encoder is also not conditioned on $\mathbf{c}$ as in the original semi-supervised AAE. Thus, the tested system is based on:

$$\mathcal{L}^{\mathrm{LP}}_{\mathrm{SS-AAE}}(\boldsymbol{\theta}_{\mathrm{c}}, \boldsymbol{\theta}_{\mathrm{x}}, \boldsymbol{\phi}_{\mathrm{a}}, \boldsymbol{\phi}_{\mathrm{z}}) = \beta_{\mathrm{c}}\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}} + \beta_{\mathrm{c}}\mathcal{D}_{\mathrm{c}} + \mathcal{D}_{\mathrm{z}} + \beta_{\mathrm{x}}\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}. \tag{C.5}$$

The parameters β_{x} and β_{c} are set to 1 to compare the investigated system with the vanilla AAE. However, these parameters can be also optimized in practice.

The parameters of encoder and decoder are shown in Table C.11. The performance of this classifier without and with batch normalization is shown in Table C.12 (deterministic scenario) and Table C.13 (stochastic scenario).

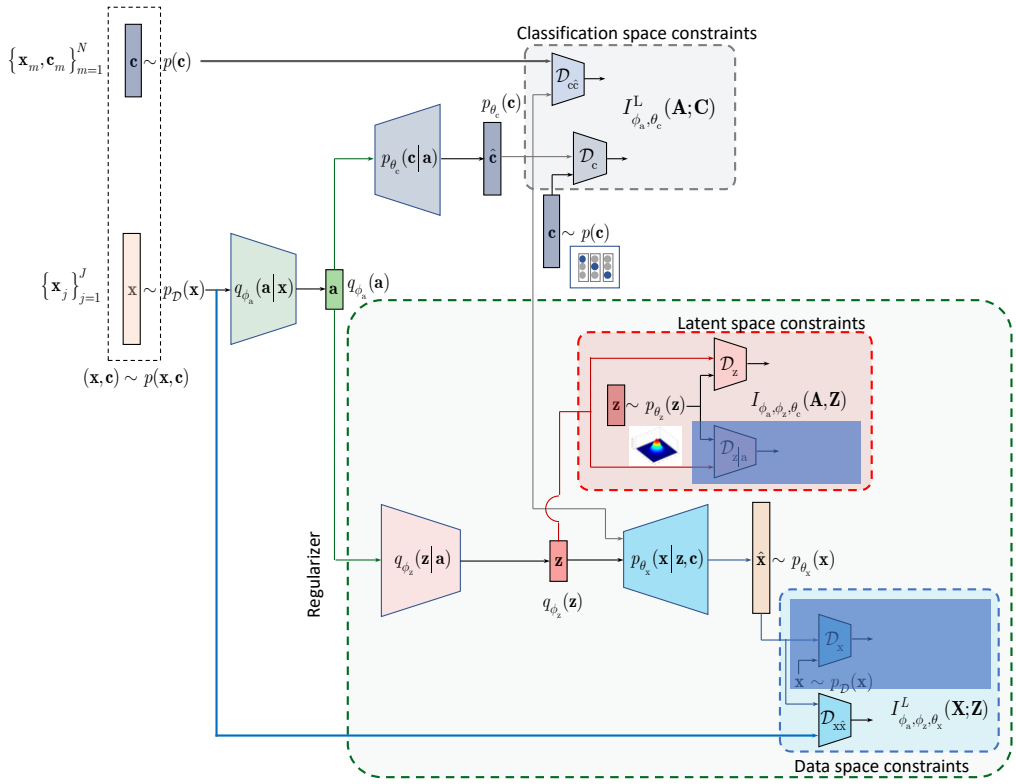


Fig. C.5 Semi-supervised classifier with learnable priors: the cross-entropy $\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}}$, MSE $\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$, class label $\mathcal{D}_{\mathrm{c}}$ and latent space regularization $\mathcal{D}_{\mathrm{a}}$. The blue shadowed parts are not used.

Table C.11 The encoder and decoder of semi-supervised classifier trained based on $\mathcal{D}_{c\hat{c}}$, $\mathcal{D}_c$ and $\mathcal{D}_z$. The encoder is trained with and without batch normalization (BN) after Conv2D layers. $\mathcal{D}_c$ and $\mathcal{D}_z$ are trained in the adversarial way.

Encoder

Size		Layer	
$28 \times 28 \times 1^*$		Input	
$14 \times 14 \times 32$		Conv2D, (BN), LeakyReLU	
$7 \times 7 \times 64$		Conv2D, (BN), LeakyReLU	
$4 \times 4 \times 128$		Conv2D, (BN), LeakyReLU	
2048		Flatten	
1024		FC, ReLU	
10	10	FC, Softmax	FC

Decoder

Size	Layer
$10 + 10$	Input
$7 \times 7 \times 128$	FC, Reshape, BN, ReLU
$14 \times 14 \times 128$	Conv2DTrans, BN, ReLU
$28 \times 28 \times 128$	Conv2DTrans, BN, ReLU
$28 \times 28 \times 64$	Conv2DTrans, BN, ReLU
$28 \times 28 \times 1$	Conv2DTrans, Sigmoid

$\mathcal{D}_z$

Size	Layer
10	Input
500	FC, ReLU
500	FC, ReLU
1	FC, Sigmoid

$\mathcal{D}_c$

Size	Layer
10	Input
500	FC, ReLU
500	FC, ReLU
1	FC, Sigmoid

Table C.12 The performance (percentage error) of **deterministic** classifier based on $\mathcal{D}_{c\hat{c}} + \mathcal{D}_c + \mathcal{D}_z + \mathcal{D}_{x\hat{x}}$ for the encoder with and without batch normalization.

Encoder model	Runs			mean	std
	1	2	3		
MNIST 100					
without BN	2.15	2.05	1.78	1.99	0.19
with BN	1.57	1.56	1.92	**1.68**	0.21
MNIST 1000					
without BN	1.55	1.47	1.53	1.52	0.04
with BN	1.37	1.34	1.73	**1.48**	0.22
MNIST all					
without BN	0.78	0.7	0.82	0.77	0.06
with BN	0.79	0.77	0.76	**0.77**	0.02

Table C.13 The performance (percentage error) of **stochastic** classifier with supervised noisy data (noise std = 0.1, # noise realization = 3) based on $\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}} + \mathcal{D}_{\mathrm{c}} + \mathcal{D}_{\mathrm{z}} + \mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$ for the encoder with and without batch normalization.

Encoder model	Runs			mean	std
	1	2	3		
MNIST 100					
without BN	1.55	3.19	2.11	2.28	0.83
with BN	1.4	1.33	1.72	**1.48**	0.21
MNIST 1000					
without BN	1.73	1.53	1.6	1.62	0.10
with BN	1.28	1.43	1.2	**1.30**	0.12
MNIST all					
without BN	0.94	0.86	0.86	0.89	0.05
with BN	0.77	0.65	0.84	**0.75**	0.10

C.6 Semi-supervised training with learnable latent space regularization and adversarial reconstruction

The semi-supervised model under investigation is similar to the previously considered model but in addition to the MSE reconstruction term representing $\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$ it also contains the adversarial reconstruction term $\mathcal{D}_{\mathrm{x}}$ as defined by (3.17) in Section 3.4. In practical implementation, the regularization of the latent space based on the adversarial term $\mathcal{D}_{\mathrm{z}}$ is considered as shown in Fig. C.6. The training is based on:

$$\mathcal{L}^{\mathrm{LP}}_{\mathrm{SS-AAE}}(\boldsymbol{\theta}_{\mathrm{c}}, \boldsymbol{\theta}_{\mathrm{x}}, \boldsymbol{\phi}_{\mathrm{a}}, \boldsymbol{\phi}_{\mathrm{z}}) = \mathcal{D}_{\mathrm{z}} + \mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}} + \mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}} + \mathcal{D}_{\mathrm{c}} + \alpha_{\mathrm{x}}\mathcal{D}_{\mathrm{x}}. \tag{C.6}$$

The parameters of encoder and decoder are shown in Table C.14. The performance of this classifier without and with batch normalization is shown in Table C.15 (deterministic scenario) and Table C.16 (stochastic scenario).

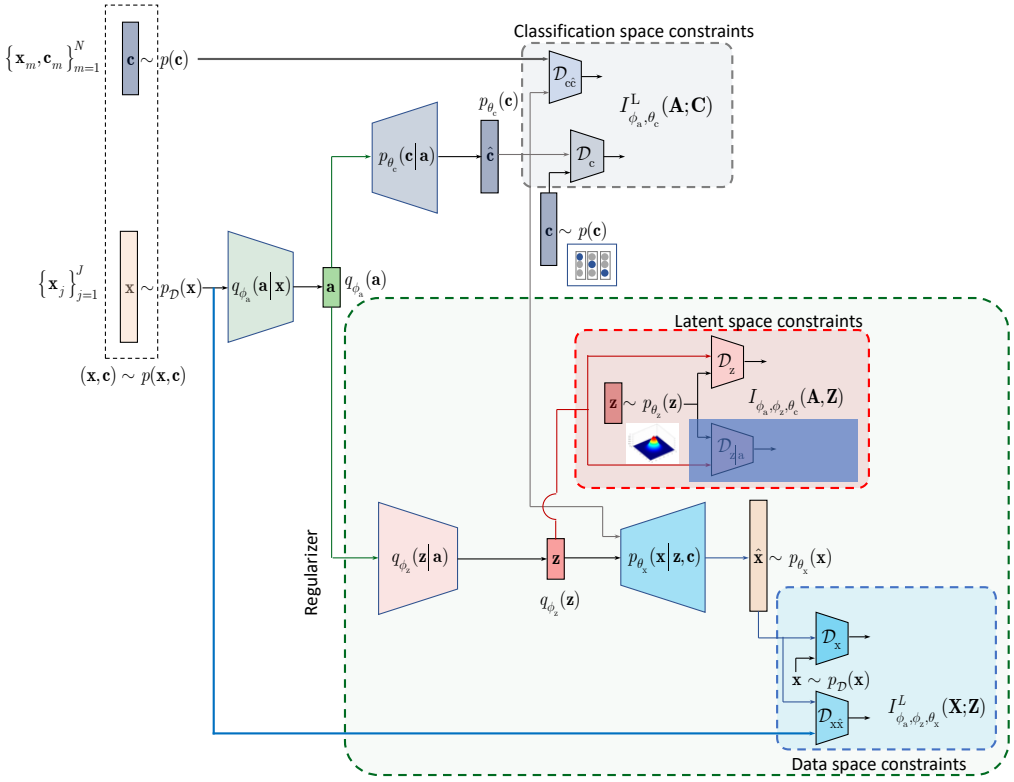


Fig. C.6 Semi-supervised classifier with learnable priors: the cross-entropy $\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}}$, MSE $\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$, adversarial reconstruction $\mathcal{D}_{\mathrm{x}}$, class label $\mathcal{D}_{\mathrm{c}}$ and latent space regularizer $\mathcal{D}_{\mathrm{z}}$. The blue shadowed parts are not used.

Table C.14 The network parameters of semi-supervised classifier trained based on $\mathcal{D}_{c\hat{c}}$, $\mathcal{D}_c$ and $\mathcal{D}_z$. The encoder is trained with and without batch normalization (BN) after Conv2D layers. $\mathcal{D}_c$ and $\mathcal{D}_z$ are trained in the adversarial way.

Encoder

Size		Layer	
$28 \times 28 \times 1$		Input	
$14 \times 14 \times 32$		Conv2D, (BN), LeakyReLU	
$7 \times 7 \times 64$		Conv2D, (BN), LeakyReLU	
$4 \times 4 \times 128$		Conv2D, (BN), LeakyReLU	
2048		Flatten	
1024		FC, ReLU	
10	10	FC, Softmax	FC

Decoder

Size	Layer
$10 + 10$	Input
$7 \times 7 \times 128$	FC, Reshape, BN, ReLU
$14 \times 14 \times 128$	Conv2DTrans, BN, ReLU
$28 \times 28 \times 128$	Conv2DTrans, BN, ReLU
$28 \times 28 \times 64$	Conv2DTrans, BN, ReLU
$28 \times 28 \times 1$	Conv2DTrans, Sigmoid

$\mathcal{D}_z$

Size	Layer
10	Input
500	FC, ReLU
500	FC, ReLU
1	FC, Sigmoid

$\mathcal{D}_c$

Size	Layer
10	Input
500	FC, ReLU
500	FC, ReLU
1	FC, Sigmoid

$\mathcal{D}_x$

Size	Layer
$28 \times 28 \times 1$	Input
$14 \times 14 \times 64$	Conv2D, LeakyReLU
$7 \times 7 \times 64$	Conv2D, LeakyReLU
$4 \times 4 \times 128$	Conv2D, LeakyReLU
$4 \times 4 \times 256$	Conv2D, LeakyReLU
4096	Flatten
1	FC, Sigmoid

Table C.15 The performance (percentage error) of **deterministic** classifier based on $\mathcal{D}_{\mathrm{c}\hat{\mathrm{c}}} + \mathcal{D}_{\mathrm{c}} + \mathcal{D}_{\mathrm{z}} + \mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}} + \alpha_{\mathrm{x}}\mathcal{D}_{\mathrm{x}}$ for the encoder with and without batch normalization.

Encoder model	α_{x}	Runs			mean	std
		1	2	3		
MNIST 100						
without BN	0.005	2.85	3.36	2.77	2.99	0.32
	0.0005	2.58	2.49	3.08	2.72	0.32
	1	19.62	19.96	15.97	18.52	2.21
with BN	0.005	1.56	1.33	1.35	**1.41**	0.13
	0.0005	1.68	1.66	2.02	1.79	0.20
	1	20.85	13.6	21.67	18.71	4.44
MNIST 1000						
without BN	0.005	2.29	2.35	2.11	2.25	0.12
	0.0005	1.69	1.88	2.24	1.94	0.28
	1	3.47	3.30	4.12	3.63	0.43
with BN	0.005	1.18	1.21	1.09	**1.16**	0.06
	0.0005	1.44	1.28	1.29	1.34	0.09
	1	4.14	2.94	2.48	3.19	0.86
MNIST all						
without BN	0.005	0.97	1.01	1.04	1.01	0.04
	0.0005	0.88	0.85	0.93	0.89	0.04
	1	1.31	1.28	1.47	1.35	0.10
with BN	0.005	0.81	0.83	0.75	0.80	0.04
	0.0005	0.73	0.78	0.75	**0.75**	0.03
	1	0.88	0.86	1.27	1.00	0.23

Table C.16 The performance (percentage error) of **stochastic** classifier with supervised noisy data (noise std = 0.1, # noise realization = 3) based on $\mathcal{D}_{c\hat{c}} + \mathcal{D}_c + \mathcal{D}_z + \mathcal{D}_{x\hat{x}} + \alpha_x \mathcal{D}_x$ for the encoder with and without batch normalization.

Encoder model	α_x	Runs			mean	std
		1	2	3		
MNIST 100						
without BN	0.005	2.45	3.04	2.67	2.72	0.30
	0.0005	2.63	2.3	2.45	2.46	0.17
with BN	0.005	1.34	1.21	6.4	2.98	2.96
	0.0005	1.35	1.51	1.93	**1.60**	0.30
MNIST 1000						
without BN	0.005	2.31	2.26	2.2	2.26	0.06
	0.0005	1.71	2.16	1.86	1.91	0.23
with BN	0.005	1.23	1.31	1.10	**1.21**	0.11
	0.0005	1.42	1.62	1.37	1.47	0.13
MNIST all						
without BN	0.005	0.93	1.01	1.05	1.00	0.06
	0.0005	0.92	0.83	0.88	0.88	0.05
with BN	0.005	0.88	0.86	0.91	0.88	0.03
	0.0005	0.77	0.80	0.80	**0.79**	0.02

Appendix D

Copy detection patterns

D.1 Supervised classification with respect to the HC fakes

D.1.1 Technical details of training

The base line model for the supervised classification is based on the cross-entropy term $\mathcal{D}_{c\hat{c}}$ (3.7) introduced in Section 3.3. The used model's architecture is given in Table D.1. The experiments are performed on the Indigo mobile dataset that is split into three sub-sets: *training* with 40% of data, *validation* with 10% of data and 50% of data is used for the *test*. To avoid the bias in the choice of training and test data, the classification model is trained five times on the randomly shifted data. Each time the model is trained with the learning rate equals to 1e-4, the batch of size 21 and the cross-entropy loss. The Adam is used as an optimizer. The supervised classification is performed in two scenarios: *(i)* five class classification and *(ii)* two class classification that are discussed in Section 4.4. Taking into account the sufficiently small amount of data available for the training, in case of two class classification the next data augmentations are used:

- the rotation on 90°, 180° and 270°;
- the gamma correction with $\gamma \in [0.4, 1.3]$ with step 0.2.

Table D.1 The architecture of the model used for the supervised classification and trained with respect to the cross-entropy term $\mathcal{D}_{c\hat{c}}$, where c equals to 5 for the five class classification scenario and to 2 for the two class classification case.

Supervised classifier	
Size	Layer
$330 \times 330 \times 3$	Input
$326 \times 326 \times 32$	Conv2d, ReLU
$108 \times 108 \times 32$	MaxPooling2D
$104 \times 104 \times 16$	Conv2d, ReLU
$34 \times 34 \times 16$	MaxPooling2D
$30 \times 30 \times 8$	Conv2d, ReLU
$10 \times 10 \times 8$	MaxPooling2D
800	Flatten
512	Dense, ReLU
128	Dense, ReLU
64	Dense, ReLU
16	Dense, ReLU
c	Dense, Softmax

D.1.2 Five class supervised classification

Table D.2 Two classes (original, fake) classification error in % on the test sub-set for a model trained in a supervised way on the printed original codes and four types of fakes.

Run	Original	Fake #1 (white, gray) & Fake #2 (white, gray)
	P_{miss}	P_{fa}
1	0.00	0.00
2	0.00	0.35
3	0.00	0.00
4	0.00	0.35
5	0.00	0.71
mean	0.00	0.28
std	0.00	0.30

Table D.3 Three classes (original, fake #1, fake #2) classification error in % on the test sub-set for a model trained in a supervised way on the printed original codes and four types of fake.

Run	Original	Fake #1 (white, gray)	Fake #2 (white, gray)
	P_{miss}	P_{fa}	P_{fa}
1	0.00	0.00	0.00
2	0.00	1.06	0.00
3	0.00	0.35	0.71
4	0.00	0.71	0.00
5	0.00	1.77	1.06
mean	0.00	0.78	0.35
std	0.00	0.68	0.50

Table D.4 Five classes (original, fake # 1 white, fake # 1 gray, fake #2 white and fake # 2 gray) classification error in % on the test sub-set for a model trained in a supervised way on the printed original codes and four types of fakes.

Run	Original	Fake #1 white	Fake #1 gray	Fake #2 white	Fake #2 gray
	P_{miss}	P_{fa}	P_{fa}	P_{fa}	P_{fa}
1	0.00	20.57	21.28	22.70	7.09
2	0.00	18.44	20.57	7.09	19.15
3	0.00	17.73	22.70	15.60	9.93
4	0.00	23.40	21.28	15.60	7.80
5	0.00	36.17	21.99	23.40	12.77
mean	0.00	23.26	21.56	16.88	11.35
std	0.00	7.55	0.81	6.62	4.89

D.1.3 Two class supervised classification

Table D.5 Two classes (original, fake) classification error in % on the test sub-set for a model trained in a supervised way on the printed originals and fakes #1 white. The test is performed for all type of fakes.

Run	Original	Fake #1 white	Fake #1 gray	Fake #2 white	Fake # 2 gray
	P_{miss}	P_{fa}	P_{fa}	P_{fa}	P_{fa}
1	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.71	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00
mean	0.00	0.00	0.14	0.00	0.00
std	0.00	0.00	0.32	0.00	0.00

Table D.6 Two classes (original, fake) classification error in % on the test sub-set for a model trained in a supervised way on the printed originals and fakes #1 gray. The test is performed for all type of fakes.

Run	Original	Fake #1 white	Fake #1 gray	Fake #2 white	Fake # 2 gray
	P_{miss}	P_{fa}	P_{fa}	P_{fa}	P_{fa}
1	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00
mean	0.00	0.00	0.00	0.00	0.00
std	0.00	0.00	0.00	0.00	0.00

Table D.7 Two classes (original, fake) classification error in % on the test sub-set for a model trained in a supervised way on the printed originals and fakes #2 white. The test is performed for all type of fakes.

Run	Original	Fake #1 white	Fake #1 gray	Fake #2 white	Fakes # 2 gray
	P_{miss}	P_{fa}	P_{fa}	P_{fa}	P_{fa}
1	0.00	99.29	100	0.00	0.00
2	0.00	99.29	100	0.00	0.00
3	0.00	99.29	100	0.00	0.00
4	0.00	100.00	100	0.00	0.00
5	0.00	99.29	100	0.00	0.00
mean	0.00	99.43	100	0.00	0.00
std	0.00	0.32	0.00	0.00	0.00

Table D.8 Two classes (original, fake) classification error in % on the test sub-set for a model trained in a supervised way on the printed originals and fakes #1 white. The test is performed for all type of fakes.

Run	Original	Fake #1 white	Fake #1 gray	Fake #2 white	Fake # 2 gray
	P_{miss}	P_{fa}	P_{fa}	P_{fa}	P_{fa}
1	0.00	99.29	100	0.00	0.00
2	0.00	98.58	99.29	0.00	0.00
3	0.00	99.29	100	0.00	0.00
4	0.00	100	100	0.00	0.00
5	0.00	99.29	100	0.00	0.00
mean	0.00	99.29	99.86	0.00	0.00
std	0.00	0.50	0.32	0.00	0.00

D.2 One class classification with respect to the HC fakes

D.2.1 OC-SVM in the spatial domain

The one class classification of the PGC in the spatial domain is based on the OC-SVM trained with respect to the Pearson correlation and Hamming distance between the original printed codes and the corresponding references (digital or physical). If needed the binarization of the real valued printed codes and physical references is performed via simple thresholding with respect to an optimal threshold determined via Otsu method [2] individually for each code. The experiments are performed on the Indigo mobile dataset that is split into *(i)* the training sub-set with 40% of the data, *(ii)* validation sub-set with 10% of the data and *(iii)* the test sub-set with 50% of the data. To avoid the bias in the training data selection, the OC-SVM is trained five times on the randomly shifted data. The python *OneClassSVM* method from the *sklearn* package is used with the next training parameters:

- kernel="rbf";
- gamma=0.1;
- nu=0.03 for the digital templates and nu=0.1 for the physical references.

The detailed information about each run classification error is given in Table D.1.

Fig. D.1 The OC-SVM classification error in % on the test sub-set in the spatial domain with respect to the Pearson correlation and Hamming distance between the printed codes and the corresponding references (digital or physical).

Run	Original	Fake #1 white	Fake #1 gray	Fake #2 white	Fake #2 gray
	P_{miss}	P_{fa}	P_{fa}	P_{fa}	P_{fa}
Train on the grayscale data with respect to the *digital* templates					
1	2.8	1.4	2.8	0	0
2	4.3	2.8	4.3	0	0
3	2.8	5.7	5.7	0	0
4	3.5	0.7	2.8	0	0
5	2.1	2.1	3.5	0	0
mean	3.1	2.54	3.82	0	0
std	0.83	1.93	1.22	0	0
Train on the rgb data with respect to the *digital* templates					
1	2.8	2.1	0.7	0	0
2	4.3	3.5	3.5	0	0
3	2.1	2.1	2.1	0	0
4	3.5	1.4	0	0	0
5	1.4	1.4	0.7	0	0
mean	2.82	2.1	1.4	0	0
std	1.14	0.86	1.4	0	0
Train on the grayscale data with respect to the *physical* references					
1	9.3	33.6	40	4.3	1.4
2	11.4	37.1	42.1	0	0.7
3	18.6	38.6	35	0	0
4	9.3	25	37.9	3.6	2.1
5	8.6	45	47.9	0.7	1.4
mean	11.44	35.86	40.58	1.72	1.12
std	4.14	7.34	4.86	2.07	0.8
Train on the rgb data with respect to the *physical* references					
1	8.6	30	40.7	3.6	1.4
2	13.6	32.1	42.1	0	0.7
3	16.4	30	30.7	0	0
4	8.6	25	37.1	2.9	1.4
5	8.6	42.1	47.1	0.7	1.4
mean	11.16	31.84	39.54	1.44	0.98
std	3.64	6.30	6.11	1.69	0.63

D.2.2 One class classification from the IB point of view

D.2.2.1 Mutual information decomposition tricks

According to the definition (4.5) in Section 4.5.2, the mutual information $I(\mathbf{A};\mathbf{T})$ can be decomposed as:

$$I(\mathbf{A};\mathbf{T}) = \mathbb{E}_{p(\mathbf{a},\mathbf{t})}\left[\log \frac{p(\mathbf{t}|\mathbf{a})}{p_{\mathcal{D}}(\mathbf{t})}\right] = -\mathbb{E}_{p_{\mathcal{D}}(\mathbf{t})}\left[\log p_{\mathcal{D}}(\mathbf{t})\right] + \mathbb{E}_{p(\mathbf{a},\mathbf{t})}\left[\log p(\mathbf{t}|\mathbf{a})\right]. \tag{D.1}$$

The first term of this decomposition corresponds to the entropy of references $H_{\mathcal{D}}(\mathbf{T}) = -\mathbb{E}_{p_{\mathcal{D}}(\mathbf{t})}\left[\log p_{\mathcal{D}}(\mathbf{t})\right]$. In the second term the transition probability $p(\mathbf{t}|\mathbf{a})$ is unknown. At the same time, it can be written as:

$$\mathbb{E}_{p(\mathbf{a},\mathbf{t})}\left[\log p(\mathbf{t}|\mathbf{a})\right] = \int_{\mathbf{t}}\int_{\mathbf{a}} p(\mathbf{a},\mathbf{t}) \log p(\mathbf{t}|\mathbf{a})\ d\mathbf{t}\ d\mathbf{a}. \tag{D.2}$$

The expectation with respect to the joint distribution $p(\mathbf{a},\mathbf{t})$ can also be defined via the marginalization $p(\mathbf{a},\mathbf{t}) = \int_{\mathbf{x}} p(\mathbf{x},\mathbf{a},\mathbf{t})d\mathbf{x} = \int_{\mathbf{x}} p(\mathbf{t},\mathbf{x})q_{\phi_{\mathrm{a}}}(\mathbf{a}|\mathbf{x})d\mathbf{x}$. Combing these results, one can obtain:

$$\begin{aligned}\mathbb{E}_{p(\mathbf{a},\mathbf{t})}\left[\log p(\mathbf{t}|\mathbf{a})\right] &= \int_{\mathbf{t}}\int_{\mathbf{a}}\int_{\mathbf{x}} p(\mathbf{t},\mathbf{x}) q_{\phi_{\mathrm{a}}}(\mathbf{a}|\mathbf{x}) \log p(\mathbf{t}|\mathbf{a})\ d\mathbf{t}\ d\mathbf{a}\ d\mathbf{x} \\ &= \mathbb{E}_{p(\mathbf{t},\mathbf{x})}\left[\mathbb{E}_{q_{\phi_{\mathrm{a}}}(\mathbf{a}|\mathbf{x})}\left[\log p(\mathbf{t}|\mathbf{a})\right]\right].\end{aligned} \tag{D.3}$$

To overcome the problem of unknown $p(\mathbf{t}|\mathbf{a})$, a variational distribution $p_{\boldsymbol{\theta}_t}(\mathbf{t}|\mathbf{a})$ parametrized via a set of parameters $\boldsymbol{\theta}_t$ is applied to approximate $p(\mathbf{t}|\mathbf{a})$:

$$\begin{aligned}\mathbb{E}_{p(\mathbf{a},\mathbf{t})}\left[\log p(\mathbf{t}|\mathbf{a})\right] &= \mathbb{E}_{p(\mathbf{t},\mathbf{x})}\left[\mathbb{E}_{q_{\phi_{\mathrm{a}}}(\mathbf{a}|\mathbf{x})}\left[\log p(\mathbf{t}|\mathbf{a})\frac{p_{\boldsymbol{\theta}_t}(\mathbf{t}|\mathbf{a})}{p_{\boldsymbol{\theta}_t}(\mathbf{t}|\mathbf{a})}\right]\right] \\ &= \mathbb{E}_{p(\mathbf{t},\mathbf{x})}\left[\mathbb{E}_{q_{\phi_{\mathrm{a}}}(\mathbf{a}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}_t}(\mathbf{t}|\mathbf{a})\right]\right] + \mathbb{E}_{p(\mathbf{a},\mathbf{t})}\left[\log \frac{p(\mathbf{t}|\mathbf{a})}{p_{\boldsymbol{\theta}_t}(\mathbf{t}|\mathbf{a})}\right],\end{aligned} \tag{D.4}$$

where in the second term the expectation defined in (D.3) is used.

$$\begin{aligned}\mathbb{E}_{p(\mathbf{a},\mathbf{t})}\left[\log \frac{p(\mathbf{t}|\mathbf{a})}{p_{\boldsymbol{\theta}_t}(\mathbf{t}|\mathbf{a})}\right] &= \mathbb{E}_{p(\mathbf{a})}\left[\mathbb{E}_{p(\mathbf{t}|\mathbf{a})}\left[\log \frac{p(\mathbf{t}|\mathbf{a})}{p_{\boldsymbol{\theta}_t}(\mathbf{t}|\mathbf{a})}\right]\right] \\ &= \mathbb{E}_{p(\mathbf{a})}\left[D_{\mathrm{KL}}\left(p(\mathbf{t}|\mathbf{A}=\mathbf{a})\|p_{\boldsymbol{\theta}_t}(\mathbf{t}|\mathbf{A}=\mathbf{a})\right)\right] \\ &= D_{\mathrm{KL}}\left(p(\mathbf{t}|\mathbf{a})\|p_{\boldsymbol{\theta}_t}(\mathbf{t}|\mathbf{a})\right).\end{aligned} \tag{D.5}$$

Since the KL-divergence $D_{\mathrm{KL}}\left(p(\mathbf{t}|\mathbf{a})\|p_{\boldsymbol{\theta}_t}(\mathbf{t}|\mathbf{a})\right) \geq 0$, the (D.4) can be lower bounded as:

$$\mathbb{E}_{p(\mathbf{a},\mathbf{t})}\left[\log p(\mathbf{t}|\mathbf{a})\right] \geq \mathbb{E}_{p(\mathbf{t},\mathbf{x})}\left[\mathbb{E}_{q_{\phi_{\mathrm{a}}}(\mathbf{a}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}_t}(\mathbf{t}|\mathbf{a})\right]\right]. \tag{D.6}$$

Therefore, the mutual information (D.1) can be lower bounded as $I(\mathbf{A};\mathbf{T}) \geq I_{\boldsymbol{\theta}_t,\boldsymbol{\phi}_\mathrm{a}}(\mathbf{A};\mathbf{T})$, where the lower bound is defined as:

$$\begin{aligned} I_{\boldsymbol{\theta}_t,\boldsymbol{\phi}_\mathrm{a}}(\mathbf{A};\mathbf{T}) &\triangleq H_{\mathcal{D}}(\mathbf{T}) + \mathbb{E}_{p(\mathbf{t},\mathbf{x})}\left[\mathbb{E}_{q_{\boldsymbol{\phi}_\mathrm{a}}(\mathbf{a}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}_t}(\mathbf{t}|\mathbf{a})\right]\right] \\ &= H_{\mathcal{D}}(\mathbf{T}) - H_{\boldsymbol{\theta}_t,\boldsymbol{\phi}_\mathrm{a}}(\mathbf{T}|\mathbf{A}), \end{aligned} \tag{D.7}$$

where $H_{\boldsymbol{\theta}_t,\boldsymbol{\phi}_\mathrm{a}}(\mathbf{T}|\mathbf{A}) = -\mathbb{E}_{p(\mathbf{t},\mathbf{x})}\left[\mathbb{E}_{q_{\boldsymbol{\phi}_\mathrm{a}}(\mathbf{a}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}_t}(\mathbf{t}|\mathbf{a})\right]\right]$.

D.2.2.2 Technical details of training

The one class classification based IB principle discussed in Section 4.5.2 includes four terms: $\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}}$, $\mathcal{D}_{\mathrm{t}}$, $\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$ and $\mathcal{D}_{\mathrm{x}}$. The estimation and reconstruction models are trained with respect to the $\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}}$ and $\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$ term correspondingly and are based on the UNet architecture given in Table D.9. The KL-divergence terms $\mathcal{D}_{\mathrm{t}}$ and $\mathcal{D}_{\mathrm{x}}$ are implemented in a form of density ration estimator. The architecture of the corresponding models is given in TableD.10.

The investigation of the one class classification of the PGC in the DNN processing domain is performed on the Indigo mobile dataset that is split into three subsets: training with 40% of data, validation with 10% of data and 50% of data is used for the test. To avoid the bias in the choice of training and test data, the classification model is trained five times on the randomly shifted data. Each time the model is trained with the learning rate equals to 1e-4, the batch of size 18 and the MSE loss. The Adam is used as an optimizer. Taking into account the sufficiently small amount of data available for the training the next data augmentations are used:

- the rotation on 90°, 180° and 270°;
- the gamma correction with $\gamma \in [0.5, 1.2]$ with step 0.1.

The one class classification of the PGC in the DNN processing domain is based on:

- the equation (4.18) introduced in Section 4.5.2.1;
- the equation (4.19) introduced in Section 4.5.2.3;
- the OC-SVM trained with respect to the Hamming distance between the original digital templates and corresponding DNN estimations, and the ℓ_2 distance between the printed query codes and the corresponding DNN reconstructions. The python *OneClassSVM* method from the *sklearn* package is used with the next training parameters:
  - kernel="rbf";
  - gamma=0.1;
  - nu=0.0005.

Table D.9 The architecture of the estimation and reconstruction models, where the number of input and output channels c equals to 1 in the case of the $\mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}}$ and to 3 in the case of the $\mathcal{D}_{\mathrm{x}\hat{\mathrm{x}}}$.

UNet model

Size	Layer
$256 \times 256 \times c$	Input
$256 \times 256 \times 64$	Conv2d, ReLU
$256 \times 256 \times 64$	Conv2d, ReLU
$128 \times 128 \times 64$	MaxPooling2D
$128 \times 128 \times 128$	Conv2d, ReLU
$128 \times 128 \times 128$	Conv2d, ReLU
$64 \times 64 \times 64$	MaxPooling2D
$64 \times 64 \times 128$	Conv2d, ReLU
$64 \times 64 \times 128$	Conv2d, ReLU
$32 \times 32 \times 128$	MaxPooling2D
$32 \times 32 \times 256$	Conv2d, ReLU
$32 \times 32 \times 256$	Conv2d, ReLU
$32 \times 32 \times 256$	Dropout
$16 \times 16 \times 256$	MaxPooling2D
$16 \times 16 \times 256$	Conv2d, ReLU
$16 \times 16 \times 256$	Conv2d, ReLU
$16 \times 16 \times 256$	Dropout
$32 \times 32 \times 256$	Conv2DTranspose, ReLU
$32 \times 32 \times 512$	Concatenate
$32 \times 32 \times 256$	Conv2d, ReLU
$32 \times 32 \times 256$	Conv2d, ReLU
$64 \times 64 \times 128$	Conv2DTranspose, ReLU
$64 \times 64 \times 256$	Concatenate
$64 \times 64 \times 128$	Conv2d, ReLU
$64 \times 64 \times 128$	Conv2d, ReLU
$128 \times 128 \times 128$	Conv2DTranspose, ReLU
$128 \times 128 \times 256$	Concatenate
$128 \times 128 \times 128$	Conv2d, ReLU
$128 \times 128 \times 128$	Conv2d, ReLU
$256 \times 256 \times 64$	Conv2DTranspose, ReLU
$256 \times 256 \times 128$	Concatenate
$256 \times 256 \times 64$	Conv2d, ReLU
$256 \times 256 \times 64$	Conv2d, ReLU
$256 \times 256 \times 16$	Conv2d, ReLU
$256 \times 256 \times c$	Conv2d, Sigmoid

Table D.10 The architecture of the discriminator based on the $\mathcal{D}_{\mathrm{t}}$ and $\mathcal{D}_{\mathrm{x}}$ terms, where the number of the input channels c equals to 1 in case of $\mathcal{D}_{\mathrm{t}}$ and to 3 in case of $\mathcal{D}_{\mathrm{x}}$.

Discriminator model

Size	Layer
$256 \times 256 \times c$	Input
$128 \times 128 \times 128$	Conv2d, LeakyReLU
$64 \times 64 \times 64$	Conv2d, BatchNormalization, LeakyReLU
$32 \times 32 \times 32$	Conv2d, BatchNormalization, LeakyReLU
$16 \times 16 \times 16$	Conv2d, BatchNormalization, LeakyReLU
$16 \times 16 \times 8$	Conv2d, BatchNormalization, LeakyReLU
$14 \times 14 \times 1$	Conv2d, LeakyReLU
196	Flatten
1	Dense, Sigmoid

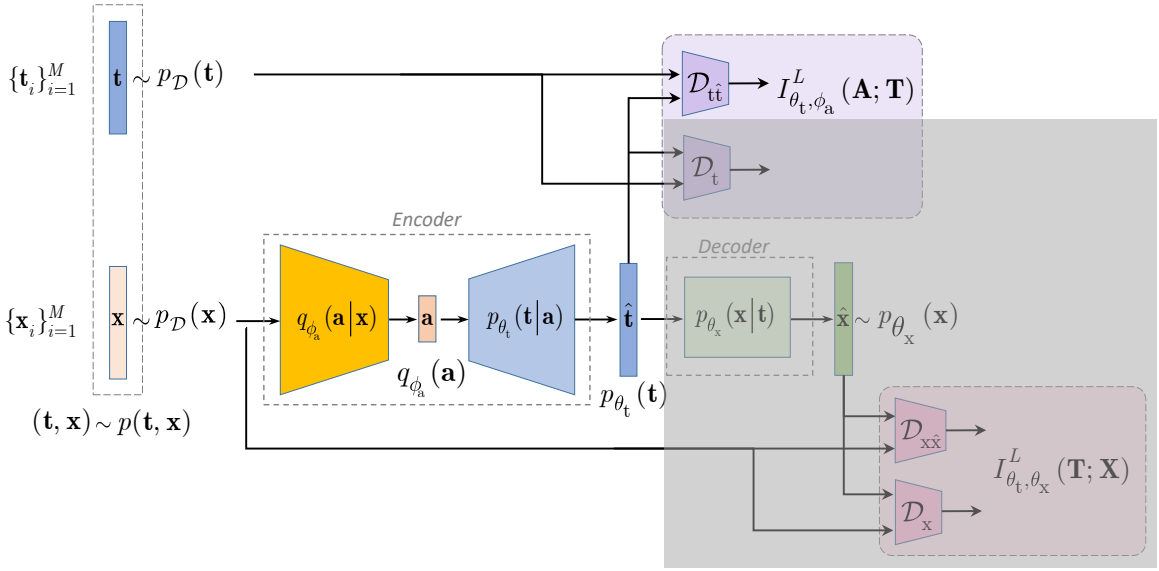


Fig. D.2 The general scheme of the system used in the first scenario and trained with respect the $\mathcal{D}_{t\hat{t}}$ term. The gray shadowed regions are not used.

D.2.2.3 The PGC authentication in the first scenario

The optimization problem (4.14) introduced in Section 4.5.2 and schematically shown in Fig. D.2 is:

$$\mathcal{L}_1(\boldsymbol{\phi}_a, \boldsymbol{\theta}_t) = -\beta_t \mathcal{D}_{t\hat{t}} \tag{D.8}$$

Taking into account the remark (4.9) in Section 4.5.2:

$$\begin{aligned} \mathcal{L}_1(\boldsymbol{\phi}_a, \boldsymbol{\theta}_t) &= -\beta_t \left(-\lambda_a \mathbb{E}_{p(\mathbf{t},\mathbf{x})} \left[\mathbb{E}_{q_{\boldsymbol{\phi}_a}(\mathbf{a}|\mathbf{x})} \left[\|\mathbf{t} - g_{\boldsymbol{\theta}_t}(\mathbf{a})\|_2 \right] \right] \right) \\ &= \alpha_a \mathbb{E}_{p(\mathbf{t},\mathbf{x})} \left[\mathbb{E}_{q_{\boldsymbol{\phi}_a}(\mathbf{a}|\mathbf{x})} \left[\|\mathbf{t} - g_{\boldsymbol{\theta}_t}(\mathbf{a})\|_2 \right] \right], \end{aligned} \tag{D.9}$$

where $g_{\boldsymbol{\theta}_t}(\mathbf{a})$ denotes the decoder.

In practical implementation the parameter α_a is set to 1. The architecture of the estimator model is given in Table D.9. The performance of the one class classification based on the (4.18) introduced in Section 4.5.2.1 is given in Table D.11. The decision constant γ_1 determined on the validation sub-set to be equal to 2 symbols. Taking into account that, due to the nature of the used trained model, the output estimation is real valued but not binary, at the inference stage to measure the Hamming distance the final estimation $\hat{\mathbf{t}}$ is obtained after the additional thresholding with a threshold 0.5.

Table D.11 The one class classification error in % on the test sub-set with respect to the first scenario optimization problem.

Run	Original P_{miss}	Fake #1 white P_{fa}	Fake #1 gray P_{fa}	Fake #2 white P_{fa}	Fake #2 gray P_{fa}
			Based on the equation (4.18)		
1	0.00	7.80	9.93	0.00	0.00
2	0.00	5.67	7.80	0.00	0.00
3	0.00	2.84	8.51	0.00	0.00
4	0.00	9.22	11.35	0.00	0.00
5	0.00	6.38	3.55	0.00	0.00
mean	0.00	6.38	8.23	0.00	0.00
std	0.00	2.40	2.95	0.00	0.00

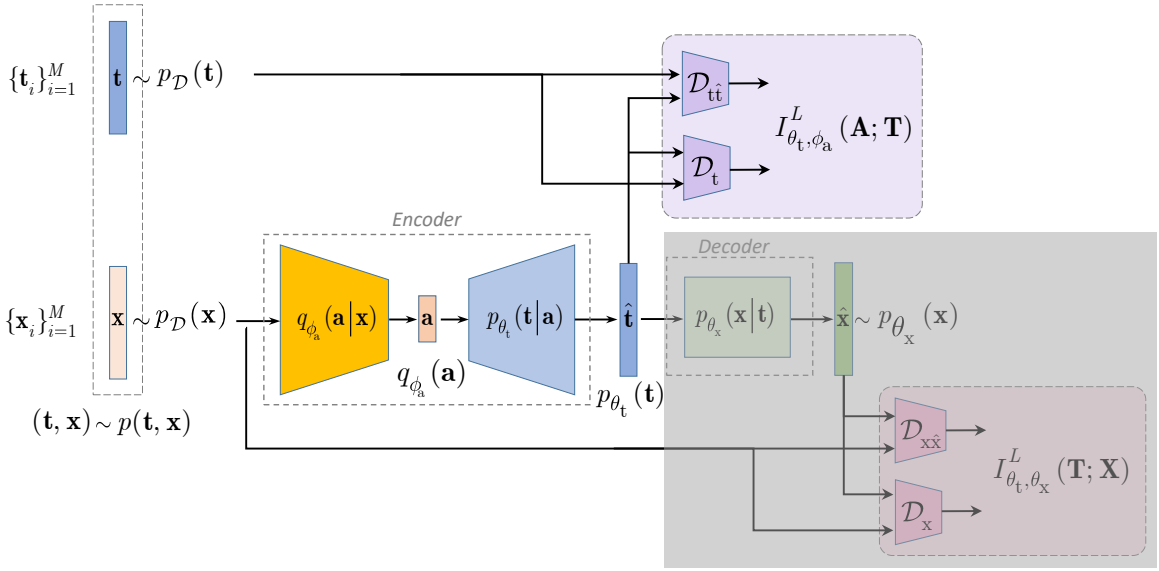


Fig. D.3 The general scheme of the system used in the second scenario and trained with respect to the $\mathcal{D}_{t\hat{t}}$ and $\mathcal{D}_t$ terms. The gray shadowed regions are not used.

D.2.2.4 The PGC authentication in the second scenario

The optimization problem (4.15) introduced in Section 4.5.2 and schematically shown in Fig. D.3 is:

$$\mathcal{L}_2(\boldsymbol{\phi}_a, \boldsymbol{\theta}_t) = -\beta_t \mathcal{D}_{t\hat{t}} + \beta_t \mathcal{D}_t \tag{D.10}$$

Taking into account the remark (4.9) in Section 4.5.2:

$$\begin{aligned}\mathcal{L}_2(\boldsymbol{\phi}_a, \boldsymbol{\theta}_t) &= -\beta_t \left(-\lambda_a \mathbb{E}_{p(\mathbf{t},\mathbf{x})} \left[\mathbb{E}_{q_{\boldsymbol{\phi}_a}(\mathbf{a}|\mathbf{x})} \left[\|\mathbf{t} - g_{\boldsymbol{\theta}_t}(\mathbf{a})\|_2\right]\right) + \beta_t D_{\mathrm{KL}}\left(p_{\mathcal{D}}(\mathbf{t}) \| p_{\boldsymbol{\theta}_t}(\mathbf{t})\right) \\ &= \alpha_a \mathbb{E}_{p(\mathbf{t},\mathbf{x})} \left[\mathbb{E}_{q_{\boldsymbol{\phi}_a}(\mathbf{a}|\mathbf{x})} \left[\|\mathbf{t} - g_{\boldsymbol{\theta}_t}(\mathbf{a})\|_2\right] + \beta_t D_{\mathrm{KL}}\left(p_{\mathcal{D}}(\mathbf{t}) \| p_{\boldsymbol{\theta}_t}(\mathbf{t})\right),\end{aligned} \tag{D.11}$$

where $g_{\boldsymbol{\theta}_t}(\mathbf{a})$ denotes the decoder.

In practical implementation the parameter α_a is set to 1, β_t equals to 0.01. The KL-divergence term is implemented in a form of density ratio estimator [132]. The architecture of the estimator model trained with respect to the $\mathcal{D}_{t\hat{t}}$ term is given in Table D.9. The architecture of the discriminator model trained with respect to the $\mathcal{D}_t$ term is given in Table D.10. At the inference stage to measure the Hamming distance the final estimation $\hat{\mathbf{t}}$ is obtained after the additional thresholding with a threshold 0.5. The performance of the one class classification based on the (4.18) introduced in Section 4.5.2.1 is given in Table D.12. The decision constant γ_1 determined on the validation sub-set to be equal to 2 symbols.

Table D.12 The one class classification error in % on the test sub-set with respect to the second scenario optimization problem.

Run	Original P_{miss}	Fake #1 white P_{fa}	Fake #1 gray P_{fa}	Fake #2 white P_{fa}	Fake #2 gray P_{fa}
			Based on the equation (4.18)		
1	0.00	5.67	7.80	0.00	0.00
2	0.00	7.80	7.80	0.00	0.00
3	0.00	9.22	8.51	0.00	0.00
4	0.00	5.67	8.51	0.00	0.00
5	0.00	5.67	2.84	0.00	0.00
mean	0.00	6.81	7.09	0.00	0.00
std	0.00	1.63	2.40	0.00	0.00

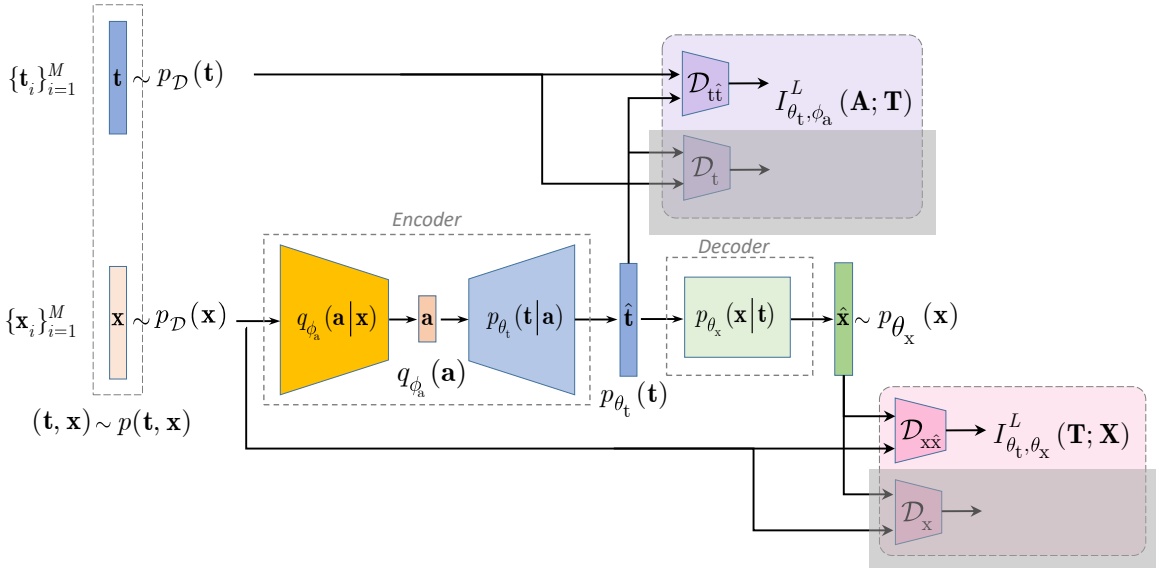


Fig. D.4 The general scheme of the system used in the third scenario and trained with respect to the $\mathcal{D}_{t\hat{t}}$ and $\mathcal{D}_{x\hat{x}}$ terms. The gray shadowed regions are not used.

D.2.2.5 The PGC authentication in the third scenario

The optimization problem (4.16) introduced in Section 4.5.2 and schematically shown in Fig. D.3 is:

$$\mathcal{L}_3(\boldsymbol{\phi}_a, \boldsymbol{\theta}_t, \boldsymbol{\theta}_x) = -\beta_t \mathcal{D}_{t\hat{t}} - \beta_x \mathcal{D}_{x\hat{x}} \tag{D.12}$$

Taking into account the remark (4.9) and the equation (4.11) in Section 4.5.2:

$$\begin{aligned}
\mathcal{L}_3(\boldsymbol{\phi}_a, \boldsymbol{\theta}_t, \boldsymbol{\theta}_x) &= -\beta_t \left(-\lambda_a \mathbb{E}_{p(\mathbf{t},\mathbf{x})} \left[\mathbb{E}_{q_{\boldsymbol{\phi}_a}(\mathbf{a}|\mathbf{x})} \left[\|\mathbf{t} - g_{\boldsymbol{\theta}_t}(\mathbf{a})\|_2\right]\right]\right) \\
&\quad - \beta_x \left(-\lambda_t \mathbb{E}_{p(\mathbf{t},\mathbf{x})} \left[\mathbb{E}_{p_{\boldsymbol{\theta}_t}(\mathbf{t}|\mathbf{x})} \left[\|\mathbf{x} - g_{\boldsymbol{\theta}_x}(\mathbf{t})\|_2\right]\right]\right) \\
&= \alpha_a \mathbb{E}_{p(\mathbf{t},\mathbf{x})} \left[\mathbb{E}_{q_{\boldsymbol{\phi}_a}(\mathbf{a}|\mathbf{x})} \left[\|\mathbf{t} - g_{\boldsymbol{\theta}_t}(\mathbf{a})\|_2\right]\right] \\
&\quad + \alpha_t \mathbb{E}_{p(\mathbf{t},\mathbf{x})} \left[\mathbb{E}_{p_{\boldsymbol{\theta}_t}(\mathbf{t}|\mathbf{x})} \left[\|\mathbf{x} - g_{\boldsymbol{\theta}_x}(\mathbf{t})\|_2\right]\right],
\end{aligned} \tag{D.13}$$

where $g_{\boldsymbol{\theta}_t}(\mathbf{a})$ denotes the decoder with respect to the latent space $\mathbf{a}$ and $g_{\boldsymbol{\theta}_x}(\mathbf{t})$ denotes the reconstruction model from the $\mathbf{t}$ to $\mathbf{x}$.

In practical implementation the parameter α_a is set to 1. To ensure the proper reconstruction quality the parameter α_t is set to 25. The architecture of the estimation and reconstruction models trained with respect to the $\mathcal{D}_{t\hat{t}}$ and $\mathcal{D}_{x\hat{x}}$ terms correspondingly is given in Table D.9. At the inference stage to measure the Hamming distance the final estimation $\hat{\mathbf{t}}$ is obtained after the additional thresholding with a threshold 0.5. The performance of the one class classification is given in Table D.12. The decision constant γ_1 in (4.18) and (4.19) determined on the validation sub-set to be equal to 2 symbols. The γ_2 in (4.19) determined

Table D.13 The one class classification error in % on the test sub-set with respect to the third scenario optimization problem.

Run	Original P_{miss}	Fake #1 white P_{fa}	Fake #1 gray P_{fa}	Fake #2 white P_{fa}	Fake #2 gray P_{fa}
			Based on the equation (4.18)		
1	0.00	1.42	0.71	0.00	0.00
2	0.00	2.13	1.42	0.00	0.00
3	0.00	1.42	0.71	0.00	0.00
4	0.00	1.42	2.13	0.00	0.00
5	0.00	1.42	0.00	0.00	0.00
mean	0.00	1.56	0.99	0.00	0.00
std	0.00	0.32	0.81	0.00	0.00
			Based on the equation (4.19)		
1	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00	0.00
3	0.00	1.42	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00
average	0.00	0.28	0.00	0.00	0.00
std	0.00	0.64	0.00	0.00	0.00
			Based on the OC-SVM		
1	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00	0.00
3	0.71	0.00	0.00	0.00	0.00
4	0.71	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00
mean	0.28	0.00	0.00	0.00	0.00
std	0.39	0.00	0.00	0.00	0.00

to be equal to 0.0017. The detailed information about the OC-SVM training is given in Appendix D.2.2.2.

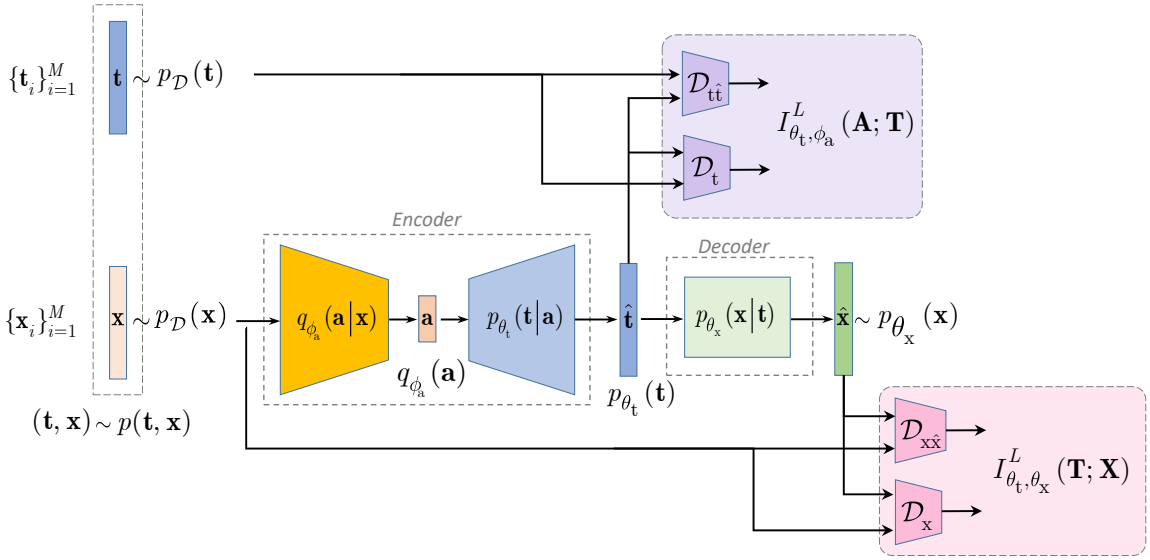


Fig. D.5 The general scheme of the system used in the fourth scenario and trained with respect to the $\mathcal{D}_{t\hat{t}}$, $\mathcal{D}_t$, $\mathcal{D}_{x\hat{x}}$ and $\mathcal{D}_x$ terms. The gray shadowed regions are not used.

D.2.2.6 The PGC authentication in the fourth scenario

The optimization problem (4.17) introduced in Section 4.5.2 and schematically shown in Fig. D.5 is:

$$\mathcal{L}_4(\boldsymbol{\phi}_a, \boldsymbol{\theta}_t, \boldsymbol{\theta}_x) = -\beta_t \mathcal{D}_{t\hat{t}} + \beta_t \mathcal{D}_t - \beta_x \mathcal{D}_{x\hat{x}} + \beta_x \mathcal{D}_x \tag{D.14}$$

Taking into account the remark (4.9) and the equation (4.11) in Section 4.5.2:

$$\begin{aligned}
\mathcal{L}_4(\boldsymbol{\phi}_a, \boldsymbol{\theta}_t, \boldsymbol{\theta}_x) &= -\beta_t \left(-\lambda_a \mathbb{E}_{p(\mathbf{t},\mathbf{x})} \left[\mathbb{E}_{q_{\boldsymbol{\phi}_a}(\mathbf{a}|\mathbf{x})} \left[\|\mathbf{t} - g_{\boldsymbol{\theta}_t}(\mathbf{a})\|_2\right]\right) + \beta_t D_{\mathrm{KL}}\left(p_{\mathcal{D}}(\mathbf{t}) \| p_{\boldsymbol{\theta}_t}(\mathbf{t})\right) \right. \\
&\quad \left. - \beta_x \left(-\lambda_t \mathbb{E}_{p(\mathbf{t},\mathbf{x})} \left[\mathbb{E}_{p_{\boldsymbol{\theta}_t}(\mathbf{t}|\mathbf{x})} \left[\|\mathbf{x} - g_{\boldsymbol{\theta}_x}(\mathbf{t})\|_2\right]\right) + \beta_x D_{\mathrm{KL}}\left(p_{\mathcal{D}}(\mathbf{x}) \| p_{\boldsymbol{\theta}_x}(\mathbf{x})\right) \right. \\
&= \alpha_a \mathbb{E}_{p(\mathbf{t},\mathbf{x})} \left[\mathbb{E}_{q_{\boldsymbol{\phi}_a}(\mathbf{a}|\mathbf{x})} \left[\|\mathbf{t} - g_{\boldsymbol{\theta}_t}(\mathbf{a})\|_2\right]\right] + \beta_t D_{\mathrm{KL}}\left(p_{\mathcal{D}}(\mathbf{t}) \| p_{\boldsymbol{\theta}_t}(\mathbf{t})\right) \\
&+ \alpha_t \mathbb{E}_{p(\mathbf{t},\mathbf{x})} \left[\mathbb{E}_{p_{\boldsymbol{\theta}_t}(\mathbf{t}|\mathbf{x})} \left[\|\mathbf{x} - g_{\boldsymbol{\theta}_x}(\mathbf{t})\|_2\right]\right] + \beta_x D_{\mathrm{KL}}\left(p_{\mathcal{D}}(\mathbf{x}) \| p_{\boldsymbol{\theta}_x}(\mathbf{x})\right),
\end{aligned} \tag{D.15}$$

where $g_{\boldsymbol{\theta}_t}(\mathbf{a})$ denotes the decoder with respect to the latent space $\mathbf{a}$ and $g_{\boldsymbol{\theta}_x}(\mathbf{t})$ denotes the reconstruction model from the $\mathbf{t}$ to $\mathbf{x}$.

In practical implementation the parameter α_a is set to 1. To ensure the proper reconstruction quality the parameter α_t is set to 25. The parameters β_t and β_x equal to 0.01 The architecture of the estimation and reconstruction models trained with respect the $\mathcal{D}_{t\hat{t}}$ and $\mathcal{D}_{x\hat{x}}$ terms is given in Table D.9. The KL-divergence terms are implemented in a form of density ratio estimator [132]. The architecture of the discriminator models trained with respect to the $\mathcal{D}_t$ and $\mathcal{D}_x$ terms is given in Table D.10.

Table D.14 The one class classification error in % on the test sub-set with respect to the fourth scenario optimization problem.

Run	Original P_{miss}	Fake #1 white P_{fa}	Fake #1 gray P_{fa}	Fake #2 white P_{fa}	Fake #2 gray P_{fa}
			Based on the equation (4.18)		
1	0.00	1.42	1.42	0.00	0.00
2	0.00	4.26	2.84	0.00	0.00
3	0.00	1.42	4.26	0.00	0.00
4	0.00	3.55	2.13	0.00	0.00
5	0.00	1.42	0.00	0.00	0.00
mean	0.00	2.41	2.13	0.00	0.00
std	0.00	1.38	1.59	0.00	0.00
			Based on the equation (4.19)		
1	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.71	0.00	0.00
4	2.84	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00
mean	0.57	0.00	0.14	0.00	0.00
std	1.27	0.00	0.32	0.00	0.00
			Based on the OC-SVM		
1	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00
4	0.71	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00
mean	0.14	0.00	0.00	0.00	0.00
std	0.32	0.00	0.00	0.00	0.00

Similarly to the previous scenarios, at the inference stage to measure the Hamming distance the final estimation $\hat{\mathbf{t}}$ is obtained after the additional thresholding with a threshold 0.5. The performance of the one class classification is given in Table D.12. The decision constant γ_1 in (4.18) and (4.19) determined on the validation sub-set to be equal to 2 symbols. The γ_2 in (4.19) determined to be equal to 0.008. The detailed information about the OC-SVM training is given in Appendix D.2.2.2.

D.3 ML fakes authentication

D.3.1 ML fakes production: technical details of training

The ML fake production is performed with respect to the (4.25) introduced in Section 4.6.1 as:

$$\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}) = \underset{\boldsymbol{\phi},\boldsymbol{\theta}}{\operatorname{argmin}} -\beta \mathcal{D}_{\mathrm{t}\hat{\mathrm{t}}}. \tag{D.16}$$

Taking into account the remark (4.9) in Section 4.5.2 and (4.23) in Section 4.6.1

$$\begin{aligned} \mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}) &= -\beta \left(-\lambda \mathbb{E}_{p(\mathbf{t},\mathbf{x})} \left[\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{a}|\mathbf{x})} \left[\|\mathbf{t} - g_{\boldsymbol{\theta}}(\mathbf{a})\|_2 \right] \right] \right) \\ &= \alpha \mathbb{E}_{p(\mathbf{t},\mathbf{x})} \left[\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{a}|\mathbf{x})} \left[\|\mathbf{t} - g_{\boldsymbol{\theta}}(\mathbf{a})\|_2 \right] \right] \end{aligned} \tag{D.17}$$

where $g_{\boldsymbol{\theta}}(\mathbf{a})$ denotes the decoder.

In practical implementation the parameter α is set to 1. The performance of the estimator model is investigated under two architectures: *(i) LinearBN* introduced in Table D.15 and *(ii) ConvBN* given in Table D.16. The practical validation is performed on the DP1C and DP1E datasets that are split into three sub-sets: *(i)* training with 100 codes, *(ii)* validation with 50 codes and *(iii)* test sub-set with 234 codes. Taking into account the sufficiently small amount of data available for the training, the codes in the training sub-sets are split into non-overlapping blocks of size 24×24. To avoid the bias in the choice of training and test data, the estimation models are trained three times on the randomly shifted data. Each time the models are trained with the learning rate equal to 1e-3, the batch of size 64 and the MSE loss. The Adam is used as an optimizer. At the inference stage, to guarantee the binarity of the estimated codes, the outputs of the estimation models are binarized via simple thresholding method with an optimal threshold estimated on the validation sub-sets.

The estimation error in % on the test sub-sets with respect to the symbol wise Hamming distance between the original digital templates and the corresponding estimations for both datasets is given in Table D.17.

Table D.15 The architecture of the *LinearBN* estimation model.

LinearBN	
Size	Layer
576 ($24 \times 24 \times 1$)	Input
256	Linear, BatchNorm1d, Tanh
128	Linear, BatchNorm1d, ReLU
36	Linear
128	Linear, BatchNorm1d, Tanh
256	Linear, ReLU
576	Linear, ReLU

Table D.16 The architecture of the *ConvBN* estimation model.

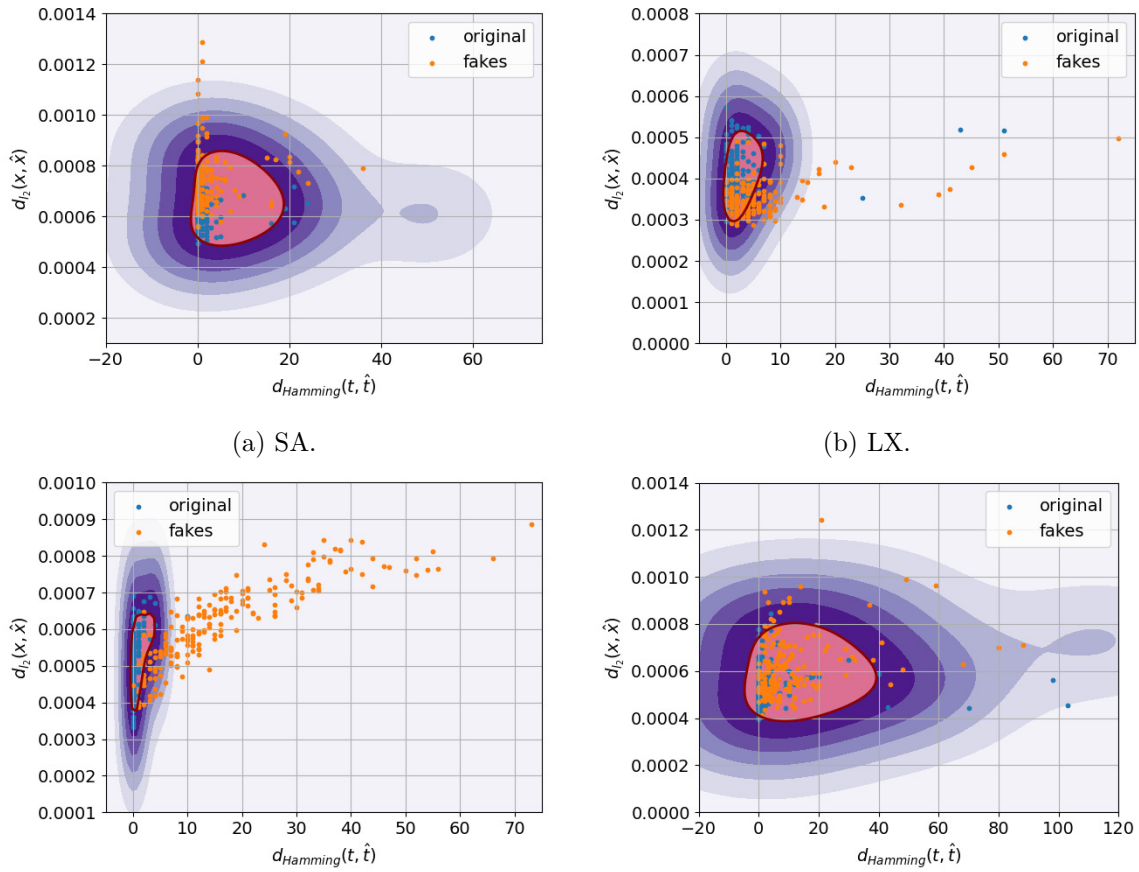
ConvBN	
Size	Layer
$24 \times 24 \times 1$	Input
$22 \times 22 \times 64$	Conv2d, BatchNorm2d, Tanh
$10 \times 10 \times 32$	Conv2d, BatchNorm2d, Tanh
$8 \times 8 \times 16$	Conv2d, BatchNorm2d, ReLU
$4 \times 4 \times 8$	Conv2d, BatchNorm2d, Tanh
$6 \times 6 \times 16$	ConvTranspose2d, BatchNorm2d, ReLU
$12 \times 12 \times 32$	ConvTranspose2d, BatchNorm2d, Tanh
$24 \times 24 \times 1$	ConvTranspose2d, BatchNorm2d, ReLU

Table D.17 The estimation error in % on the test sub-sets with respect to the symbol wise Hamming distance between the original digital templates and the corresponding estimations.

		DP1E				**DP1E**			
Train on	Run	SA	LX	CA	HP	SA	LX	CA	HP
					Hand-crafted				
Full area		2.88	5.21	1.48	1.93	2.88	5.21	1.48	1.93
Half area		0.85	1.76	0.66	0.88	0.85	1.76	0.66	0.88
					LinearBN				
	1	0.14	0.97	0.12	0.32	0.14	1.47	0.20	0.56
	2	0.14	1.15	0.13	0.43	0.16	1.45	0.20	0.67
SA	3	0.16	1.26	0.14	0.39	0.18	1.53	0.22	0.59
	mean	0.15	1.13	0.13	0.38	0.16	1.48	0.21	0.61
	std	0.01	0.15	0.01	0.06	0.02	0.04	0.01	0.06
	1	0.30	0.36	0.19	0.37	0.35	0.74	0.33	0.70
	2	0.30	0.34	0.21	0.37	0.41	0.65	0.41	0.73
LX	3	0.24	0.31	0.21	0.39	0.45	0.74	0.36	0.70
	mean	0.28	0.34	0.20	0.38	0.40	0.71	0.36	0.71
	std	0.03	0.03	0.01	0.01	0.05	0.05	0.04	0.02
	1	0.25	1.09	0.12	0.34	0.17	1.21	0.13	0.50
	2	0.22	0.96	0.13	0.35	0.17	1.42	0.13	0.44
CA	3	0.20	1.49	0.13	0.39	0.16	1.32	0.13	0.48
	mean	0.22	1.18	0.13	0.36	0.16	1.32	0.13	0.47
	std	0.03	0.27	0.01	0.03	0.01	0.10	0.00	0.03
	1	0.35	1.67	0.17	0.40	0.16	1.19	0.15	0.42
	2	0.30	1.23	0.12	0.27	0.27	1.27	0.15	0.64
HP	3	0.25	1.04	0.13	0.27	0.24	1.21	0.15	0.61
	mean	0.30	1.31	0.14	0.31	0.22	1.23	0.15	0.56
	std	0.05	0.32	0.03	0.08	0.06	0.04	0.00	0.12
					ConvBN				
	1	0.10	0.60	0.17	0.34	0.12	1.16	0.24	0.62
	2	0.11	1.02	0.26	0.54	0.10	1.67	0.40	1.02
SA	3	0.14	0.70	0.22	0.40	0.24	1.23	0.25	0.73
	mean	0.12	0.77	0.21	0.43	0.15	1.35	0.30	0.79
	std	0.02	0.22	0.04	0.10	0.08	0.28	0.09	0.21
	1	0.52	0.13	0.44	0.84	0.53	0.23	0.78	1.25
	2	0.85	0.27	0.83	1.27	0.46	0.46	0.65	1.35
LX	3	0.54	0.28	0.53	0.81	0.30	0.57	0.29	0.87
	mean	0.64	0.23	0.60	0.97	0.43	0.42	0.58	1.16
	std	0.18	0.08	0.20	0.26	0.12	0.17	0.25	0.25
	1	0.24	1.06	0.09	0.31	0.15	0.79	0.09	0.41
	2	0.32	1.52	0.13	0.35	0.25	1.02	0.17	0.49
CA	3	0.30	0.85	0.14	0.43	0.23	0.99	0.14	0.59
	mean	0.29	1.14	0.12	0.36	0.21	0.93	0.13	0.50
	std	0.04	0.34	0.02	0.06	0.05	0.12	0.04	0.09
	1	0.19	1.64	0.10	0.18	0.11	0.81	0.09	0.17
	2	0.27	0.61	0.17	0.20	0.10	0.73	0.07	0.17
HP	3	0.25	1.07	0.23	0.26	0.12	0.71	0.07	0.20
	mean	0.23	1.11	0.17	0.22	0.11	0.75	0.08	0.18
	std	0.04	0.52	0.07	0.04	0.01	0.05	0.01	0.02

D.3.2 One class classification with respect to the ML fakes

The one class classification of the PGC based on the OC-SVM with respect to the ML fakes produced by the *ConvBN* is performed in a way similar to the one class classification of the PGC with respect to the HC fakes discussed in Section 4.5.2.4 and Appendix D.2.2. The obtained performance of the one class classification is given in Table D.18. The examples of the OC-SVM decision boundaries for the different printers in the DP1E dataset are given in Fig. D.6. For the DP1C dataset the examples of the OC-SVM decision boundaries are given in Fig. 4.31 in Section 4.6.3.



(a) SA.

(b) LX.

(c) CA.

(d) HP.

Fig. D.6 The examples of the OC-SVM decision boundaries for the different printers in the DP1E dataset.

Table D.18 The OC-SVM classification error in % on the test sub-sets of DP1C and DP1E datasets with respect to the ML attacks based on the *ConvBN* model.

		DP1C		**DP1E**	
	Run	P_{miss}	P_{fa}	P_{miss}	P_{fa}
SA	1	4.20	93.80	7.30	81.20
	2	9.90	88.90	9.90	86.50
	3	11.60	84.30	7.30	72.90
	4	5.20	90.40	8.90	79.70
	5	6.80	79.70	8.30	70.80
	average	7.54	87.42	8.34	78.22
	std	3.13	5.50	1.11	6.38
LX	1	8.90	72.50	19.80	51.90
	2	30.70	39.30	9.90	60.90
	3	39.60	33.20	6.20	75.50
	4	6.20	86.80	17.70	49.50
	5	20.30	57.30	18.80	64.10
	average	21.14	57.82	14.48	60.38
	std	14.19	22.39	6.06	10.40
CA	1	4.70	43.80	12.00	4.70
	2	11.50	31.80	31.80	7.30
	3	17.20	26.70	6.20	6.20
	4	5.20	47.40	7.80	18.80
	5	16.10	28.10	9.40	17.70
	average	10.94	35.56	13.44	10.94
	std	5.87	9.44	10.48	6.75
HP	1	7.80	73.90	6.20	86.80
	2	10.60	59.80	6.20	79.20
	3	4.70	89.50	8.30	75.90
	4	5.30	88.60	6.20	79.70
	5	7.30	87.20	7.80	83.30
	average	7.14	79.80	6.94	80.98
	std	2.33	12.86	1.03	4.18

D.3.3 Supervised classification with respect to the ML fakes

The supervised classification with respect to the ML fakes produced by the *ConvBN* model is performed in the same way as described in Section 4.4 with only one difference that for each printer there exists only one type of fakes. The obtained classification error is given in Table D.19.

Table D.19 The supervised classification error in % on the DP1C and DP1E test sub-sets with respect to the ML attacks based on the *ConvBN* model.

		DP1C		**DP1E**	
	Run	P_{miss}	P_{fa}	P_{miss}	P_{fa}
SA	1	23.44	14.58	13.54	35.41
	2	47.92	32.29	22.92	35.42
	3	35.94	41.67	25.52	29.17
	4	41.14	37.50	24.47	35.42
	5	22.40	18.75	34.37	23.95
	average	34.17	28.96	24.16	31.87
	std	11.12	11.79	7.42	5.19
LX	1	31.25	42.19	11.46	16.67
	2	35.42	11.98	4.69	6.77
	3	23.44	32.81	9.89	2.08
	4	30.73	39.06	11.45	17.70
	5	31.25	36.98	16.15	7.81
	average	30.42	32.60	10.73	10.21
	std	4.34	12.02	4.11	6.73
CA	1	0.00	14.58	1.04	4.68
	2	2.08	10.42	0.00	5.21
	3	5.73	1.56	0.00	1.56
	4	1.04	10.93	0.00	10.41
	5	2.60	9.37	0.00	5.73
	average	2.29	9.37	0.21	5.52
	std	1.33	4.20	1.62	3.38
HP	1	11.98	18.23	8.33	21.35
	2	3.13	22.92	3.13	19.79
	3	12.50	7.29	13.02	4.17
	4	7.29	25.52	11.97	21.35
	5	7.81	29.17	4.17	9.89
	average	8.54	20.62	8.12	15.31
	std	3.84	8.45	4.46	7.85

D.4 Digital templates estimation

To be coherent with the results obtained in Appendix D.3.1 and discussed in Section 4.6.2 the digital templates estimation from the printed counterparts is based on the (D.17) in Appendix D.3.1.

In practical implementation the parameter α in (D.17) is set to 1. The architecture of the estimation model used for the estimation of the codes with the symbol size 5×5 and 4×4 is given in Table D.20, and in Table D.21 for the codes with the symbols of size 3×3. The experiments are performed on the Indigo scanner dataset that is split into three sub-sets: the training with 40% of data, validation with 10% of data and 50% of data is used for the test. To avoid the bias in the choice of training and test data, the estimation models are trained three times on the randomly shifted data. Each time the models are trained with the learning rate equals to 1e-4, the batch of size 18 and the MSE loss. The Adam is used as an optimizer. Taking into account the sufficiently small amount of data available for the training, each code is split into 4 sub-blocks of the size suitable as input to the corresponding training models, namely, the codes with the symbol size 5×5 and 4×4 are split into blocks of size 256×256 and the codes with the symbol size 3×3 are split into blocks of size 128×128. To each sub-block additionally there are applied the next data augmentation:

- the rotation on 90°, 180° and 270°;
- the gamma correction with $\gamma \in [0.5, 1]$ with step 0.35.

The obtained results are given in Tables D.22 - D.24

Table D.20 The architecture of the model used for the digital templates estimation from the printed counterparts with the symbol size 5×5 and 4×4.

UNet model

Size	Layer
$256 \times 256 \times 1$	Input
$256 \times 256 \times 64$	Conv2d, ReLU
$256 \times 256 \times 64$	Conv2d, ReLU
$128 \times 128 \times 64$	MaxPooling2D
$128 \times 128 \times 128$	Conv2d, ReLU
$128 \times 128 \times 128$	Conv2d, ReLU
$64 \times 64 \times 64$	MaxPooling2D
$64 \times 64 \times 128$	Conv2d, ReLU
$64 \times 64 \times 128$	Conv2d, ReLU
$32 \times 32 \times 128$	MaxPooling2D
$32 \times 32 \times 256$	Conv2d, ReLU
$32 \times 32 \times 256$	Conv2d, ReLU
$32 \times 32 \times 256$	Dropout
$16 \times 16 \times 256$	MaxPooling2D
$16 \times 16 \times 256$	Conv2d, ReLU
$16 \times 16 \times 256$	Conv2d, ReLU
$16 \times 16 \times 256$	Dropout
$32 \times 32 \times 256$	Conv2DTranspose, ReLU
$32 \times 32 \times 512$	Concatenate
$32 \times 32 \times 256$	Conv2d, ReLU
$32 \times 32 \times 256$	Conv2d, ReLU
$64 \times 64 \times 128$	Conv2DTranspose, ReLU
$64 \times 64 \times 256$	Concatenate
$64 \times 64 \times 128$	Conv2d, ReLU
$64 \times 64 \times 128$	Conv2d, ReLU
$128 \times 128 \times 128$	Conv2DTranspose, ReLU
$128 \times 128 \times 256$	Concatenate
$128 \times 128 \times 128$	Conv2d, ReLU
$128 \times 128 \times 128$	Conv2d, ReLU
$256 \times 256 \times 64$	Conv2DTranspose, ReLU
$256 \times 256 \times 128$	Concatenate
$256 \times 256 \times 64$	Conv2d, ReLU
$256 \times 256 \times 64$	Conv2d, ReLU
$256 \times 256 \times 16$	Conv2d, ReLU
$256 \times 256 \times 1$	Conv2d, Sigmoid

Table D.21 The architecture of the model used for the digital templates estimation from the printed counterparts with the symbol size 3×3.

UNet model	
Size	Layer
$128 \times 128 \times 1$	Input
$128 \times 128 \times 64$	Conv2d, ReLU
$128 \times 128 \times 64$	Conv2d, ReLU
$64 \times 64 \times 64$	MaxPooling2D
$64 \times 64 \times 128$	Conv2d, ReLU
$64 \times 64 \times 128$	Conv2d, ReLU
$32 \times 32 \times 128$	MaxPooling2D
$32 \times 32 \times 128$	Conv2d, ReLU
$32 \times 32 \times 128$	Conv2d, ReLU
$16 \times 16 \times 128$	MaxPooling2D
$16 \times 16 \times 256$	Conv2d, ReLU
$16 \times 16 \times 256$	Conv2d, ReLU
$16 \times 16 \times 256$	Dropout
$8 \times 8 \times 256$	MaxPooling2D
$8 \times 8 \times 256$	Conv2d, ReLU
$8 \times 8 \times 256$	Conv2d, ReLU
$8 \times 8 \times 256$	Dropout
$16 \times 16 \times 256$	Conv2DTranspose, ReLU
$16 \times 16 \times 512$	Concatenate
$16 \times 16 \times 256$	Conv2d, ReLU
$16 \times 16 \times 256$	Conv2d, ReLU
$32 \times 32 \times 128$	Conv2DTranspose, ReLU
$32 \times 32 \times 256$	Concatenate
$32 \times 32 \times 128$	Conv2d, ReLU
$32 \times 32 \times 128$	Conv2d, ReLU
$64 \times 64 \times 128$	Conv2DTranspose, ReLU
$64 \times 64 \times 256$	Concatenate
$64 \times 128 \times 128$	Conv2d, ReLU
$64 \times 128 \times 128$	Conv2d, ReLU
$128 \times 128 \times 64$	Conv2DTranspose, ReLU
$128 \times 128 \times 128$	Concatenate
$128 \times 128 \times 64$	Conv2d, ReLU
$128 \times 128 \times 64$	Conv2d, ReLU
$128 \times 128 \times 16$	Conv2d, ReLU
$128 \times 128 \times 1$	Conv2d, Sigmoid

Table D.22 The estimation error in % based on the symbol wise Hamming distance between the original digital templates with the symbol's size 5×5 and the corresponding estimations.

Scanned codes symbol size	Run	Average % of error symbols	% of codes with error symbols = 0	% of codes with error symbols ≤ 2	% of codes with error symbols > 2
ML trained on the HC half area estimations					
7×7	1	0.3910	40.00	25.33	34.67
	2	0.6512	6.76	17.57	75.68
	3	0.6727	8.11	15.54	76.35
	mean	0.5716	18.29	19.48	62.23
	std	0.1568	18.82	5.17	23.87
5×5	1	0.0135	71.62	22.30	6.08
	2	0.0206	68.24	22.97	8.78
	3	0.0073	77.03	21.62	1.35
	mean	0.0138	72.30	22.30	5.41
	std	0.0067	4.43	0.68	3.76
ML trained on the original digital template					
7×7	1	0.0261	72.30	20.27	7.43
	2	0.0313	77.03	16.22	6.76
	3	0.0185	75.00	16.22	8.78
	mean	0.0253	74.77	17.57	7.66
	std	0.0065	2.37	2.34	1.03
5×5	1	0.0016	93.92	6.08	0.00
	2	0.0016	93.24	6.76	0.00
	3	0.0012	95.27	4.73	0.00
	mean	0.0015	94.14	5.86	0.00
	std	0.0002	1.03	1.03	0.00

Table D.23 The estimation error in % based on the symbol wise Hamming distance between the original digital templates with the symbol's size 4×4 and the corresponding estimations.

Scanned codes symbol size	Run	Average % of error symbols	% of codes with error symbols = 0	% of codes with error symbols ≤ 2	% of codes with error symbols > 2
ML trained on the HC half area estimations					
5×5	1	2.3454	0.00	0.00	100
	2	2.3811	0.00	0.00	100
	3	2.2964	0.00	0.67	99.33
	mean	2.3410	0.00	0.22	99.78
	std	0.0425	0.00	0.38	0.38
4×4	1	3.0582	0.00	0.00	100
	2	3.2698	0.00	0.00	100
	3	3.0638	0.00	0.00	100
	mean	3.1306	0.00	0.00	100
	std	0.1206	0.00	0.00	0.00
ML trained on the original digital template					
5×5	1	0.0015	96.00	3.33	0.67
	2	0.0014	96.00	3.33	0.67
	3	0.0020	93.33	6.00	0.67
	mean	0.0016	95.11	4.22	0.67
	std	0.0003	1.54	1.54	0.00
4×4	1	0.0014	96.00	3.33	0.67
	2	0.0018	94.00	5.33	0.67
	3	0.0017	94.67	4.67	0.67
	mean	0.0016	94.89	4.44	0.67
	std	0.0002	1.02	1.02	0.00

Table D.24 The estimation error in % based on the symbol wise Hamming distance between the original digital templates with the symbol's size 3×3 and the corresponding estimations.

Scanned codes symbol size	Run	Average % of error symbols	% of codes with error symbols = 0	% of codes with error symbols ≤ 2	% of codes with error symbols > 2
ML trained on the HC half area estimations					
4×4	1	10.1771	0.00	0.00	100
	2	10.0940	0.00	0.00	100
	3	10.4815	0.00	0.00	100
	mean	10.2509	0.00	0.00	100
	std	0.2040	0.00	0.00	0.00
3×3	1	8.5802	0.00	0.00	100
	2	8.2882	0.00	0.00	100
	3	8.7097	0.00	0.00	100
	mean	8.5260	0.00	0.00	100
	std	0.2159	0.00	0.00	0.00
ML trained on the original digital template					
4×4	1	0.0069	76.67	21.33	2.00
	2	0.0115	72.00	24.67	3.33
	3	0.0341	73.33	23.33	3.33
	mean	0.0175	74.00	23.11	2.89
	std	0.0146	2.40	1.68	0.77
3×3	1	0.0093	70.00	28.00	2.00
	2	0.0118	69.33	27.33	3.33
	3	0.0109	68.67	26.67	4.67
	mean	0.0107	69.33	27.33	3.33
	std	0.0013	0.67	0.67	1.33