University of Geneva
Department of Computer Science

Master's Thesis

Fine-grained identification of pharmaceutical products based on local descriptors

Author:
Oscar Albert Taïsen Dabrowski

Supervisors:
Prof. Sviatoslav Voloshynovskiy
Dr. Taras Holotyak

October 12, 2017

Acknowledgements

I would like to express my deepest thanks to Prof. Sviatoslav Voloshynovskiy for giving me the opportunity to work on this project, for which I have great interest. I would also like to thank him for opening my mind to the passionating field of computer vision and image processing, by his teachings. Moreover, I would like to thank Dr. Taras Holotyak for his help, his advices and his sense of humor. I would also like to thank both of them for proof-reading my thesis.

Thanks to Olga Taran, Shideh Rezaeifar, Jonathan Schlechten, Nabil Stendardo and Maurits Diephuis for their help, advices and for the pleasant working atmosphere they have always provided. Thanks also to my friends Alexis Beck, Sabri Fernana, Jeremy Morel and Kamil Dobosz for the pleasant discussions that we had.

Finally I would also like to thank my parents, and especially my mother for her support, her advices and her everlasting presence by my side.

Contents

List of Symbols

X	Scalar random variable
$\underline{X}, \vec{X}$	Vector or matrix random variable
$\underline{x}, \vec{x}$	Vector, realization of a random variable $\underline{X}, \vec{X}$
x	Scalar, realization of a random variable X
$\mathcal{X}$	Alphabet of a random variable X
$\underline{\hat{x}}$	Vector that is being estimated or measured
$\hat{x}$	Scalar that is being estimated or measured
$\underline{x}^T$	Transpose operation on a vector
$\triangleq$	Equal by definition

Chapter 1

Introduction

The saying ”the devil is in the details”, sometimes attributed to Friedrich Nietzsche, describes well the problem of fine-grained recognition. That is to say, distinguishing between very similar objects having only minor differences. Although, the general problem of physical object recognition is a well known one, and many algorithms in the field of computer vision exist, it still remains challenging to detect minor details on very similar objects. Distinguishing between very distinct classes (such as cats versus dogs, cats versus cars, etc.) is quite well handled by state-of-the-art techniques. However, when it comes to differentiating two medical packages of identical design, produced by the same pharmaceutical company, but with just a single character difference (e.g., dosage of 100 mg versus 150 mg), then the problem becomes quite more difficult (Figure 1.1).

In this thesis, we present a new database of pharmaceutical packages and investigate the problem of their identification based on images taken by consumer mobile phones. The problem of correct identification of medical packages is the first step towards detection of counterfeit drugs. This problem is of great importance. It has been stated in [1], that counterfeit drugs worldwide represent up to 15% of all drugs sold, and this percentage is estimated to an amount as high as 50% in some Asian and African countries. Consequently, it seems obvious that this can lead to serious health problems and put many human lives at risk. We should add that many medicines are quite expensive and therefore attractive to drug counterfeiters. Moreover, with today's technologies it is relatively easy to produce reasonably credible forgeries.

Compared to other products, pharmaceutical packages are a particular class of physical objects with their own specificities. This leads us back to the fine-grained recognition problem. Indeed, from a very coarse point of view, a medical package is generally a white rectangular box with some minor variations of size and color. Therefore, it can be said that the visual context of pharmaceutical packages is quite poor. Moreover, as discussed before, it becomes even more challenging to identify subtle differences of dosage or number of pills written on packages of the same drug. Another problem can arise due to different recognition conditions. Indeed, we cannot guarantee that end users will identify their package in the same lighting conditions as those used at the enrollment. Therefore, a significant brightness variability will most likely be observed and can lead to incorrect identification. For example, the text showing the number of pills might be overexposed and not identifiable on the image. An example can be seen in Figure 1.2. This could lead to a case of false acceptance or a miss.

FIGURE 1.1: An example of two very similar packages (they differ only by the dosage of active ingredients, i.e., 100 mg vs. 150 mg).

FIGURE 1.2: An example of a problematic image on which the text showing the number of pills in the package is garbled by glares.

1.1 State-of-the-art

The task of object identification by means of computer vision algorithms is a well known one, and many different paths have been explored by the scientific community. However, in some situations, such as *fine-grained recognition*, this task remains very challenging. In this section, we will present an overview of different existing identification methods. Firstly, the distinction between so-called *invasive* and *non-invasive* protection techniques will be made. Then, approaches related to our work will be explored. Finally, different popular machine-learning algorithms will be briefly described.

1.1.1 Invasive techniques

Invasive approaches require to modify or mark each product that is to be protected beforehand. Such kind of approaches are used for the protection of banknotes for example. In the case of pharmaceutical packages, this kind of technique raises a significant problem, namely, packages released before the implementation of the protection will remain unprotected. Moreover, this kind of approach could require complex fabrication systems for the serialization of each product and also to directly work in coordination with industrial printing partners.

Watermarking

The technique described here does not refer to the usual visible watermarks such as the ones which can be found printed on special papers (e.g., logo of a company) or even visible digital watermarks inserted in photos to show the owner of the copyright and provide a non-watermarked version in exchange of payment. Watermarking techniques [2, 3] and steganography, which in its primitive form, goes back to ancient Greece, are closely related in their way to proceed. They are both data-hiding techniques, however, they fundamentally differ in the application that they address, as described in [4]. Indeed, when using steganography with images, one aims at communicating a secret message hidden in a host image, the latter being considered as a "decoy". However, in watermarking we do care a lot about the content being watermarked (i.e., the pharmaceutical package in our case). Understanding this subtle distinction among data-hiding techniques is important, as in the case of copyright or anti-counterfeiting protection, a very small watermark can be sufficient (e.g., a serial number encoded in only 64 bits). Moreover, it can be publicly known that the products are watermarked, and indeed we might even want it to be known and advertise it as an anti-copy protection feature.

Technically, applied to pharmaceutical packages, a watermarking technique can be used to hide a secret code (watermark) inside specific groups of pixels of a digital image which will then be printed on a physical package. This can be viewed as adding noise [5] to the image. The detection phase, in that case, can be thought of measuring correlation between a known watermark and the one being examined. It is important to know that state-of-the-art watermarking techniques can be made robust to many distortions such as geometric transformations, noise, compression artifacts, "copy attack"[2], etc. Indeed the so-called "analogue hole" is sealed: "an embedded watermark can be retrieved even after analogue copies" [6].

The main restriction of watermarking techniques is related to the poor graphical content of pharmaceutical packages that generally does not contain texture rich images. It is known that the entropy of text symbologies is significantly lower and it is more difficult to hide and modulate them [7, 8].

Additionally, since the targeted recognition is based on mobile phones, the existing text data hiding methods can rarely cope with that. Therefore, the recent trends rely more on non-invasive techniques.

1.1.2 Non-invasive techniques

The non-invasive techniques encompass ones which do not modify the products and consequently rely on the detection and comparison of some specific characteristics or features present on them. These so-called non-invasive techniques have a significant advantage over the invasive ones, as they do not need any preprocessing of the products to add security features. Moreover, they are applicable to products manufactured at any given past time, not only those created after the implementation of security features.

Forensics

The forensics approach, relies on the detection of features intrinsically present on products due to the manufacturing processes and on which the producer has no direct control in principle. They can be seen as random artifacts created during manufacturing. A specific arrangement of dots produced by printers can be used to decide whether some document was printed on a laser or inkjet printer. It is possible to derive what is called a "signature" for a specific device [9]. Indeed, for instance, scanners and digital cameras are equipped with a sensor, which replaces the photosensitive argentic films of analogue cameras. The digital sensors are in general either a CMOS[1] (Complementary Metal–Oxide–Semiconductor) or CCD[2] (Charge-Coupled Device) electronic device. In summary, these sensors can be thought of as an array of photosensitive elements. Images taken by these devices are subjected to "sensor noise" because the manufacturing process always produces some minor flaws. In [9] two types of noise are described, i.e., *sensor noise* caused by imperfections (e.g., dead pixels), and *pattern noise* (produced by "dark currents", i.e., sort of small uncontrollable bursts of current). This allows to derive a "forensic signature" and gives us the possibility of discriminating for example two images produced by different cameras.

Microstructures

The term "microstructure" refers to natural or artificial random patterns that can be observed on physical objects at the microscopic level [10]. These features are extremely difficult to reproduce (i.e., make a clone) and are therefore also called PUF (Physical Unclonable Feature) [10]. We would like to mention the FAMOS dataset [11], which was created at the University of Geneva and contains 5000 unique microstructures. The usage of such features to identify packages is obviously an interesting solution, but requires images taken at a close range ("acquired at micrometer scale" [12]) or magnified with an additional lens. In our project, we target a batch object recognition that does not require to enroll a photo from each physical object.

[1] https://en.wikipedia.org/wiki/Active_pixel_sensor
[2] https://en.wikipedia.org/wiki/Charge-coupled_device

Fingerprinting

The fingerprinting approach we are talking about consists of computing a small sequence of bits from an image to allow its identification [13, 14, 15]. This bit sequence is called a "robust hash" like in cryptography or a "fingerprint". Yet, we must keep in mind that unlike in cryptography, this "robust hash" is built in such a way that small modifications of the image results in similar hashes when compared to the original, but a very different image should produce a very different hash. This is also known as "perceptual hashing" [15]. Mostly, it is true for simplified additive linear models.

Practically, one has to create a database of fingerprints computed from a database of images. Then when a probe image is presented to the system for identification, the same fingerprinting algorithm is applied to the probe and the perceptual hash will be compared to those in the database. The index of the fingerprint with the smallest distance will be returned by the identification system. Given a binary fingerprint, the distance metric is typically the Hamming distance, i.e., the number of bits that differ between the two fingerprints. Technically, this operation can be implemented by the XOR (eXclusive OR) operator. Therefore, we have an identification problem, which can be formulated in the following way: considering a database of M objects, let x_m be an enrolled object and $b_m^x \in \{0,1\}^L$ its binary fingerprint with $1 \leqslant m \leqslant M$, and L is the length of the fingerprint. The multiple hypothesis testing problem consists of finding the corresponding index $j \in [1; M]$ to the probe image y_i with binary fingerprint b_i^y following the above mentioned criterion. The probe image y is considered to be passing through a channel[3] such as $p(y|x)$ [12] :

$$\begin{cases} H_0 : p(y|H_0) = p(y|x'), \\ H_m : p(y|H_m) = p(y|x_m), 1 \leqslant m \leqslant M, \end{cases} \tag{1.1}$$

where x' is a random vector with no relation to the objects in the database. The multiple hypothesis is an $(M+1)$ hypothesis problem, where we have M hypothesis for each object in the database and one extra for the case where a probe with no relation to the database is presented.

1.1.3 Machine learning and computer vision

A commonly accepted definition of machine learning, describes it as the: "field of study that gives computers the ability to learn without being explicitly programmed" [16, 17].

It is important to recall the pioneering work of Marvin Minsky and Seymour Papert about artificial neural networks [18]. The topic of their book was the Perceptron, which was considered as one of the most basic kind of artificial neural network. A single neuron (recieving multiple inputs and providing a single output) implemented with a Heaviside Step function as its activation function, was called a Perceptron [19]. It was essentially a binary classifier that could only classify linearly separable data.

The modern machine learning techniques deal with both hand-crafted features and more recently with automatically extracted image features. In both cases SVM and neural networks are used. We will consider briefly their properties.

[3]The term "channel" refers to the theoretical noisy channel model invented by Claude Shannon through which bits have a probability (intrinsic to the channel) to be flipped.

Global hand-crafted descriptors

In this section we discuss a holistic representation of images and their implementations. The most straight-forward way to compare two images $\underline{x}$ and $\underline{y}$, considered as 2 synchronized n-dimensional vectors[4], would be to compute their pixel-wise difference: $|\underline{x} - \underline{y}|$. The lower the result, the more similar the images. This approach however, has many flaws. The main problem being that the image is taken as a whole and its constituents (e.g., house, car, tree, etc.) cannot be isolated and considered as features. Moreover, we should add that such a simplistic approach is of course not robust to geometric transformations, noise or intensity changes.

Other global methods include comparing image histograms instead of the raw pixel values. It is quite clear that this approach is very coarse and at best can give us information such as whether an image is brighter or darker or if a given color is more present (if considering RGB histograms for example). It is interesting to note, however, that as the image histogram does not take into account pixel positions, an image having suffered a random permutation of its pixels (but keeping all intensities the same) will produce the same histogram. This suggests that such a technique is characterized by a low distinguishability, which is a natural price of geometric invariance.

More advanced algorithms exist, namely global descriptors such as GIST [20], which is advertised by its authors to be effective for recognition of natural scenes. (The name of this descriptor comes from an English slang term: "the gist", which means "the main point of something" or "the general meaning of something"). It is also possible to compute HoG (Histogram of Gradients) [21] or CHoG (Compressed Histograms of Gradients) [22] on a whole image as a global descriptor. An impressive application of HoG applied to the detection of human silhouettes is presented in [21].

To conclude this section, we would like to mention that these approaches lack the ability to focus on local features of the images, and are far too coarse for *fine-grained* recognition. Indeed, pharmaceutical packages are mostly whitish rectangles, and we are not interested in the recognition of the general shape or color but in small local details such as text or graphics showing number of pills in the package or active ingredients, etc.

Local hand-crafted descriptors

Approaches based on local descriptors usually proceed in two steps for their computation. First, keypoints are detected, then they are described in a compact vector of values. Usually, descriptors are computed for small patches of pixels (e.g., 16 by 16) around a keypoint.

The matching procedure is based on the predominance of similar local descriptors: if a large enough number of pairs of local descriptors are considered similar, then there is a good probability of match between two images. The similarity measure can be computed by Euclidean distance or cosine distance between two descriptor vectors.

[4] An N by M pixels image (i.e., an NxM matrix of pixels) can be "vectorized" in a (N*M)x1 column vector or a 1x(N*M) row vector.

Many descriptors have been developed since the oldest of them and still "one of the best in terms of its performance" as stated in [23], namely, the Scale Invariant Feature Transform (SIFT) [24, 25].

Also, we should distinguish between two families: the binary and the non-binary descriptors. Descriptors such as SIFT [24, 25], SURF (Speeded Up Robust Features) [26], GLOH (Gradient Location and Orientation Histogram) [27] are non-binary and based on the computation of histograms of gradients. Conversely, BRISK [28], BRIEF [29], AKAZE [30], FREAK [31] and ORB [32] are example of binary descriptors.

SURF uses integral images to improve computation speed (and its built-in detector uses the determinant of a Hessian for blob detection). GLOH is similar to SIFT but it is reducing the size of the descriptor to a vector of dimension 64 instead of 128.

Binary descriptors such as ORB, BRIEF or BRISK, can overcome the burden of computing histograms of gradients, which is time consuming, and instead produce binary strings based on the intensities in local patches. Moreover, these binary strings can be compared using the Hamming distance, which can be implemented very efficiently with a single XOR instruction.

In the scope of this project, the SIFT algorithm was chosen. The built-in SIFT detector was used. It is based on the DoG (Difference of Gaussians). SIFT exhibits a reasonable invariance to scaling, rotation and viewpoint variations. The SIFT descriptor computes histograms of gradients for small image patches. The invariance to rotation is achieved by always orienting the patches in the direction of the dominant gradient. Moreover, the SIFT implementation used in the scope of this project, allows the adjustment of various parameters of the detector. We have used the peak threshold (PeakThresh) parameter, which allows to control (roughly) how many keypoints are detected. In the following chapters, we will present two state-of-the-art approaches. Firstly, based on geometrical alignment of SIFT keypoints, secondly based on soft clustering of the descriptors (without taking geometrical information into account). Finally, it should be pointed out that SIFT applies to grayscale images.

Support vector machines

The Support Vector Machine (SVM) [33] is an algorithm for linear binary classification. The typical example presented in many tutorials is the separation of 2D data points with a straight line (more generally a hyperplane[5]). If the data is not linearly separable in its original space, the feature dimensionality increase and the kernel trick [34] can be used. The general idea is to switch to the representation of data in a higher dimensional space in which linear separability would be possible.

This machine learning technique could be applied to clouds of feature vectors (such as the ones created by local descriptor algorithms presented at the end of section 1.1.3). In the context of pharma package recognition (although this was not a method which we experimented with), one SVM for each unique package would be considered. The binary linear classifier would separate the feature vectors from the considered package versus those from *all* others. For example, given 1000 medical

[5] If we consider that we are placed in a 2D space, then hyperplanes in such a space are just lines (i.e., 1-dimensional hyperplanes).

packages, a thousand SVMs would be needed, using one-vs-all encoding. Given a probe package image $\underline{y}$, one would need to go through all 1000 SVMs and choose the one which could discriminate the features of $\underline{y}$ from all the other features of dissimilar images. Hopefully, it would be the correct one, otherwise a false positive would occur. However, if no SVM could classify it (i.e., it would be classified as dissimilar by all SVMs), then we would have an undefined situation and it would make sense to consider that this item is not in the database. Of course, if it was in reality corresponding to a database object, then a miss (false negative) would have occurred

Artificial neural networks

Inspired by the human brain, with its billions of neurons and synaptic connections, scientists developed artificial neural networks (ANNs) and *deep* ANNs [35] mimicking some of the aspects of the learning processes in our brains. Although, they are not yet - and by far - comparable to the human brain, they provide very interesting results and are now widely used, among others, by some large companies such as Google, Amazon or Facebook.

Without pretending to be exhaustive, we would like to mention some of today's state-of-the-art deep neural networks such as GoogLeNet [36], AlexNet [37] and "Very Deep Convolutional Networks for Large-Scale Visual Recognition" (VGG net) [38]. Moreover, it is interesting to mention tools like TensorFlow[6], which is a powerful framework provided by Goolge to help developers build deep neural nets. We can mention some very impressive achievements made possible by usage of deep neural networks such as Google DeepMind's AlphaGo [39] program beating a Go champion named Lee Sedol. Also, an impressive accomplishment in the field of "super resolution" should be mentioned: the "PixelCNN" presented in [40].

Although, the most impressive results about the deep nets are reported recently, the history of their development goes back as far as the 1960s [41, 42, 43]. This recent progress is mainly due to the appearance of modern powerful computers and availability of massive amount of training data.

To briefly give an overview of the fundamentals of ANNs, we can say that it is conceptually constituted by neurons and synaptic connections such as we can observe in Figure 1.3. When "fully connected" each neuron is essentially[7] doing a dot product between its inputs (from other neurons) and their synaptic weights. (Indeed we can consider a vector of inputs $\underline{x}$ and a vector of synaptic weights $\underline{w}$ and their dot product $\sum_{i=1}^{n} x_i w_i$). The result of this scalar product is then passed to an activation function (such as a step function, sigmoid, hyperbolic tangent, arctangent, rectified linear unit (ReLU), ...), which provides a non-linear[8] response. Let f be the non-linear activation function, then a neuron's response is given by $f(\sum_{i=1}^{n} x_i w_i)$. This value is then passed to the next neural layer, if any. Finally, the output layer of the network can contain, for example, m neurons whose values represent probabilities (the activation functions returns a value between 0 and 1) of belonging to one of m classes.

[6] https://www.tensorflow.org/
[7] In practice, a bias value may need to be added.
[8] It is known that this non-linearity is important to deal with complex models.

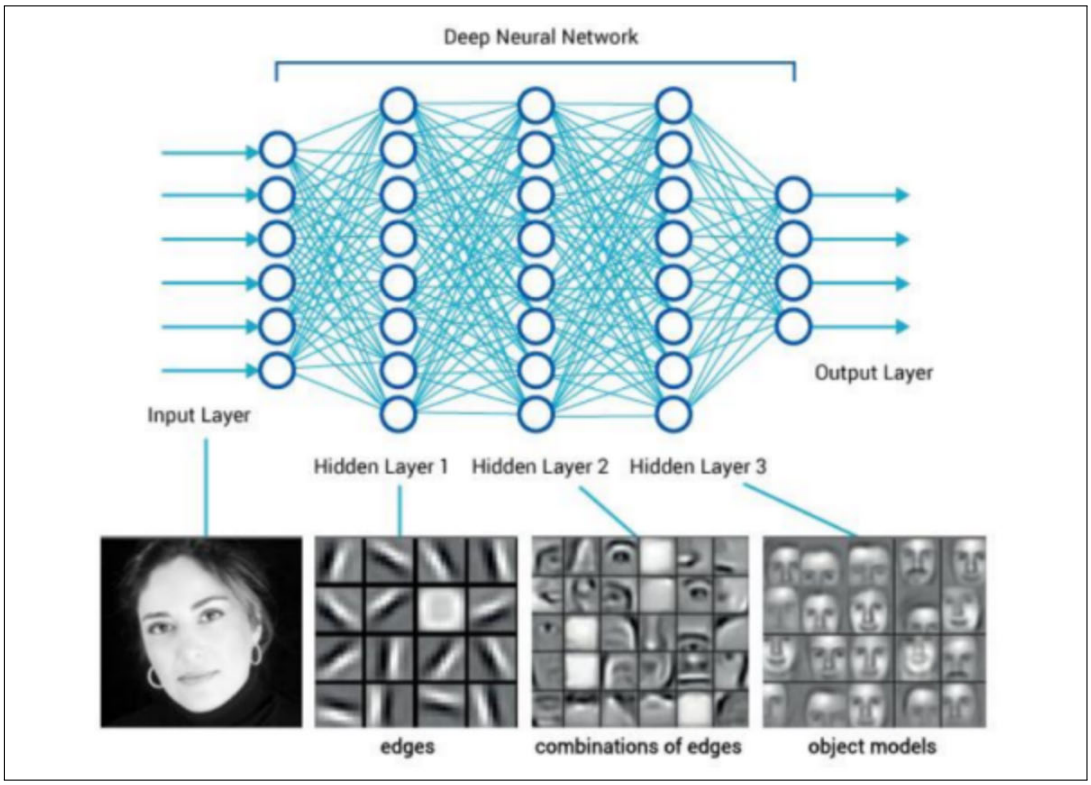

FIGURE 1.3: Intuitive representation of a deep neural network[9].

An essential concept of ANNs is *training* (provided that they have the ability to learn, which is possible using algorithms that will be briefly described afterwards). A well known theory describing simplified mechanisms, which can be used in computer programs to simulate synaptic plasticity to some extent is proposed by Hebb's rule in [44]. An intuitive summary of this rule is given by Siegrid Löwel's sentence: "neurons wire together if they fire together" [45]. Indeed, it is the ability to teach ANNs, from a training set of labeled values (i.e., supervised learning), to produce correct outputs, that makes them particularity powerful at dealing with problems such as classification and pattern recognition. A common algorithm for training is "backpropagation" [46, 47]. Roughly, the idea is to propagate errors backwards (i.e., from output back to input) and modify synaptic weights until a good performance is achieved, i.e., training set objects are assigned their true label by the network (e.g., good classification of labeled images).

The difference with standard ANNs and deep neural networks resides in the number of hidden layers. Indeed, following their input layer, deep ANNs can have many hidden layers, which enables them to detect "much more complex features" than shallow ones [48]. A conceptual representation of such a deep ANN is given in Figure 1.3.

Convolutionnal neural networks (CNNs) [49, 50, 51] are a category of deep ANNs that include the usage of the mathematical convolution operation. It is mostly used in computer vision to deal with images where a 2D-convolution with a small kernel (e.g., a 3x3 matrix) can represent a filtering operation (such as Sobel [52], Canny [53] or Prewitt [54] for edge detection, Gaussian blur, etc.). The "LeNet-5" [55]

[9]Source: https://leonardoaraujosantos.gitbooks.io/artificial-inteligence/content/neural_networks.html [Accessed: 25 May 2017].

convolutional neural network, designed to recognize handwritten digits, is shown in Figure 1.4.

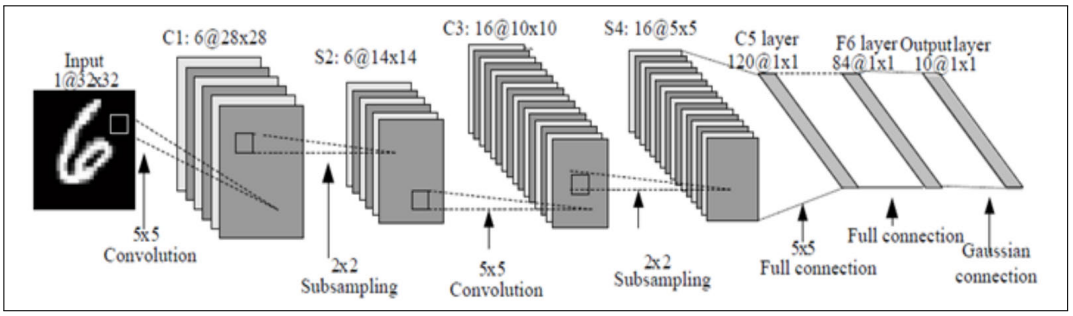

FIGURE 1.4: Architecture of the LeNet-5 convolutional neural network [55, 56].

1.1.4 OCR

Usually, the first step of an OCR (Optical Character Recognition) algorithm is to binarize an input image by using some kind of thresholding method (e.g., "Otsu's method" [57]). Subsequently, characters are segmented and compared to a template (from a codebook or library of character models) [58]. A simple matching procedure could, for example, consider the codebook entries having the lowest Hamming distances with the probe character as the best candidates for a match. More advanced methods based on machine learning techniques are used in the Tesseract OCR engine [59]. It uses training data to perform "linguistic analysis" used for the recognition of words.

SIFT descriptors have been used for optical font recognition (OFR) in [60]. The authors mention the challenges they face with languages such as Farsi/Arabic using cursive and connected letters. It is a significantly more challenging task because connected letters are harder to segment compared to fonts with detached characters. The same problem arises in European languages such as English or French when cursive fonts are used.

We must mention that no OCR based algorithms were investigated in this thesis, mainly because we have a significant number of enrolled packages originating from different countries. For example, Arabic and Cyrillic characters are found on many packages as shown in Figure 1.5. Moreover, texts are sometimes printed in fancy fonts (e.g., similar to handwritten text), especially some pharmaceutical companies names. Furthermore, many packages also contain some images or logos which are important for their identification and obviously would be discarded by OCR-based algorithms. Therefore, these are the reasons why we favored a more universal approach based on local descriptors.

1.1.5 Overview of existing databases

Overview

In this section, we present various databases publicly available for testing machine learning algorithms. The list of datasets mentioned is by far not exhaustive and only some of the most well-known ones are discussed. Moreover, a longer list is available at

FIGURE 1.5: Examples of packages with Arabic and Cyrillic writings.

https://en.wikipedia.org/wiki/List_of_datasets_for_machine_learning_research.

The California Institute of Technology (Caltech) provides the Caltech 101 [61] database containing images of 101 categories of objects http://www.vision.caltech.edu/Image_Datasets/Caltech101/. They advertise having around 40-800 images per category taken at a resolution of 300x200 pixels. By providing this database to the public, it opens opportunities for scientists to test image recognition algorithms. Recently, the Caltech 256 database has been released with 30607 images [62].

The ImageNet database [63] available at http://www.image-net.org/ provides a very large amount of images to researchers. It is based on WordNet [64] which is a database of words organized in a structured manner. As described on http://www.image-net.org/, the ImageNet database provides around 1000 images for each WordNet "synset" (which is similar to a concept, regrouping synonyms of a word or a category of word of the English language). Moreover, the ImageNet community proposes each year a challenge called the "Large Scale Visual Recognition Challenge" [65], which encourages scientists to test their image recognition algorithms on their database.

The ETH Zurich (Swiss Federal Institute of Technology in Zurich) provides various datasets stated as available to the scientific community "for research purposes" at http://www.vision.ee.ethz.ch/en/datasets/. Among others, there is a very large database of faces [66] at https://data.vision.ee.ethz.ch/cvl/rrothe/imdb-wiki/ claiming to have more than 500k images with face and gender labels.

The MNIST Database [67] consists of handwritten digits. This database has a training set of 60000 images and a testing set of 10000 images as claimed in http://yann.lecun.com/exdb/mnist/.

The Amsterdam Library of Object Images (ALOI) [68] which contains 1000 enrolled objects in different lighting conditions and viewing angles. The images are publicly available for scientific research at http://aloi.science.uva.nl/.

Also, we would also like to mention the Stanford mobile visual search data set [69] which contains images of various text documents, CD and DVD covers and other objects alike.

Finally, the closest dataset to ours is probably the Stanford Dogs dataset [70]. The authors claim that it is specifically designed for "fine-grained image categorization". It contains 20580 images of 120 breeds of dogs. A sample of 6 images of one of these 120 classes is shown in Figure 1.6.

FIGURE 1.6: Sample images from the Mexican hairless class.

PharmaPack

The above mentioned databases provide large amounts of images of various objects (e.g., apples, toys, cars, etc.). However, it seems that they all lack to provide a significant amount of images of the same *semantic category* of objects. Moreover, none of these databases specifically target pharmaceutical packages. Therefore, we introduce a new database: PharmaPack, which aims to provide a sufficient number of semantically similar pharmaceutical products to allow the computer vision community to develop and test fine-grained recognition algorithms. Moreover, we should add that images are acquired using modern mobile phones, to provide training data as close as possible to which an end user might produce.

We would like to mention that, up to our best knowledge, no database such as the one we provide, currently exists. Having asserted this, we should point out the

specificities of the PharmaPack database. Firstly, as already discussed, a specific category of objects is targeted: pharmaceutical packages. Some of their particularities can be summarized in the following way:

- *fine-grained recognition*: the goal is to accurately recognize each unique package (e.g., "Xanax 0.5 mg, 30 pills" from "Xanax 1 mg, 30 pills" both from the same pharmaceutical company and having identical designs). Such differences are very small and sometimes even difficult to notice by a human observer, at first glance. Whilst recent methods based on deeps nets, for example, show impressive results on the recognition of noticeably distinguishable classes, the recognition of minor details in order to identify similar objects remains a challenging task;
- *visual context of packages*: it is quite poor, as pharmaceutical packages mainly contain text and some logos, rarely some images. Moreover, packages of the same companies will tend to have similar logos and text fonts, therefore, extracted local descriptors have a high probability to have a low distance between each other;
- *recognition conditions*: depending on the lighting of the environment where the image is taken, the perspective and the distance at which the camera is held, the resulting image can vary quite a lot between each shot. This is of course a significant problem, and is also why we need multiple images of the same package taken from different angles;
- *memory footprint and complexity*: the size of descriptors should be as small as possible to decrease complexity (comparisons between potentially millions of candidate descriptors) as well as number of bytes to be transfered (e.g., in the context of using wireless channels to transfer data between a mobile phone and a server). Moreover, if the system is further extended towards the detection of fake packages, the complexity (in CPU time and memory) of the recognition and detection algorithms will most likely increase.

1.2 Objectives

As mentioned before, due to the lack of databases providing a large number of semantically similar objects suitable for the testing of fine-grained recognition algorithms, our first objective was to build such a database. This task required to collect a large amount of pharmaceutical packages. Subsequently, it entailed building an enrollment setup corresponding to our needs and establish protocols, conventions and interfaces for the organization and management of the database. Finally, we defined the objective of enrolling more than 1000 unique packages, acquired by modern mobile phones.

The second part of this thesis is devoted to fine-grained recognition of pharmaceutical packages. The main objective is to develop an algorithm that generates a list of candidates for each query. Moreover, our goal was to establish a baseline using local descriptor based recognition algorithms, needed for performance benchmarking. Two different methods were investigated. One is based on the geometric information (i.e., positions) provided by SIFT keypoints associated with descriptors. The other investigates a non-geometric approach based on descriptor aggregation. Experimental results will be presented in the form of separability histograms and ROC (Receiver

Operating Characteristic) curves. The latter are computed following the decision rule [71]:

$$\begin{aligned} P_D &= \Pr\{S(i,j) \geqslant \gamma | \mathcal{H}_i\} \\ P_{FA} &= \Pr\{S(i,j) > \gamma | \mathcal{H}_{\bar{i}}\} \end{aligned} \tag{1.2}$$

where P_D and P_{FA} are the probabilities of correct detection and false acceptance respectively, γ is a threshold, $S(i,j)$ is a measure of similarity between a feature vector $\underline{f}_j(i)$, $1 \leqslant j \leqslant 54$, and a probe $\underline{q}$, $\mathcal{H}_i$ and $\mathcal{H}_{\bar{i}}$ are the correct and incorrect hypothesis respectively. This methodology is based on statistical hypothesis testing [72].

1.3 Motivations

The main motivation of this project was the problem of counterfeit pharma products, which is a serious danger for health. Moreover, it is known that fake medicines have infiltrated markets worldwide. Furthermore, up to our best knowledge, there does not exist any mobile applications allowing the identification and detection of fake pharmaceutical packages. Therefore, one of our motivations is to make a first step towards this direction.

Moreover, a significant motivation was the lack of a publicly available database containing a sufficient amount of semantically similar objects. We hope that PharmaPack can fill this gap and we believe that it can benefit to the computer vision community for developing and testing fine-grained recognition algorithms.

1.4 Main contributions

The contributions of this work can be summarized in the following way:

1. the creation of a physical setup for the enrollment of pharmaceutical packages by modern consumer mobile phones from different viewpoints;
2. the creation of a database of medical packages providing a significant number of images of similar packages, suitable for the development and testing of fine-grained recognition algorithms;
3. the development of methods based on local descriptors for identification of pharmaceutical packages and the investigation of their recognition abilities;
4. a comparative analysis of the recognition performance for various scenarios based on different testing sets and local descriptor-based approaches;
5. application and investigation of object recognition methods in the scope of fine-grained recognition based on local descriptors.

1.5 Structure of the thesis

After the introduction which consists of an overview of different state-of-the-art approaches, we present in this section the structure of the thesis.

The experimental setup used to perform the enrollment of pharma packages is presented in chapter 2. Then, the issues encountered and the solutions found thereafter are described. The conventions for the organization of the database are presented in section 2.5 and all the technical parameters used in order to allow the reproduction of a similar setup are given in section 2.6. Subsequently, section 2.7 describes the implementation of the database and introduces a Web interface to allow its querying to display images. Afterwards, the configuration of a "private cloud" is presented in section 2.8, allowing automatic file transfer from mobile phones to an external wireless hard drive.

An approach based on geometrical matching of local descriptors is presented in chapter 3. Subsequently, the experimental results obtained for this technique are given in section 3.2. In section 3.4 we present some examples of false acceptance and investigate their origin, in order to illustrate the problem of fine-grained recognition. Finally, considerations with regards to complexity and feasibility of fine-grained recognition, based on geometrical matching, are presented in section 3.5.

In chapter 4, we present methods based on aggregation of local descriptors, which does not take into account any geometrical information provided by keypoints. The experimental results obtained are presented in section 4.2. Then, in section 4.3, the geometrical and non-geometrical approaches are compared. This chapter is concluded in section 4.4 with considerations about complexity and feasibility of fine-grained recognition using local descriptor aggregation.

Finally, this thesis is concluded in chapter 5.

Chapter 2

Experimental setup

2.1 Introduction

This chapter presents how the experimental setup was built and configured. Firstly, the physical enrollment installation is presented. All the conventions (i.e., file naming template, file transfer protocols, etc.) and parameters used are briefly presented. Subsequently, we describe the structure of the database and introduce a Web interface developed with the aim to allow the convenient execution of predefined SQL queries.

The enrollment process required to collect a large number of pharmaceutical packages from trusted sources. Their anonymity is guaranteed (e.g., remaining labels with patient's names were removed).

With regard to photographic quality, we briefly present issues and solutions that were found. Camera parameters are presented as well as various conventions, e.g., for keeping packages inside the camera's field of view, etc.

Finally, we describe the implementation of a system which we refer to as "private cloud", which allows us to automatically transfer images from phones to an external hard drive.

2.2 Description of the experimental setup

In this section we describe the physical installation, which was used to create the enrollment database and conduct the experiments.

The installation includes a cube-shaped metal structure on which three Samsung mobile phones are mounted (with adjustable sticks) and two controllable LED lamps mounted on a metal rail on-top of the structure. A rotating platform controlled by a LEGO step-motor is placed under the phones, and illuminated by LED lamps. This motor is itself controlled by a Matlab script by which we synchronize the rotation with the triggering of photographs through an application installed on the phones. We will refer to this setup as "the CUBE". It is shown in Figure 2.1 and Figure 2.2

The "first iteration" of experiments was dedicated to produce (relatively small sets) of sample images and evaluate their subjective quality with regards to recognition using image processing algorithms.

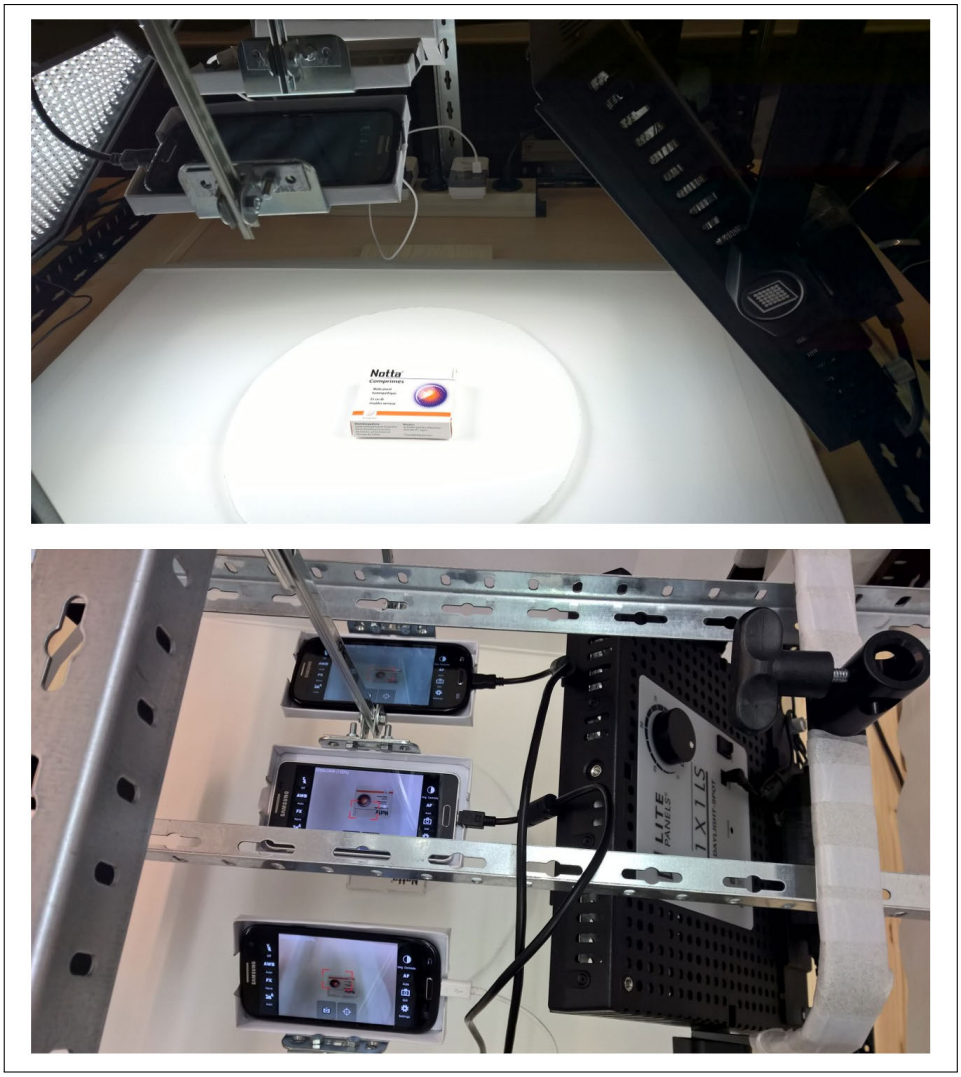

FIGURE 2.1: The CUBE setup with a white background on the rotating platform.

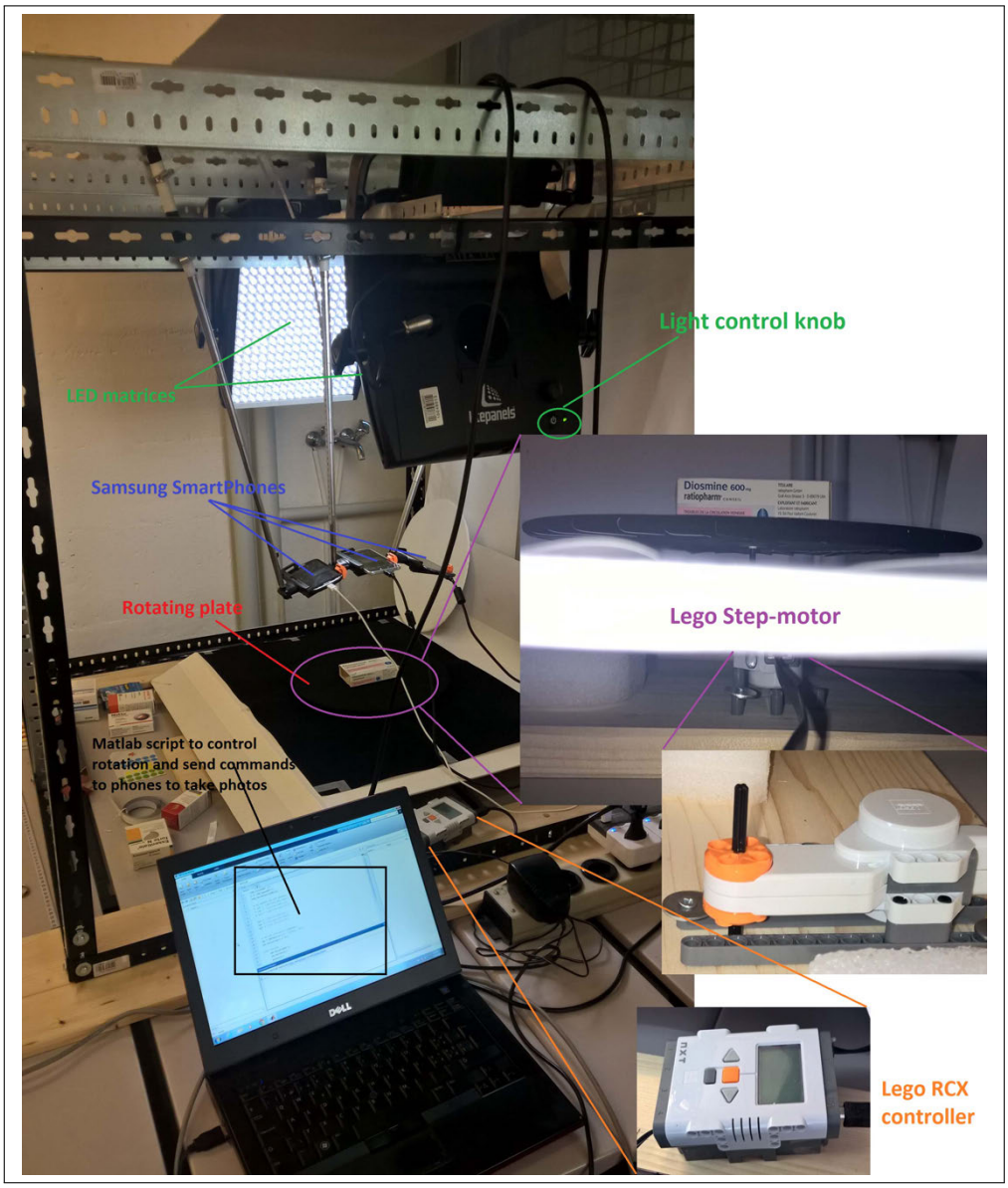

FIGURE 2.2: The CUBE setup with a black coated background.

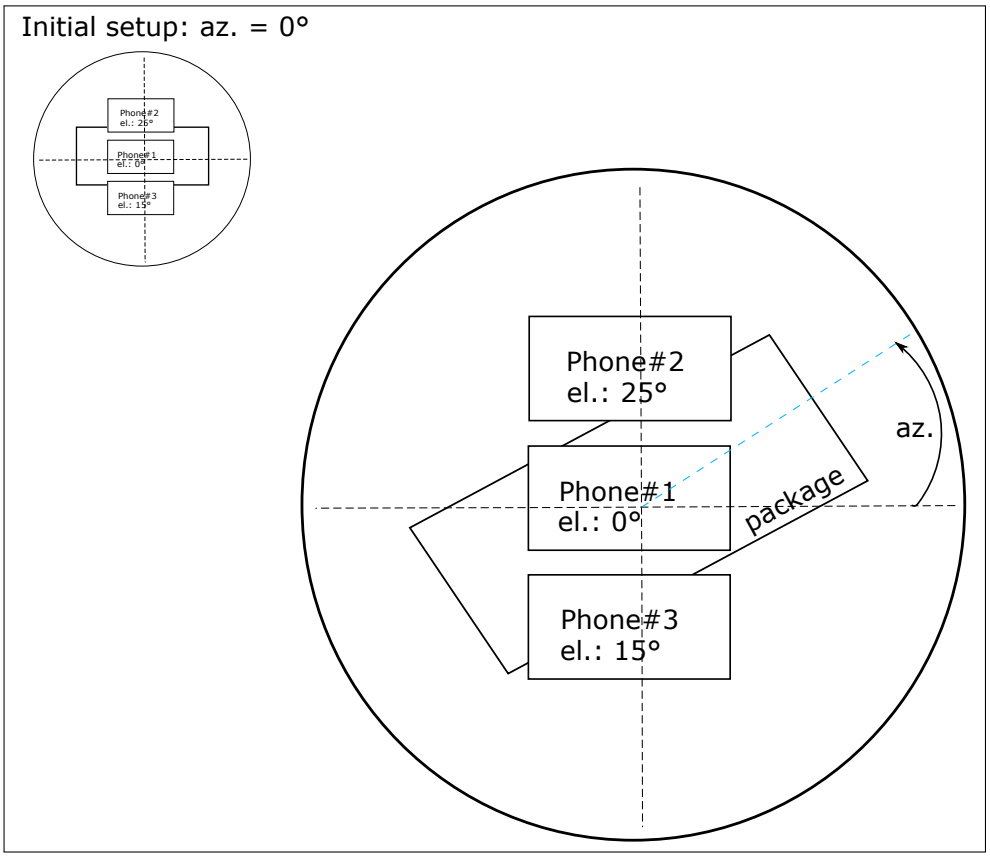

FIGURE 2.3: Diagram of the geometry of the phone's placement with regards to the rotating platform.

The phones, symbolized by small rectangles and the package being photographed represented by a larger rectangle are schematically shown in Figure 2.3. Phones have each their own (fixed) elevation, to simulate user's behavior of holding their phones to photograph a package, resulting in a different amount of projection in the image. Simultaneously, these data can be used for the training at the enrollment stage. The elevation at 0 degree (el=0) represents the "normal" conditions (no projection), i.e., holding phone straight on top of package. The elevation of 15 degrees represents a slight deviation, and 25 degrees is the maximal deviation we simulate, leading to the maximum perspective projection of the package on images. Initially, the azimuth is at 0 degrees and the platform is rotated by steps of 20 degrees, thus ending at az=360, to obtain 18 images per phone and therefore 18*3=54 images per package. Figure 2.4 shows images taken by the 3 different phones and Figure 2.5 shows all 54 enrolled images for one package.

TABLE 2.1: Summary of phone parameters.

Phone ID	Phone model	Nb photos/package	Elevation (degrees)
♯ 1	Samsung Galaxy Alpha (SM-G850F)	18	0
♯ 2	Samsung Galaxy S 3 (GT-I9300)	18	25
♯ 3	Samsung Galaxy S 3 (GT-I9300)	18	15

FIGURE 2.4: Phone 1 (el=0°, top), phone 2 (el=25°, center), phone 3 (el=15°, bottom). (Example from the enrollment database).

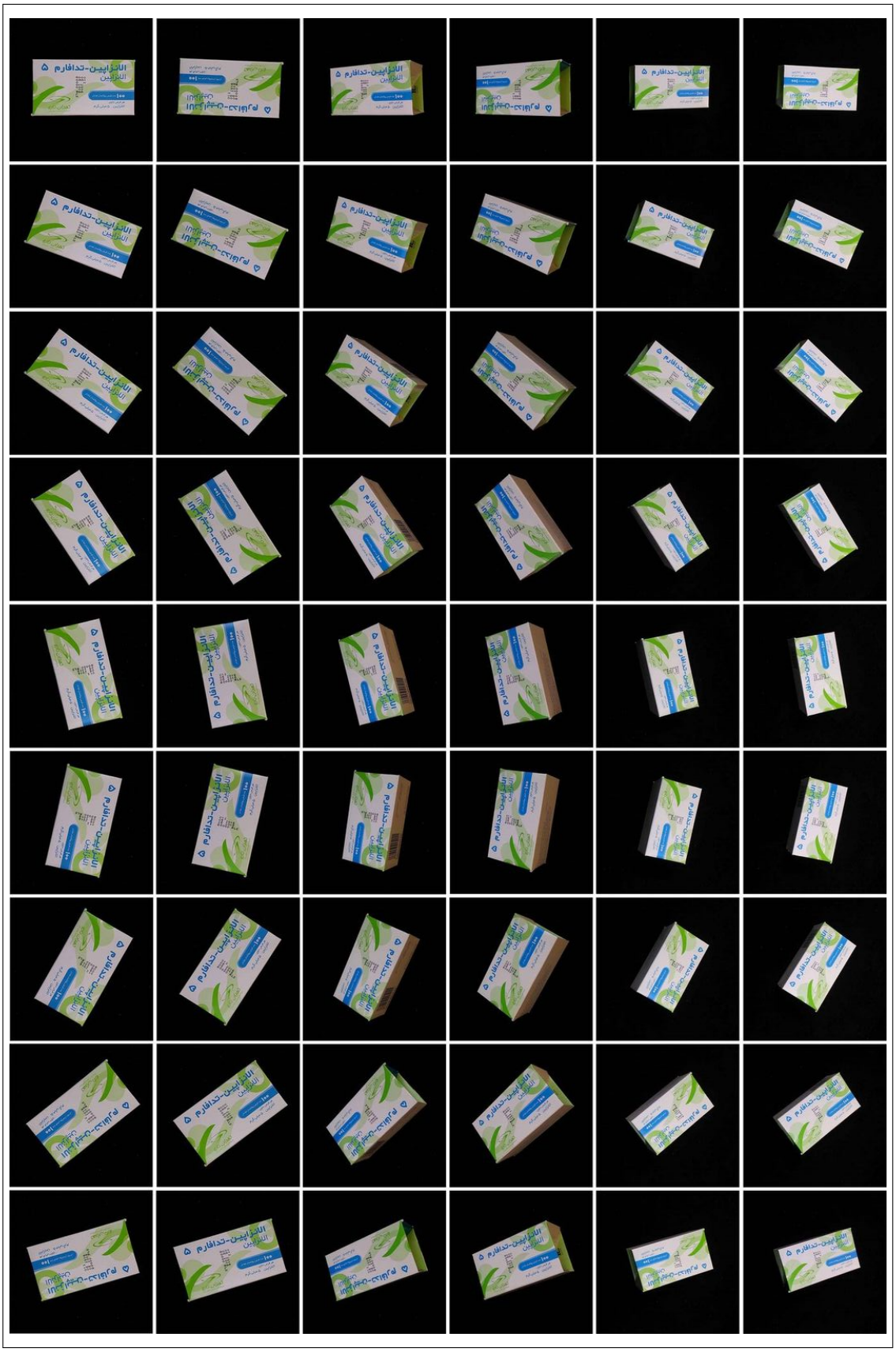

FIGURE 2.5: Example of 54 enrolled images for 1 unique package. Each pair of rows (starting from top) contain 18 images taken by phone 1, 2 and 3, in this order.

2.3 Lighting

2.3.1 White background

The first set of acquisitions performed on a white background, produced images such as shown in Figure 2.6. The problem with such images was the visible shadows and non uniform luminosity on the white background. The objective was to perform high quality acquisitions in the enrollment database. Therefore, such images could not be considered satisfactory in the scope of a high quality database of enrolled images used for object recognition. No optimal configuration entailing a white background, despite trying different light intensities and angles/positions, has been found.

FIGURE 2.6: Shadows on white background.

2.3.2 Black (anti-glare) background

The solution retained was to use a black (anti-glare) velvet-like material for coating the rotating platform and background below it. This allowed us to remove the problematic shadows. However, with the type of camera (and camera software) built-in the mobile phones, an unavoidable automatic adjustment takes place given the percentage of ”black” with regards to the area occupied by the packages in the images. Indeed, photography theory stipulates that, in *most situations*, a good exposure[1] is achieved when the averages brightness measured by the camera's metering system tends to a middle gray ("18% gray"). This value of ”18% is considered to be the "average reflectance" of objects in typical situations [73]. Perceptually, we think of it as being a gray value half way between pure white and pure black. The fact that it is reflecting 18% light and not 50%, as one would intuitively think, stems from the logarithmic perception of light by our eyes. Therefore, if 50% of the incident light was reflected, then we would perceived a much brighter gray. It is demonstrated in [74] that a gray card (such as the ones used to calibrate cameras) with 18% reflectance corresponds to a CIE 1976 L*a*b* (CIELAB)[75] color space value of L*=50 a*=0 b*=0, i.e., a mid-gray of a (perceived) brightness of approximately[2] 50%. Regardless of what elements make up the scene being photographed, the camera's auto-exposure

[1] We recall that exposure is controlled by 3 parameters: aperture size, shutter speed and ISO (how much the signal is amplified).

[2] Also demonstrated in https://en.wikipedia.org/wiki/Lightness.

system will make sure that the average scene intensity is middle gray (on the other hand, in some circumstances the auto-exposure system will compare the scene to a built-in database of thousands of photos to produce a smarter exposure[3]).

As mentioned in [76], using mid-gray or light gray backgrounds would provide better results because they have less effect on exposure as the camera's light meter will measure mostly an average gray and will not consider the need to over or under-expose. However, as mentioned before, even a mid-gray background would generate too much shadows. The most problematic situation, when using a black surface, was over-exposed images such as shown in Figure 2.7 (right). Most consumer mobile phones (such as the ones used in this project) do not allow users to have full control on settings such as shutter speed and aperture size. Therefore, we must rely on the Exposure Value (EV) setting which can only take a range of values between -2 and 2.[4]

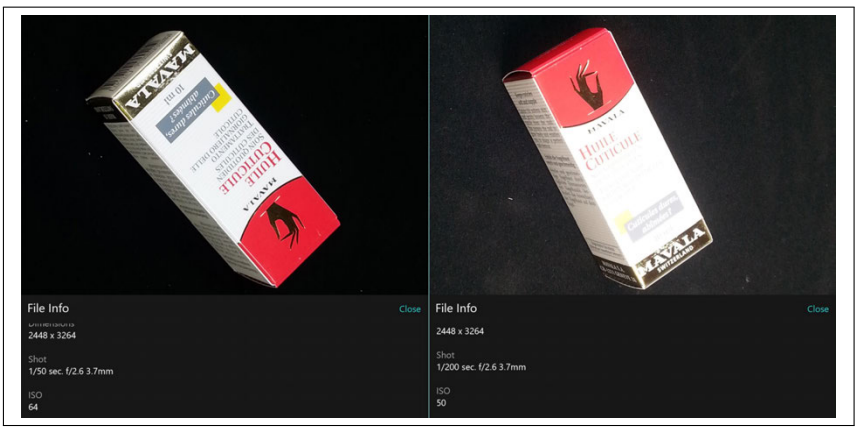

FIGURE 2.7: Correctly exposed image (left), overexposed image (right).

Other issues we were facing concerned specular reflections (changing reflectivity depending on angle) of some packages having very reflective material, as shown in Figure 2.7 (right, golden shiny band).

Figure 2.8, shows 3 correctly exposed images (top row) of a white sheet of paper, a middle gray square printed on a sheet of paper and the black velvet-like background used for shooting images of pharma packages. On the bottom row, the same were photographed, however, this time leaving the automatic metering settings. This experimentally shows that one cannot rely on fully automatic settings. Indeed, the camera's software is built in such a way that the best exposure is considered achieved when the average intensity in the image tends to a mid-gray.

As shown in Figure 2.10, the exposure is corrected by a low ISO setting and an EV=-2. This is correct for most situations, yet there are specific packages where a slight over-exposition is noticed. Indeed, as mentioned before, we do not have control on all camera parameters (e.g., shutter speed), and therefore the phone still adjusts

[3] http://www.nikon.com/news/2009/1014_d3s_01.htm

[4] Lower EV values resulting in a lower exposure (darker images).

[5] These values are only given in order to show that they are comparatively similar, however a RGB or greyscale value by itself is almost meaningless if not linked to an absolute color space, such as CIELAB, as a given RGB value of [127,127,127] on a given screen could be perceived as the same gray as, e.g., [140,140,140] on another device depending on the display technology, etc.

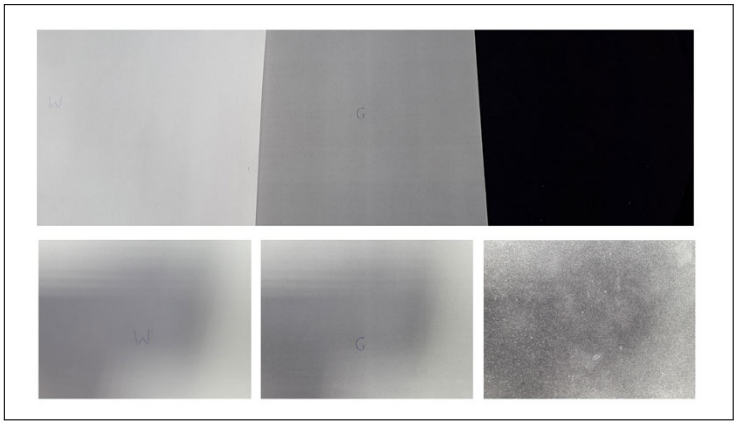

FIGURE 2.8: Results of the "Average gray experiment". Top row images correspond to the bottom row column-wise. (Mean intensity values from left to right : 149.9, 155.8, 160.6)[5].

FIGURE 2.9: Example of incorrect (left) and correct (right) exposure when leaving default camera settings. (left = image ♯2, right = image ♯1)

FIGURE 2.10: Corrected image ♯2 with EV setting = -2.

TABLE 2.2: Parameters of correct, incorrect, and corrected exposure.

Image ID	Exposure	Aperture size	Shutter speed	ISO	Exposure bias
♯1	Correct	f/2.2	1/33 sec.	320	0 step
♯2	Incorrect	f/2.2	1/17 sec.	1250	0 step
♯2	Corrected	f/2.2	1/17 sec.	100	-2 step

automatically some settings. If the shot was still too bright, in rare problematic cases, the LED lights had to be slightly dimmed.

It should be observed, that among the collected pharmaceutical packages, a small number of almost completely black or very dark ones can be found. Likewise, some packages are coated with very glossy materials, which, due to their high reflexivity, causes unacceptable image quality (e.g., almost indistinguishable logos or texts). Some examples of such problematic packages are shown in Figure 2.11. However, clearly, a large majority of packages are similar to Figure 2.6, which is quite a typical pharma package.

FIGURE 2.11: Examples of problematic packages.

2.4 Confidentiality

A large number of pharmaceutical packages was needed to enroll 1000 unique ones. Indeed, in every batch of packages, a high percentage entailed already enrolled products. Obviously, some medicines are more common than others, such as ordinary painkillers (e.g., Aspirin, etc.), whereas some tranquilizers or opioids such as morphine based drugs are quite less frequently obtained. These packages have been collected from trusted sources and no patient's personal information is recorded in the database.

2.5 Naming conventions

2.5.1 Description

To be able to uniquely identify packages for the reproducible research, naming conventions have been defined. They will be described in this section.

It should be emphasized that the project entails a *digital* database as well as a *physical* database. Indeed, the *physical* packages are kept in a safe location. They are stored in labeled storage boxes and sorted by their unique identification number. This ensures the possibility to create new datasets with different parameters (e.g., higher resolution, different background, etc.).

2.5.2 Hierarchical structure

When the project was started, packages were only identified by a unique package number and a sample number. Two packages of the same product but with different designs, like in Figure 2.12, were not distinguished. During a second phase, when enough packages were collected and many different designs were encountered, we introduced a hierarchical structure to take into account these specificities (see Figure 2.13).

This new model is designed like a 3-layer tree structure. The top node of the model corresponds to *metadata* concerning the product (i.e., name, brand, number of pills, etc.). The second level concerns the product design: a package can have the same meta-data but have a very different design as shown in Figure 2.12. The last level of the tree takes care of individual samples of a given product (e.g., "Cipralex antidepressant, Lundbeck, design 1" -> sample 1, sample 2, etc.). Indeed, we have collected many samples of the same products as well as a few products with significantly different designs, although the product inside is exactly the same (name, pharmaceutical company, number of pills, active components inside, etc.). We should add that in some rare cases, we could notice minor modifications between samples such as a line of text shifted to another side of a package, although the global design remained the same. Therefore, in such a situation, they were enrolled as yet another sample but with a comment (in the database) describing the minor difference(s) with regards to the other samples. This is an important element towards the fine-grained recognition.

FIGURE 2.12: Example of 2 products with same metadata but different design.

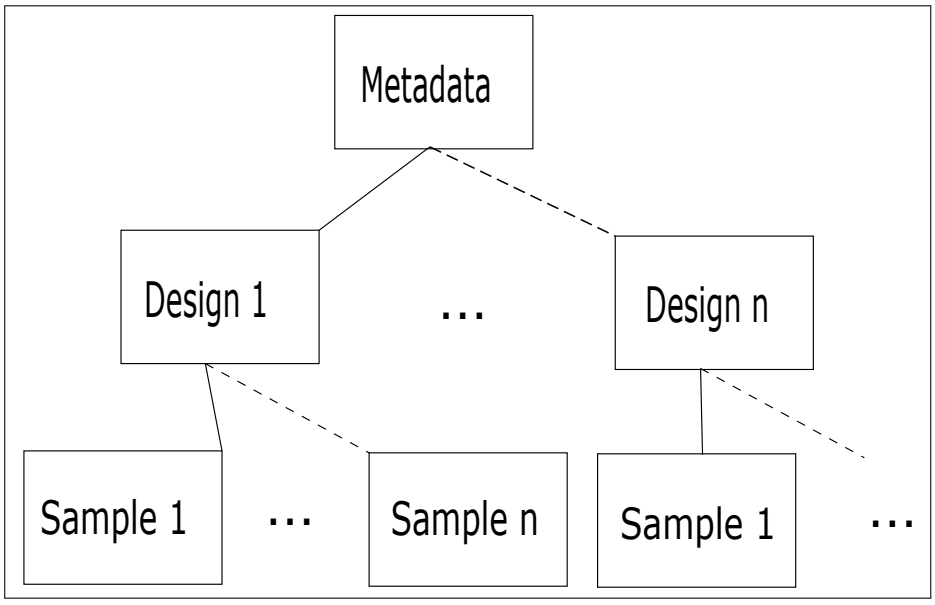

FIGURE 2.13: Diagram of the 3 layer tree structure for package naming.

2.5.3 Naming template

To proceed uniformly with each file and to be able to quickly search (e.g., with regular expressions), we defined a naming template containing the following elements: phone number, package identifier, sample number, class of package, angle and side.

Formally, the naming template is the following:

- Ph♯ : phone number (1-3);
- P♯♯♯♯: package number (e.g., 'P0001');
- S♯♯♯: sample number (when multiple copies of the same product, e.g., 2 boxes of "Aspirin" (same product but another sample));
- C♯: class of the package on the image (1-4)(classes are w.r.t. distance of phone to package);
- az♯♯♯: azimuth (angle of platform rotation: 20-360);
- side♯: which side is enrolled (side 1: main side).

As an example, this is the filename of the package in Figure 2.14 :

`PFP_Ph1_P0069_D02_S001_C3_az020_side1.jpg`

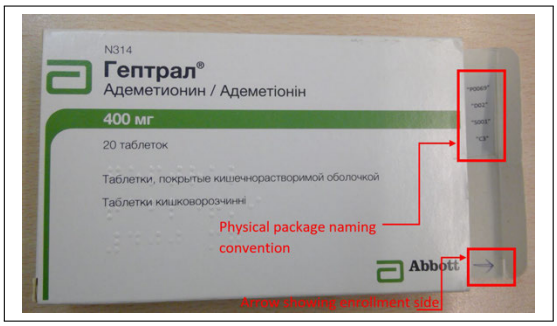

FIGURE 2.14: Example of physical package naming convention (label shows: P0069, D02, S001, C3).

Finally, it should be mentioned that file naming is done automatically by a Visual Basic Script (.vbs), which takes a list of package identifiers (IDs), and the rest of the above information (encoded in VBS arrays), and renames packages accordingly. Exactly 18 rotations of 20 degrees are performed in such a way that there are 54 images per package, considering that we have 3 phones as described in section 2.1. Given this fixed parameter, we can use a *for* loop with a counter to determine the angle at which the image was taken as shown in the following **pseudo-code** (the actual implementation is in VBScript):

```
Do Until files.lastFile
        If cnt==18 Then
                cnt=0
                cntId=cntId+1
        Endif
```

```
        cnt=cnt+1
        azimuth=cnt*20
        renameCurrentFile "PFP_" & PhoneID(cntId) &
                Design(cntId) &
                Sample(cntId) &
                Class(cntId)  &
                "_az"+azimuth &
                "_side1.jpg"
Loop
```

2.6 Reproducibility

2.6.1 Description

In this section, we give implementation details and figures which are aimed at allowing our experiments to be reproduced.

2.6.2 Classification

Given the constraint of maintaining a similar scale for all packages on all images, regardless of the package's physical size, 4 different size classes were defined. This also handles the constraint of keeping the whole package visible on the images (no clipping). Thus, it is required to keep a higher distance between the phone and package for larger ones, and lowering the distance for smaller packages. Therefore, classes are characterized by the distance between the rotating platform and the phones, depending on the package's dimensions. The following table[6] presents each class's characteristics (i.e., distance from platform to phone).

Phone \ Class	C1	C2	C3	C4
Phone 1	25[cm]	28[cm]	36[cm]	50[cm]
Phone 2	20[cm]	22[cm]	28[cm]	35[cm]
Phone 3	23[cm]	26[cm]	36[cm]	43[cm]

2.6.3 Camera software

The software installed on phones, called "CameraPro", allows wireless control of the integrated camera via Wi-Fi technology. As explained in subsection 2.3.2, with the default camera settings, the quality of resulting images is unsatisfactory. Therefore, the CameraPro settings that were used are the following:

- resolution: 2448 × 3264 (8 Megapixels);
- exposure compensation (EV): -2.0;
- flash: off;
- AWB: auto;

[6] Please note that the values given are approximative because of the variance of the package's dimensions (width × height × depth), thus we keep a flexibility of about ± 5 [cm].

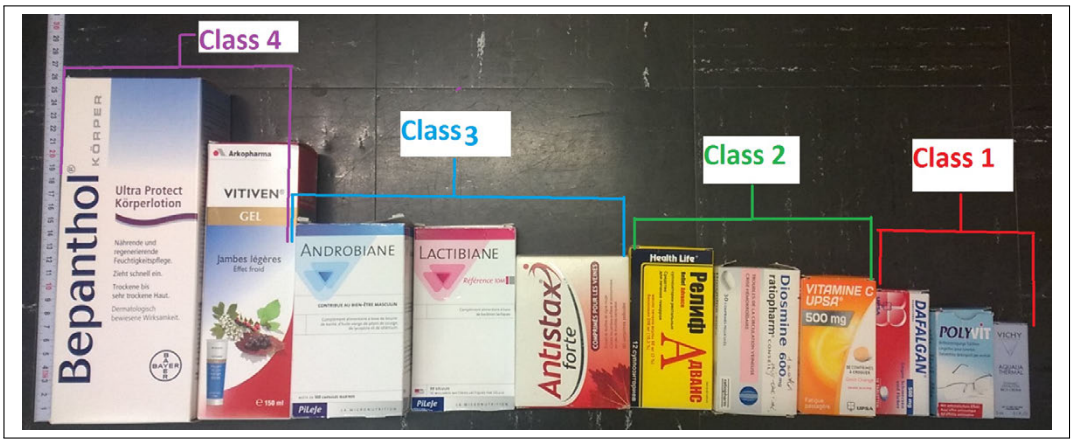

FIGURE 2.15: Reference packages for each class: *approximative* range of package sizes from largest to smallest. Classes were defined empirically by comparison of all gathered packages to be enrolled in the database.

- ISO[7]: auto;
- brightness: 0;
- sharpness: 0;
- saturation: 0;
- anti-banding mode(AB): 50Hz.

Although some settings were manually changed, efforts have been made to leave as much as possible the default ones in order to acquire images in conditions closest as possible to those of a standard user.

2.7 Database

2.7.1 Description

This section is dedicated to the organization and interfaces of PharmaPack. As previously mentioned, a database of physical packages collected from various sources was created. Afterwards, these physical packages were enrolled in the digital database, i.e., images were acquired and metadata was inserted.

2.7.2 Implementation

The digital database was implemented with the MySQL[8] RDBMS (Relational Database Management System) and a Web interface was implemented with the PHP[9] scripting language.

A Web server running on http://gallager.unige.ch/Master/ provides the Web interface to the database. Figure 2.20 shows an HTML page with a Web form and

[7] In particular cases, the ISO value can be manually increased or decreased (e.g., when encountering problematic/special packages).
[8] https://www.mysql.com/
[9] https://secure.php.net/

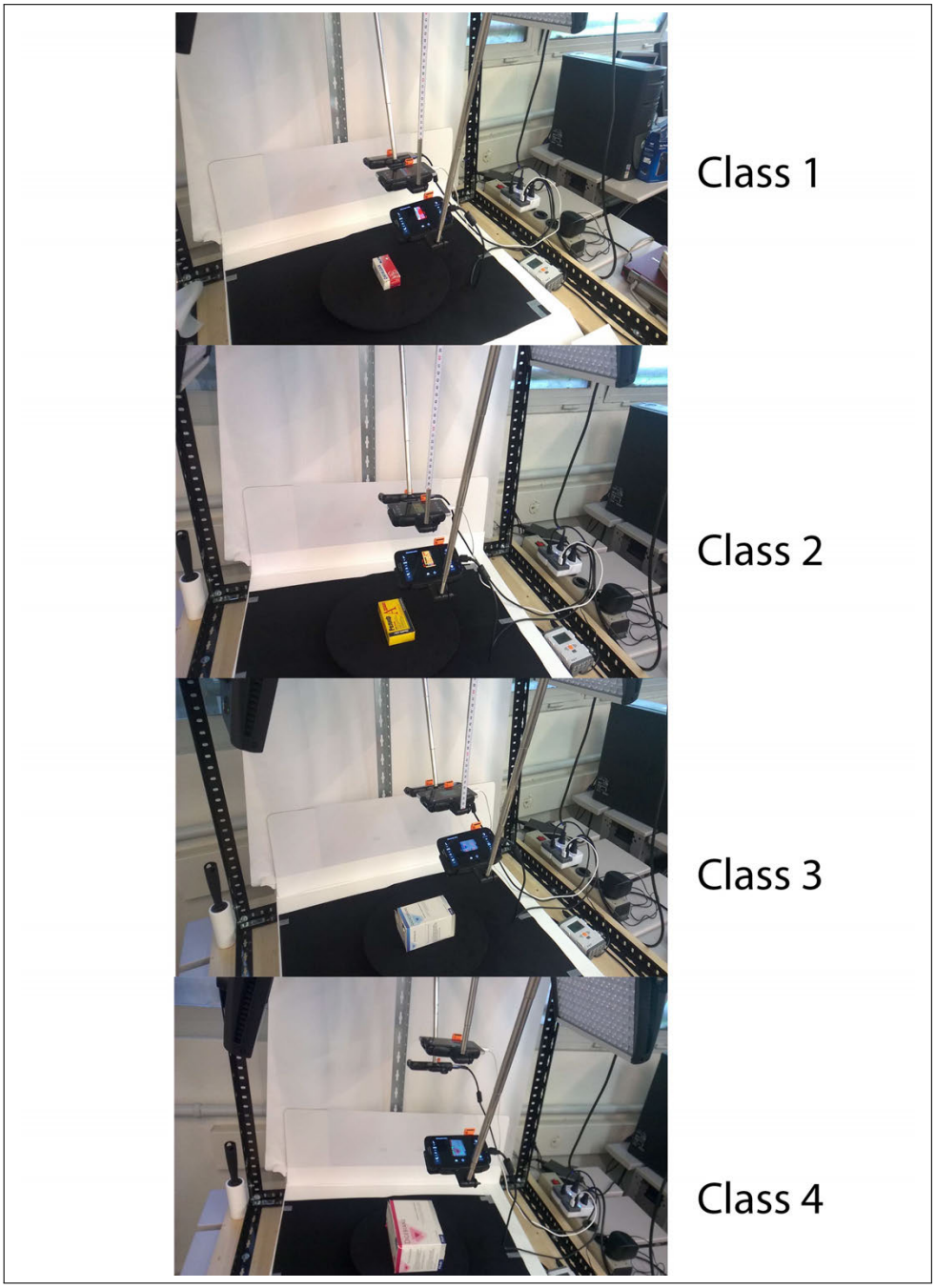

FIGURE 2.16: View of packages of each class shown in the enrollment setup.

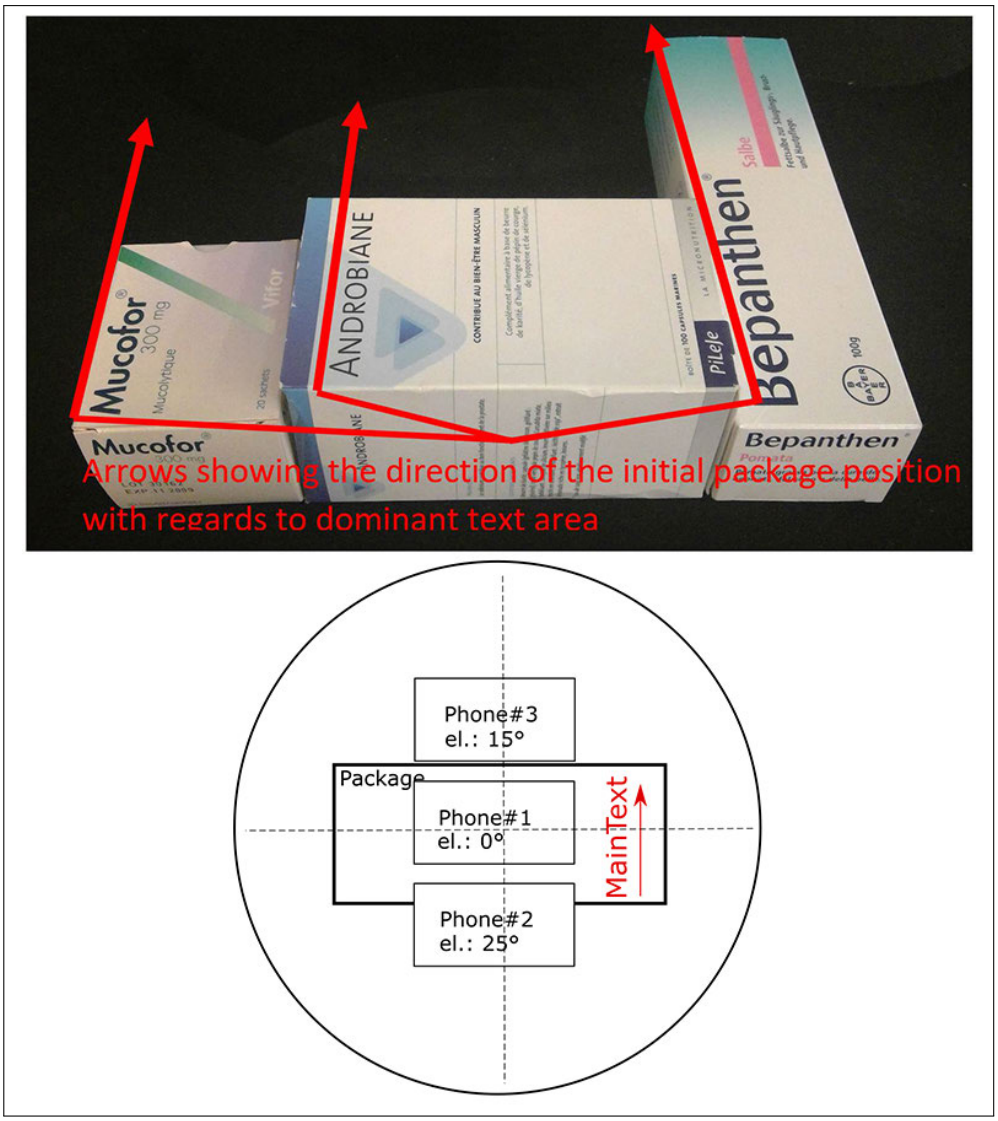

FIGURE 2.17: Convention for the initial position of packages on the rotating platform (convention shown for a sample of three packages having significantly different shapes).

options to provide a convenient way to search in the database. The HTML form is then submitted to a PHP script, implementing the required SQL queries, which is executed on the server. The response is transmitted to the client as shown in Figure 2.19. Various parameters can be adjusted through the Web interface to change the way that the results are presented, e.g., choosing to display thumbnails (and customize their size) instead of links, etc.

The MySQL database consists of two tables, namely **Boxes** and **Classes**. The primary key of the Boxes table is composite and expressed in the SQL language as:

```
ALTER TABLE ‘boxes‘
   ADD PRIMARY KEY (‘Number‘,‘Design‘)
```

This composite key allows us to implement the constraint that a product is uniquely identified by its number and design as described by the hierarchical structure defined in subsection 2.5.2. Indeed, such constraints reduces risks of errors. Other fields are shown in Figure 2.18. The Classes table contains the four shape categories that were defined in order to cover all the possible different package sizes. It should be pointed out that the CID (Class IDentifier) field in the Boxes table contains a foreign key, which references a primary key in the Classes table.

Image files were not stored directly in the MySQL database, although this could be an option using the BLOB (Binary Large OBject) or LONGBLOB[10] SQL datatypes, which may contain binary data. Conversely, one could store the file paths and names in SQL table fields. The approach used for PharmaPack consists of retrieving files with PHP scripts using regular expressions. The latter provide a language-independent solution. An example showing such regular expressions is shown below (PHP language):

```
foreach ($phone1Files as $fileName) {
   if(@preg_match('/_Ph1_P(0)*'.
   $row["Number"].'_D(0)*'.
   $row["Design"].
   '/',$fileName)){

   //code to present the data in
   //an HTML table
   ...

}
```

The above PHP code fragment loops through all images taken by Phone 1 and matches the ones corresponding to the query, i.e., the package number inside $row[" Number "]. The structured naming template defined earlier allows us to do this very conveniently.

Finally, we should add that regular expressions have the advantage of flexibility compared to a "hard coded" file name in a database field. On the other hand, storing BLOB binary data directly would most likely excessively overload the MySQL database, considering the number of files and their size. This would make database management cumbersome. Therefore, a logical separation between metadata and file

[10] https://dev.mysql.com/doc/refman/5.7/en/blob.html

data (images) was favored (cf. Figure 2.13). This has the advantage of allowing the use of regular expressions. The latter providing a quick and reliable way to extract relevant files in case of a naming template modification.

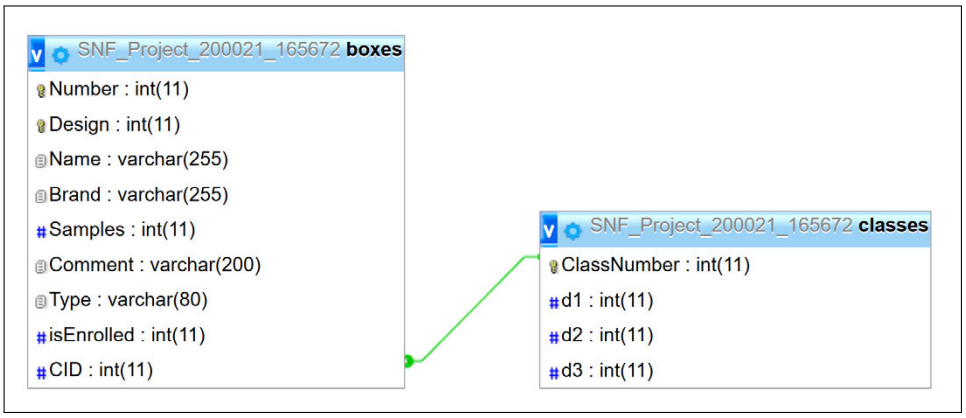

FIGURE 2.18: Relational view of tables in the database (diagrams created from phpMyAdmin database manager).

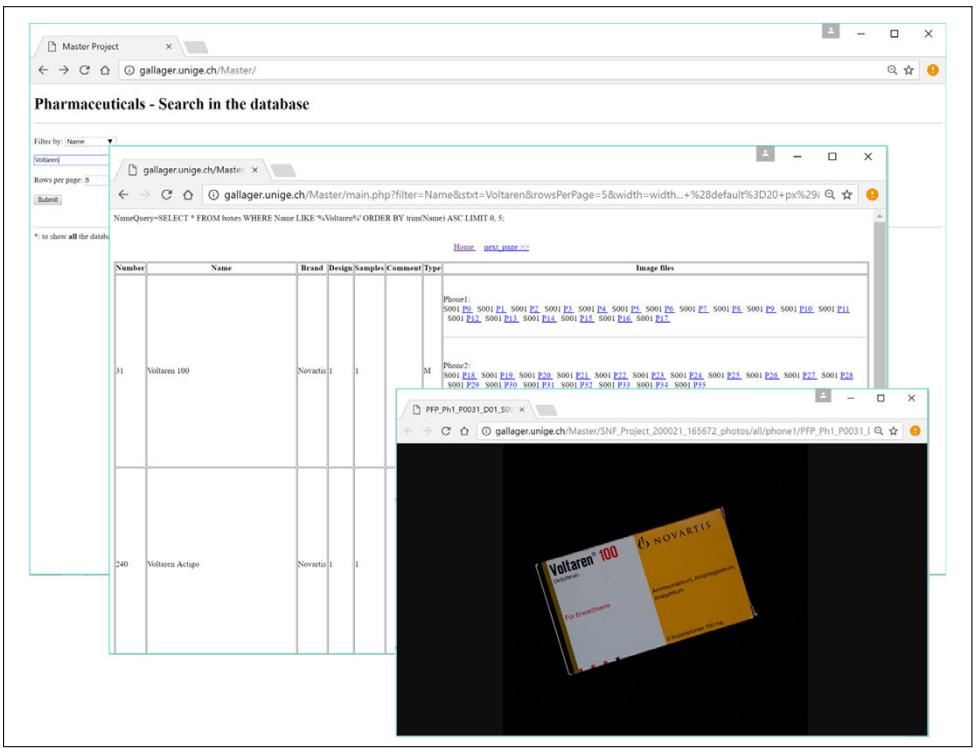

FIGURE 2.19: Sample database search. This is a sequence of 3 screenshots to show the process at time t, t+1, t+2 (submit filled form to server, choose thumbnail or link from results page, display image).

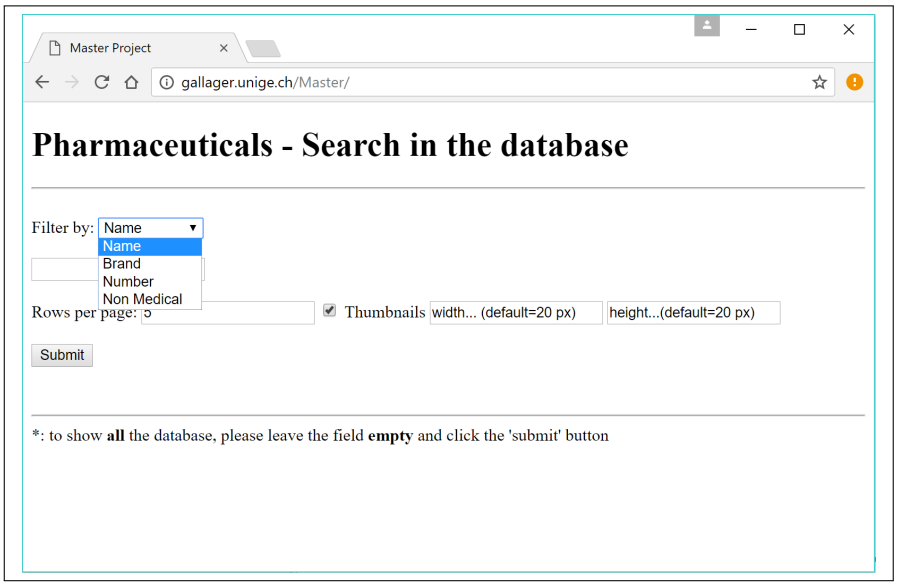

FIGURE 2.20: Search options in the Web interface.

2.8 Private cloud configuration and phone synchronization

To improve performance, a ”private cloud” was installed to automatically synchronize the 3 mobile phone’s internal memory at a regular frequency with a hard drive equipped with a Wi-Fi network card.

The three mobile phones used in this project were connected with private static IP addresses to the private network and an application for FTP (File Transfer Protocol) synchronization was installed on each of them in order to allow automatic transfer of files using the FTP protocol.

FIGURE 2.21: The Western Digital MyPassport ”smart” hard drive.

The ”private cloud” was password protected and accessible through a local IP address of the type **192.168.X.Y**. We should add that access to the Internet was possible within this network through sharing and bridge configuration of the Wi-Fi card integrated in the hard drive.

Technical details regarding applications are the following:

- Phone synchronization application: FolderSync (OS: Android);
- storage and network hardware: Western Digital MyPassport Wireless (1 TB);
- synchronization frequency: 5 minutes (minimum allowed by application).

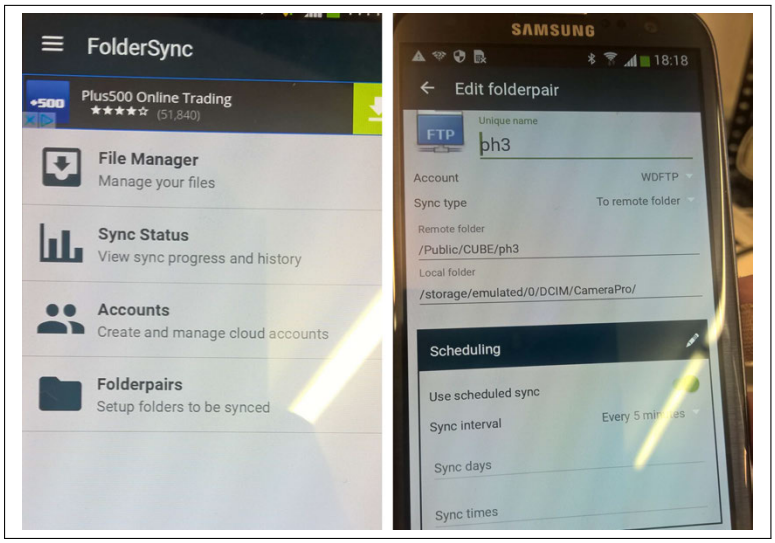

FIGURE 2.22: Configuration of the "FolderSync" app. on Android.

Finally, it can be added that a NAS (Network Attached Storage)[11] could have been an interesting option instead of a basic hard drive with integrated Wi-Fi capability. A NAS would provide Ethernet connectivity for faster bandwidth, larger storage capabilities, and more features (such as automated backups). However, this obviously comes at a higher financial cost. Moreover, an Android application with faster synchronization capabilities would be preferable in order to have an instantaneous preview of images, for an improved enrollment performance.

[11] https://en.wikipedia.org/wiki/Network-attached_storage

Chapter 3

Package identification based on geometrical matching of local descriptors

3.1 Introduction

In this chapter, we present experimental results achieved with the use of the RANSAC [77] (Random Sample Consensus) algorithm integrated with OpenCV[1] functions to find homographies aligning matching keypoints (via their SIFT [24, 25] descriptors) from pairs of testing and training set images.

As opposed to the Bag of Words (BoW) model approach, which does not keep geometrical information, we have used an approach based on RANSAC to find a homography, given the geometrical information contained in SIFT keypoints. Homographies that are able to align a good percentage of keypoints will be considered as having been able to find a decent geometrical consistency. Moreover, we will consider some properties of the homography matrix to further discard some wrong (degenerate) matches. Indeed, as the results will show, a considerable improvement is achieved with these additional tests.

3.1.1 The RANSAC algorithm

The Random Sample Consensus algorithm, first presented in [77], is an algorithm which estimates parameters of a mathematical model in order to find the best approximation of the parameters so that the model fits a dataset containing both inliers and outliers.

One of the simplest examples, which can be given is probably the problem of fitting a line to a cloud of points. Figure 3.1 illustrates such a problem, where we have a dataset (cloud of points) containing outliers.

We can see in Figure 3.1, that the standard "Least Squares" method, which finds the line having the lowest quadratic distance to all points, produces a very bad fitting. Indeed, the **outliers** in this case are the cause of this bad fit. However, RANSAC is able to detect them (shown as red squares in Figure 3.1) and outputs the parameters of the line y=mx+b, which fits best the dataset while ignoring the outliers. RANSAC takes iteratively a sample of a few random points and fits a line to it. Then, it estimates the number of outliers given a threshold. At the end of a fixed number of

[1] https://github.com/kyamagu/mexopencv

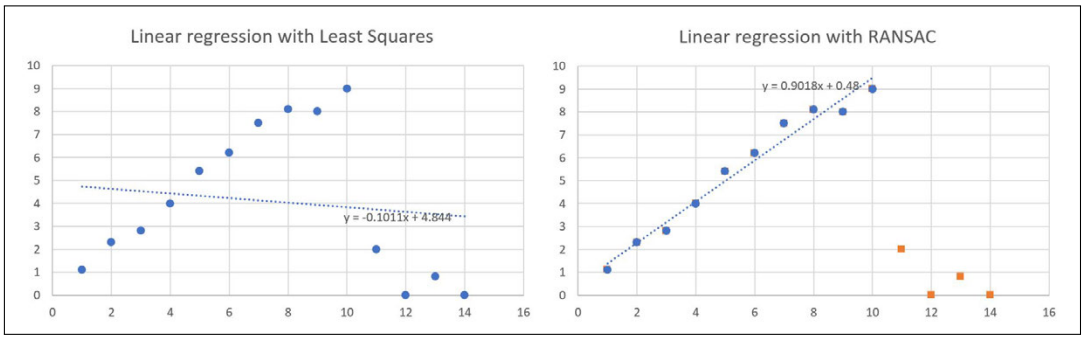

FIGURE 3.1: Fitting a line to a cloud of points containing outliers. Example inspired from [77] .

iteration the best model (with the lowest number of outliers) is returned.

We will use this algorithm to find the best homography aligning two images given a set of matching keypoints. Therefore, we are not searching for the parameters of a line (slope, y-intercept). What we are looking for, are the best values of a projective transform characterized by a homography matrix. RANSAC is well suited for this problem, as it is a general algorithm to find a mathematical model given outliers in a dataset.

3.1.2 Homography

To summarize, a *homography* is a geometrical transformation that maps points in a source image I_{src} to the corresponding points in a transformed image J_{dst}. With $\underline{H}$ being a transformation which can be represented by a matrix of scalar values. In the scope of this project, given two images of possibly matching packages, we try to *find* the *homography matrix* $\underline{H}$, which aligns both images. This is done by using the geometrical information contained in SIFT keypoints. Indeed, each of the latter contains (x, y) image coordinates. Ideally, if there is a match, then the images should be well aligned, whereas if they are dissimilar, no consistent alignment should be found. In Figure 3.2 we can see an example of alignment[2] of a test set image to an enrollment image.

The homography matrix for the alignment of the *test* image to the *enrollment* one, in Figure 3.2, is the following:

$$\underline{H} = \begin{pmatrix} -0.9 & -0.1 & 857.4 \\ 0.1 & -0.9 & 1185.3 \\ 0.0 & 0.0 & 1.0 \end{pmatrix}. \tag{3.1}$$

Homogeneous coordinates are used, thus producing a 3×3 matrix. This allows the third column to hold the translation values. The 2×2 sub-matrix $\underline{H}(1:2, 1:2)$ holds the linear transform coefficients (scaling, shearing, rotation) and the last row of $\underline{H}$ holds the projective coefficients, if any. This is an example of a *good* homography, as it can be seen in Figure 3.2. Indeed the source image as well as the target in enrollment set are taken from a frontal point of view, i.e., almost no projection and the scale is substantially the same (i.e., close to 1). We can see that the absolute value of the diagonal elements of $\underline{H}(1:2, 1:2)$, which contain the scaling (homothety)

[2]To display the aligned image, the OpenCV function cv.warpPerspective(src,H) is used; it applies the matrix $\underline{H}$ to remap all the keypoints from *src* to align it to *dst*.

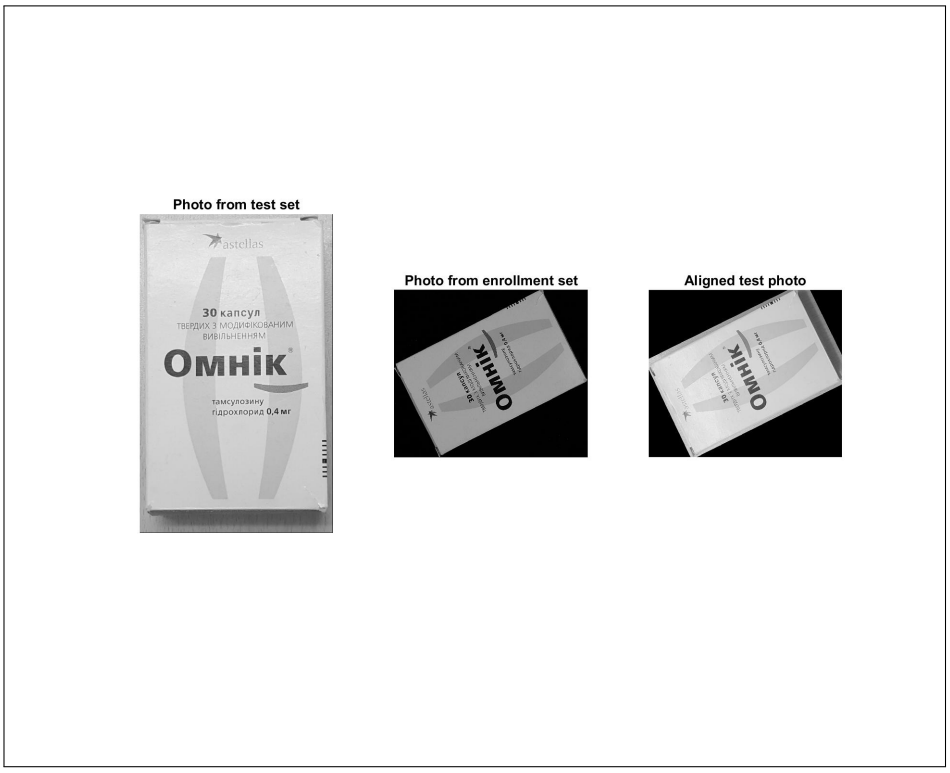

FIGURE 3.2: Alignment with homography. ${p_i}^{Enr} \approx H * {p_i}^{Test}$

coefficients, have a value of 0.9. A useful indication of this *goodness* is the value of the determinant:

```
det(H(1:2,1:2)) = 0.82
```

According to [78] it should not be negative or above a (context dependent) threshold. Not having any recommended values, thresholds were empirically chosen:

```
th1 < det(H(1:2,1:2)) < th2
```

where

```
th1=0.4 and th2=6
```

.

In this context, one may think of the determinant as being almost the square of the scaling value of the 2×2 linear transform matrix, because it is known that for 2D matrices, it is simply computed as the difference of the product of diagonals. Following (3.1), $det(\underline{H}) = 0.82$. Therefore, it can be remarked that $det(\underline{H}) \simeq (-0.9)^2$. A determinant of 6 (which is the uppermost empirical threshold), would correspond to a scaling factor of $\sqrt{6} \approx 2.45$. That would almost increase the images size by 2.5. This can be already considered as a large scaling factor in the context of comparing images of packages acquired at a roughly similar distance. Indeed, testing set images may only undergo a minor scaling effect, namely to simulate an end user's normal behavior.

In the following sections, the experimental results will show that this strategy indeed works reasonably well.

3.1.3 Overview of the matching process

A schematic description of the matching process can be seen in Figure 3.3. At the end of this process, the percentage of match between pairs of images is estimated. Moreover, histograms and ROC curves are provided to allow performance comparison between the different experiments that were conducted. Furthermore, we will sometimes present pseudo-code fragments (in a Matlab-like syntax) to clarify the description of our experiments.

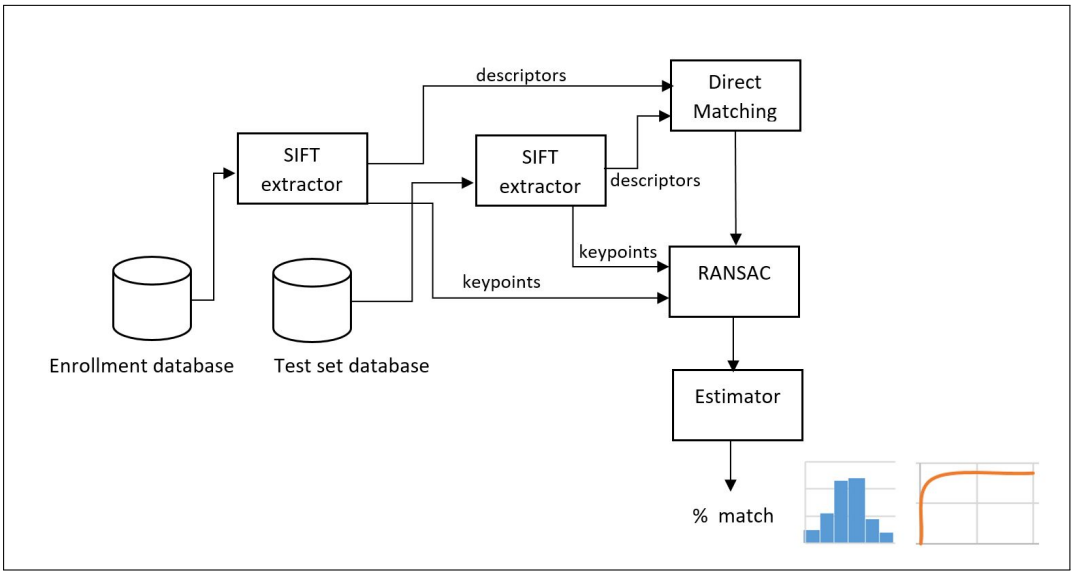

FIGURE 3.3: Diagram showing the matching process with RANSAC.

3.2 Experimental results

In this section, we start by describing the definitions and metrics as well as the algorithms used to perform the experiments. Subsequently, we present the obtained statistics.

3.2.1 Definition of similarity

During the experiments performed in this project, in order to present our results we will refer to "similarity", "semi-similarity" and "dissimilarity" measures. Our definition is the following:

1. similarity: two packages (i.e., 1 testing set package vs. 1 enrollment set package) are considered to be *similar* only if the dosage of active ingredients, the quantity of medicine doses (e.g., pills, tablets, etc.), the pharmaceutical company producing the product are *identical*. Moreover, both packages have the *same graphical design* and the enrollment set image is shot from a *frontal* point of view (i.e., from Phone1);

2. semi-similarity: specifies that two packages are "similar" (in the sense of the previous definition) but with the enrollment-set image having undergone a projection (i.e., shot by Phone2 or Phone3, e.g., Figure 2.4);

3. dissimilarity: any pair of packages that satisfies neither of the two previous definitions (e.g., the two packages in Figure 1.1 are considered "dissimilar" because of the dosage difference[3]).

We recall that the enrollment database contains images of 1000 unique packages, and that the enrollment setup entails 3 phones taking each 18 images per package. Therefore, a total of 54 images per package is achieved. We recall that among those 54 images, 18 are frontal (Phone1), and 36 are projected (Phone2+Phone3). Subsequently, the enrollment database contains a total of 54000 images.

The same similarity definitions are used in all the experiments performed. Therefore, the histograms and ROC (Receiver Operating Characteristic) curves presented in the experimental results will all refer to those definitions.

3.2.2 Data sets

The testing sets **PharmaPack_R_I_S1** and **PharmaPack_R_I_S2** consist both of **300** packages in 8 different positions: rotated, projected and slightly scaled, for a total of 300*8=**2400 images**.

The training set is a subset taken from the **PharmaPack** database (enrollment set[4]). It consists of the 300*54=16200 matching[5] images. Moreover, the training set contains 300 randomly chosen images among the subset[6] of dissimilar ones. That

[3] Although they seem *similar* (in the sense of having only *minor* differences), at first glance, from the point of view of an external observer.

[4] Phone1 takes photos of a frontal (normal view) above the packages. Phone2 takes severely projected photos and Phone3 much less projected but not frontal.

[5] The term "matching" is used in the sense of providing the ground truth for the comparison of similar and semi-similar packages.

[6] (1000*54 enrollment images) - (300*54 matching images) = 37'800 dissimilar images.

is to say, the 300 randomly chosen images have the constraint to be dissimilar with regards to *all* images in the testing set (cf. Figure 3.6–Figure 3.8).

All the images in the enrollment set are cropped. The training set keypoints and descriptors are extracted from these.

The testing set consists of both cropped and non-cropped images. It is interesting to have both, considering that an end user might shoot photos on a complex background, for which the image might be difficult to crop[7].

We must add that the **PharmaPack_R_I_S2** contains images of hand-held packages, whereas **PharmaPack_R_I_S1** consists of images of packages placed on a wooden table background (fixed position). Henceforth, we will refer to these testings sets as **S1** and **S2**.

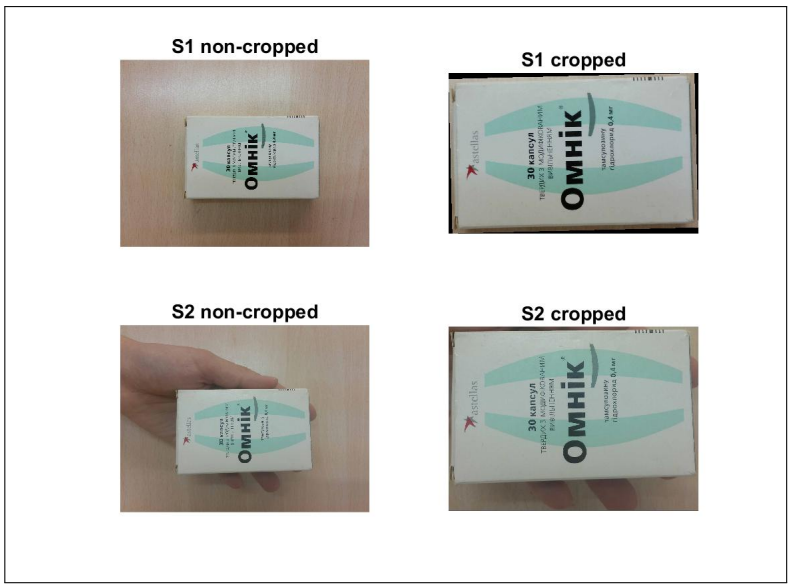

FIGURE 3.4: Example of cropped and non-cropped for datasets **PharmaPack_R_I_S1** and **PharmaPack_R_I_S2**.

[7]However, in the scope of the deployment of a mobile phone app, this should be discouraged for end users, as a complex background will most likely produce descriptors unrelated to the package and therefore increase the probability of incorrect identification.

FIGURE 3.5: Example of images from the recognition (testing) datasets **PharmaPack_R_I_S1** (left) and **PharmaPack_R_I_S2** (right). Image from [71].

3.2.3 Description of experiments

The comparison setup is illustrated in Figure 3.6, Figure 3.7 and Figure 3.8. Please note, that the average number of descriptors/image (`avg_nbDescriptors`) was arbitrarily set to 1000. This was done so to provide a lower bound on the number of comparisons. In practice it is most likely much larger (e.g., for a peakThresh=0.01 parameter, the number of descriptors/image can go as high as 4000, see subsection 3.2.5), yet this lower bound provides a general idea: the order of magnitude is in **billions** of descriptor comparisons.

As we can observe from the aforementioned figures, a huge number of comparisons is needed because of the high amount of descriptors per image. Therefore, in the scope of this thesis, it was not feasible to perform the experiments on the whole database for complexity reasons (CPU time and memory). However, considering that if we could achieve good performances with the setup described in Figure 3.6, Figure 3.7 and Figure 3.8, we might hope to get reasonably close ones for the full scale system.

3.2.4 Algorithmic approach

During our experiments, we present the statistics obtained for *similar*, *semi-similar* and *dissimilar* classes of packages (cf. Figure 3.6 and Figure 3.8). The approach consists of:

- extracting descriptors and keypoints for all images of the datasets;
- finding matching descriptors (and their corresponding keypoints) by brute force[8] using the pairwise Euclidean distance;
- using RANSAC, finding a homography matrix that aligns a maximum of keypoints;
- computing the percentage of inliers (w.r.t. homography).

It should be mentioned that for the direct matching (i.e., brute force matching), we have used "Lowe's criterion" [25], which considers a match, if the ratio of the second best match to the first is larger than a given threshold (thresh=1.5).

The following code snippet illustrates how an homography is found using the OpenCV libraries:

```
[H,mask]=cv.findHomography(src,dst,'Ransac','RansacReprojThreshold',3);
```

where $\underline{H}$ contains the homography matrix and **mask** contains a list of inliers. The same methodology has been used for the other experiments, however complementary tests with regards to the determinant of $\underline{H}$ were performed.

When filtering homographies based on the determinant's value, the following constraints were set, thus considering that a homography if valid, only if:

- thresh1 $< det(\underline{H}) <$ thresh2

[8] That is to say, comparing every descriptor vector from the source image to every descriptor vector of the candidate image (we will later refer to this as *direct matching*).

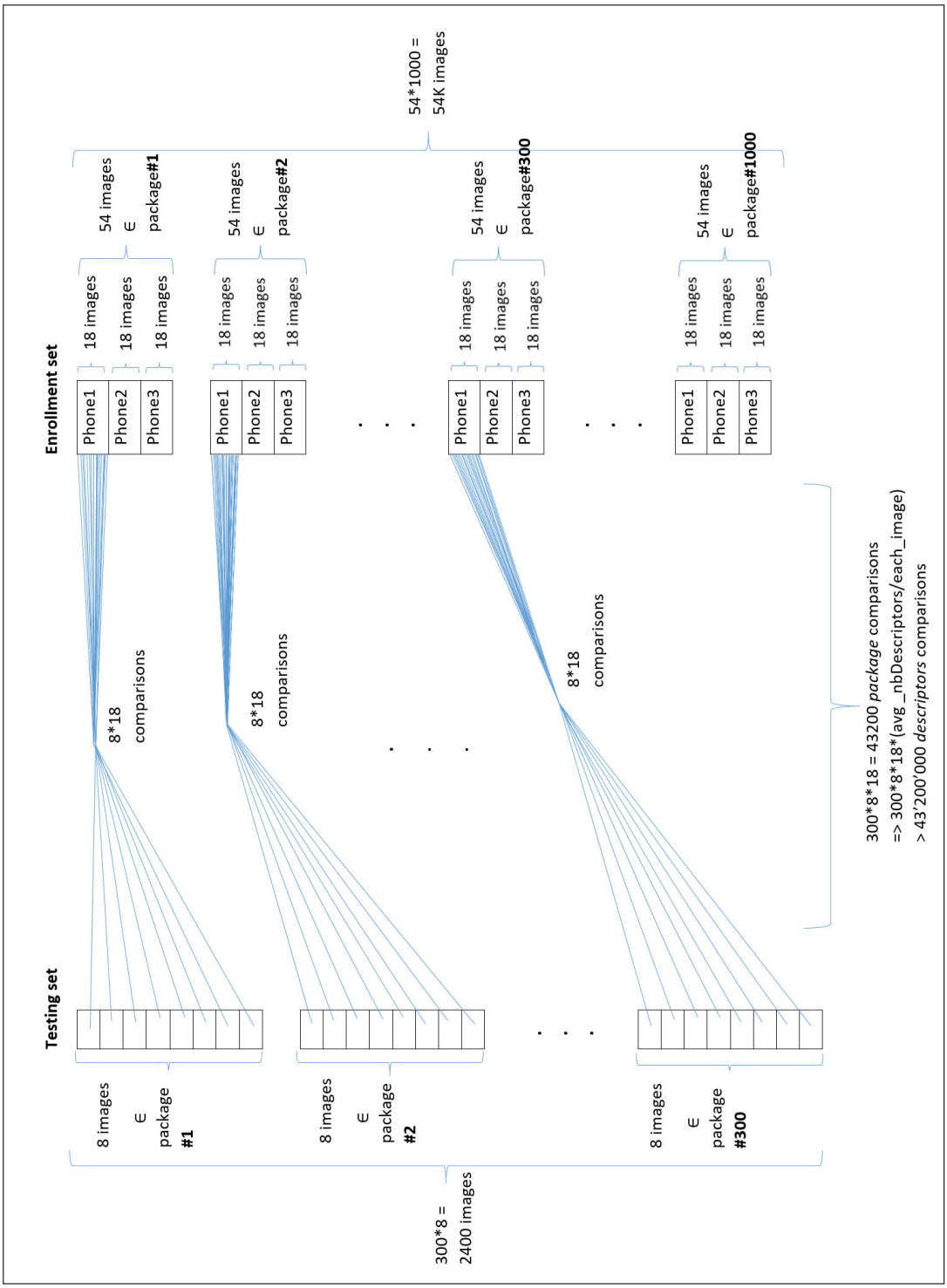

FIGURE 3.6: Comparison setup for **similar** packages.

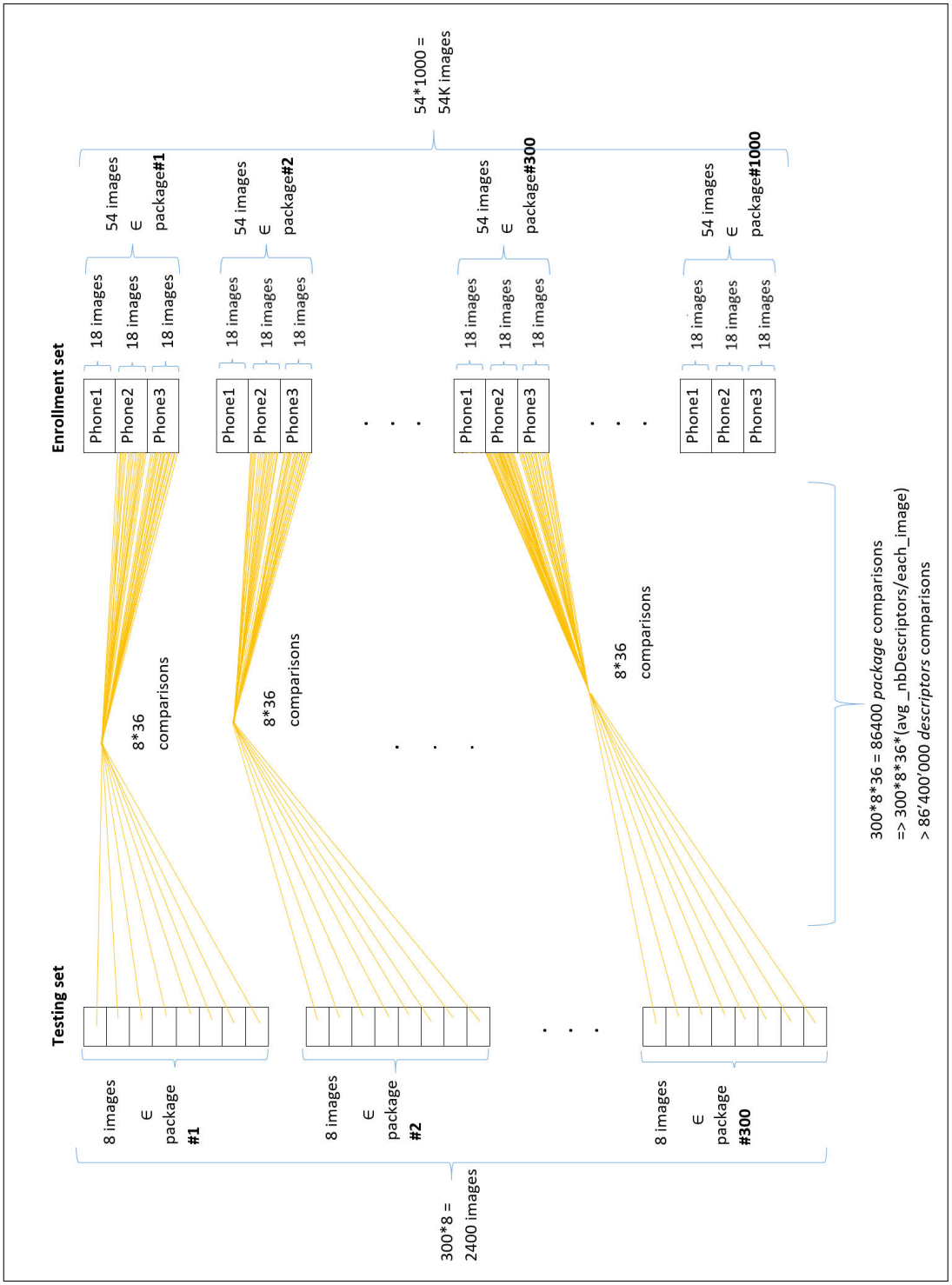

FIGURE 3.7: Comparison setup for **semi-similar** packages.

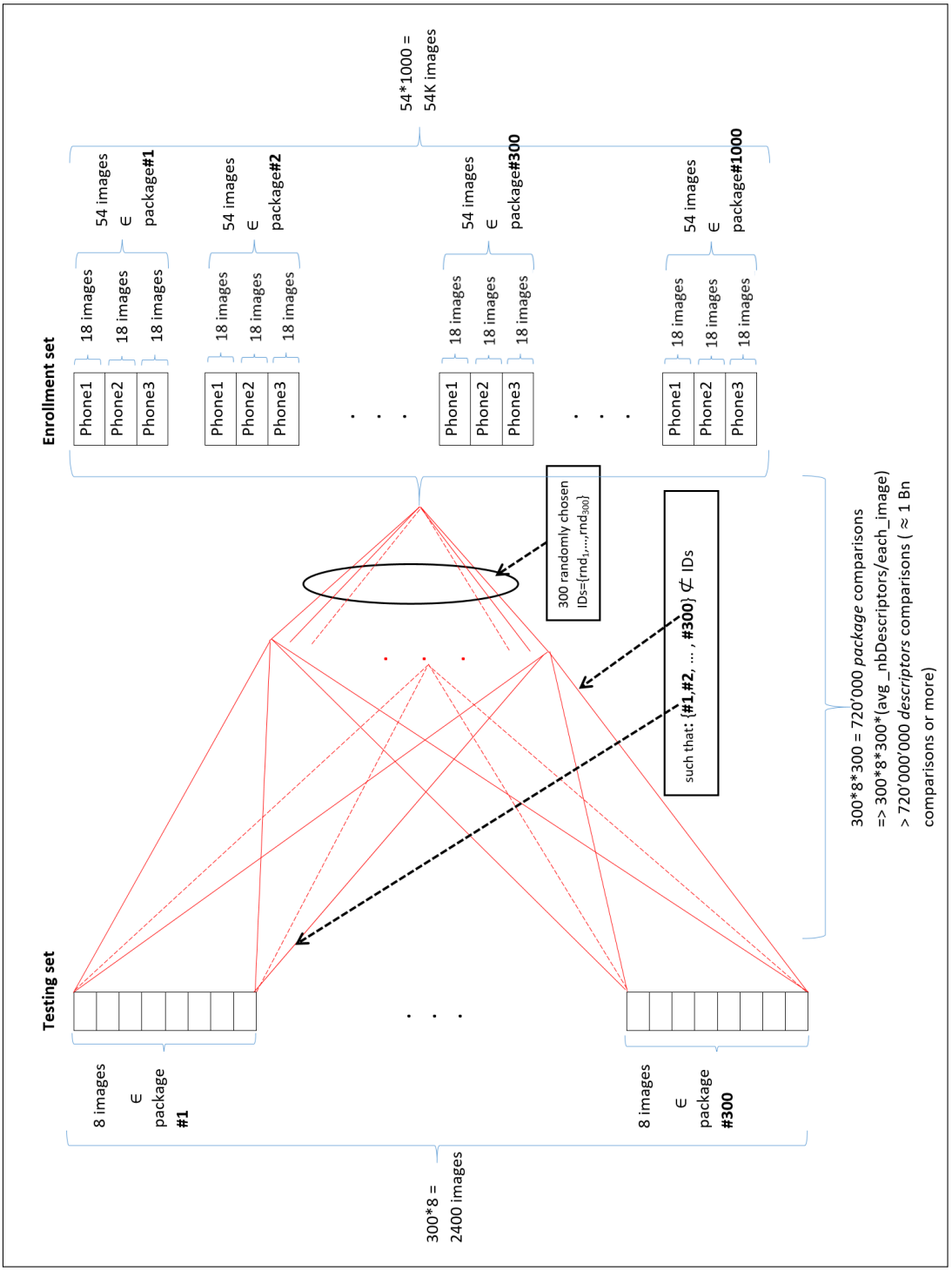

FIGURE 3.8: Comparison setup for **dissimilar** packages.

- isNaN($det(\underline{H})$)= false[9]

with thresh1=0.4 and thresh2=6. In such a way, we ensure that we do not accept homographies that are inconsistent. The thresholds were found empirically although we knew that it should not be negative, too large or too small.

3.2.5 Experiment 1 (S1 dataset)

This first experiment was achieved using SIFT descriptors extracted with a peakThresh[10]=0.01, yielding a varying number of descriptors, roughly up to 4000 descriptors/image. Please, note that the exact number of descriptors generated for each image depends also on its contents (e.g., no descriptors would be generated for a hypothetical image in which every pixel would have the same intensity value).

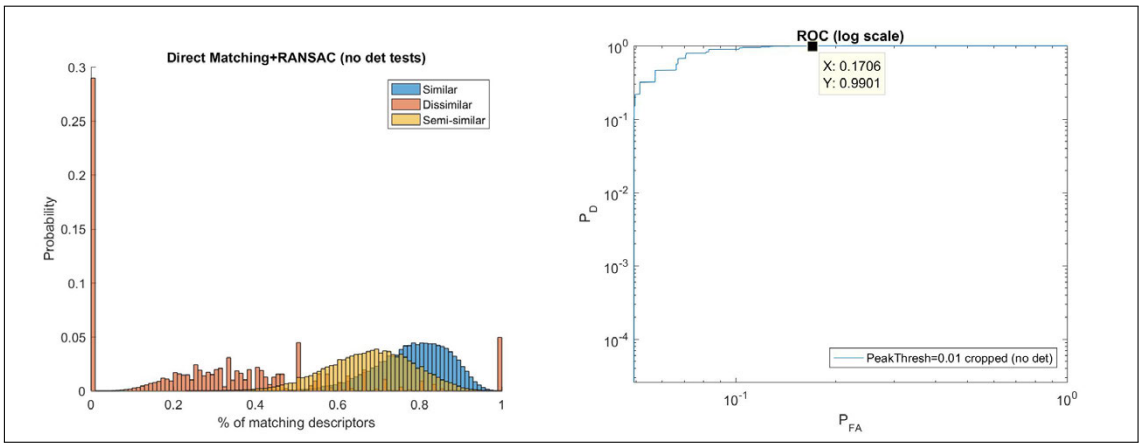

FIGURE 3.9: Separability graph and ROC curve : RANSAC with peakThresh=0.01, **no** constraints for det(H).

"Equal error rate strategy" (EER) [72], i.e., finding $P_{FA} = P_M$[11], the pseudo-code snippet illustrates[12] how we find them:

```
max(find(abs(X-(1-Y))<10^-3)) = index
1-Y(index) = PM
X(index) =  PFA
```

"Neyman-Pearson strategy" to find several P_M for fixed $P_{FA} = 10^{-1}, 10^{-2}, 10^{-3}$:

```
max(find(X<10^-1))=index
Y(index) = PD
1-Y(index) = PM

max(find(X<10^-2))=index2
Y(index2)=PD2
1-Y(index2)=PM2

...
```

[9] NaN: "Not a Number" can happen if there is a division by 0 or a number too large or considered close to infinity.

[10] PeakThresh: peak threshold parameter controlling the sensitivity of the built-in SIFT detector.

[11] Probability of false acceptance as close as possible to probability of miss, we are not always able to have them exactly equal

[12] Given that we use here a function that returns us the true positives in Y and the false positives in X.

It is also possible to read the values directly on the ROC curve. It can be observed from the latter that a P_{FA} much lower than 4.9% cannot be achieved. However, as we can see in Figure 3.10, the P_M=84% (i.e., P_D = 16%) is unacceptable. Therefore, in our experiments, instead of the classical "Neyman-Pearson" approach, we will give P_{FA} values for fixed P_M values. The latter can be read on the ROC curves. Alternatively, they can be found by $min(find((1 - Y) == 0))$, in Matlab.

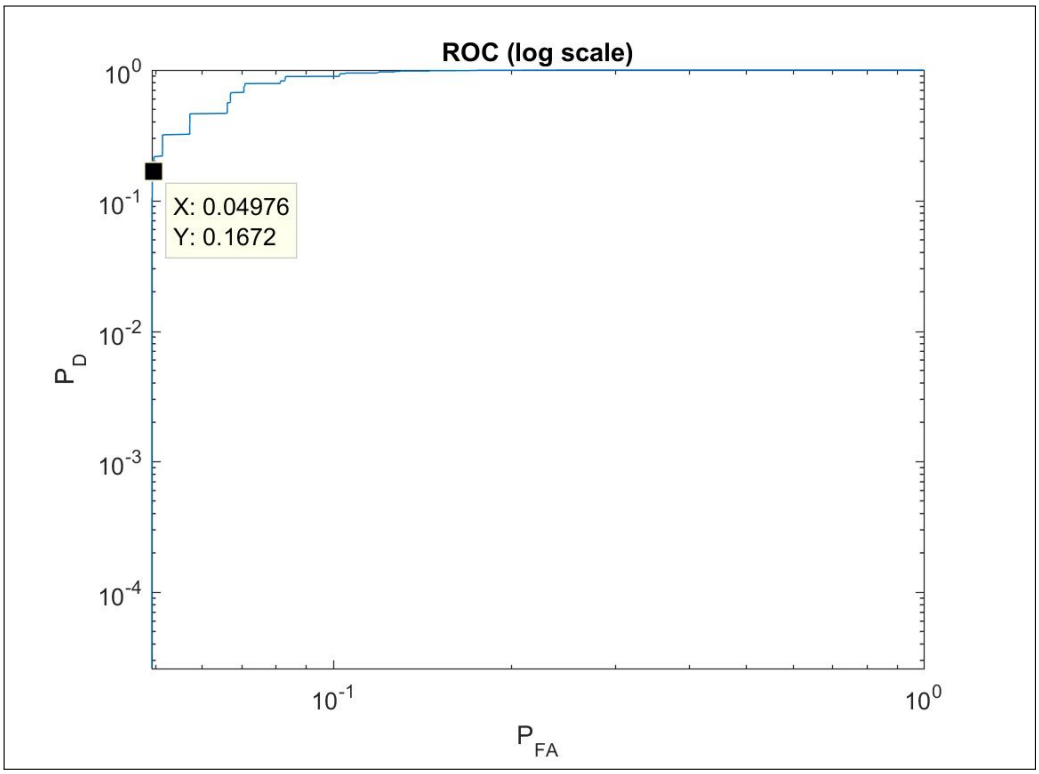

FIGURE 3.10: ROC curve readings.

TABLE 3.1: Summary of statistics for Experiment 1. Probabilities given for the EER strategy (P_{FA}=P_M).

P_{FA}	P_D	P_M
10.26%	89.74%	10.26%

TABLE 3.2: Experiment 1: statistics in % for several values of probability of miss.

Prob. of miss P_M	Prob. of false acceptance P_{FA}	Prob. of correct detection P_D
0%	35%	100%
0.01%	30%	99.99%
0.1%	23%	99.9%
1%	17%	99%
5%	11%	95%
10%	10%	90%
95%	4%	5%

3.2.6 Experiment 2 (S1 dataset)

This experiment was performed using SIFT descriptors extracted with peakThresh=0.01 as before. However, all filtering tests on the matrix determinant were applied this time.

A better separability of the *similar* and *dissimilar* distributions can be observed. However, the *semi-similar* distribution almost completely overlaps the *similar* one. Therefore, as the two latter distributions cannot be distinguished, we will henceforth ignore the *semi-similar* one. Subsequent results will therefore be presented with regards to *similar* vs. *dissimilar* distributions.

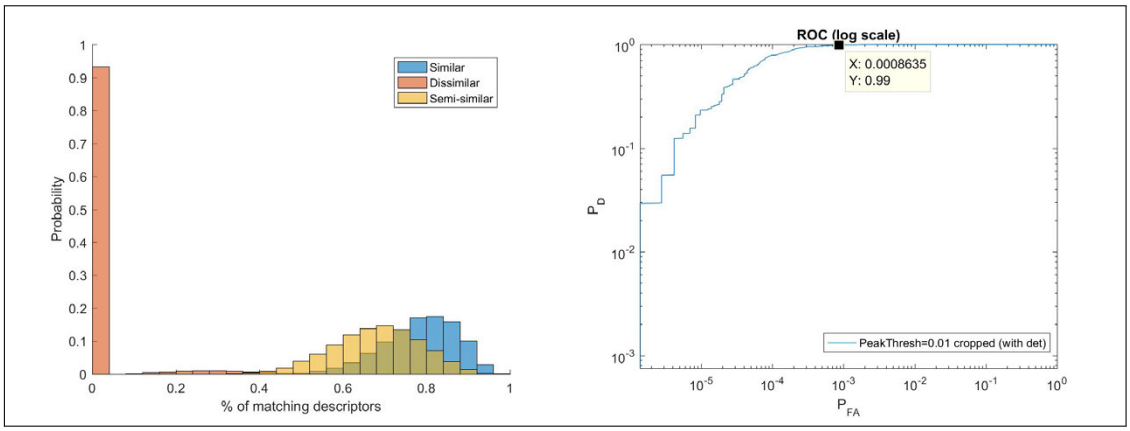

FIGURE 3.11: ROC curve and separability graph for experiment 2: RANSAC with peakThresh=0.01 and **all constraints** applied w.r.t. det(H).

TABLE 3.3: Summary of statistics for Experiment 2. Probabilities given for the EER strategy (P_{FA}=P_M).

P_{FA}	P_D	P_M
0.26%	99.75%	0.25%

TABLE 3.4: Experiment 2: statistics in % for several values of probability of miss.

Prob. of miss P_M	Prob. of false acceptance P_{FA}	Prob. of correct detection P_D
0%	1.7%	100%
0.01%	1.1%	99.99%
0.1%	0.5%	99.9%
1%	0.08%	99%
5%	0.03%	95%
95%	0.00027%	5%

3.2.7 Experiment 3 (S1 dataset)

For this experiment we used SIFT descriptors extracted with a custom peak threshold yielding up to 900-1000 descriptors/image. All tests with regards to det(H) were used.

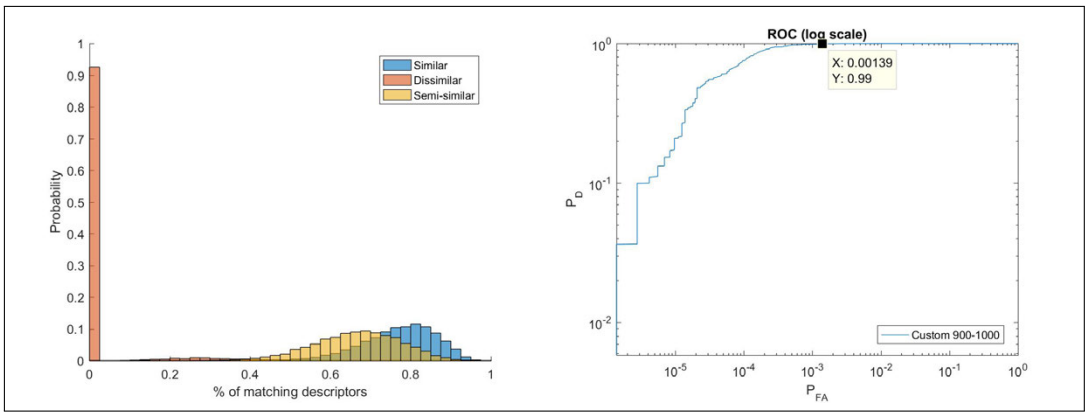

FIGURE 3.12: ROC curve and separability graph for experiment 3: RANSAC (900-1000 descriptors/image).

TABLE 3.5: Summary of statistics for Experiment 3. Probabilities given for the EER strategy (P_{FA}=P_M).

P_{FA}	P_D	P_M
0.32%	99.69%	0.31%

TABLE 3.6: Experiment 3: statistics in % for several values of probability of miss.

Prob. of miss P_M	Prob. of false acceptance P_{FA}	Prob. of correct detection P_D
0.05%	2.9%	99.95%
0.1%	0.85%	99.9%
1%	0.14%	99%
5%	0.04%	95%
95%	0.00027%	5%

3.2.8 Experiment 4 (S1 dataset)

For this experiment we used SIFT descriptors extracted with a custom peakThresh to have 500-600 descriptors/image. All homography determinant tests were performed. The probability of detection $P_D = 99.8\%$ with a $P_{FA} = 1.35\%$ was the best achievable in this experiment and is shown in the first row of Table 3.8.

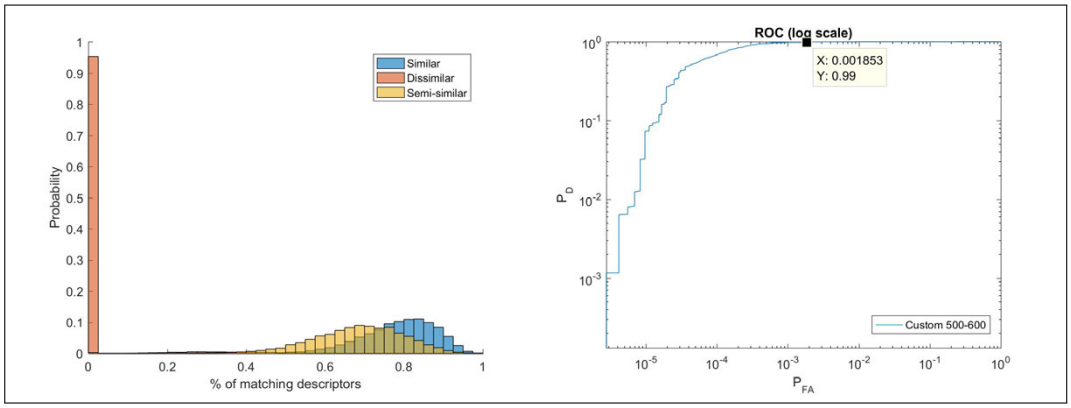

FIGURE 3.13: ROC curve and separability graph for experiment 4: RANSAC (500-600 descriptors/image).

TABLE 3.7: Summary of statistics for Experiment 4. Probabilities given for the EER strategy (P_{FA}=P_M).

P_{FA}	P_D	P_M
0.41%	99.58%	0.42%

TABLE 3.8: Experiment 4: statistics in % for several values of probability of miss.

Prob. of miss P_M	Prob. of false acceptance P_{FA}	Prob. of correct detection P_D
0.2%	1.35%	99.8%
1%	0.18%	99%
5%	0.04%	95%
95%	0.00097%	5%

3.2.9 Experiment 5 (S1 dataset)

For this experiment we used SIFT descriptors extracted with a custom peakThresh in such a way to have 300-400 descriptors/image. All tests on the determinant were performed. We should add that, as shown in Table 3.10, no higher correct detection probability than $P_D = 99.2\%$ was achievable.

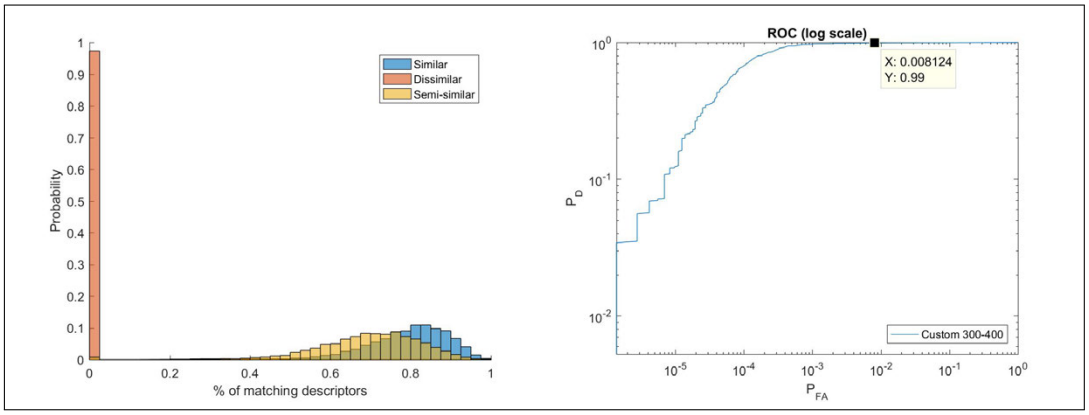

FIGURE 3.14: ROC curve and separability graph for experiment 5: RANSAC (300-400 descriptors/image).

TABLE 3.9: Summary of statistics for Experiment 5. Probabilities given for the EER strategy (P_{FA}=P_M).

P_{FA}	P_D	P_M
0.95%	99.04%	0.96%

TABLE 3.10: Experiment 5: statistics in % for several values of probability of miss.

Prob. of miss P_M	Prob. of false acceptance P_{FA}	Prob. of correct detection P_D
0.8%	2.09%	99.2%
1%	0.8%	99%
5%	0.06%	95%
95%	0.00027%	5%

3.2.10 Direct matching experiment (S1 dataset)

In this section we briefly present the results of direct matching (DM) **without using geometrical information** of keypoints. In this experiment, the number of matching descriptors for the *similar*, *semi-similar* and *dissimilar* classes are compared using Lowe's criterion for filtering unreliable matches. The statistics obtained in Table 3.11, show a relatively high probability of false acceptance of 4.26%, given a 100% correct detection ($P_M = 0\%$). Furthermore, a $P_{FA} = 2\%$ is obtained with a 0.1% probability of miss. These results should be compared with those given in Table 3.4 in order to observe the advantage provided by RANSAC alignment over DM alone. Indeed, RANSAC based matching roughly halves the probability of false acceptance compared to DM.

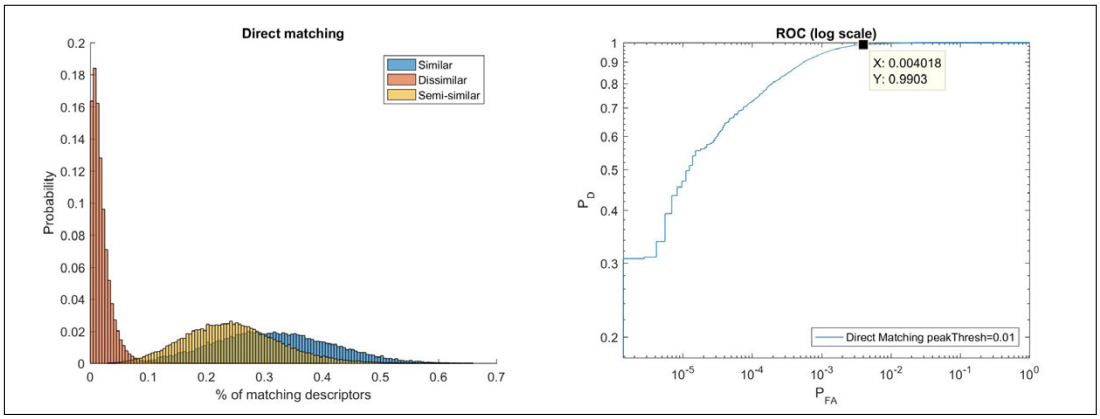

FIGURE 3.15: ROC curve and separability graph for direct matching (no geometrical information).

TABLE 3.11: Direct Matching: statistics in % for several values of probability of miss.

Prob. of miss P_M	Prob. of false acceptance P_{FA}	Prob. of correct detection P_D
0%	4.26%	100%
0.01%	3.5%	99.99%
0.1%	2%	99.9%
1%	0.4%	99%
5%	0.1%	95%
95%	0.0012%	5%

3.2.11 Non-cropped images: experiment

This section is dedicated to presenting the obtained statistics for experiments based on **non-cropped** images of the **S1** testing set. The descriptors were extracted with a peakThresh=0.01 and RANSAC was used to find the best homographies as for the previous experiments.

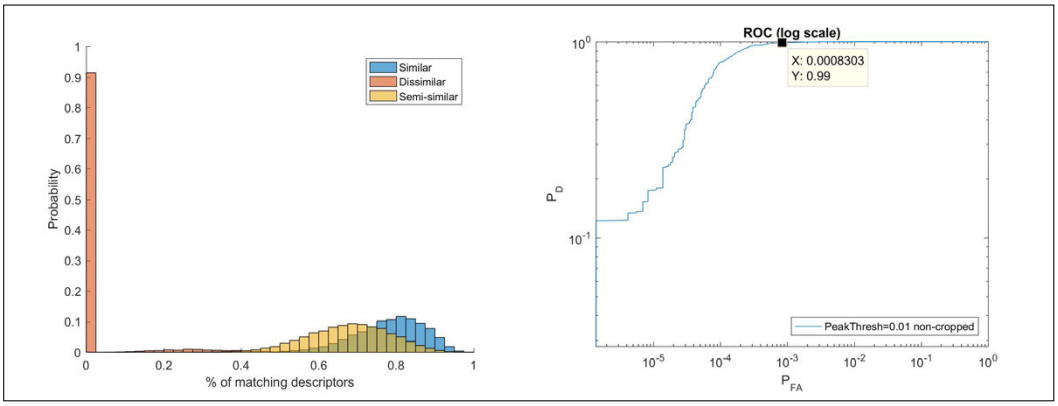

FIGURE 3.16: ROC curve and separability graph for non-cropped experiment: RANSAC with peakThresh=0.01.

TABLE 3.12: Summary of statistics for non-cropped experiment. Probabilities given for the EER strategy (P_{FA}=P_M).

P_{FA}	P_D	P_M
0.28%	99.73%	0.27%

TABLE 3.13: Non-cropped experiment: statistics in % for several values of probability of miss.

Prob. of miss P_M	Prob. of false acceptance P_{FA}	Prob. of correct detection P_D
0.01%	1.3%	99.99%
0.1%	0.57%	99.9%
1%	0.08%	99%
5%	0.029%	95%
95%	0.00013%	5%

We can see that the probabilities of miss and false acceptance increase slightly compared to the cropped version in subsection 3.2.6 , but only by a small amount.

3.2.12 ROC comparisons (S1 dataset)

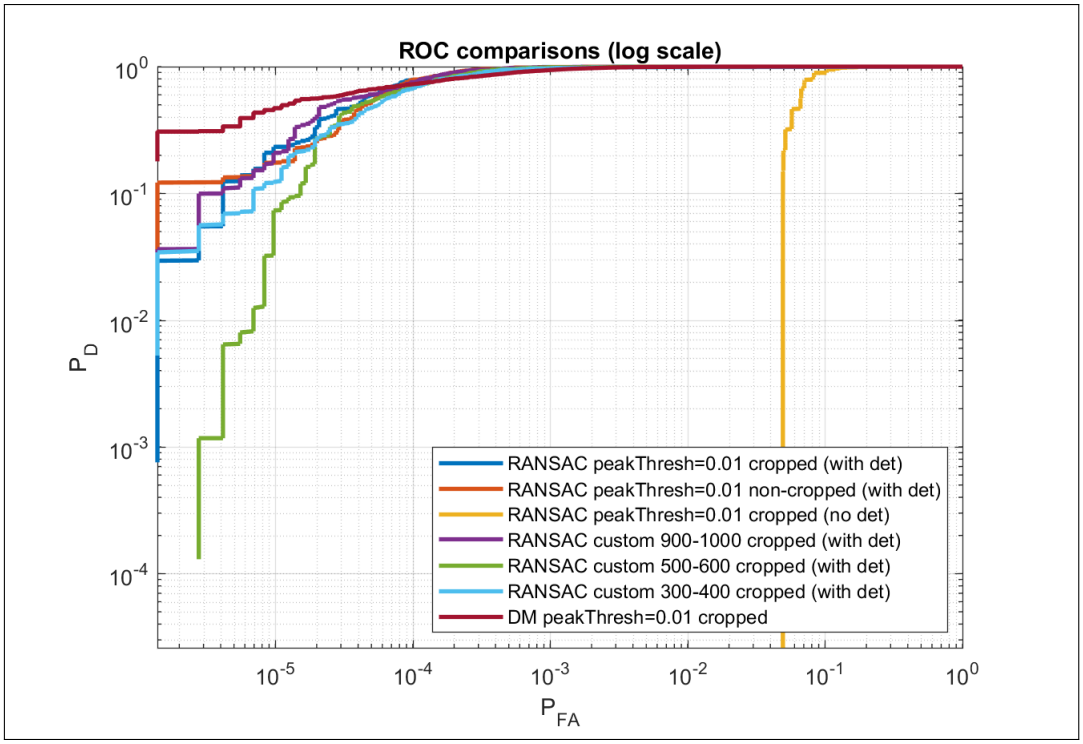

FIGURE 3.17: ROC comparisons for RANSAC local feature matching (S1 dataset).

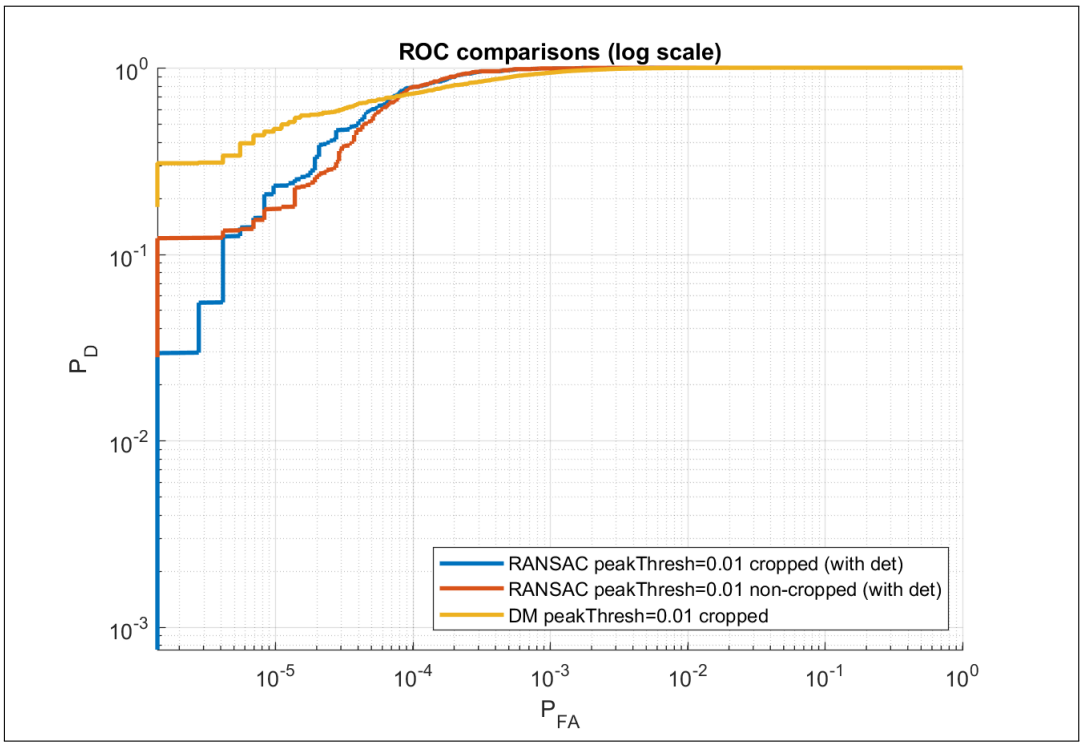

FIGURE 3.18: ROC comparisons for RANSAC local feature matching (S1 dataset). In this figure, the emphasis is on the recognition performance difference between non-cropped and cropped.

3.2.13 Comparison between S1 and S2 datasets

In this section the performance of RANSAC applied to **PharmaPack_R_I_S1** and **PharmaPack_R_I_S2** is investigated. Figure 3.19 shows a superior performance with the S1 dataset. We recall that S2 contains images of hand-held packages whereas in S1, they are in a fixed position. Furthermore, performance regarding non-cropped packages is worse than for their cropped counterpart, although not by much.

Please, note that in the context of this project, the leftmost part the ROC curve, corresponding to low P_D values should not be taken into account, as it would result in too many errors. One should therefore focus on P_{FA} values matching P_D values that are at least above 90%.

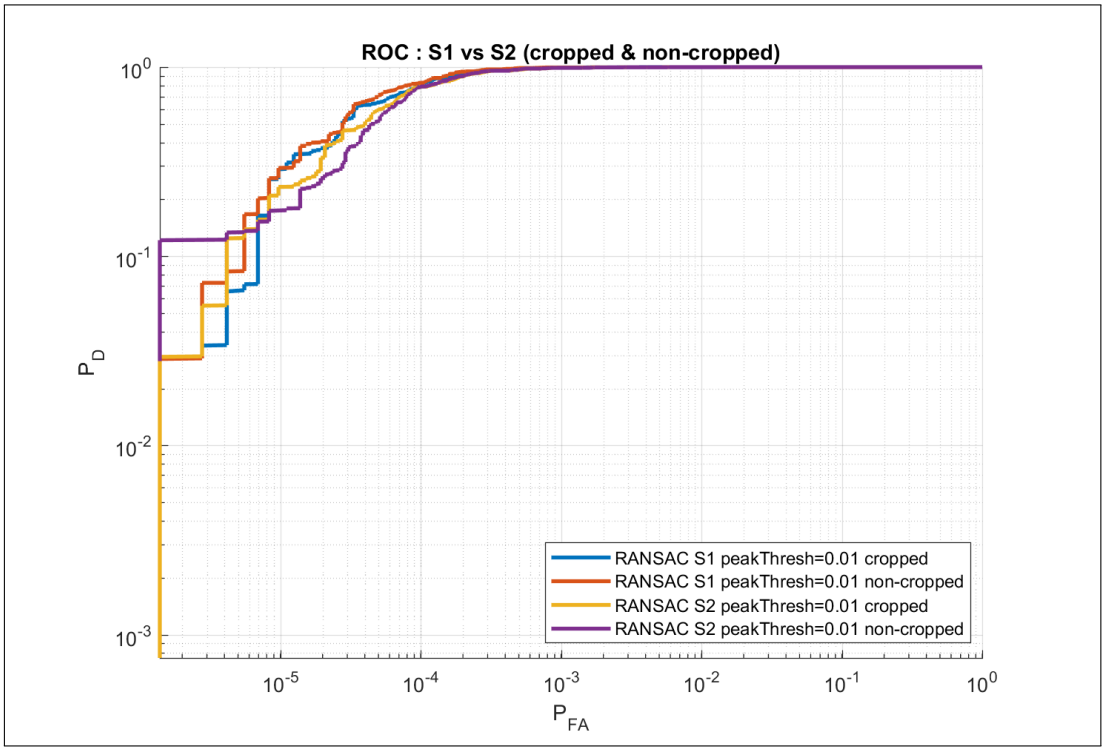

FIGURE 3.19: ROC comparisons.

3.3 Execution times

TABLE 3.14: Execution times for several experiments with up to 4000 descriptors/image (peakThresh=0.01).

Experiment	Exec. time [s]	Exec. time [m]	Exec. time [h]	Number of image comparisons (cf. Figure 3.6 - Figure 3.8)
RANSAC Dissimilar	16667	277.8	4.6	720000
RANSAC Semi-similar	4424.5	73.74	1.23	86400
RANSAC Similar	4162.7	69.37	1.15	43200
DM Dissimilar	6873.8	114.6	1.9	720000
DM Semi-similar	4726.3	78.77	1.3	86400
DM Similar	4351.2	72.52	1.2	43200

3.4 Examples of false positives

In this section, we illustrate some cases of false positives, i.e., for two *dissimilar* packages, a higher percentage of matching descriptors is found than for the true corresponding package. In the examples presented below, two types of false positives can be distinguished: those originating from the comparison of two almost identical packages, i.e., the problem of fine-grained recognition (cf. Figure 3.20), and those stemming from an insufficient descriptor coverage (cf. Figure 3.22).

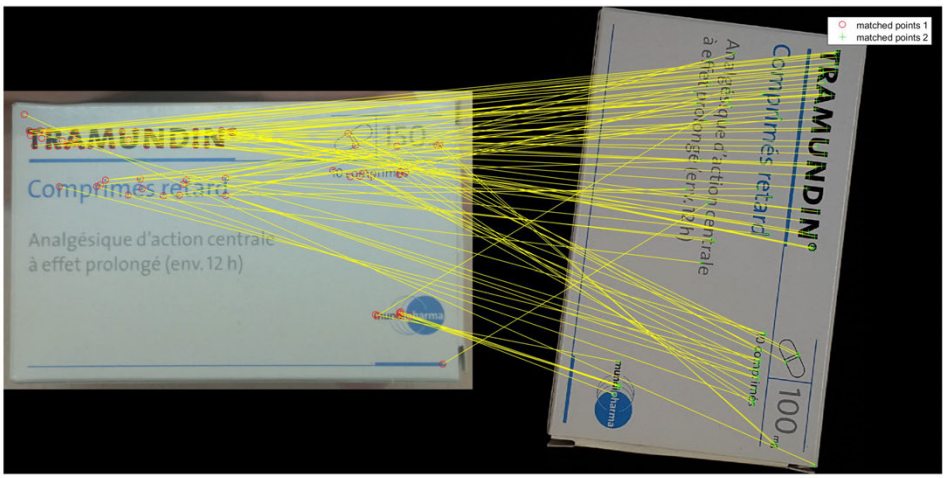

FIGURE 3.20: Example of SIFT matches. The high amount of matching features comes from the fact that these packages are almost identical. Indeed, they differ only by the dosage (150 mg vs. 100 mg). No descriptors were matched on the area corresponding to this information.

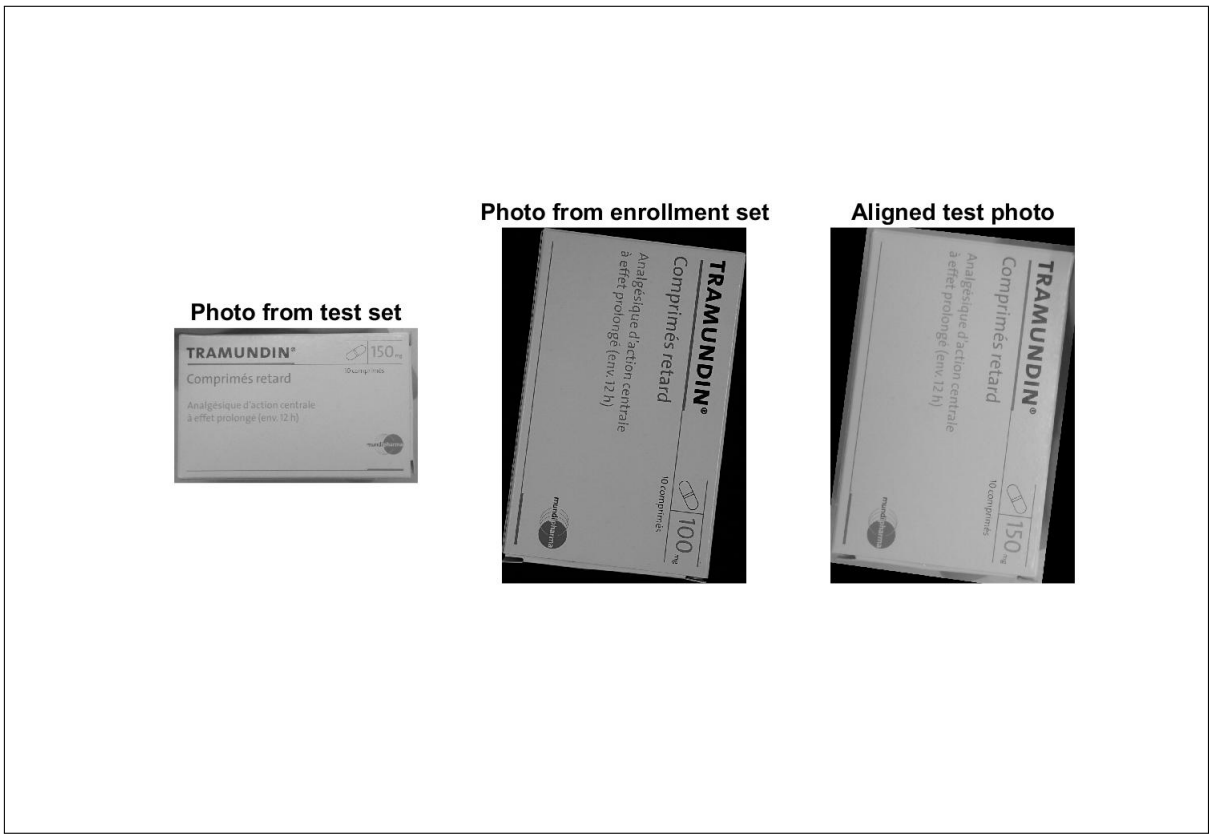

FIGURE 3.21: Geometrical alignment with RANSAC. The testing set image was perfectly aligned by RANSAC to the enrollment set one (middle). The packages are very similar due to only a minor difference in their dosage of the active ingredient.

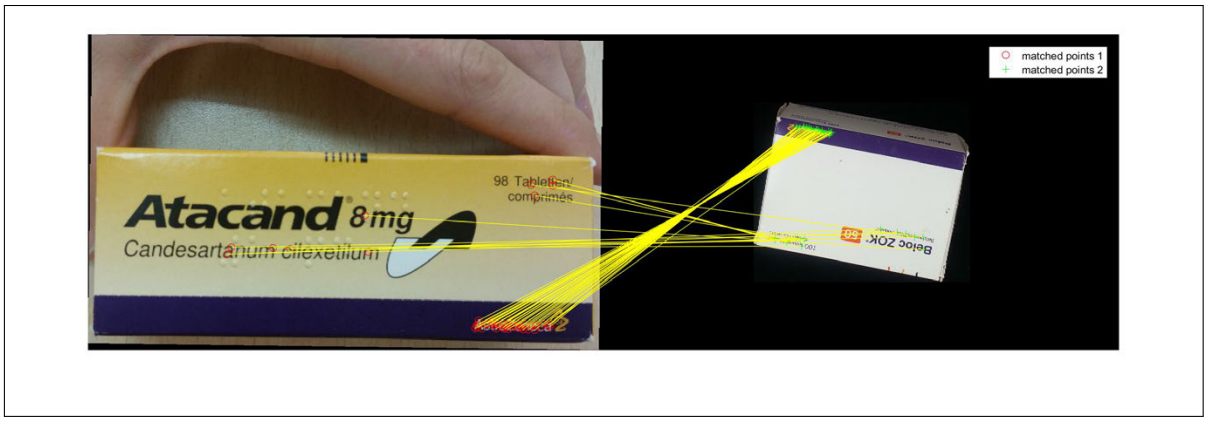

FIGURE 3.22: Example of SIFT matching for dissimilar packages. Correctly matching features result from the similar descriptors found on the pharma company name "AstraZeneca" (on violet strip).

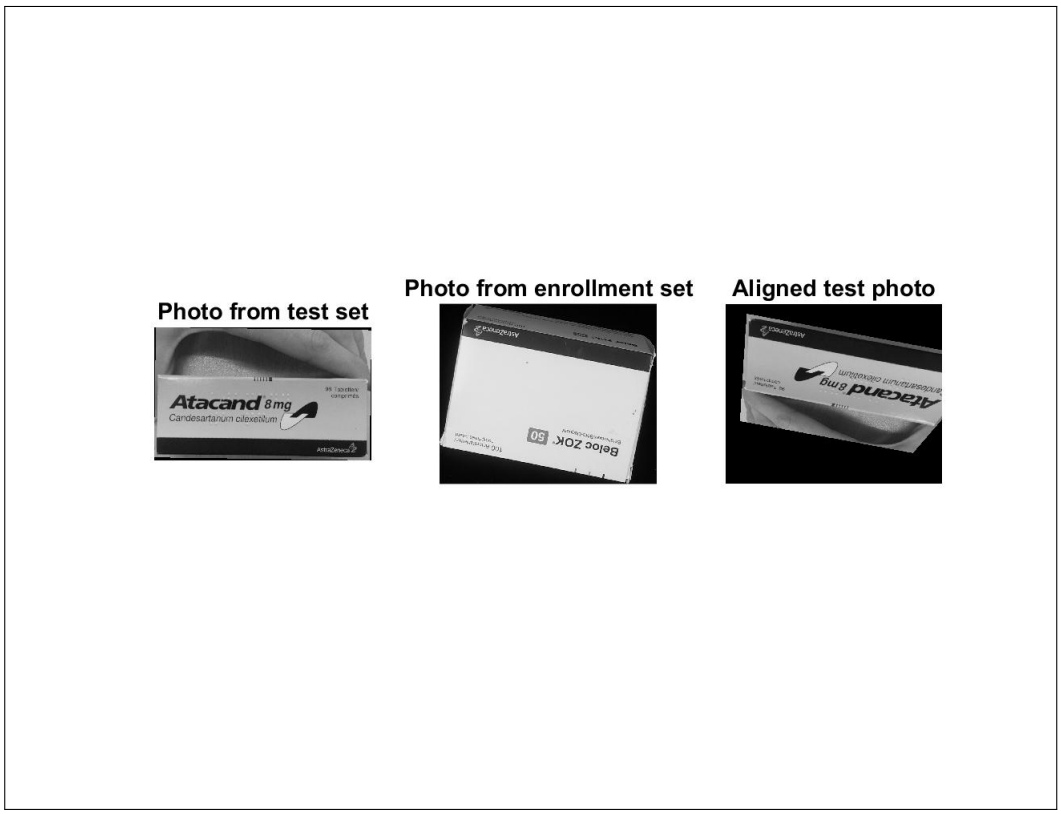

FIGURE 3.23: Geometrical alignment found with RANSAC for dissimilar packages. RANSAC was still able to perform an alignment due to the matching descriptors on the company name. The homography matrix has rotated the test set image by ∼180 degrees to match the text "AstraZeneca".

3.5 Conclusions

From the experimental results, it can be observed that compared to Direct Matching without geometrical information, the approach based on RANSAC provides a significant improvement. We recall that the RANSAC algorithm, preceded by a brute force matching, was used to match keypoints between two images. Given "source points" from an enrolled image and "destination points" from a probe image, RANSAC tries to find the best homography matrix aligning these keypoints. The RANSAC algorithm iteratively takes random subsets of points to achieve the best results.

We can see in Table 3.4, that for 0.01% of miss (i.e., P_D =99.99%), a probability of false acceptance of 1.1% is obtained. This is an improvement compared to the results of direct matching without geometrical information shown in Table 3.11. For the same P_D =99.99% a P_{FA}=3.5% is obtained (i.e., more than 3 times higher). Likewise, with other values of decreasing P_D we have respectively lower P_{FA}.

It is also interesting to notice that a significant improvement can be achieved when the information provided by the **determinant** of the homography matrix is taken into account. Indeed, in Table 3.2 a probability of false acceptance at 30% can be observed, the latter dropping down to 1.1% in Table 3.4 (both for a probability of detection of 99.99%). Moreover, it can be seen (if not taking into account the determinant value), from the separability graphs in Figure 3.9 that the *dissimilar* distribution has many high spikes overlapping the *similar* distribution. Thus, without the determinant tests applied to the homography matrix, they cannot be well separated.

Finally, experiments 3-5 show the results obtained with a lower number of descriptors per image. We have adjusted the peakThresh parameter of the SIFT detector in such a way as to have approximately the number of descriptors in the range stated for each experiment. A higher peakThresh produces a lower number of descriptors. The worst performance can be observed in experiment 5 (300-400 descriptors/image), as shown in Figure 3.17. Nevertheless, it is interesting to investigate the performance with a lower number of descriptors because it reduces complexity.

However, although a P_D=99.99% with a P_{FA}=1.1% can be achieved, the approach consisting of a direct matching followed by RANSAC alignment is not conceivable for a large database. Firstly, because of the huge computational complexity, which is not compatible with the requirements of a fast identification system using a mobile phone application. Moreover, even with the considerable amount of descriptors (up to 4000 descriptors/image) extracted per each image when using a peakThresh parameter at 0.01, a error-less identification is still not achieved.

Some examples of false positives have been presented in section 3.4. The challenge of fine-grained recognition has been illustrated there. By essence, local features are not very discriminative. It is by extracting multiple of such local descriptors that one can experience an advantage in using this kind of approach. Therefore, we can surmise with reasonable confidence, that it is the lack of descriptor coverage which has led to many cases of false positives or misses. However, massively increasing the number of extracted features will lead to dramatic complexity increasing.

As a future improvement, one could investigate the use of other types of more *discriminative* descriptors, such as *non-local* or *semi-local* [79] descriptors. One could therefore investigate the use of the SketchPrint [80] descriptors. To roughly summarize the idea, it uses a description of a line between two keypoints. This produces much more "meaningful" descriptors and thus less of them are needed to characterize an object. However, the keypoint pairs need to be very robust, and a filtering stage is required.

Another possible improvement would be to extract text regions from images and compute descriptors only for these regions. In such a way, we could concentrate only on the most meaningful parts and hopefully construct more discriminative descriptors. In the scope of this approach, a technique such as the "Stroke Width Transform" [81] (SWT), considered to be among state-of-the-art techniques for natural text extraction, could be investigated and tested on the PharmaPack database.

Chapter 4

Package identification without geometrical matching

4.1 Introduction

To investigate a possibility to reduce and identify complexity, we will consider an approach based on descriptors aggregation. Once we have a "short" list of similar packages, the RANSAC matcher might be applied only to this list thus drastically reducing the complexity.

We will present experimental results achieved with the use of the Fisher vectors [82] [83] (FV) with Gaussian Mixture Model [84][85] (GMM) clustering. Fisher vector is a way to encode information about an image by aggregating statistics about local image descriptors (SIFT [25] [24]).

Much like the Bag of Words [86] (BoW) model, or, as one should say, Bag of Visual Words (BoVW), in the case of images, FV produces a single aggregated vector which describes an image and can be compared by Euclidean distance to other FVs, stored, for example, in a database, and determine the closest match. Similarly to BoW, the Fisher vector does not store geometrical information about the SIFT keypoints (x,y location in image). We will show in the experimental results, that this loss of information has an impact on the precision of the representation and therefore on the quality of the results. Moreover, anyway this approach might be of interest for accelerated matching as a pre-stage for RANSAC.

4.1.1 Gaussian mixture model

The Gaussian mixture model (GMM) aims at finding the parameters of K Gaussian distributions that will model the distribution of a cloud of points. The goal is to perform a *soft clustering* as opposed to *hard clustering*, which can be achieved by algorithms such as K-means. With soft clustering, each vector belongs to multiple clusters at the same time but with different probabilities. With an algorithm such as K-means, a hard clustering is performed and each vector belongs to one unique cluster. Using GMM for soft clustering allows us to take into account more subtle situations, where, for example, some vectors might be almost at equal distances to multiple clusters. This approach is known to produce better results than K-means.

More precisely, in this project N 128-dimensional SIFT vectors will be clustered to produce an aggregated descriptor. However, in order to develop an intuition behind the GMM, we present an example with 2D vectors (cf. Figure 4.1, Figure 4.2 and Figure 4.3). It can be generalized to D-dimensional vectors, such as 128-dimensional

SIFT vectors. For the case of multivariate Gaussian distributions, one should keep in mind that they are parametrized by a vector of means $\underline{\mu}$ and a covariance matrix $\underline{\Sigma}$. Indeed the covariance matrix is a generalization of the variance of the univariate Gaussian distribution. A multivariate model is required, as a 128-dimensional SIFT vector can be considered as as a random high dimensional vector, in the context of the GMM. Therefore, we seek a Gaussian mixture that will most faithfully model the distribution of the cloud of random vectors.

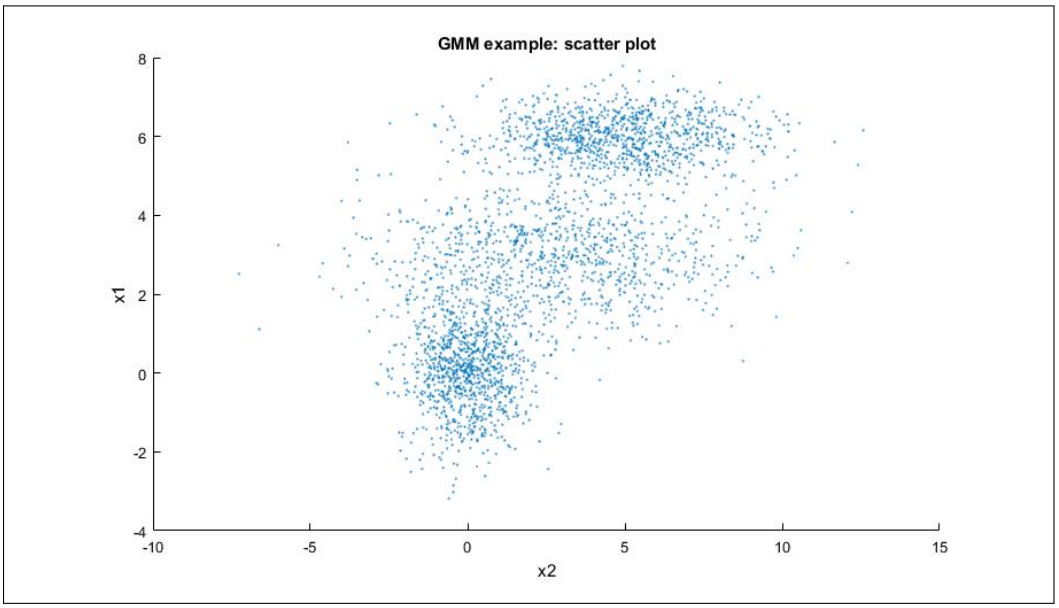

FIGURE 4.1: GMM example: scatter plot (each point can be thought of representing a 2D vector).

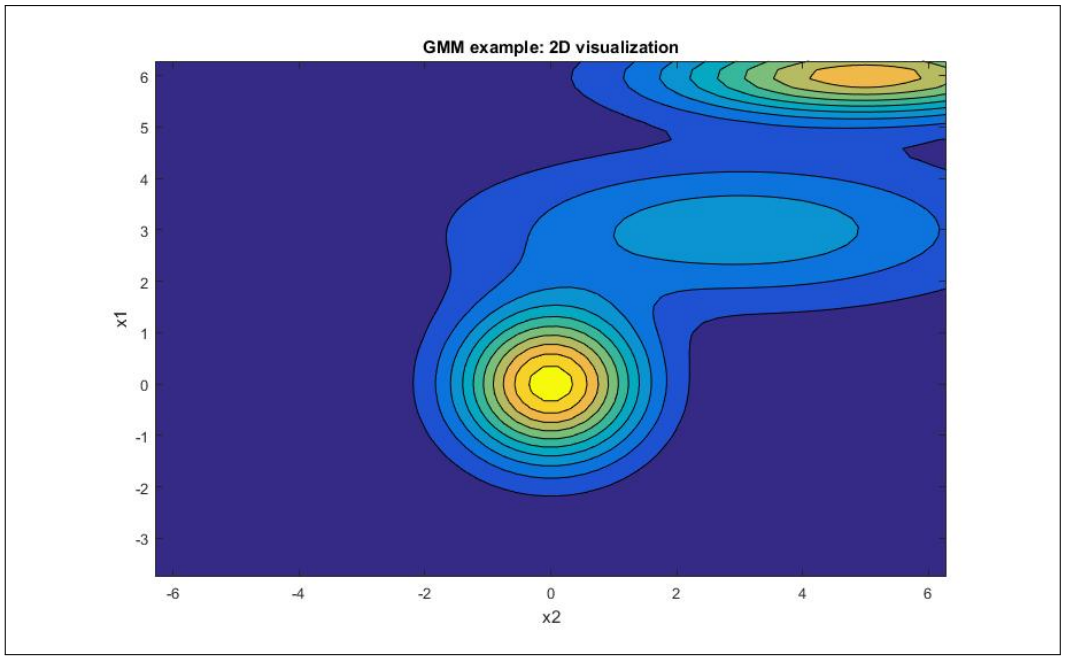

FIGURE 4.2: GMM example: 2D visualization of a mixture of 3 Gaussians fitting the 3 cloud of points of Figure 4.1 .

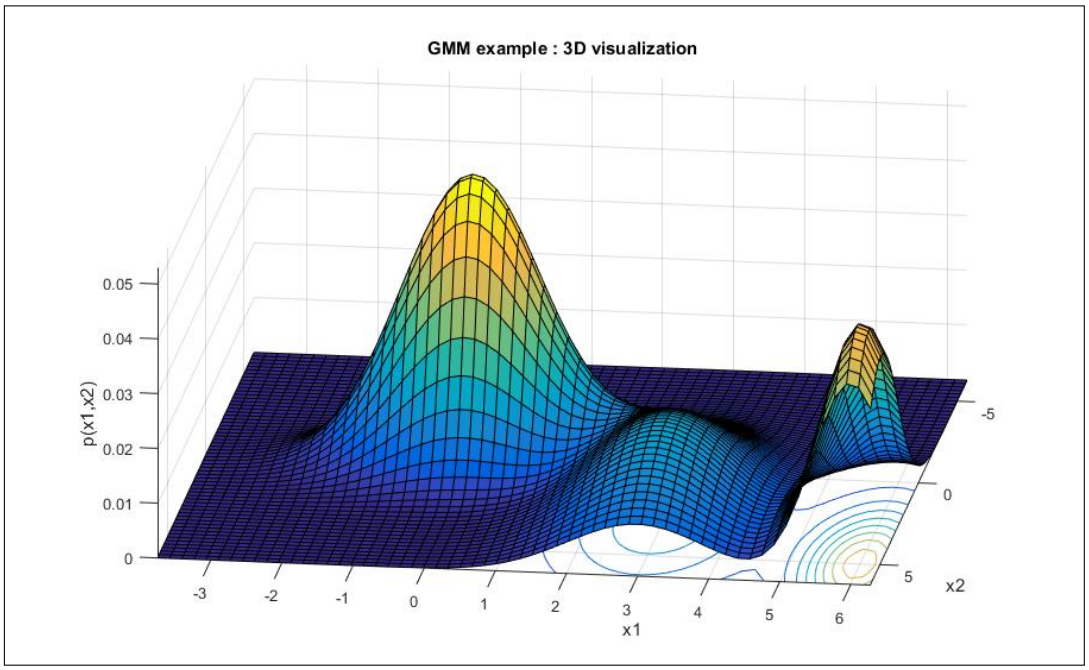

FIGURE 4.3: GMM example (inspired by [87]): 3D visualization of a mixture of 3 Gaussian distributions fitting the 3 clouds of points of Figure 4.1. (Note that the figure is rotated to show the underlying 2D level curves of Figure 4.2 representing the Gaussians in 2D).

4.1.2 Expectation maximization algorithm

The Expectation Maximization (EM) algorithm [88][87] was applied to estimate parameters of a GMM fitting a dataset of extracted SIFT features. We will briefly describe the mathematical intuition behind it in this section.

Let $\Theta = \{w_k, \underline{\mu}_k, \underline{\Sigma}_k; \forall k \in \{1, ..., K\}\}$ be the parameters of K Gaussian distributions constituting the GMM. Let $w_k \in \mathbb{R}$ be the weight of the k^{th} component, $\underline{\mu}_k \in \mathbb{R}^{128}$ the mean and $\underline{\Sigma}_k = \begin{bmatrix} \sigma^2_{k,1} & & \\ & \ddots & \\ & & \sigma^2_{k,128} \end{bmatrix}, \underline{\Sigma}_k \in \mathbb{R}^{128\times128}$ the diagonal covariance matrix, where $k = 1, ..., K$. Typically, all K classes will have the same weight at the beginning.

Let $\underline{X} = \{\underline{x}_1, ..., \underline{x}_n\}$, $\underline{x}_i \in \mathbb{R}^{128}$, be the data points (SIFT descriptors) to be clustered. Let $p(\underline{x}_i|\underline{\mu}_k, \underline{\Sigma}_k)$ be the probability that $\underline{x}_i$ originates from the k^{th} Gaussian distribution. The likelihood function is:[1] $\mathcal{L}(\Theta; \underline{X}) = \mathrm{p}(\underline{X}|\Theta)$. According to [83], an independence assumption is made. Therefore, the SIFT vectors $\underline{x}_i$, $i \in \{1, ..., n\}$, are considered statistically independent, i.e., $\underline{x}_1 \perp\!\!\!\perp \underline{x}_2 \perp\!\!\!\perp ... \perp\!\!\!\perp \underline{x}_n$:

$$\mathcal{L}(\Theta; \underline{x}_1, ..., \underline{x}_n) = \prod_{i=1}^{n} p(\underline{x}_i|\Theta) = \prod_{i=1}^{n} \sum_{k=1}^{K} w_k p(\underline{x}_i|\underline{\mu}_k, \underline{\Sigma}_k). \tag{4.1}$$

[1] https://en.wikipedia.org/wiki/Expectation-maximization_algorithm

In order to simplify computations, the log-likelihood can be considered[2]:

$$\ell(\Theta; \underline{x}_1, ..., \underline{x}_n) = \ln \mathcal{L}(\Theta; \underline{x}_1, ..., \underline{x}_n), \tag{4.2}$$

let:

$$\ell\ell \triangleq \ell(\Theta; \underline{x}_1, ..., \underline{x}_n). \tag{4.3}$$

The objective is to find the parameters $\hat{\Theta} = \{\hat{w}_k, \hat{\underline{\mu}}_k, \hat{\underline{\Sigma}}_k; \forall k \in \{1, ..., K\}\}$ maximizing the likelihood that the data points (SIFT vectors) originate from the GMM, i.e., finding:

$$\hat{\Theta} = argmax_{\Theta}\ \ell\ell. \tag{4.4}$$

In order to perform a soft clustering of the data, the EM algorithm was used. The intuition behind it is briefly summarized in the following way:

1) randomly initialize K cluster centers (K Gaussian distributions with random covariance matrices $\hat{\underline{\Sigma}}_k$);

2) calculate the likelihood that each vector belongs to each class;

3) calculate mean $\hat{\underline{\mu}}_k$, covariance matrix $\hat{\underline{\Sigma}}_k$ and weight $\hat{w}_k$ for each class w.r.t. new assignments;

4) repeat the above procedure from step (2) until mean and variance stabilize or, anyways, for a fixed number of iterations.

The EM algorithm alternates between an "Expectation" step (2) and a "Maximization" step (3). The former calculates the likelihoods, whilst the latter estimates the means, variances and weights given the assignments. An implementation of the EM algorithm based on VLFeat[3] libraries [89] was achieved.

Once the EM has finished, we obtain a clustered dataset with soft assignments of the SIFT vectors to each Gaussian component. This clustering results in a vocabulary (codebook) similar to one which could be obtained with a BoW approach using K-means. However, clustering with soft assignments is in general superior to hard assignments because it allows to take into account the dependencies to all classes (clusters), resulting in more subtle assignments.

Finally, it should be pointed out that, the covariance matrices are diagonal[4]. Otherwise, the algorithm would have to deal with full covariance matrices $\in \mathbb{R}^{128\times128}$ for SIFT features, which would become computationally expensive.

[2] https://en.wikipedia.org/wiki/Maximum_likelihood_estimation
[3] http://www.vlfeat.org/api/gmm.html
[4] http://www.vlfeat.org/overview/gmm.html

4.1.3 Fisher vector encoding

In this section, we summarize how the Fisher Vector (FV) encoding [90] is performed. The first step was to extract SIFT descriptors from images in the training (enrollment) set. Subsequently, the data was clustered to produce a dictionary of visual words, in the form of a GMM (using the EM algorithm).

The Fisher Vector of some image I is a concatenation of gradients with respect to parameters of each class. It produces a fixed length output regardless of the number of descriptors of an image. Its size is only dependent on the number of clusters of the visual vocabulary and the dimensionality of the feature vectors. Given $\underline{S} = \{\underline{x}_1, ..., \underline{x}_N\}$, a set of N SIFT descriptors extracted from some image I, $\hat{\Theta}_k = \{\hat{w}_k, \hat{\underline{\mu}}_k, \hat{\underline{\Sigma}}_k\}$ the parameters of the k^{th} GMM component, $k \in \{1, ..., K\}$, the Fisher Vector is defined as:

$$\underline{\Phi}(\underline{S}) = \begin{bmatrix} \vdots \\ \frac{\partial \ell\ell}{\partial \underline{\mu}_k} \\ \\ \frac{\partial \ell\ell}{\partial \underline{\Sigma}_k} \\ \\ \vdots \end{bmatrix}, \tag{4.5}$$

such that each gradient is formed by its partial derivatives with respect to means and covariances:

$$\frac{\partial \ell\ell}{\partial \underline{\mu}_k} = \begin{bmatrix} u_{1,k} \\ . \\ . \\ . \\ u_{D,k} \end{bmatrix}, \tag{4.6}$$

$$\frac{\partial \ell\ell}{\partial \underline{\Sigma}_k} = \begin{bmatrix} v_{1,k} \\ . \\ . \\ . \\ v_{D,k} \end{bmatrix}, \tag{4.7}$$

where $D = 128$ for SIFT descriptors. The components of the previous partial derivatives are:

$$u_{jk} = \frac{1}{N\sqrt{\hat{w}_k}} \sum_{i=1}^{N} q_{ik} \frac{x_{ij} - \hat{\mu}_{jk}}{\hat{\sigma}_{jk}}, \tag{4.8}$$

$$v_{jk} = \frac{1}{N\sqrt{2\hat{w}_k}} \sum_{i=1}^{N} q_{ik} \left(\left(\frac{x_{ij} - \hat{\mu}_{jk}}{\hat{\sigma}_{jk}} \right)^2 - 1 \right), \tag{4.9}$$

where:

$$q_{ik} = \frac{\exp\left[-\frac{1}{2} (\underline{x}_i - \hat{\underline{\mu}}_k)^T \hat{\underline{\Sigma}}_k^{-1} (\underline{x}_i - \hat{\underline{\mu}}_k) \right]}{\sum_{t=1}^{K} \exp\left[-\frac{1}{2} (\underline{x}_i - \hat{\underline{\mu}}_t)^T \hat{\underline{\Sigma}}_k^{-1} (\underline{x}_i - \hat{\underline{\mu}}_t) \right]}. \tag{4.10}$$

The latter Equation 4.10 describes the probability of a descriptor $\underline{x_i}$ to be assigned to the k^{th} Gaussian component. Therefore, the aggregated descriptor of SIFTs, of some image I, can be presented as a column Fisher Vector such as[5]:

$$\underline{\Phi}(\underline{S}) = \begin{bmatrix} u_{1,1} \\ v_{1,1} \\ . \\ . \\ . \\ u_{D,1} \\ v_{D,1} \\ . \\ . \\ . \\ u_{1,K} \\ v_{1,K} \\ . \\ . \\ . \\ u_{D,K} \\ v_{D,K} \end{bmatrix}, \tag{4.11}$$

where $D = 128$. Consequently, The dimensionality of the FV is $2DK$. In the case of SIFT descriptors it is $2 \times 128 \times K$. Therefore, the FV has a fixed length which depends only on the number of K clusters, chosen for the GMM, regardless of how many descriptors were generated by the SIFT algorithm for an image I. Compared to a brute force matching method, where all pairs of descriptors of 2 given images must be compared, we can see that a single comparison per image is needed with the FV approach. Therefore, this allows to significantly reduce the complexity of the matching procedure. The FV encoding was chosen because it is quite popular and is currently considered as one of the state-of-the-art aggregation methods for local descriptors.

[5] http://www.vlfeat.org/api/fisher-fundamentals.html

4.1.4 Generalized overview

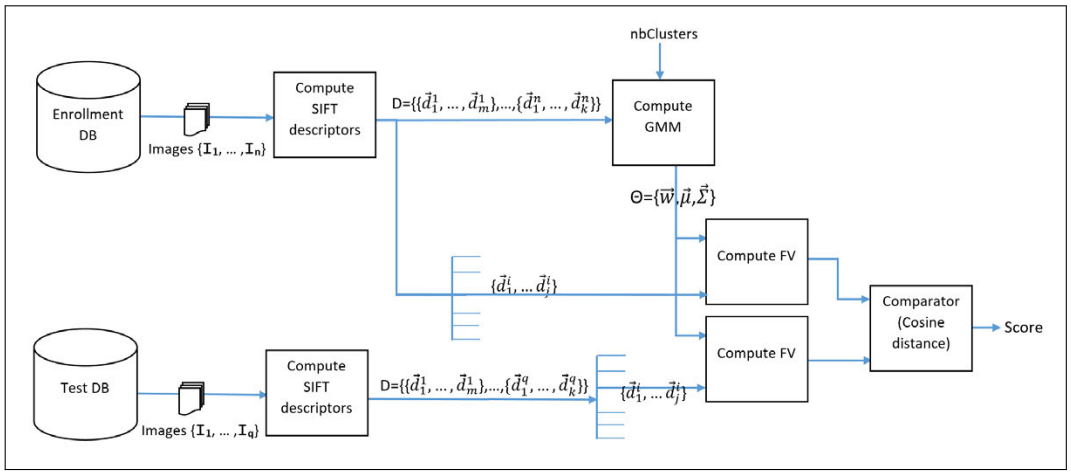

FIGURE 4.4: Generalized diagram of the process of image recognition by Fisher vector representation.

It should be emphasized, that as shown in Figure 4.4, the *same* Gaussian Mixture Model (GMM), with given parameters (number of clusters), was used to train the enrollment dataset and to compute FV from the testing set. This is obvious, of course, as the GMM can be viewed as a codebook of visual words (but *softly* assigned). One would ideally hope to find very similar "codewords" in the computed testing set FVs, as those assigned in the GMM to its match, in such a way that the distance to the match would be the smallest, as explained before. GMMs were trained with different number of clusters on the same enrollment subset and Fisher Vectors were computed from them for the testing set.

The similarity metric used was the following:

$$dist(\underline{x}, \underline{y}) = 1 - \frac{\underline{x} \cdot \underline{y}^T}{\sqrt{(\underline{x} \cdot \underline{x}^T)(\underline{y} \cdot \underline{y}^T)}}, \tag{4.12}$$

given that $\underline{x}$ and $\underline{y}$ are *row* vectors[6]. We would like to add that it is more convenient to use 1-$cos\angle(\underline{x}, \underline{y})$ because negative values can be avoided.

[6]Otherwise, if they are column vectors the transpose operator for the scalar product should of course be applied on the first operand.

4.2 Experimental results

This section presents experimental results obtained using an approach based on Fisher Vectors and Gaussian Mixture Model. Please, note that the exact same experimental setup and definition of "similarity" and "dissimilarity" were used than those presented in subsection 3.2.1.

4.2.1 Description

The obatined results and statistics using the following parameters[7] will be presented:

- PharmaPack S1 with 32 cl. GMM
- PharmaPack S1 (cropped) with 64 cl. GMM
- PharmaPack S1 (cropped) with 128 cl. GMM
- PharmaPack S1 (cropped) with 256 cl. GMM
- PharmaPack S1 (cropped) with 512 cl. GMM
- PharmaPack S1 (non-cropped) with 256 cl. GMM
- PharmaPack S2 (cropped) with 256 cl. GMM

Moreover, we used SIFT descriptors extracted with a value of peakThresh=0.01, and custom peakThresh[8] parameter values (higher than 0.01) chosen in such a way as to obtain maximal number of features of approximately:

- 900-1000 features max.
- 500-600 features max.
- 300-400 features max.

This should reveal the impact of the number of features on system performance. We will present ROC curves and separability histograms to show the influence of these different parameters.

Furthermore, we will investigate the influence of cropping and the presence of skin texture and shadows in the background (for hand-held packages).

[7] S1: wooden background, S2: hand-held

[8] We recall that the PeakThresh parameter controls the sensitivity of the SIFT detector, and as it's value is increased, less features will be detected and vice-versa.

4.2.2 Experiment 1

In this section we present results for peakThresh=0.01 and GMMs trained with 32, 64, 128, 256 and 512 clusters. The testing set is **PharmaPack_R_I_S1** (wooden background, cropped).

32 clusters

To further detail our results, we will present numerical statistics in the form of tabulated percentages at the end of each experiment.

TABLE 4.1: Summary of statistics for Experiment 1 with 32 clusters. Probabilities given for the EER strategy ($P_{FA} \simeq P_M$).

P_{FA}	P_D	P_M
5.3%	94.68%	5.3%

TABLE 4.2: Experiment 1 (32 clusters): statistics in % for several values of probability of miss.

Prob. of miss P_M	Prob. of false acceptance P_{FA}	Prob. of correct detection P_D
0%	78%	100%
0.01%	66%	99.99%
0.1%	41%	99.9%
0.5%	24%	99.5%
1%	17%	99%
5%	5.6%	95%
10%	10%	90%
95%	0.005%	5%

64 clusters

TABLE 4.3: Summary of statistics for Experiment 1 with 64 clusters. Probabilities given for the EER strategy ($P_{FA} \simeq P_M$).

P_{FA}	P_D	P_M
3.62%	96.39%	3.61%

TABLE 4.4: Experiment 1 (64 clusters): statistics in % for several values of probability of miss.

Prob. of miss P_M	Prob. of false acceptance P_{FA}	Prob. of correct detection P_D
0%	70%	100%
0.01%	60%	99.99%
0.1%	33%	99.9%
0.5%	15.4%	99.5%
1%	10.25%	99%
5%	2.5%	95%
10%	1%	90%
95%	0.06%	5%

128 clusters

TABLE 4.5: Summary of statistics for Experiment 1 with 128 clusters. Probabilities given for the EER strategy ($P_{FA} \simeq P_M$).

P_{FA}	P_D	P_M
2.31%	97.70%	2.30%

TABLE 4.6: Experiment 1 (128 clusters): statistics in % for several values of probability of miss.

Prob. of miss P_M	Prob. of false acceptance P_{FA}	Prob. of correct detection P_D
0%	53%	100%
0.01%	40%	99.99%
0.1%	20%	99.9%
0.5%	8.4%	99.5%
1%	4.8%	99%
5%	1.1%	95%
10%	0.5%	90%
95%	0.02%	5%

256 clusters

TABLE 4.7: Summary of statistics for Experiment 1 with 256 clusters. Probabilities given for the EER strategy ($P_{FA} \simeq P_M$).

P_{FA}	P_D	P_M
1.71%	98.30%	1.70%

TABLE 4.8: Experiment 1 (256 clusters): statistics in % for several values of probability of miss.

Prob. of miss P_M	Prob. of false acceptance P_{FA}	Prob. of correct detection P_D
0%	50%	100%
0.01%	39%	99.99%
0.1%	16.4%	99.9%
0.5%	4.9%	99.5%
1%	2.7%	99%
5%	0.6%	95%
10%	0.3%	90%
95%	0.01%	5%

512 clusters

TABLE 4.9: Summary of statistics for Experiment 1 with 512 clusters. Probabilities given for the EER strategy ($P_{FA} \simeq P_M$).

P_{FA}	P_D	P_M
1.28%	98.72%	1.28%

TABLE 4.10: Experiment 1 (512 clusters): statistics in % for several values of probability of miss.

Prob. of miss P_M	Prob. of false acceptance P_{FA}	Prob. of correct detection P_D
0%	51%	100%
0.01%	34.4%	99.99%
0.1%	10.8%	99.9%
0.5%	2.9%	99.5%
1%	1.5%	99%
5%	0.44%	95%
10%	0.29%	90%
95%	0.07%	5%

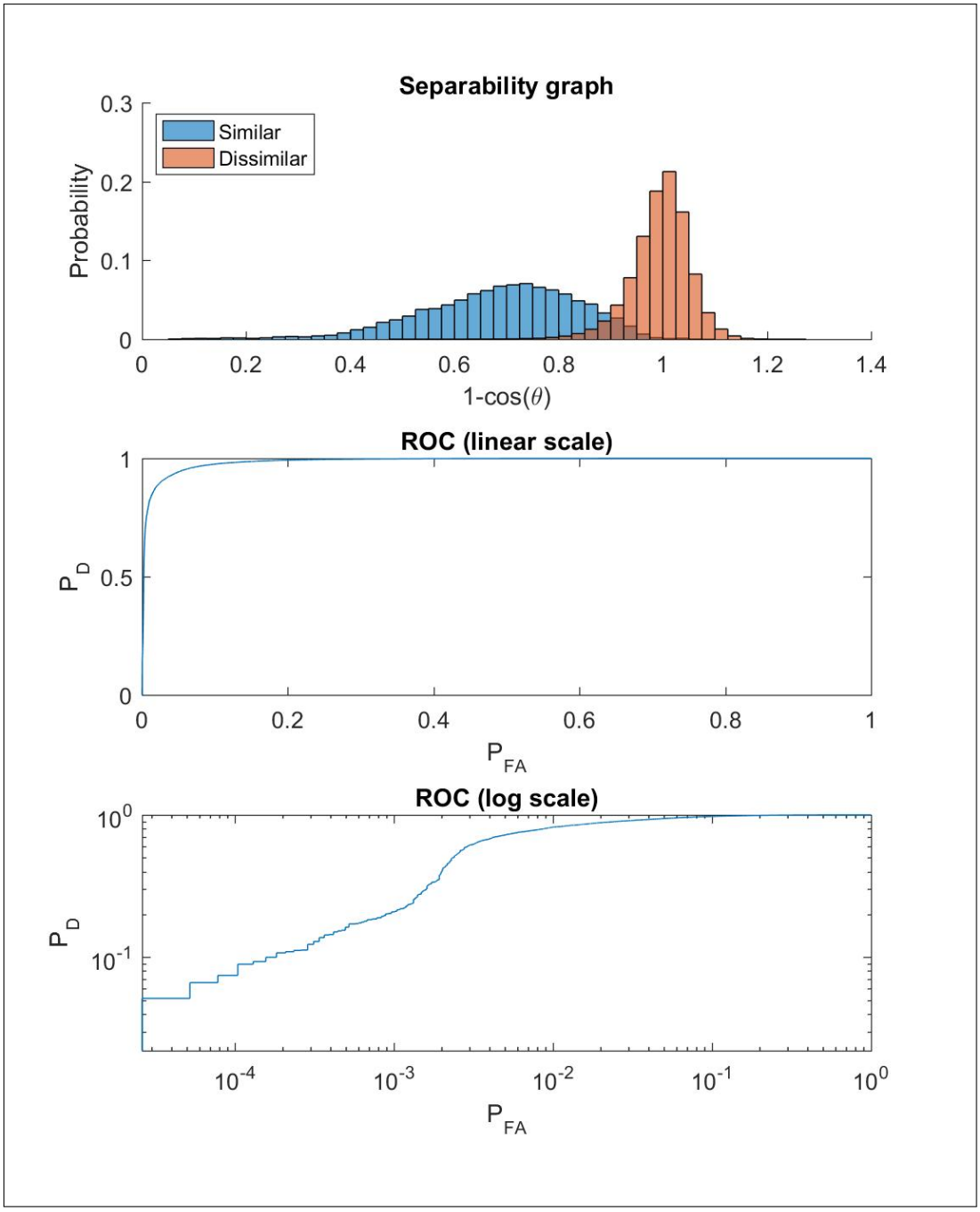

FIGURE 4.5: Separability plot and ROC curves for S1 and a trained GMM of **32 clusters**.

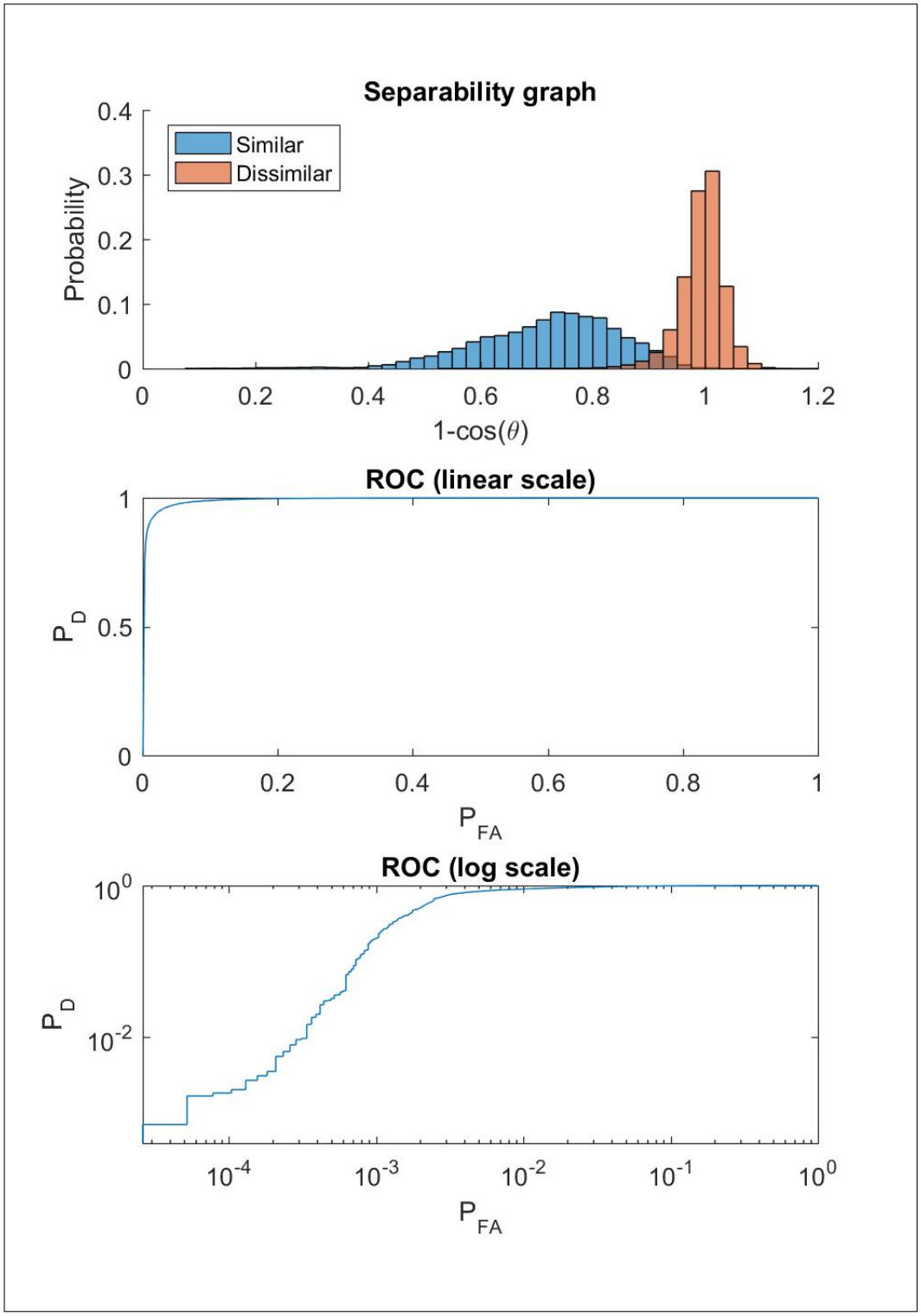

FIGURE 4.6: Separability plot and ROC curves for S1 and a trained GMM of **64 clusters**.

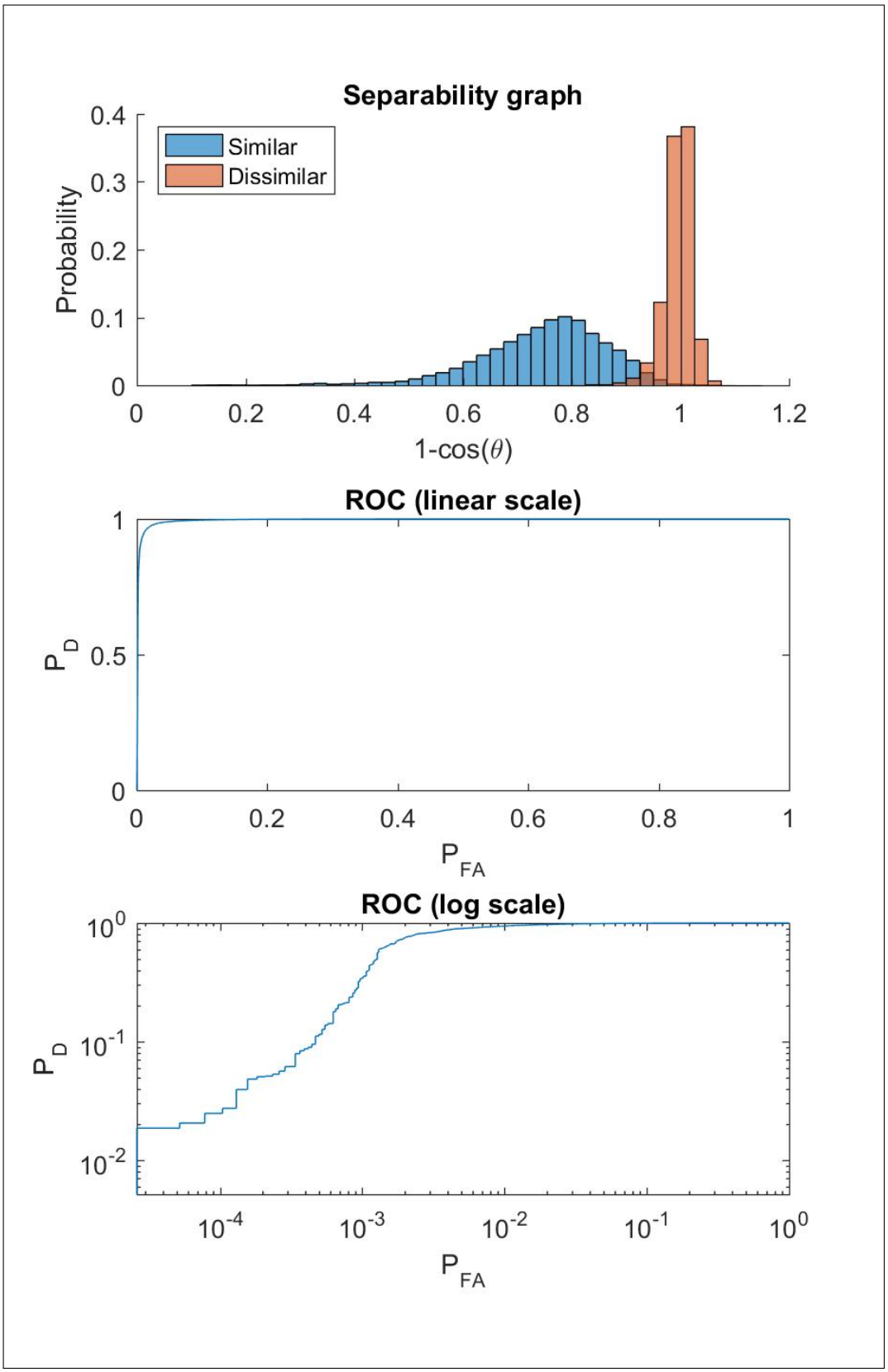

FIGURE 4.7: Separability plot and ROC curves for S1 and a trained GMM of **128 clusters**.

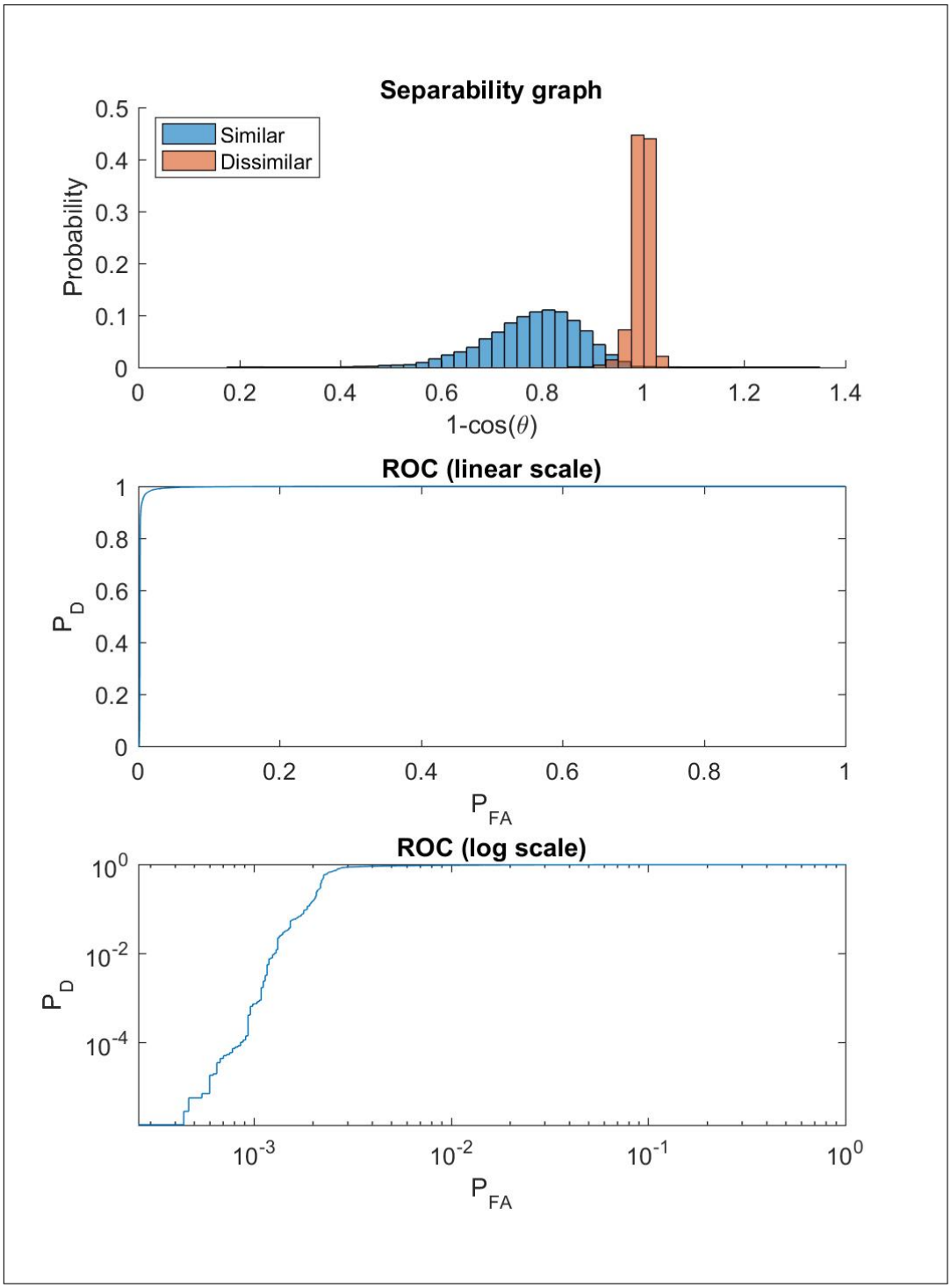

FIGURE 4.8: Separability plot and ROC curves for S1 and a trained GMM of **256 clusters**.

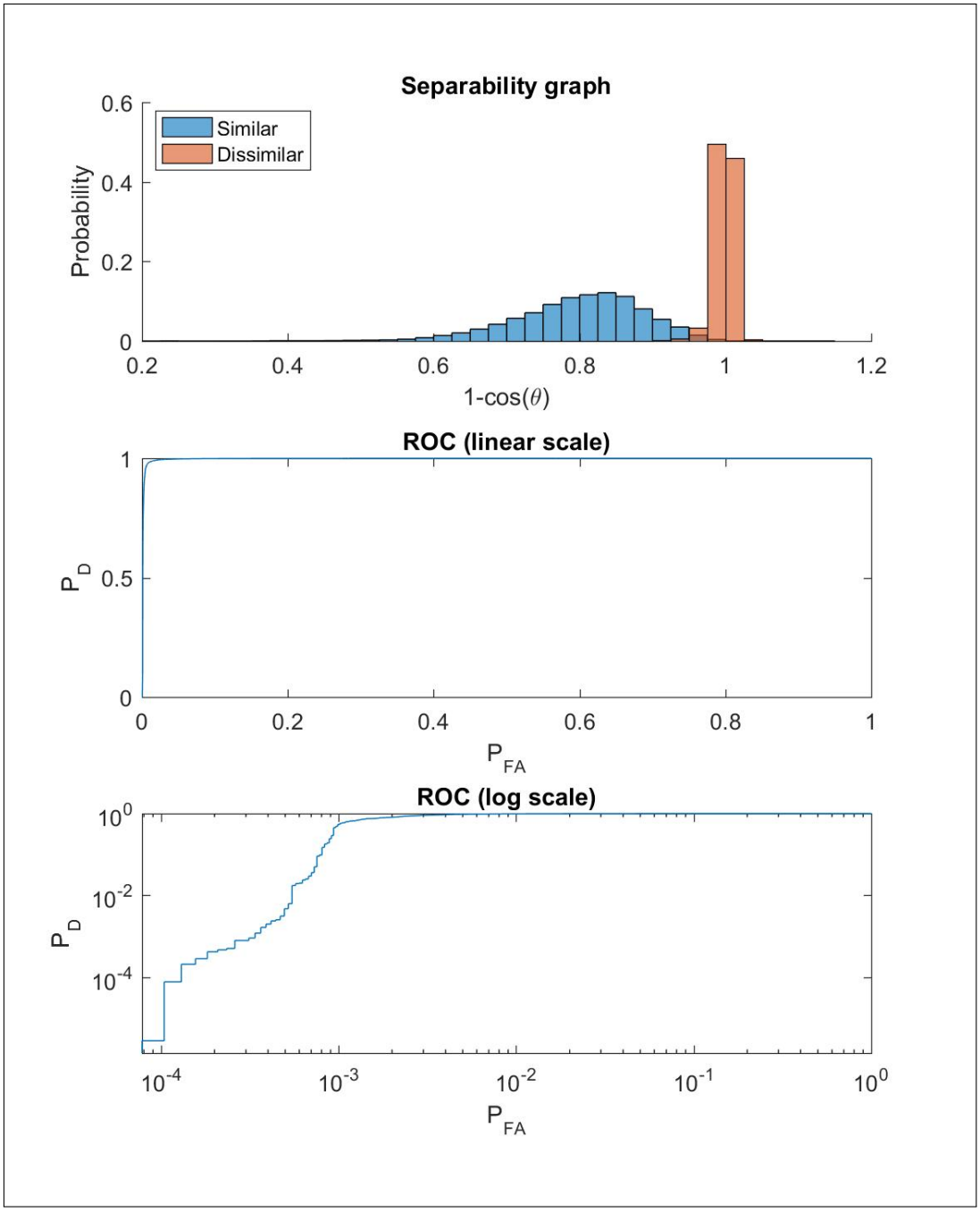

FIGURE 4.9: Separability plot and ROC curves for S1 and a trained GMM of **512 clusters**.

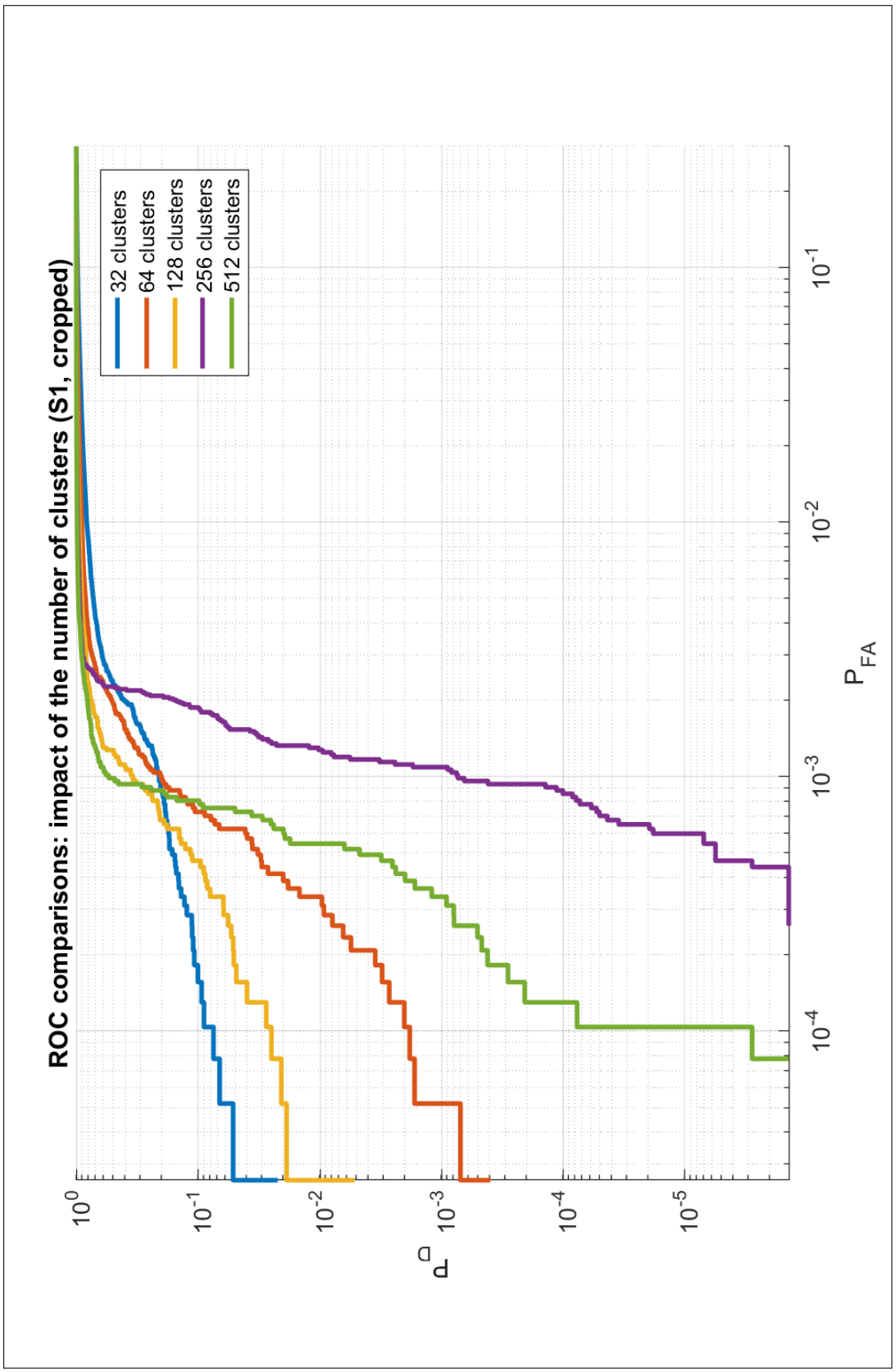

FIGURE 4.10: Experiment 1: comparisons between ROCs for different numbers of clusters. A higher number of clusters results in a better performance (for the same number of descriptors/image, i.e., peakThresh=0.01 yields up to about 4000 descriptors/image).

We will briefly draw some conclusions from Experiment 1. Firstly, it can be observed in Figure 4.5 - Figure 4.9 that the *similar* and *dissimilar* distributions tend to be better separated as the number of clusters increases. Likewise, it can be observed in Figure 4.10 that the ROC curves show the best performances for the highest amount of clusters (i.e., 256 and 512 cl.). Moreover, a smaller performance gap between 256 and 512 clusters can be observed. It should be emphasized that we are interested in the upper right slice of the ROC curves (i.e., for values of P_D) that are higher than 90%). Indeed, any lower probabilities of correct detection can be considered as unacceptable for practical usage since they will generate many misses.

4.2.3 Experiment 2

We present the results obtained with 256 clusters for the PharmaPack_R_I_S1 testing dataset and custom SIFT peakThresh parameters in such a way as to obtain the following (approximate) number of descriptors per image[9] :

- 900-1000 descriptors/image
- 500-600 descriptors/image
- 300-400 descriptors/image

The number of clusters chosen was 256 because considering the results obtained with peakThresh=0.01, this was considered as a good trade-off between computational complexity and performance.

This experiment will allow us to assess the impact of a lower number of descriptors per image. It is interesting to estimate the performance with a low number of descriptors with regards to complexity.

TABLE 4.11: Summary of statistics for Experiment 2 with 256 clusters. Probabilities given for the EER strategy ($P_{FA} \simeq P_M$).

nbClusters	nbDescr.	P_{FA}	$\boldsymbol{P_D}$	P_M
256	900-1000	2.77%	**97.24%**	2.76%
256	500-600	2.34%	**97.67%**	2.33%
256	300-400	3.08%	**96.93%**	3.07%

TABLE 4.12: Experiment 2 (256 clusters): statistics in % for several values of probability of miss.

nbDescr.	Prob. of miss P_M	Prob. of false acceptance P_{FA}	Prob. of correct detection P_D
900-1000	0.01%	51%	99.99%
900-1000	0.1%	29.9%	99.9%
900-1000	0.5%	12.2%	99.5%
900-1000	1%	7.1%	99%
900-1000	5%	1.5%	95%
900-1000	10%	0.7%	90%

[9]Please note that these numbers are a higher bound, not lower bound, as some images may intrinsically produce less descriptors, due for example, to low contrast or mostly uniform patterns.

TABLE 4.13: Experiment 2 (256 clusters): statistics in % for several values of probability of miss.

nbDescr.	Prob. of miss P_M	Prob. of false acceptance P_{FA}	Prob. of correct detection P_D
500-600	0.01%	48%	99.99%
500-600	0.1%	22%	99.9%
500-600	0.5%	8.5%	99.5%
500-600	1%	5%	99%
500-600	5%	0.9%	95%
500-600	10%	0.4%	90%

TABLE 4.14: Experiment 2 (256 clusters): statistics in % for several values of probability of miss.

nbDescr.	Prob. of miss P_M	Prob. of false acceptance P_{FA}	Prob. of correct detection P_D
300-400	0.01%	46%	99.99%
300-400	0.1%	24.7%	99.9%
300-400	0.5%	9.7%	99.5%
300-400	1%	6.6%	99%
300-400	5%	2.2%	95%
300-400	10%	1.4%	90%

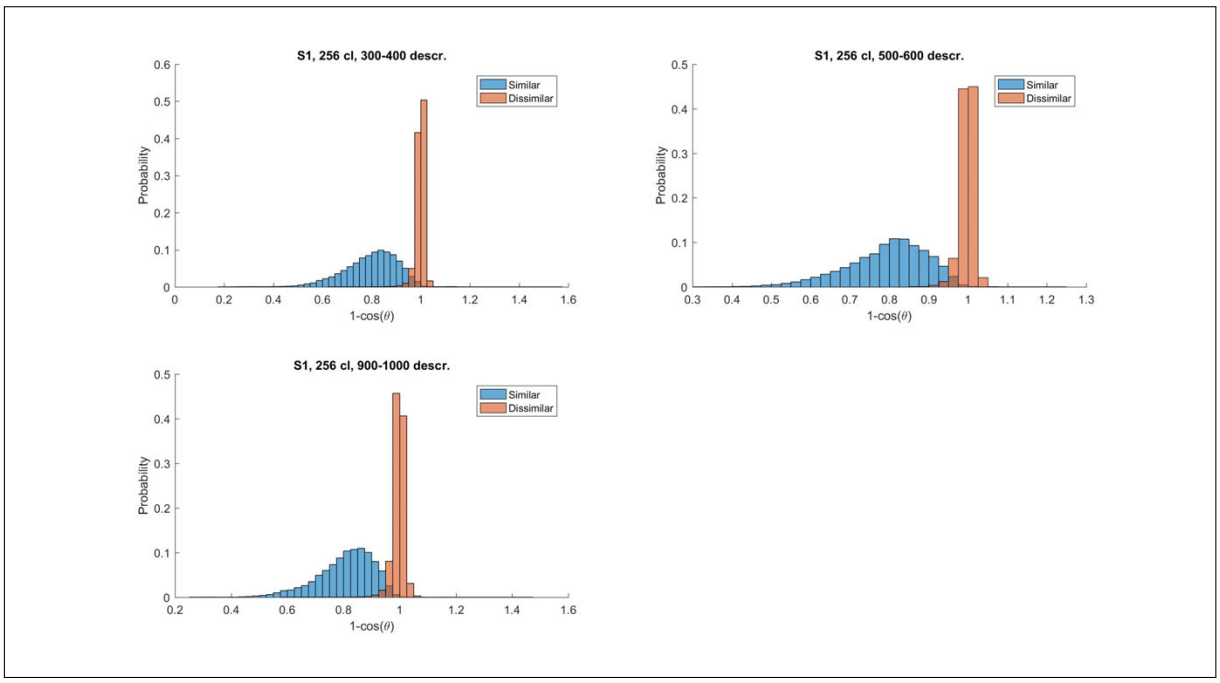

FIGURE 4.11: Separability plots for Experiment 2.

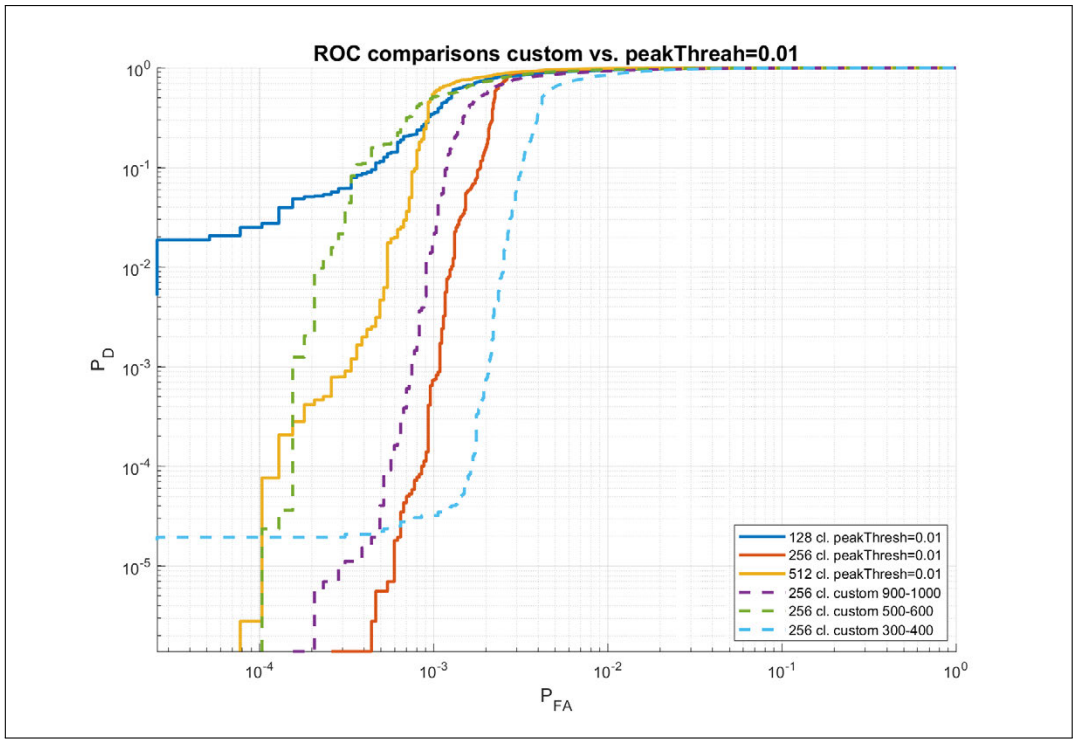

FIGURE 4.12: ROC curves for Experiment 2 plotted on the full log-log scale (0 to 1). It is not clear from this graph, which parameters provide the best performance.

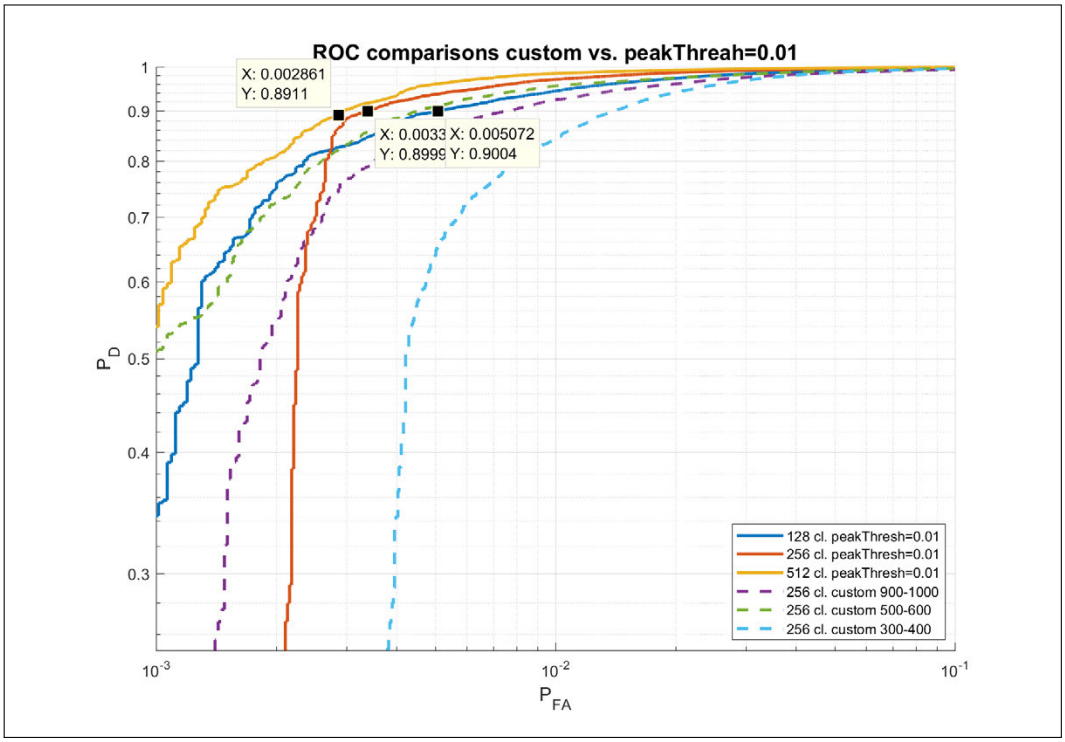

FIGURE 4.13: Zoom on the useful part of the ROC curve of Figure 4.12 (Experiment 2). The data cursors on the figure shows the probabilities of false acceptance for a fixed probability of miss ($\simeq$90%). It can be observed that the best performance, for the ROC curve slice which is over $P_M = 90\%$, is reached with 512 cl. with PeakThresh=0.01 (up to 4000 descriptors/image).

Figure 4.13 allows us to draw conclusions with regards to the impact of the number of descriptors/image for a fixed number of clusters (i.e., 256 cl.). Moreover, to have a better overview, previous results from Experiment 1 are also shown next to those from Experiment 2.

As mentioned previously, we only consider the "useful" part of the ROC curves. That is to say, the approximately two thirds of the full scale ROC of Figure 4.12. Indeed, any percentage of true positives lower than 90% is considered too low for a practical usage. The data cursors in Figure 4.13 show us that the best performance is reached for 512 and 256 cl. with peakThresh=0.01. Then, 128 cl. and peakThresh=0.01 competes with 256 clusters with custom 500-600 descriptors/image.

The performance behavior with regards to the lower number of descriptors, i.e., 300-1000 max. descriptors/image, for a fixed number of clusters, is, however, less clear. What can be observed is that a too low number of descriptors/image (i.e., less than 1000) results in the worst performance.

4.2.4 Experiment 3

The impact of the background was investigated in this experiment by comparing cropped versus non-cropped images. Indeed, it seemed likely that the performance would be worse with non-cropped images, which show the background. The testing dataset used was **PharmaPack_R_I_S1** for **non-cropped** images. A peakThresh=0.01 and 256 clusters were used and the results were compared to those obtained with (S1) cropped images.

TABLE 4.15: Summary of statistics for Experiment 3 with 256 clusters. Probabilities given for the EER strategy ($P_{FA} \simeq P_M$).

P_{FA}	P_D	P_M
1.7%	98.3%	1.7%

TABLE 4.16: Experiment 3 (256 clusters): statistics in % for several values of probability of miss.

Prob. of miss P_M	Prob. of false acceptance P_{FA}	Prob. of correct detection P_D
0%	60%	100%
0.01%	45%	99.99%
0.1%	18.8%	99.9%
0.5%	5.2%	99.5%
1%	2.8%	99%
5%	0.6%	95%
10%	0.3%	90%
95%	0.01%	5%

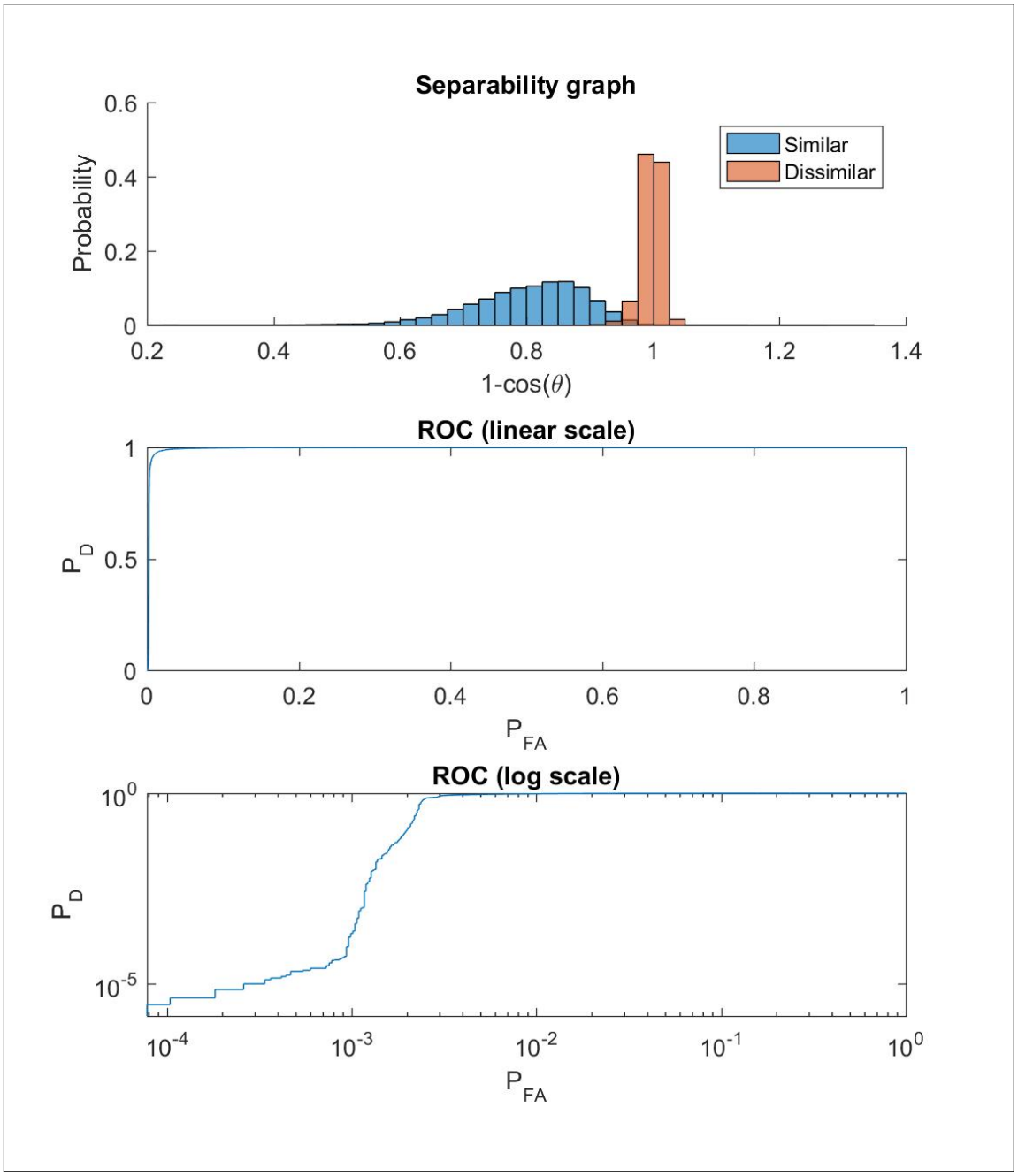

FIGURE 4.14: ROC and separability plots for Experiment 3 (dataset S1 (non-cropped), peakThresh=0.01, 256 cl.).

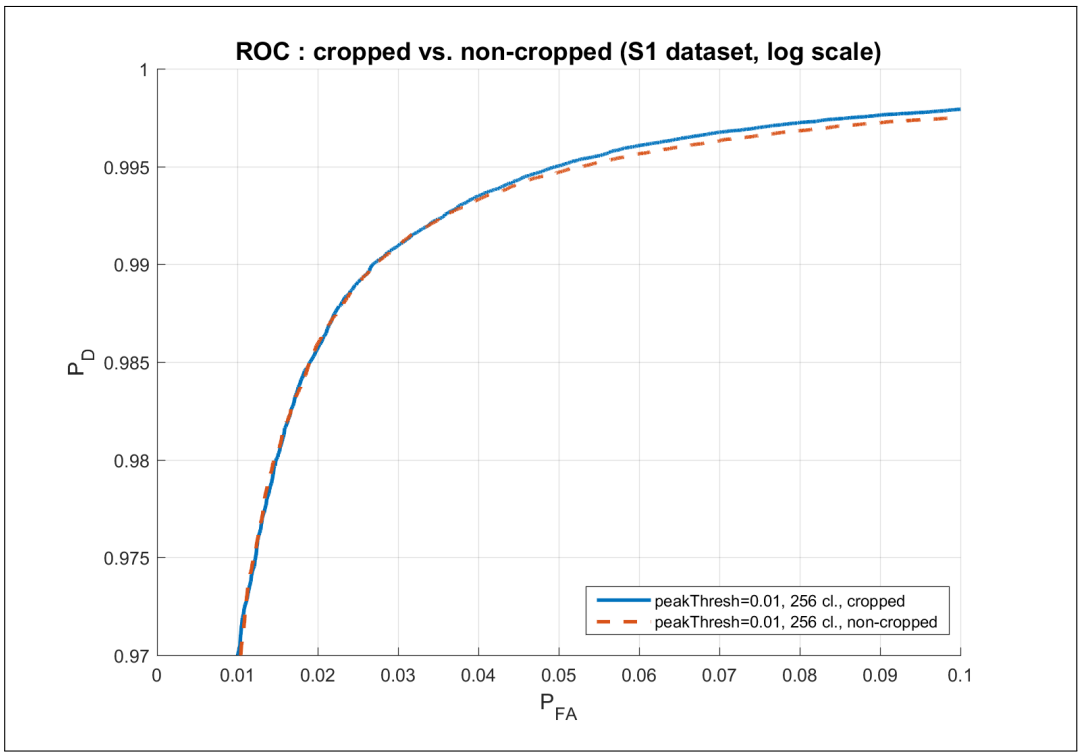

FIGURE 4.15: Performance comparison between cropped and non-cropped for Experiment 3.

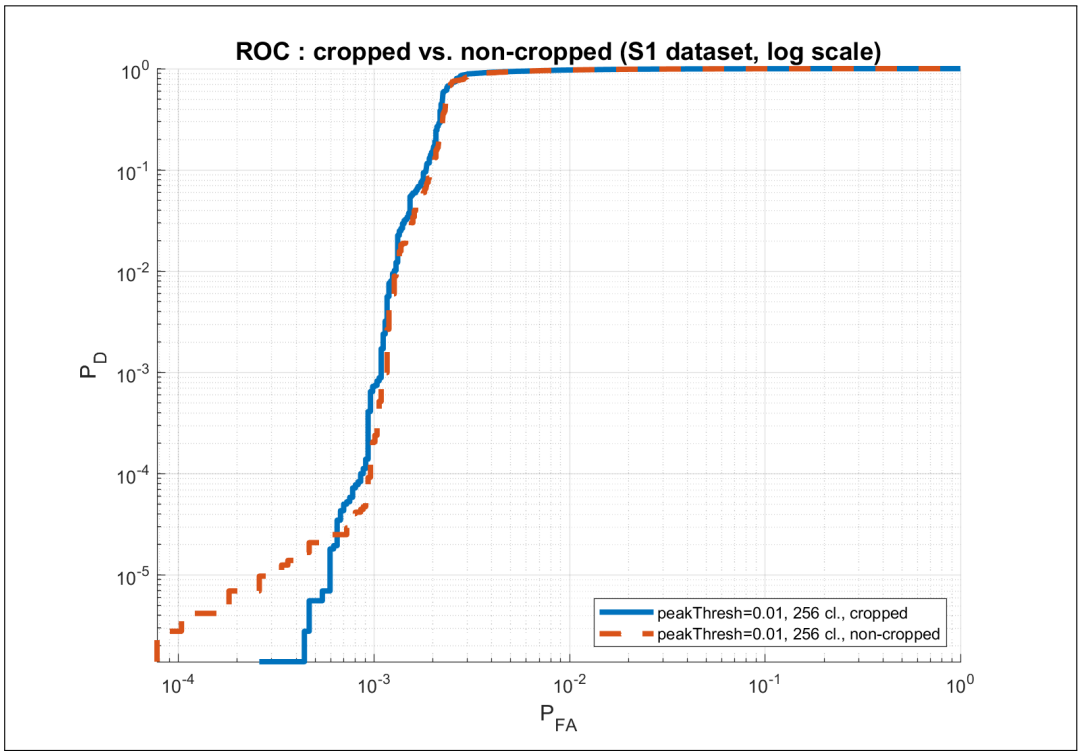

FIGURE 4.16: Performance comparison between cropped and non-cropped for Experiment 3. Almost no difference can be noticed. However, when observing closely the ROC curves and comparing the statistics in Table 4.8 vs. Table 4.16, it seems that the performance with cropped images is slightly better.

In Experiment 3, we have investigated the recognition performance of cropped vs. non-cropped images. As it can be observed in Figure 4.16, only a minor difference in performance between cropped and non-cropped images is noticed. However, as expected, we have slightly better results with the cropped ones. This is most likely due to the fact that non-cropped images might produce descriptors in the non-cropped (background) area, thus, increasing the risk of false acceptance. Nevertheless, the performance gap is minor, probably due to the wooden background being a relatively uniform area.

Moreover, it seems likely that on a more complex or cluttered background, the performance could significantly decrease, due to a higher risk of descriptor production leading in potentially higher probability of false acceptance.

4.2.5 Experiment 4

Through this experiment, we have investigated the influence of images of hand-held packages versus those laid on a wooden table (fixed position). For this experiment we have used the **PharmaPack_R_I_S2** testing dataset (hand-held). We have used again peakThresh=0.01 and 256 clusters and compared it to the *non* hand-held results.

TABLE 4.17: Summary of statistics for Experiment 4 with 256 clusters. Probabilities given for the EER strategy ($P_{FA} \simeq P_M$).

P_{FA}	P_D	P_M
2.47%	97.54%	2.46%

TABLE 4.18: Experiment 4 (256 clusters): statistics in % for several values of probability of miss.

Prob. of miss P_M	Prob. of false acceptance P_{FA}	Prob. of correct detection P_D
0%	59%	100%
0.01%	46%	99.99%
0.1%	23.5%	99.9%
0.5%	9.4%	99.5%
1%	5.5%	99%
5%	1.3%	95%
10%	0.8%	90%
95%	0.3%	5%

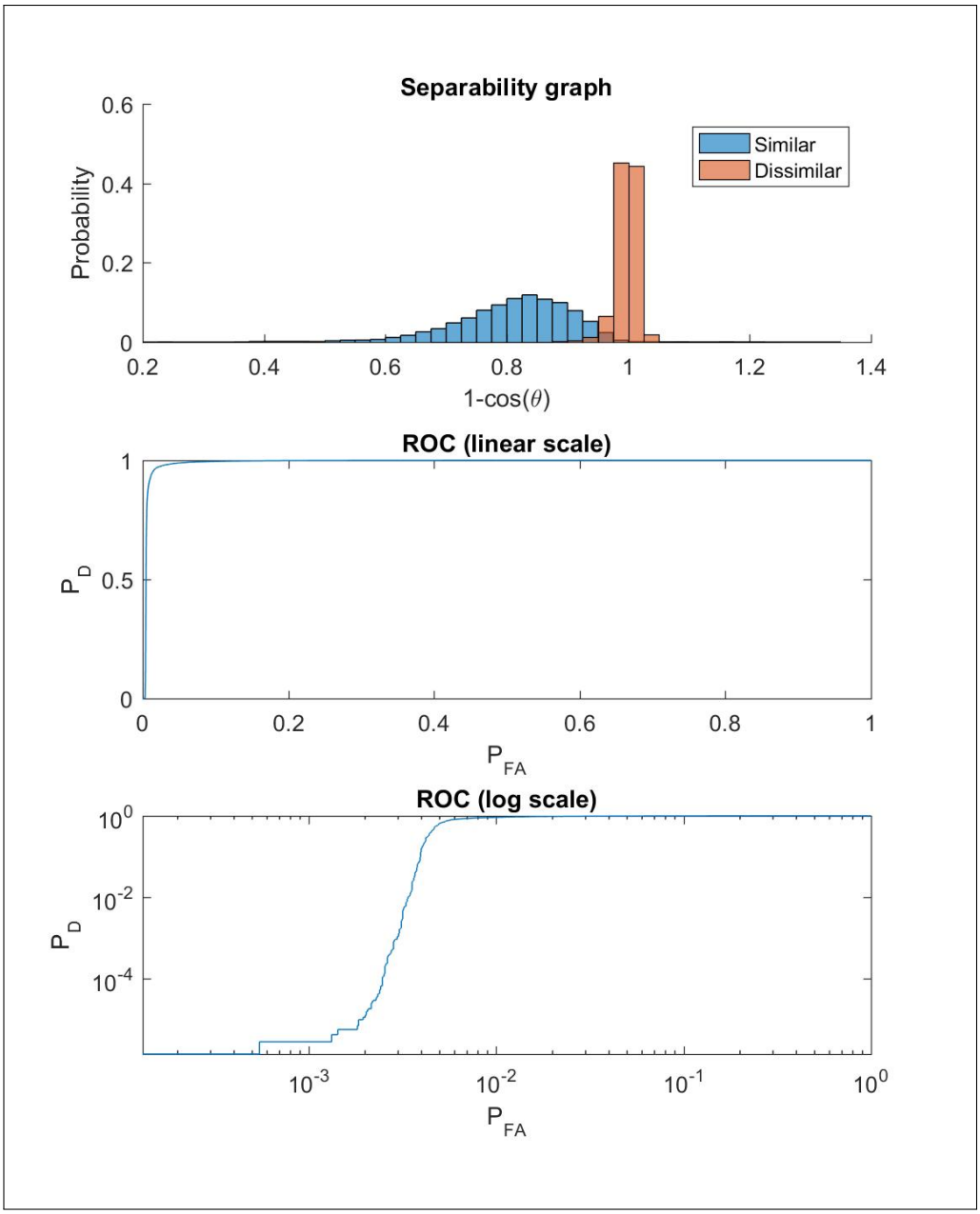

FIGURE 4.17: Histograms and ROCs for Experiment 4 (dataset S2 (hand-held, cropped), peakThresh=0.01, 256 cl.) .

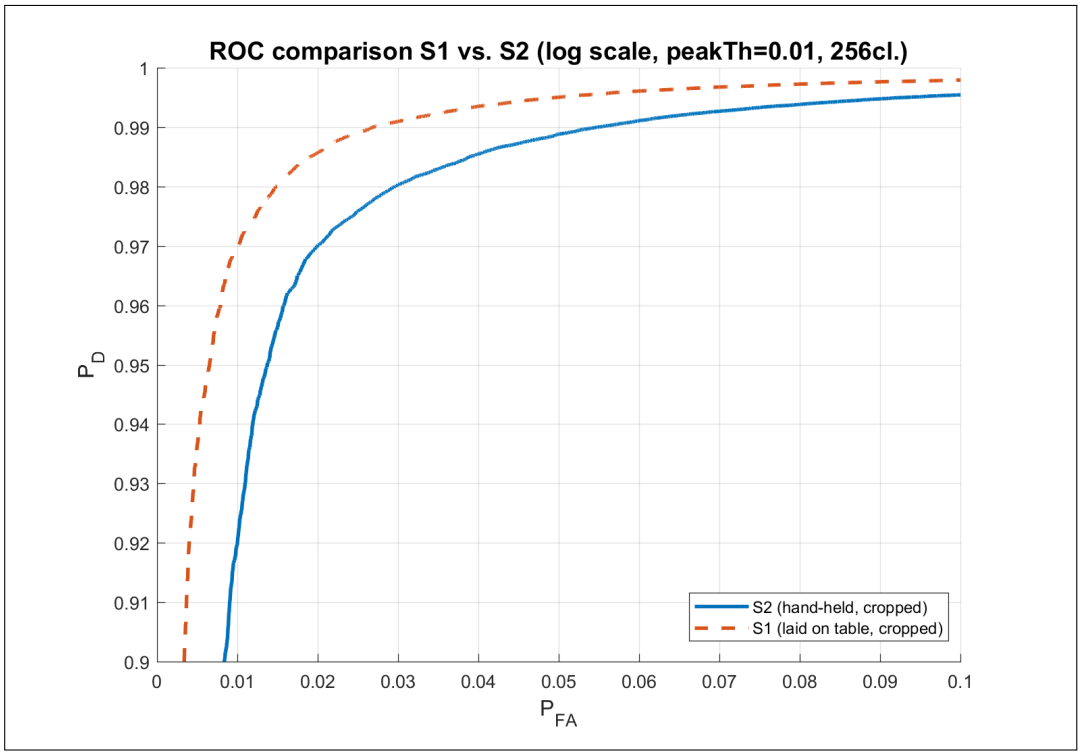

FIGURE 4.18: Comparison between hand-held and fixed position (laid on table) for Experiment 4. A better performance is observed for the fixed position.

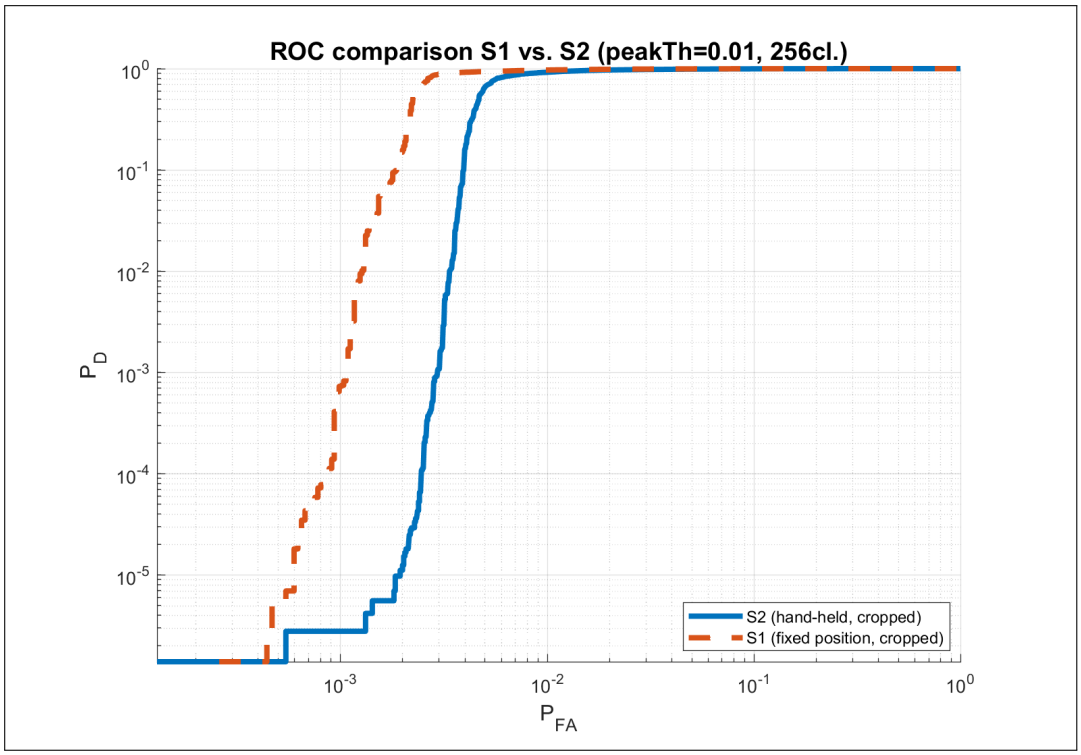

FIGURE 4.19: Comparison between hand-held and fixed position (laid on table) for Experiment 4. It can be observed that the fixed position yields better performance.

We can conclude from the fourth experiment that the fixed position is superior to the hand-held one. This is clearly visible in Figure 4.18 and Figure 4.19. It is likely that the lower recognition performance with the S2 (hand-held) dataset is caused by motion blur and glares (due to changing light conditions). Thus, with these lower quality images, the risk of false acceptance is increased. Moreover, it is likely that very similar packages will be more sensitive to this.

4.2.6 Performance rate

Due to high computational complexity in both CPU time and memory, we could not afford to train Gaussian Mixture Models with more than 512 clusters. Indeed, the MATLAB scripts were running on a shared server with the following hardware specifications:

- Intel Xeon CPU E5-2680 v3 @ 2.50GHz (12 cores)
- 378 GB of RAM

We have observed that computing GMMs with a significantly higher number of clusters drastically increased the complexity. Moreover, after performing the experiments, it could be observed that the increase of performance when doubling the number of clusters, seemed to follow a logarithmic curve. This is illustrated by Figure 4.20. It can be noticed that the *trend-line* (dashed line) fitting the empirical curve is logarithmic. Therefore, it can be expected that even if significantly increasing the number of clusters, the performance rate will tend to slow down and possibly stabilize not far from the 512-cluster performance.

It should be pointed out that the graph in Figure 4.20 illustrates the performance increase obtained in subsection 4.2.2 with the SIFT detector set at peakThresh=0.01 and 5 different numbers of clusters, namely: 32, 64, 128, 256 and 512.

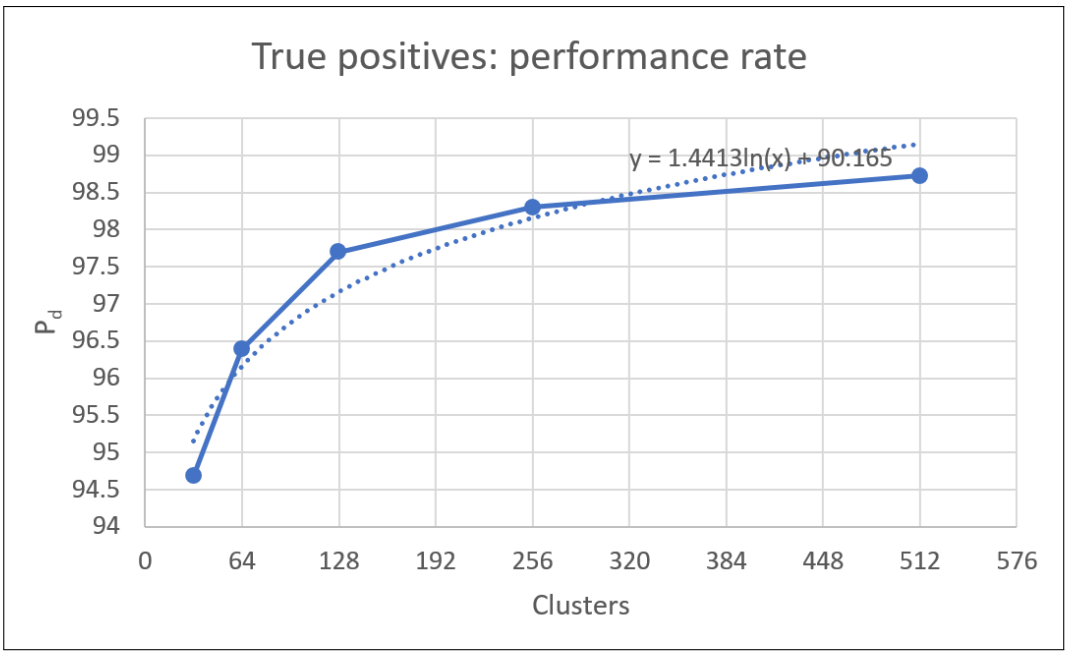

FIGURE 4.20: Performance rate with regards to the number of clusters of the GMM.

4.3 Additional observations

4.3.1 Geometrical matching approach vs. non-geometrical approach

In this section we present a comparison of SIFT+RANSAC performance vs. SIFT+FV encoding with GMM clustering to investigate the impact and importance of taking into account geometrical information provided by SIFT keypoints.

We can observe in Figure 4.21 that the performance of RANSAC is indeed better than the FV, as we had expected. This clearly shows the advantage of taking into account the geometrical information of SIFT keypoints.

Moreover, we should mention the impact on complexity of both approaches. The Fisher Vector approach produces a single aggregated descriptor vector for each image, whereas for the RANSAC approach all SIFT descriptors and their associated keypoints must be kept.

The complexity of RANSAC matching compared to FV matching is considerably higher. Indeed, for the FV approach, two images can be compared by their corresponding FV descriptors, using Euclidean or Cosine distance. On the contrary, for the RANSAC approach, all descriptors of both images must be compared and matched by brute force, subsequently a geometrical alignment using the RANSAC algorithm must be performed using keypoints associated with the matching descriptors. Only then, a percentage of inlier keypoints (with regards to the homography matrix found by RANSAC) can be computed to estimate the similarity of both images. Therefore, we can see that the better performance provided by the RANSAC-based approach comes at a high price in terms of complexity.

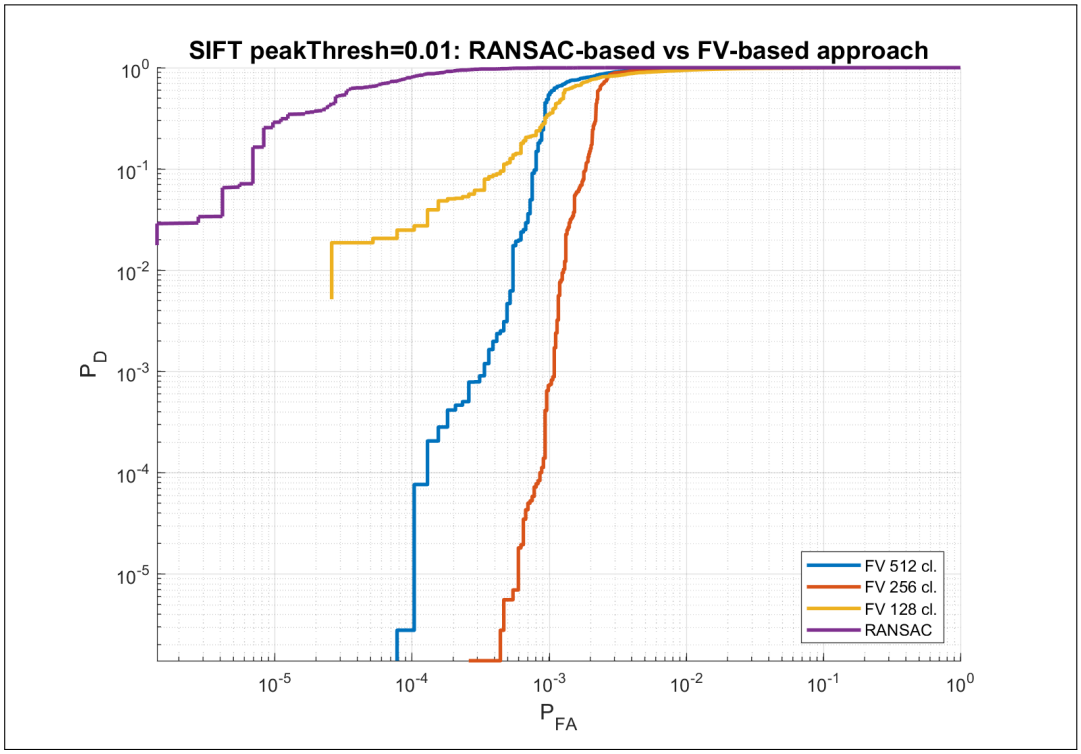

FIGURE 4.21: ROC comparisons: geometrical matching approach vs. descriptor aggregation approach.

4.3.2 Distribution of SIFT features on images

It is an interesting question which descriptors (how which regions) contribute most to the success of a unique identification. Therefore, in this section we present an experiment conducted on enrolled images in order to show the distribution of SIFT features with regards to increasing the peakThreshold parameter. As discussed previously, the peakThresh parameter controls the sensitivity of the SIFT detector. The lower it is set, the more features the algorithm will detect. Having more features allows to get closer to a *fine-grained recognition*, (i.e., detecting very fine details, such as small text area indicating the number of pills in a package, etc.) but this comes at the cost of a high computational complexity in both CPU time and memory. At the same time, a lot of descriptors are detected in the flat regions.

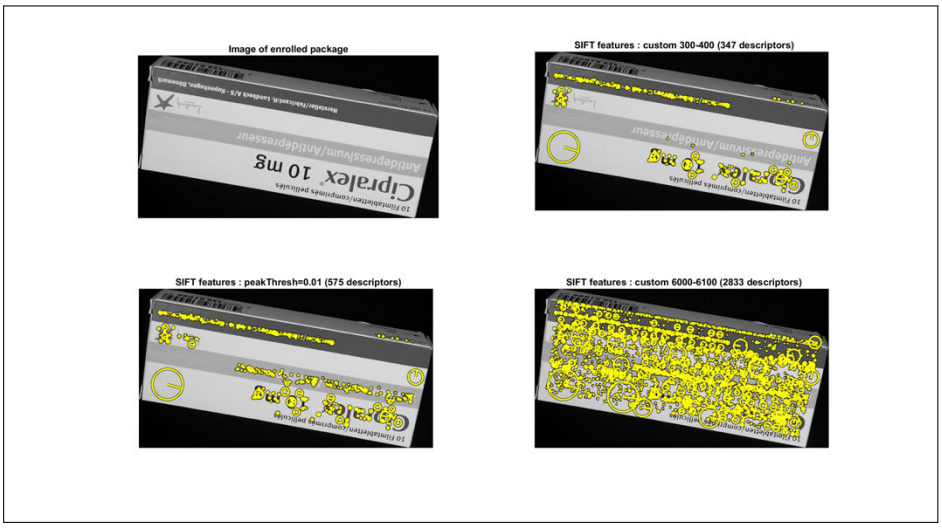

FIGURE 4.22: Distribution of detected SIFT features on the same image for different peakThreshold parameters.

In Figure 4.22, it can be observed that with peakThresh adjusted to obtain a maximum of 300 to 400 features, 347 effective descriptors were extracted. Moreover, it can be noticed that they do not cover some text areas that could be potentially important for the identification of the package.

On the image with peakThresh=0.01, 575 descriptors were extracted. We can see that with this parameter, more areas are covered. For example, the whole text area with "Antidepressivum/Antidépresseur", which has a relatively low contrast, is fully covered by descriptors.

Finally, with a peakThresh adjusted to yield up to 6000-6100 descriptors/image, we can see[10] that 2833 descriptors were extracted. It is interesting to obverse that now almost all of the package area is **uniformly** covered. Indeed, the SIFT descriptors do not seem to aggregate only on specific areas. This is a good property in the context of package recognition. Other type of descriptors are known to have a non-uniform distribution, such as aKaZe [30] and tend to aggregate on specific areas, however, this is not further investigated here.

[10]The exact number of keypoints can be seen by computing in Matlab the size of the matrix containing the extracted keypoints from the image.

4.3.3 SIFT with color information

We briefly investigate some potential problems with regards to the fact that SIFT does not take color information into account.

In Figure 4.23 we can see that many keypoints of the package with red text on the left are matched with point to the right on the package with dark blue text. This can lead to a case of false acceptance as SIFT only considers grayscale version of the images. This seems to be, however, not often the case in the context of pharma package recognition, as indeed in many scenarios, not using color could actually be an advantage. For example, a significant color shifting (e.g., change of color temperature due to different lighting conditions) could occur especially for mobile phone based recognition. If taken into account, this information could produces errors as well.

However, we could mention the existence of a version of the SIFT algorithm which is designed to use color information, namely CSIFT [91]. If color would be considered as needed to be taken into account, one could use the latter mentioned descriptor, or conversely apply the original SIFT algorithm successively to the 3 color channels (R,G,B) of a given image, and produce 3 sets of descriptors for each image. This approach has not been investigated but would clearly add more complexity to the process.

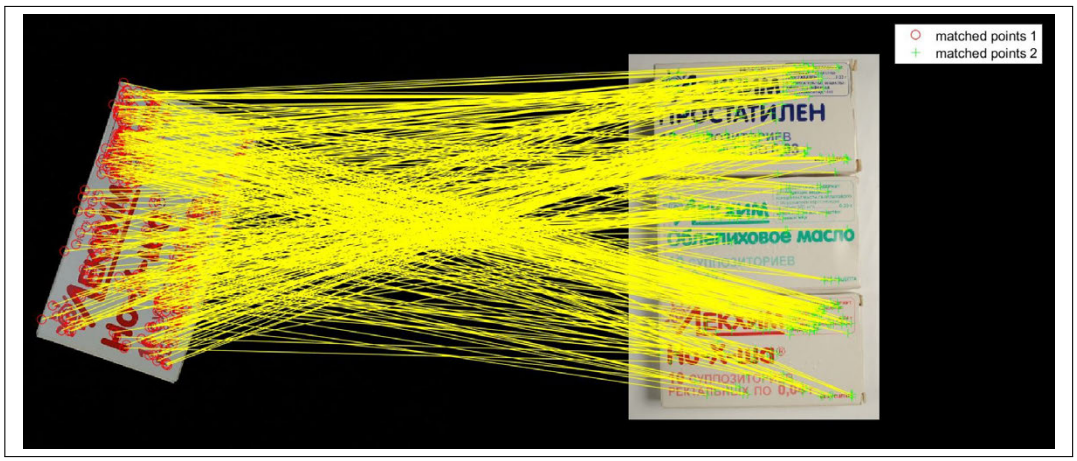

FIGURE 4.23: Example illustrating the risk of false positive or miss due to SIFT lacking color information.

4.4 Conclusions

The experimental results show that increasing the number of clusters of the GMM improves the identification performance. This is illustrated in Figure 4.10. However, the rate of increase of true positives seems to behave logarithmically, as illustrated by Figure 4.20. Therefore, it will most likely not be very useful to increase significantly the number of clusters, as the performance rate will tend to decrease after 256 clusters. Moreover, the more clusters, the higher the complexity, therefore the computational burden of computing the GMM.

It should be added that experiments investigating the impact of the number of descriptors/image for a fixed number of clusters, reveal an unclear behavior for a low number of descriptors per image. It can be concluded that a significant number of descriptors per image (i.e., up to 4000 with peakThresh=0.01), with the highest number of clusters, leads to the best performance in this setup.

With regards to the influence of cropping, we can observe in Figure 4.15 and Figure 4.16, that the performance is quite similar, yet slightly better for the cropped images, as expected. The small increase of performance is most likely due to the fact that not many SIFT descriptors are extracted on the flat (table) surface.

Concerning the comparison between S1 and S2 datasets, we can observe a better performance for S1. This is understandable considering that hand-held packages are most likely not perfectly cropped due to the presence of skin texture and package shadow projection, thus possibly leading to the creation of descriptors unrelated to the package. This difference of performance can be seen in Figure 4.18 and Figure 4.19.

From all these experiments, we can draw a conclusion with regards to *fine-grained* detection: the GMM approach is clearly not accurate enough. Indeed, it fails to obtain and acceptably low percentage of false positives (for a correct detection probability higher than 99%).

However, this approach can be interesting when considering a ”pipeline model”, where we would use increasingly more discriminative (and with higher computational complexity) algorithms at each stage of the aforementioned ”pipeline”. For example, the first stage could consist of a simple and fast recognition method, which would describe the image in a *hollistic* manner, such as a Sum of Squared distances (SSD) between images, image histograms, or for example a global descriptor such as GIST [20]. Subsequently, a basic Bag of Words (BoW) + K-means could be applied and further refined with a clustering by Gaussian Mixture Models. We would have less candidates on a list at each stage of the pipeline. Hopefully, after enough refinements, a geometrical re-ranking [92] method could be applied with a high number of SIFT descriptors extracted from the remaining candidate images, thus possibly leading to a good recognition performance.

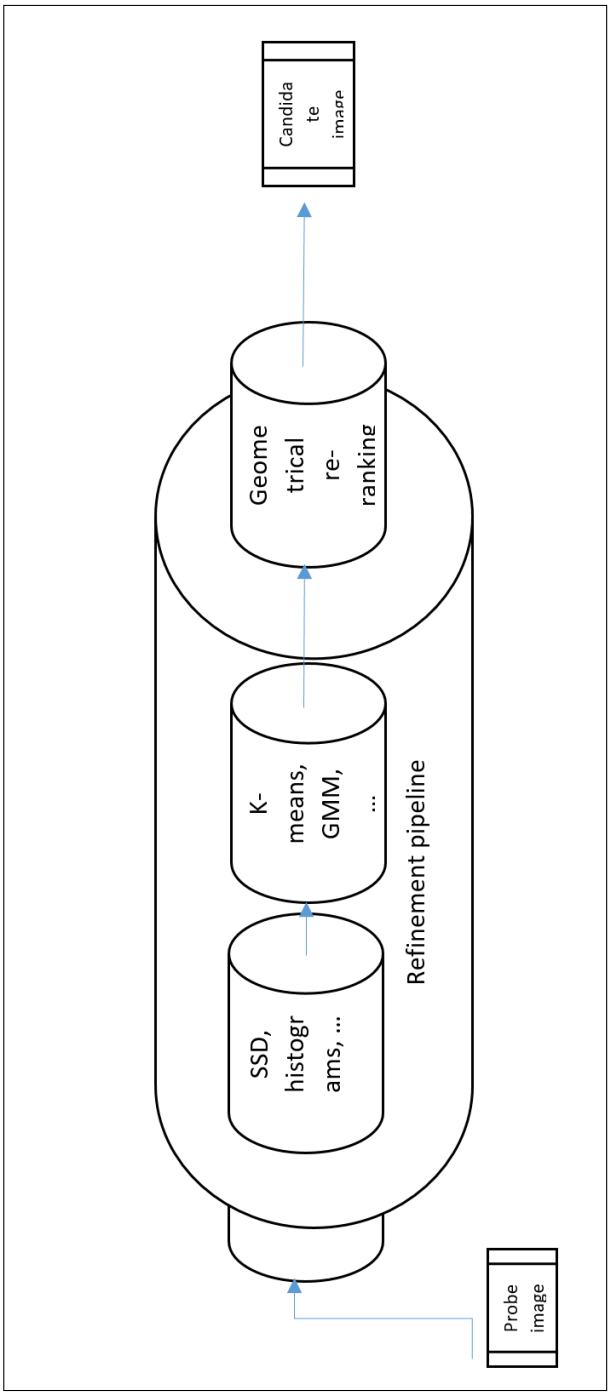

FIGURE 4.24: Illustration of the "pipelining" recognition process.

Chapter 5

Conclusions

In this thesis, we presented PharmaPack, a new database of pharmaceutical packages containing more than 60'000 images of 1000+ unique packages as well as multiple samples of same products. We presented the enrollment setup consisting of three mobile phones positioned at different angles to capture a range of possible viewpoints at which an end user might photograph a package for identification. Moreover, shooting 18 images per phone for a 360 degrees coverage of packages, resulted in 54 images per package. With 8 MegaPixels photos (JPEG compressed), the current size of the database has reached about 65 GB. As it will be released for public usage by the scientific community, we hope to have filled a gap in the already wide amount of computer vision datasets. Indeed, up to our best knowledge there is no database with a large enough amount of objects of the same class, moreover with multiple images of the same objects, in the scope of fine-grained recognition acquired by modern mobile phones.

Moreover, two well-known state-of-the-art approaches were experimented on various subsets of the database in order to investigate the feasibility of fine-grained recognition and to have a baseline for comparison with regards to future work with other algorithms. It can be concluded that approaches such as BoW are largely insufficient to provide acceptable results in terms of false positive and false negative rate. We recall that the investigated method, namely Fisher vectors encoding relies on soft clustering of SIFT descriptors using Gaussian mixture models. Moreover, this approach discards any geometric information about SIFT keypoints. Therefore, not so surprisingly, the results are relatively poor.

The second investigated approach, based on geometrical alignment of keypoints using RANSAC, produced better results than the non-geometrical method. Indeed, it seems that an important amount of information is lost if positions of keypoints are not taken into account. The drawbacks of this approach are the complexity (CPU time) required for direct matching of descriptors followed by RANSAC alignment, and the amount of stored data. Moreover, a 100% recognition still cannot be achieved with a small enough probability of false positives. This is clearly due to the fact that the local descriptor based methods investigated in this thesis, are not well adapted to the problem of fine-grained recognition. Figure 3.20 provides examples, which show that it was often packages of the same drug but with very minor differences (e.g., number of pills) that were falsely recognized. The best result achieved with RANSAC alignment, was 1.7% probability of false acceptance for a 100% probability of correct detection (as shown in Table 3.4). This percentage may seem low, however, we must think of it in the perspective of a very large database of, for example, 100'000 or even 1'000'000 images. Therefore, a 1.7% amount of error would result in 1700 or 17'000

falsely recognized packages, which is clearly not acceptable.

As already discussed, correct identification of pharmaceutical packages is the first and essential step towards detection of counterfeit products. We have shown the challenge of fine-grained recognition through our experiments and pointed out the need of more sophisticated or specialized techniques to deal with this problem. Future work can be dedicated to the creation of specific descriptors based on text extraction techniques, for example.

Bibliography

[1] R. Cockburn, P. N. Newton, E. K. Agyarko, D. Akunyili, and N. J. White, "The global threat of counterfeit drugs: why industry and governments must communicate the dangers," *PLoS Med*, vol. 2, no. 4, p. e100, 2005.

[2] S. Voloshynovskiy, S. Pereira, T. Pun, J. J. Eggers, and J. K. Su, "Attacks on digital watermarks: classification, estimation based attacks, and benchmarks," *IEEE communications Magazine*, vol. 39, no. 8, pp. 118–126, 2001.

[3] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, *Digital watermarking and steganography*. Morgan Kaufmann, 2007.

[4] J. Fridrich, *Steganography in digital media: principles, algorithms, and applications*. Cambridge University Press, 2009.

[5] S. Voloshynovskiy, A. Herrigel, N. Baumgaertner, and T. Pun, "A stochastic approach to content adaptive digital image watermarking," in *International Workshop on Information Hiding*, pp. 211–236, Springer, 1999.

[6] M. Yagüe, P. Wolf, M. Steinebach, and K. Diener, "Complementing drm with digital watermarking: mark, search, retrieve," *Online Information Review*, vol. 31, no. 1, pp. 10–21, 2007.

[7] R. Villán, S. Voloshynovskiy, O. Koval, J. Vila, E. Topak, F. Deguillaume, Y. Rytsar, and T. Pun, "Text data-hiding for digital and printed documents: Theoretical and practical considerations," in *Security, Steganography, and Watermarking of Multimedia Contents*, p. 607212, 2006.

[8] R. Villán, S. Voloshynovskiy, O. J. Koval, F. Deguillaume, and T. Pun, "Tamper-proofing of electronic and printed text documents via robust hashing and data-hiding.," in *Security, Steganography, and Watermarking of Multimedia Contents*, p. 65051T, 2007.

[9] P.-J. Chiang, N. Khanna, A. K. Mikkilineni, M. V. O. Segovia, S. Suh, J. P. Allebach, G. T.-C. Chiu, and E. J. Delp, "Printer and scanner forensics," *IEEE Signal Processing Magazine*, vol. 26, no. 2, pp. 72–83, 2009.

[10] S. Voloshynovskiy, T. Holotyak, and P. Bas, "Physical object authentication: Detection-theoretic comparison of natural and artificial randomness," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pp. 2029–2033, IEEE, 2016.

[11] S. Voloshynovskiy, M. Diephuis, F. Beekhof, O. Koval, and B. Keel, "Towards reproducible results in authentication based on physical non-cloneable functions: The forensic authentication microstructure optical set (famos)," in *Proceedings of IEEE International Workshop on Information Forensics and Security*, (Tenerife, Spain), December 2–5 2012.

[12] M. Diephuis and S. Voloshynovskiy, "Physical object identification based on famos microstructure fingerprinting: comparison of templates versus invariant features," in *Image and Signal Processing and Analysis (ISPA), 2013 8th International Symposium on*, pp. 119–123, IEEE, 2013.

[13] F. Farhadzadeh, S. Voloshynovskiy, and O. Koval, "Performance analysis of content-based identification using constrained list-based decoding," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 5, pp. 1652–1667, 2012.

[14] S. Ferdowsi, S. Voloshynovskiy, and D. Kostadinov, "Content identification: binary content fingerprinting versus binary content encoding," in *IS&T/SPIE Electronic Imaging*, pp. 90280P–90280P, International Society for Optics and Photonics, 2014.

[15] S. Voloshynovskiy, O. Koval, F. Beekhof, and T. Pun, "Robust perceptual hashing as classification problem: decision-theoretic and practical considerations," in *Multimedia Signal Processing, 2007. MMSP 2007. IEEE 9th Workshop on*, pp. 345–348, IEEE, 2007.

[16] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of research and development*, vol. 3, no. 3, pp. 210–229, 1959.

[17] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of research and development*, vol. 44, no. 1.2, pp. 206–226, 2000.

[18] M. Minsky and S. Papert, "Perceptrons: Anlntroduction to computational geometry," 1969.

[19] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain.," *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[20] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International journal of computer vision*, vol. 42, no. 3, pp. 145–175, 2001.

[21] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 886–893, IEEE, 2005.

[22] B. Girod, V. Chandrasekhar, D. M. Chen, N.-M. Cheung, R. Grzeszczuk, Y. Reznik, G. Takacs, S. S. Tsai, and R. Vedantham, "Mobile visual search," *IEEE signal processing magazine*, vol. 28, no. 4, pp. 61–76, 2011.

[23] S. Voloshynovskiy, M. Diephuis, and T. Holotyak, "Mobile visual object identification: from sift-bof-ransac to sketchprint," in *SPIE/IS&T Electronic Imaging*, pp. 94090Q–94090Q, International Society for Optics and Photonics, 2015.

[24] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, vol. 2, pp. 1150–1157, Ieee, 1999.

[25] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[26] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," *Computer vision–ECCV 2006*, pp. 404–417, 2006.

[27] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

[28] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 2548–2555, IEEE, 2011.

[29] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," *Computer Vision–ECCV 2010*, pp. 778–792, 2010.

[30] P. F. Alcantarilla and T. Solutions, "Fast explicit diffusion for accelerated features in nonlinear scale spaces," *IEEE Trans. Patt. Anal. Mach. Intell*, vol. 34, no. 7, pp. 1281–1298, 2011.

[31] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint," in *Computer vision and pattern recognition (CVPR), 2012 IEEE conference on*, pp. 510–517, Ieee, 2012.

[32] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 2564–2571, IEEE, 2011.

[33] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[34] M. Hofmann, "Support vector machines-kernels and the kernel trick," *An elaboration for the Hauptseminar Reading Club SVM*, 2006.

[35] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[36] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, 2015.

[37] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[38] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[39] H. S. Chang, M. C. Fu, J. Hu, and S. I. Marcus, "Google deep mind's alphago," *OR/MS Today*, 2016.

[40] R. Dahl, M. Norouzi, and J. Shlens, "Pixel recursive super resolution," *arXiv preprint arXiv:1702.00783*, 2017.

[41] A. G. Ivakhnenko and V. G. Lapa, "Cybernetic predicting devices," tech. rep., DTIC Document, 1966.

[42] A. Ivakhnenko, Y. V. Koppa, and W. S. Min, "Polynomial and logical theory of dynamic systems (part 2)," *Sov. Automat. Contr.*, vol. 15, no. 3, 4, pp. 11–30, 1970.

[43] A. G. Ivakhnenko and V. G. Lapa, "Cybernetics and forecasting techniques," 1967.

[44] D. O. Hebb, "The organization of behavior. a neuropsychological theory. john wiley and sons," *New York*, 1949.

[45] S. Lowel and W. Singer, "Selection of intrinsic horizontal connections in the visual cortex by correlated neuronal activity," *Science*, vol. 255, no. 5041, p. 209, 1992.

[46] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," tech. rep., DTIC Document, 1985.

[47] D. Rumelhart, "David e. rumelhart, geoffrey e. hinton, and ronald j. williams," *Nature*, vol. 323, pp. 533–536, 1986.

[48] "Deep Networks: Overview." http://ufldl.stanford.edu/wiki/index.php/Deep_Networks:_Overview, accessed 2017-26-05.

[49] Y. LeCun, Y. Bengio, *et al.*, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[50] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Handwritten digit recognition with a back-propagation network, 1989," in *Neural Information Processing Systems (NIPS)*.

[51] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," in *Advances in Neural Information Processing Systems 2* (D. S. Touretzky, ed.), pp. 396–404, Morgan-Kaufmann, 1990.

[52] I. Sobel and G. Feldman, "A 3x3 isotropic gradient operator for image processing," *a talk at the Stanford Artificial Project in*, pp. 271–272, 1968.

[53] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[54] J. M. Prewitt, "Object enhancement and extraction," *Picture processing and Psychopictorics*, vol. 10, no. 1, pp. 15–19, 1970.

[55] Y. LeCun *et al.*, "Lenet-5, convolutional neural networks," *URL: http://yann. lecun. com/exdb/lenet*, 2015.

[56] S. A. Radzi, "A matlab-based convolutional neural network approach for face recognition system," 2016.

[57] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[58] R. W. Smith, *The Extraction and Recognition of Text from Multimedia Document Images*. PhD thesis, University of Bristol, 1987.

[59] R. Smith, "An overview of the tesseract ocr engine," in *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, vol. 2, pp. 629–633, IEEE, 2007.

[60] M. Zahedi and S. Eslami, "Farsi/arabic optical font recognition using sift features," *Procedia Computer Science*, vol. 3, pp. 1055–1059, 2011.

[61] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," *Computer vision and Image understanding*, vol. 106, no. 1, pp. 59–70, 2007.

[62] G. H. Griffin and A. Perona, "P. the caltech-256," *Caltech Technical Report, Tech. Rep.*, 2012.

[63] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 248–255, IEEE, 2009.

[64] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[65] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[66] R. Rothe, R. Timofte, and L. V. Gool, "Deep expectation of real and apparent age from a single image without facial landmarks," *International Journal of Computer Vision (IJCV)*, July 2016.

[67] Y. LeCun, C. Cortes, and C. J. Burges, "Mnist handwritten digit database," *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, vol. 2, 2010.

[68] J.-M. Geusebroek, G. J. Burghouts, and A. W. Smeulders, "The amsterdam library of object images," *International Journal of Computer Vision*, vol. 61, no. 1, pp. 103–112, 2005.

[69] V. R. Chandrasekhar, D. M. Chen, S. S. Tsai, N.-M. Cheung, H. Chen, G. Takacs, Y. Reznik, R. Vedantham, R. Grzeszczuk, J. Bach, *et al.*, "The stanford mobile visual search data set," in *Proceedings of the second annual ACM conference on Multimedia systems*, pp. 117–122, ACM, 2011.

[70] A. Khosla, N. Jayadevaprakash, B. Yao, and F.-F. Li, "Novel dataset for fine-grained image categorization: Stanford dogs," in *Proc. CVPR Workshop on Fine-Grained Visual Categorization (FGVC)*, vol. 2, p. 1, 2011.

[71] O. Taran, S. Rezaeifar, O. Dabrowski, J. Schlechten, T. Holotyak, and S. Voloshynovskiy, "Pharmapack: mobile fine-grained recognition of pharma packages," in *EUSIPCO, Kos, Grece*, 2017.

[72] S. M. Kay, "Fundamentals of statistical signal processing, vol. ii: Detection theory," *Signal Processing. Upper Saddle River, NJ: Prentice Hall*, 1998.

[73] B. Brown, *Cinematography: theory and practice: image making for cinematographers and directors*. Taylor & Francis, 2013.

[74] A. Angus, "More on 18% Gray." http://stonerosephotos.com/blog/2010/02/more-on-18-gray/, accessed 2017-01-05.

[75] K. McLaren, "Xiii—the development of the cie 1976 (l* a* b*) uniform colour space and colour-difference formula," *Coloration Technology*, vol. 92, no. 9, pp. 338–341, 1976.

[76] G. M. Galdino, J. E. Vogel, and C. A. Vander Kolk, "Standardizing digital photography: it's not all in the eye of the beholder.," *Plastic and reconstructive surgery*, vol. 108, no. 5, pp. 1334–1344, 2001.

[77] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[78] E. Vincent and R. Laganiére, "Detecting planar homographies in an image pair," in *Image and Signal Processing and Analysis, 2001. ISPA 2001. Proceedings of the 2nd International Symposium on*, pp. 182–187, IEEE, 2001.

[79] R. Sorschag, "Object detection with semi-local features," in *Pattern Recognition-Applications and Methods*, pp. 23–35, Springer, 2013.

[80] M. Diephuis, S. Voloshynovskiy, and T. Holotyak, "Sketchprint: physical object micro-structure identification using mobile phones," in *Signal Processing Conference (EUSIPCO), 2015 23rd European*, pp. 834–838, IEEE, 2015.

[81] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 2963–2970, IEEE, 2010.

[82] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the fisher vector: Theory and practice," *International journal of computer vision*, vol. 105, no. 3, pp. 222–245, 2013.

[83] F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pp. 1–8, IEEE, 2007.

[84] K. Pearson, "Contributions to the mathematical theory of evolution," *Philosophical Transactions of the Royal Society of London. A*, vol. 185, pp. 71–110, 1894.

[85] C. Améndola, J.-C. Faugere, and B. Sturmfels, "Moment varieties of gaussian mixtures," *arXiv preprint arXiv:1510.04654*, 2015.

[86] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on statistical learning in computer vision, ECCV*, vol. 1, pp. 1–2, Prague, 2004.

[87] F. Dellaert, "The expectation maximization algorithm," tech. rep., Georgia Institute of Technology, 2002.

[88] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.

[89] A. Vedaldi and B. Fulkerson, "Vlfeat: An open and portable library of computer vision algorithms," in *Proceedings of the 18th ACM international conference on Multimedia*, pp. 1469–1472, ACM, 2010.

[90] K. Chatfield, V. S. Lempitsky, A. Vedaldi, and A. Zisserman, "The devil is in the details: an evaluation of recent feature encoding methods.," in *BMVC*, vol. 2, p. 8, 2011.

[91] A. E. Abdel-Hakim and A. A. Farag, "Csift: A sift descriptor with color invariant characteristics," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2, pp. 1978–1983, IEEE, 2006.

[92] X. Wang, M. Yang, and K. Yu, "Efficient re-ranking in vocabulary tree based image retrieval," in *Signals, Systems and Computers (ASILOMAR), 2011 Conference Record of the Forty Fifth Asilomar Conference on*, pp. 855–859, IEEE, 2011.