KU LEUVEN

Biometrics System Concepts - Assignment 2

Deniz Soysal - r0875700

April 29, 2022

1 Introduction

This assignment is divided into 2 parts : in the first part, we will implement and test a keypoint-based fingerprint and iris recognition system and fuse the two systems together The goal is to, based on an fingerprint and an iris image which identity is unknown, find the top matching identity in the database. In the second part (section 5), we will create a biometric identification system based on palmprint recognition.

The additional task chosen is therefore " J. [.] Create a biometric identification or authentication system on a modality of choice and evaluate it (6pt)".

2 Fingerprint Recognition

In this part of the assignment, given the fingerprint image of the perpetrator and a dataset of 100 fingerprint images corresponding to 100 identities, we have to find the fingerprint which is the most similar to the one of the perpetrator.

2.1 Question 1 : Is the given similarity function a good metric to quantify the distance between two fingerprints? Is it reliable enough to incriminate a suspect? What are its limitations?

The similarity scores between the perpetrator fingerprint and the fingerprints of the database are computed using a mean squared difference between the pixels values of the 2 images. Applying directly this method is not robust. Indeed, the fingerprints have different scales, orientation, translation, etc. The fingerprints needs to be aligned to generate a score that represent truly if the fingerprint are matching or not.

Moreover, looking at the difference of the pixel values is not robust, even for relatively well-aligned images : if all pixels are shifted by one to the right for example, this will have a big impact on the similarity. We have to consider extracting robust features before looking at the similarity between 2 images.

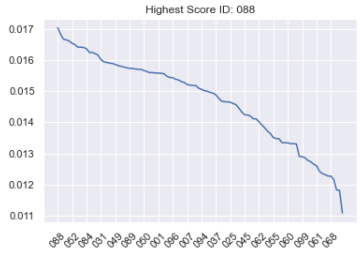


Figure 1: Similarity score obtained

As we can see at figure 1, there is no clear separations between the different identities : the similarity scores are pretty close to each other. Note that this is simply a geometric distance between the pixels. At Question 3, after extracting features, we will look at the distance between the feature vectors. At Question 5, we will use both the geometric distance and the distance between the feature vectors.

2.2 Question 2 : Are all the keypoint matches accurate? Are they expected to be? Explain why.

After having enhanced the images and extracted features using the ORB method, we use a Brute-Force Matcher to find pairs of matching features between images. The idea is quite simple : based on a distance similarity, return the pairs of keypoints with lowest distance. ORB is a binary string based descriptor, so the distance measure we use is the Hamming Distance, which computes the number of positions at which the two symbols are different.

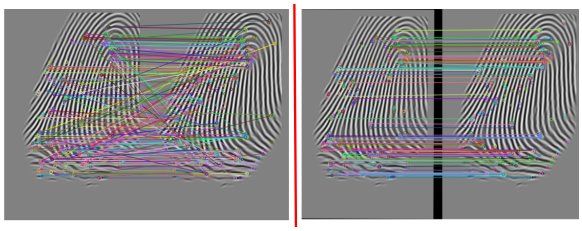


Figure 2: Brute-Force Keypoint Matching, before alignment and after alignment

As we can see at the left side of the red line of Figure 2, before alignment, the matches are not accurate at all. The reason is that the fingerprints are not aligned, and doing matches on non-aligned fingerprints is challenging : we can have translational variance and rotational variance between the images. This will make it very difficult to match the two images : we must first align them. To do so, we will use the keypoints and do affine transformations with the RANSAC method. At the right side of the red line of Figure 2, we can observe the results after aligning the two images : the matches are much more accurate.

2.3 Question 3 : Choose a global feature similarity function, (e.g. you can start from euclidean distance between the reduced sets of KeyPoints and count the values above a threshold).

The global feature similarity function we will use is to, based on the reduced sets of KeyPoints, count the values of pairs with distance above the threshold. We will compare several distance

measures, and see which one performs the best. Then, we will tune the threshold for this distance measure. According to [1], published in the International Journal of Pattern Recognition and Artificial Intelligence, authors found that Minkowski, Sorgel, Jaccard, Motyka perform well for fingerprint recognition. We will try some of these measures, with also more classical ones such as Mean Squared Difference and Euclidian Distance. The metrics we will implement and compare are Euclidian, MSS, Chebyshev, Minkowski.

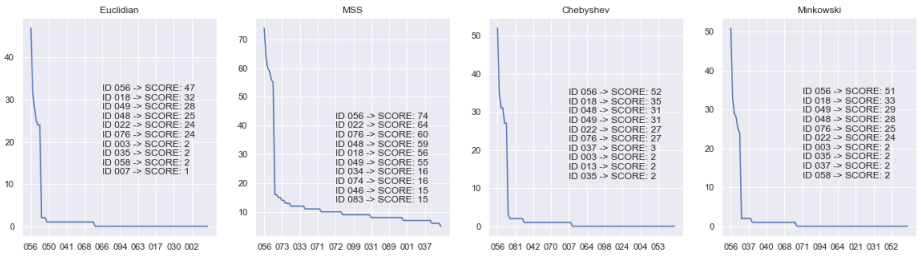


Figure 3: Comparison between the 4 metrics

From Figure 3, we can clearly see that MSS is the worst performing one : the separation between the top similarities is small compared to the other metric measures. For the 3 other metric, it is difficult to choose one from this plot. To compare them, we will normalize the similarity scores given by each metric, and look at the difference between the top similarity scores : Difference between Top 1 and Top 2, and also difference between Top 2 and Top 3. This will give how well the distance measure is able to separate the top matching pairs of image. In our case, we want to be able to decide if the matching image is the perpetrator or not, so we want a large separation between the Top 1 match and the Top 2 match. The results obtained are presented at Table 1.

	Eucl	MSS	Cheb	Mink
Top 1 - Top 2	0.319	0.145	0.327	0.353
Top 2 - Top 3	0.085	0.058	0.077	0.078

Table 1: Differences of top scores

Minkowski seems like the distance metric that is able to separate the most the top matching points ! Note that we use Minkowski with p=3 (p being the order of the norm of the difference between the 2 points). With using simply p=2, Minkowski would be the same as the Euclidian distance

2.4 Question 4 : Visualize the scores and determine a score threshold to discriminate the matching fingerprints. Explain how you determine the threshold.

We have already, at Question 3, visualized the scores for the different metrics. Here, we will try to tune the threshold, and visualize its effect. We expect that the threshold will have an effect on the False Match Rate. If we set the threshold high, the False Match Rate will decrease (because we consider the keypoint being similar if the distance is lower than the threshold). However, we may also increase the False Rejection Rate, by rejecting keypoint that are in reality similar.

We use the Minkowski Distance for different threshold values : 0.5, 1.0, 1.5, 2.0. The similarity scores are displayed at Figure 4. As we did to compare the different metrics, let's normalize the similarity scores and look at the difference between the top similarity scores. The results are displayed at Table 2.

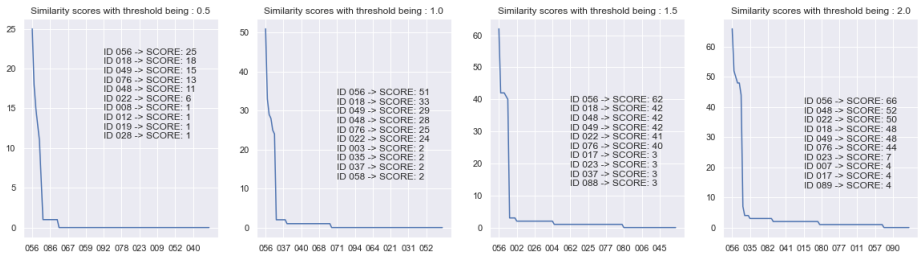


Figure 4: Comparison between different threshold values - Minkowski Distance

Treshold Value	0.5	1.0	1.5	2.0
Top 1 - Top 2	0.28	0.353	0.323	0.212
Top 2 - Top 3	0.12	0.078	0.0	0.03

Table 2: Differences of top scores

We can see that having a threshold of 1.0 allows us to have a separation of 0.353 between the Top 1 match and the Top 2 match, which is the best separation compared to the other threshold values.

2.5 Question 5 : Choose a hybrid feature similarity function that makes use of both the geometric distance and the feature distance of the keypoints. Using this hybrid function, visualize and assess the matches.

Here, we will follow the best scheme of the Global Features (use Minkowski with a threshold of 1.0), except that we will also use the geometric distance as an additional threshold. This will allow to have a more robust feature representation. The result are shown at Table 3. We can see that compared to Table 2 where we had 0.353 as best result, we are able to have a better separation of 0.375 between the Top 1 match and the Top 2 match.

	Hybrid Features
Top 1 - Top 2	0.375
Top 2 - Top 3	0.031

Table 3: Differences of top scores

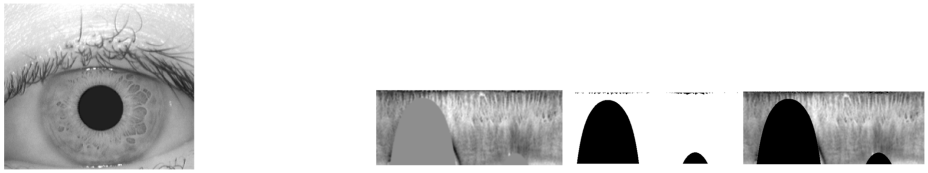
3 Iris Recognition

We can still not confidently decide who is the perpetrator... But thanks to our friends from CSI New York, we have high quality images of the iris of the perpetrator ! As we did with

the fingerprints, We can compare this iris with our database.The iris identifier has a very good uniqueness, so it should be very accurate ! Iris Recognition is one of the few biometrics that also can be used in identification.

3.1 Question 6 : Check out iris_perpetrator.png. Where do you see difficulties? What kind of similarity measures do you expect to work best?

The image of the iris of the perpetrator, displayed at Figure 5 a), has high quality, and the iris is well visible. For the perpetrator image, eye lids does not hide the iris (even if for some images in the database, it is the case). However, there are some white spots on the right side of the iris, which hides this part of the iris. This will increase the difficulty of matching the iris with the database in the case where these spots appeared after the time when the images in the database were taken.



(a) Perpetrator Iris Image (b) In order : enhanced iris, mask, segmented iris

Figure 5: Original iris image, and image after transformation

After having enhanced and normalise the image (by unwraping the circular region, results displayed at Figure 5 b)), we will construct the similarity matrix. We will use local features, with the Hamming distance (which is usually used in Iris Recognition).

3.2 Question 7 : Construct a similarity table with the iris images. Use local OR global matches in order to get scores. Use the scores you get to determine the perpetrator

The iris scores we obtain are presented at Figure 6. We can see that again, there is no clear separation between the top matches !

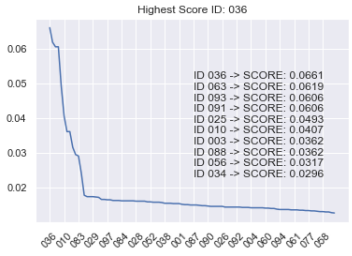


Figure 6: Iris similarity scores

This will become even more obvious after constructing the Top difference table as we did with fingerprint :

	Iris scores
Top 1 - Top 2	0.079
Top 2 - Top 3	0.024

Table 4: Differences of top scores

4 Multimodal System - Fingerprint and Iris

As we have seen, the Fingerprint scores and the Iris scores are difficult to interpret : the top matching scores are pretty close, which makes the identification quite hard. We need to find other ways to improve our identification : let's try a Multimodal System by fusing the Fingerprints and the Iris scores

4.1 Question 8 : Fuse your iris and fingerprint biometric system on the score level to solve the murder case! Do you feel confident in your prediction? How do you fuse the scores? Why?

Here, we will fuse the biometric systems on the score level. We will use the hybrid features, and normalize the values of the features before adding them. To fuse the two biometric system, we will create a new score, based on a weighted sum between the scores of the two systems. We will choose the weight to give to each score based on the results we get.

The weights we will try are : 0.4 for the fingerprint, 0.6 for the iris ; 0.5 for the fingerprint, 0.5 for the iris ; 0.6 for the fingerprint, 0.4 for the iris ; 0.7 for the fingerprint, 0.3 for the iris. The results are displayed Figure 7. The proportion "0.7 for the fingerprint, 0.3 for the iris" seems to be the one that discriminate the best the top matching score

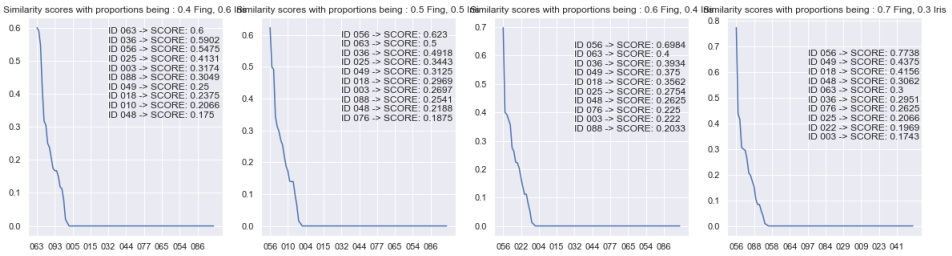


Figure 7: Iris similarity scores

This will become even more obvious after constructing the Top difference table as we did before.

Proportions	0.4*F + 0.6*I	0.5*F + 0.5*I	0.6*F + 0.4*I	0.7*F + 0.3*I
Top 1 - Top 2	0.016	0.197	0.427	0.435
Top 2 - Top 3	0.071	0.013	0.009	0.028

Table 5: Differences of top scores

Table 5 shows clearly that having a proportion of 0.7 for the fingerprint scores and 0.3 for the iris scores allows us to have a clear distinction between the Top 1 Match and the Top 2 Match. This difference of 0.435 is also better compared to all previous results obtained at Table 1, 2,

3,4, showing that **using a Multimodal system is better than the individual systems** in order to discriminate the perpetrator.

5 Additional Task J : Palmprint recognition

For this part, we will develop a **Contact Less Palm recognition system**. The dataset we will use is available at [2]. Note that the authors have published 2 different datasets : one *original* dataset for raw images, one *ROI* dataset for images where the palm have been detected and the different palms have been aligned. For the sake of simplicity, we will do our system based on the ROI dataset. For the ones interested on how to extract the ROI from the raw images, the authors have given a comprehensive explanation in their paper [3].

The framework here will be *identification* : based on an input palm print, compare it to the database and find the corresponding identity. It can be considered as a "N+1" classification setup, with "N" being the number of identities in the databse, and the "+1" corresponding to the case where the input palmprint does not belong to the database.

5.1 The Dataset

In the ROI [1] dataset, there are two folders : session 1 and session 2, corresponding to two different sessions of measurements where the palmprints have been acquired. Each folder contain 6000 images corresponding to 600 different palms (so each identity has 10 images at each session). Each palm is considered as a different identity. We will merge the two sessions of measurement, so at the end we will have a dataset of 12000 images with 600 identities. Due to limited computation capacity, we will not use the whole dataset, but only the first 100 identities [2] : so, the subdataset we will use has 2000 images from 100 identities (each identity has 20 images).

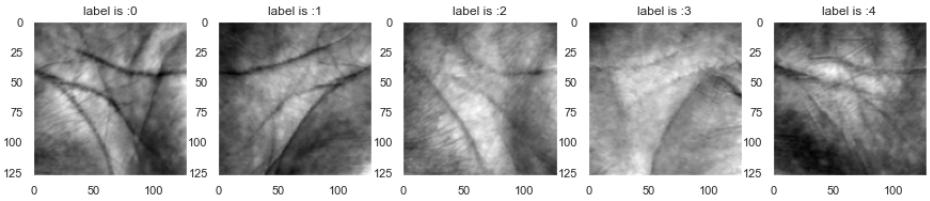


Figure 8: Some palm images, with "label" being the corresponding identity

5.2 Feature extraction

The advantage of working on the ROI dataset is that we will directly dive into the interesting part of a biometric system : feature extraction.

5.2.1 What are the features in a palmprint ?

As in fingerprint with edges, valleys, minutiae, etc, palmprints have also unique features among individuals. Note that here, we use the *palm prints*, not the *palm veins*, which is another biometric identifier. However, some systems combining the 2 identifier exist [4].

A palmprint has as features texture, wrinkles, principle lines, ridges, and minutiae points that we can use to build a feature vector. Different methods have been developped to extract the

[1] Note that ROI stands for Region Of Interest. Indeed, the palm have been detected and aligned in this dataset
[2] Images from 00001.bmp to 01000.bmp for session 1 and session 2

features of a palmprint, inspired by IrisCode of Daugman. The most popular methods for palmprint recognition are PalmCode [6],CompCode [7]. Other more classical feature extraction methods such as SIFT [8] have also been used. In general, authors use SIFT features to fuse the scores of their method with the scores of the SIFT features to have a more robust system.

However, as more classical feature extraction methods have already been exploited in the first part of this assignment, the approach we have chosen is to do a deep learning based feature extraction pipeline.

5.2.2 Feature extraction method : Deep Learning

CNN, with residual layers using triplet loss have already been applied successfully on palm print recognition [9]. I decided to create a pipeline a bit different, without using residual layers, and using an SVM at the end to classify the input palm print. Remember that we are in an identification setup : based on an input palmprint, we classify it into one of the N identities or into the "unknown" identity (so building a classifier with "N+1" classes).

Definition of the network

The network should be able to learn embeddings of the images in a robust way : translation, rotation (even if the images have been aligned), illumination changes should not have a big impact on the embeddings created from the network. Thus, using CNN is well suited. The CNN will output a feature vector representing the image, called **embedding**. The purpose of using *Triplet Loss* is to have well separated embeddings between different classes. The full pipeline is presented at Figure 9.

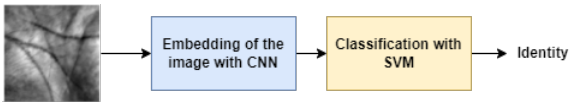


Figure 9: Palm Recognition Pipeline

Triplet Loss

Triplet Loss is used a lot, especially in Face Recognition systems. The idea is quite simple but elegant.

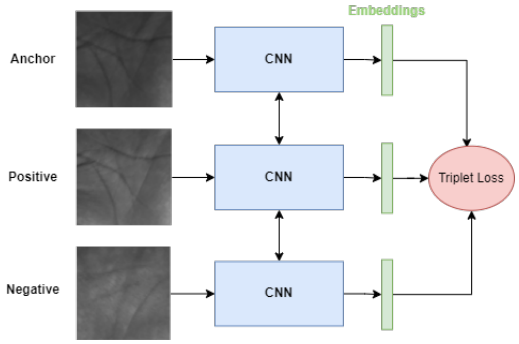


Figure 10: Training of the embedding extraction step

At Figure 10, we present how the CNNs are trained. We use Triplet Loss. The *Anchor* is one image of identity "A", *Positive* is all images from identity "A", *Negative* is all images that are not identity "A". Our goal is to train the network to maximize the distance, in the feature space, of images belonging to different classes and minimize the distance, in the feature spaces, of images belonging to the same class. The Triplet Loss is defined as :

$$\sum_{i}^{N} \|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \tag{1}$$

Where x_i represent an input sample (image), $f(x_i)$ represent the embedding computed for the input sample, α represent the margin between positive and negative pairs (threshold value), and the exponents "a,p,n" represent respectively "anchor, positive and negative"

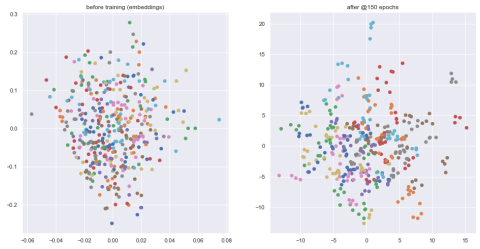
If we do not consider the threshold α, the loss function is then minimal when "distance anchor $\rightarrow$ positive" is small and "distance anchor $\rightarrow$ negative" is big, which is what we want ! Therefore, by minimizing the Triplet Loss function, we will maximize our objective.

5.2.3 Results and discussion

Training of the CNN

The Triplet Loss evolution through 150 epoch is plotted at Figure 11 a). One may wonder why the validation loss is lower than the training loss : in this case, we suppose that it was because the samples in the validation set were easier to separate than the ones in the training set. Moreover, to see if the CNN helps really to separate the classes,we have performed a Principal Component Analysis, before and after applying the CNN. The results are plotted at Figure 11 b).

(a) Triplet Loss Training



(b) PCA, without CNN and with CNN

Figure 11: Training and PCA

Even if the class are not well separable after 150 epoch, we can still see some clusters. Let's also keep in mind that we visualize the data in 2 dimensions, and we will apply a non-linear classifier to it.

To be really able to see if having the embeddings with triplet loss, let's compare the performance of an SVM on the data before applying our pipeline and the performance of an SVM on the embedding obtained by our pipeline. At Table 6, we can clearly see that extraction the embeddings of the palmprint with the CNN helps : we obtain a Rank 1 Recognition Rate of 0.9875, compared to a Rate of 0.89 when only using the SVM.

	Rank 1 Recognition Rate
SVM	0.89
CNN + SVM	0.9875

Table 6: Rank 1 Recognition Rate

CMC

By displaying the CMC curve, at Figure 12, we are able to see that our CMC curve is very steep, which shows the high performance of our system.

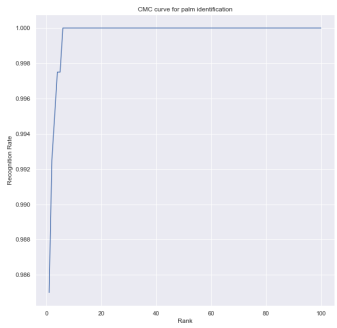


Figure 12: CMC curve

References

[1] Sangita D., Bharkad and Manesh Kokare (2011). Performance Evaluation of Distance Metrics : Application to fingerprint recognition. *International Journal of Pattern Recognition and Artificial Intelligence*

[2] https://cslinzhang.github.io/ContactlessPalm/?fbclid=IwAR27ef6k9NEFq-bAkQVV63rRy-PZM0P7uQdCrwWIQjO2JBABT3jql48Hk9k

[3] Lin Zhanga,Lida Li, Anqi Yang, Ying Shen, Meng Yang. Towards contactless palmprint recognition: A novel device, a new benchmark, and a collaborative representation based identification approach. *Pattern Recognition*

[4] Jian-Gang Wang, Wei-Yun Yau, Andy Suwandy, Eric Sung. Person recognition by fusing palmprint and palm vein images based on "Laplacianpalm" representation. *Pattern Recognition*

[6] A. Kumar, H.C. Shen Palmprint identification using palmcodes. *https://ieeexplore.ieee.org/document/1410434*

[7] A.W.-K. Kong, D. Zhang Competitive coding scheme for palmprint verification. *https://ieeexplore.ieee.org/abstract/document/1334184*

[8] Aythami Morales, Miguel A. Ferrer, Ajay Kumar Improved palmprint authentication using contactless imaging. *https://ieeexplore.ieee.org/document/5634472*

[9] Yang Liu, Ajay Kumar Contactless Palmprint Identification Using Deeply Learned Residual Features. *https://ieeexplore.ieee.org/document/8963760*