Biometrics system concepts

Assignment 3: Face recognition

Luca Stradiotti 0876133

1 Introduction

In this assignment, the focus is on implementing and testing a face recognition system based on old-school computer vision techniques and a deep learning approach.

Face recognition is one of the most used biometrics system since its deployment and implementation are easy and it does not require any interaction by the end user.

2 Reading the data

In this work the Caltech dataset is used because it contains raw images and its size is too large. The dataset consists of 445 images belonging to 26 unique people; the used version is a modification of the original one because 5 images were removed since they did not match the corresponding individual. The dataset contains many photos for each individual considering different conditions of lighting, background and facial expressions.

Figure 1: Different images of an individual varying lighting conditions, background and facial expressions

3 Face detection

Even if the Caltech dataset provides also bounding box coordinates to crop the faces, they are computed from scratch using a face detector.

In this work the HAAR Cascade face detector is used. This method was proposed by Paul Viola and Michael Jones in the paper "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. Initially, the classifier is trained on images of faces (positive examples) and images without faces (negative examples). HAAR features are extracted using the sum of pixels under rectangular areas of different shape and the concept of integral image is introduced to reduce the computational cost of this step. Then, since most of the calculated features are irrelevant, Adaboost is used: it employs a boosting strategy to train many weak classifiers to select the features with minimum error rate using for each the best threshold to classify positive and negative examples. The final classifier is a weighted sum of these weak classifiers. Finally, the concept of cascade of classifiers is introduced to check easily if a window is or not a face region. Each window of features is passed through different stages of classifiers and when a window fails a stage, it is discarded and not further processed. In this way, the computational complexity of the detector is greatly reduced.

The parameters used for the HAAR Cascade classifier are different than the original ones in the notebook because initially two faces were detected for 5 images and therefore the parameters were changed to correct these mistakes.

Original dataset images	445
Detected faces	435
Different individuals	26
Different features	2209

Table 1: Statistics of the data

As it can be observed from the above statistics, the HAAR Cascade fails in recognizing faces in 10 images of the dataset. So, in the following only 435 faces belonging to 26 individual are used by the face recognition system.

Figure 2: Cropped faces detected by the HAAR Cascade classifier

4 Feature extraction

In this section, features are extracted from each image using different techniques. The goal of these techniques is to map the image to a lower-dimensional subspace through linear and non linear-mapping. The extracted features should be invariant to lighting and positioning conditions. In this way, the computational complexity of an evaluation system is decreased since the number of features is lower than image pixels.

In this assignment, both old-school computer vision techniques and a newer deep learning approach are used.

4.1 Eigenfaces for face recognition

The Eigenfaces algorithm was proposed in the 1987 by Kirby and Sirovich in the paper "A Low-Dimensional Procedure for the Characterization of Human Faces", which is considered to be a seminal work in computer vision history. The algorithm uses Principal Component Analysis to represent the face images in a lower-dimensional space. This involves applying the eigenvalue decomposition to the dataset, keeping the eigenvectors with the largest corresponding eigenvalues (so with the largest variance). These eigenvectors are called Eigenfaces. At this point, a face can be represented as a linear combination of the retained eigenfaces. In this work, 35 eigenfaces are retained, so each face is represented by 35 features.

For this technique, the Euclidean distance is used as distance metric to compute the similarity between eigenface representations.

4.2 Fisherfaces for face recognition

Scholarpedia explains than PCA can not be enough accurate when the goal is classification rather than representation. In a classification setting, the goal is to find a feature space that minimizes the intra-class variation and maximizes the inter-class variation. This technique is known as Linear Discriminant Analysis (LDA) and it differs from PCA because it tries to model the difference between classes, which in this case are represented by individuals. The returned eigenvectors are called Fisherfaces when LDA is used to compute the feature space representation of a set of images.

Again, to determine the similarity of fisherface representations, the Euclidean distance is employed as distance metric.

4.3 LBP for face recognition

Local Binary Patterns (LBP) are a texture descriptor proposed by Ojala et al. in their 2002 paper "Multiresolution Grayscale and Rotation Invariant Texture Classification with Local Binary Patterns". This technique computes a local representation of texture by comparing each pixel with its neighbors. To do this, initially the image is converted to grayscale and the neighborhood size r is selected. Then, for each pixel a LBP value is computed thresholding on its neighbors values: if the center pixel is greater than the surrounding one 1 is returned, 0 otherwise. These LBP values are accumulated in the output LBP 2D array and in the last step the histogram this array is computed.

In this setting, the similarity of histogram representations is computed using the chi-squared distance metric.

4.4 Deep metric learning

Nowadays all the face recognition techniques are based on deep learning models to generate feature representations: a deep learning model will be used in this case to learn a projection of the input data in a lower-dimensional space.

In this section the siamese network DNN architecture, developed and presented in 2014 by Facebook researchers in the paper "DeepFace: Closing the Gap to Human-Level Performance in Face Verification", is used. This model consists of two copies of the same CNN, which share their weights, and a pair of images at a time is presented to the network. The goal of the training process is to minimize the distance between the embedded representations of the same individual and to maximize the distance between the representations of different individuals.

In this network, a shallow CNN architecture is trained with contrastive loss. Two copies of this subnetwork are used and their output are passed to a layer which calculates their euclidean distance. The model is trained with pairs of impostor or genuine images with their associated labels.

In this work the embeddings before the Euclidean-distance layer are used to compute the similarity scores among faces.

5 Distance-based scoring

For each technique previously discussed, matching scores are obtained from the computed embeddings and the specified distance metrics, using the formula:

$$matching\_score(faces_i, faces_j) = \frac{1}{1 + distance\_metric(embedded(faces_i), embedded(faces_j))} \quad (1)$$

The matching scores can be used then to be compared to a decision threshold in a 1-to-1 setting in verification mode or they can be used to rank the templates in the database in a 1-to-N setting in identification mode.

6 Evaluation

In this section 4 different systems based on the features extracted through the previously discussed techniques are compared.

6.1 Validation as verification system

The first setting is the verification. Its goal is to authenticate a claimed identity by comparing the returned matching score with a pre-defined threshold.

6.1.1 Genuine and impostor score distributions

Initially, the genuine and impostor score distributions of the different systems can be plotted to gain some first insights on the performance of these approaches.

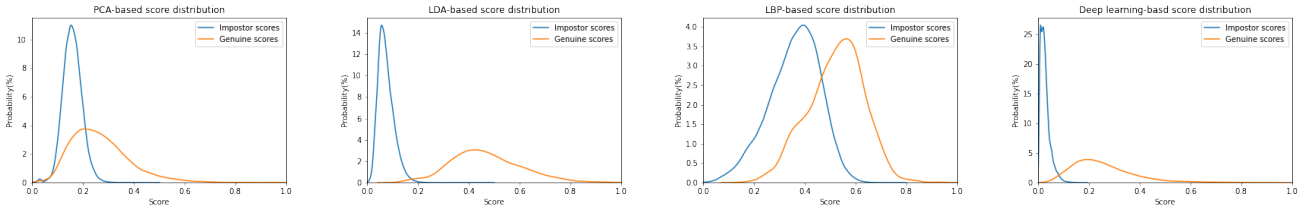

Figure 3: Score distributions of PCA, LDA, LBP and DL based matching systems

PCA and LBP based genuine and impostor score distributions overlap a lot, so even if the threshold is selected accurately, many classification errors are expected using these techniques. While, the other two systems have quite separate score distributions, so probably a system based on LDA or Deep-Learning performs better than the other two.

It should be also noticed that the class distributions is highly unbalanced: the impostor scores are 90276, while the genuine ones are only 4119.

6.1.2 F1 and accuracy as metrics

Considering accuracy as perfomance metric, the following curves can be obtained for the different feature extractors varying the threshold.

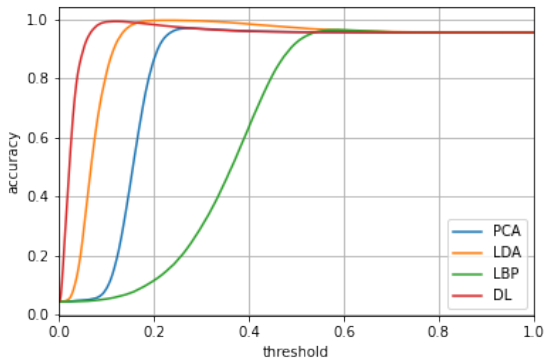

Figure 4: Accuracy of the PCA, LDA, LBP and DL based systems

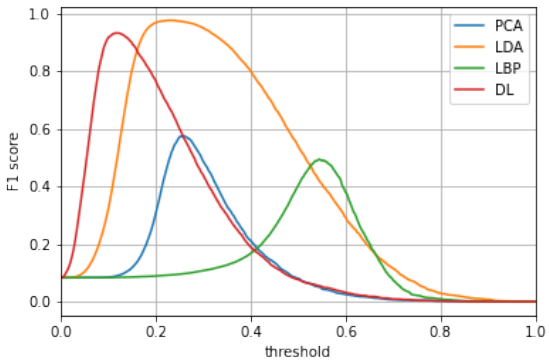

Figure 5: F1-score of the PCA, LDA, LBP and DL based systems

The results obtained using accuracy as performance metric can be visualized in Figure 4. The accuracy using the best threshold is very close to 1 for all the analyzed techniques. However, accuracy is not a good performance metric when classes are unbalanced. It can be noticed from the curves that also for threshold close to 1 the obtained accuracies are very high: this happens because in those cases most of the scores are classified as impostors, which represent the vast majority of the dataset.

Therefore, F1-score can be a more suitable performance metric in this case because it takes into account also class prior information. It corresponds to the harmonic mean of precision, which is class prior dependent, and recall, which is instead class prior independent.

The difference among the techniques can be visualized easier with this performance metric in Figure 5. LDA and Deep Learning performs better than the other approaches, the maximum F1-score for PCA and LBP based scores is quite low (≈ 0.5).

Given the above curves, an optimal threshold can be determined. It depends on the requirements of the application for which the system is realized. The threshold for which the F1-score is maximized can be a suitable choice for applications that require good performance but can tolerate some false positives. On the other hand, there exist some applications that must completely avoid false positives and therefore an higher threshold is chosen: its value must be enough high so that no impostor scores are recognized as genuine ones. In this way the false rejection rate increases but no zero-effort impostors are accepted by the system.

6.2 ROC curve

Plotting the ROC curves, which displays FPR and TPR for different threshold values, and the DET curves, which shows FPR versus FNR instead, is another way to observe the influence of the threshold. The x-axis of these curves uses a logarithmic scale to provide relatively straight lines that are easier to be compared.

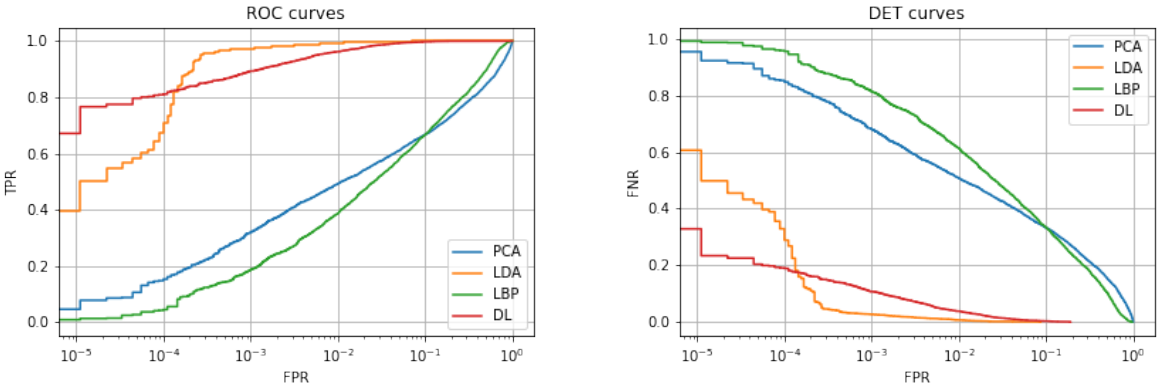

Figure 6: ROC and DET curves of the PCA, LDA, LBP and DL based systems

With these curves, it can be easily observed that the LDA-based system outperforms all the others. Moreover, PCA and LBP based systems perform quite poorly with respect to the others.

It is interesting to note also that PCA based system outperforms LBP-based one when lower FPRs are considered, while the opposite happens for higher FPRs (> 0.1); the same (but considering different values) happens respectively for DL and LDA based systems as well. Indeed biometric systems are usually evaluated using rank classifiers which

allow to compare them in different regions of the space.

Even if it is better to consider the whole curve, the performance difference of the different systems can be noticed also considering the Area Under ROC curve (AUROC).

Used technique	AUROC
PCA	0.825
LDA	0.999
LBP	0.860
DL	0.998

Table 2: AUROC performances of the PCA, LDA, LBP and DL based systems

6.3 Equal Error Rate

Another way for assessing overall system performance is to evaluate the Equal Error Rate, which reflects the operating point when FNR and FPR have the same value. In general, the lower the equal error rate value, the greater the biometric system's performance.

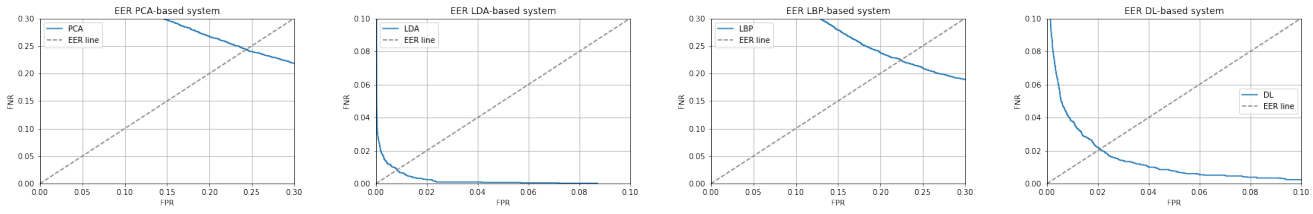

Figure 7: Equal error rate of PCA, LDA, LBP and DL based systems

Again, DL and LDA based systems greatly outperform the others since they have a lower EER.

6.3.1 Precision and Recall

Precision-recall curves explain the trade-off between a predictive model's actual positive rate (recall) and positive predictive value at various probability thresholds. They are superior than ROC curves for high unbalanced classes because they include a class prior dependant measure (precision).

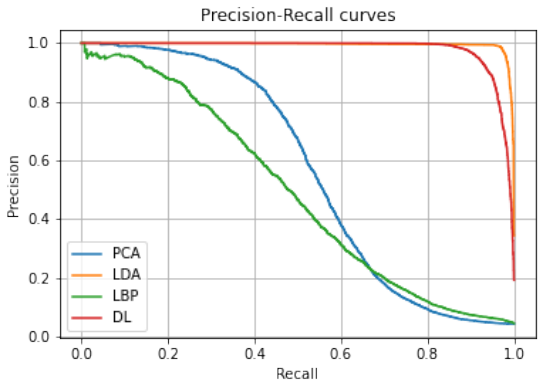

Figure 8: Precision-recall curves of the PCA, LDA, LBP and DL based systems

From these curves it can be noticed for example that PCA and LBP based systems have low precision values when recall is close to 1. This happens because a threshold, which returns for these systems a low number of false negatives (high recall), corresponds also to a high number of false positive (low precision) as it can be observed from the score distributions in Figure 3.

Precision and recall curves can be summarized using two summary measures:

- Area Under Precision-Recall (AUPR) curve, which is calculated by taking the integral over the precision-recall curve. Therefore, the higher this number, the better.
- Average Precision (AP) score, which corresponds to the weighted mean of precisions achieved at each threshold; the weights correspond to the increase in recall from the previous threshold.

Used technique	AUPR
PCA	0.568
LDA	0.995
LBP	0.491
DL	0.979

Used technique	AP
PCA	0.568
LDA	0.995
LBP	0.491
DL	0.979

Table 3: AUPR and AP of the PCA, LDA, LBP and DL based systems

The AP score is an approximation of the AUPR. In these cases they correspond for all the systems, but they are in general different.

6.4 Validation as identification system

The identification is the second setting. This is used to determine if a person is enrolled in a database by matching his/her identity.

6.4.1 CMC curves

The Cumulative Match Characteristic (CMC) curves are typically used to evaluate systems in the identification scenario. They are computed directly through the similarity matrix, obtaining for every faces the corresponding sorted matching scores. Obviously, the diagonal values (the matching score of an image with itself) are not considered.

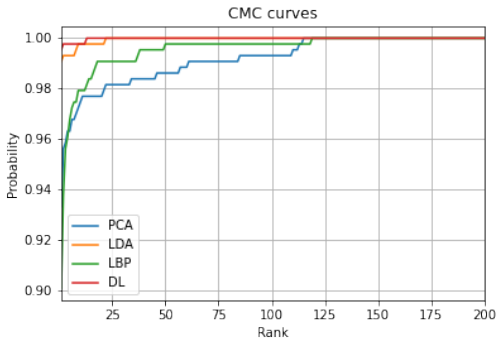

Figure 9: CMC curves of the PCA, LDA, LBP and DL based systems

Also in these setting, LDA and DL based techniques performs better than the others.

Considering the Rank-1 identification rate, the likelihood of a correct identification is returned as the best matching score, the following values are obtained:

Used technique	Rank-1
PCA	0.933
LDA	0.991
LBP	0.901
DL	0.995

Table 4: Rank-1 of the PCA, LDA, LBP and DL based systems

It can be observed from these values that when the correct identification has to be returned as the best matching score the DL-based system is the best, while for example in the verification scenario the LDA-based system always performed better.

7 Additional tasks

The implementation of the optional tasks can be found at the end of the notebook.

7.1 Implement 2 different face detectors and compare all techniques to the ground truth bounding boxes provided in CalTechFacesDirs/ImageData.mat. Look up the literature for methods to compare different face detectors. (1pt.)

In this optional task, the Haar Cascade face detector is compared with two other face detection methods:

- HOG face detector
  This method is based on 5 HOG filters and SVM. It is the fastest method on CPU and works very well for frontal and slightly non-frontal faces. Its major drawback is that it does not detect small faces (minimum detected face size is 80x80). Moreover, it does not work properly for non-frontal faces. It can be easily used importing the dlib library.
- DNN face detector
  This model is a deep learning based face detector which uses ResNet10 architecture as backbone. It is included in OpenCV. This method should be the most accurate because it works for different face orientations and also under substantial occlusion. The only drawback is that it is slower than the HOG-based face detector.

When one of the face detectors does not recognize any face in an image, an empty image (containing only 0 values) is returned. It should be noticed that the HAAR Cascade face detectors does not recognize a face in 10 images, the HOG-based in only 2 images, while the DNN-based recognizes a face in every image.

The bounding boxes returned by the face detectors are compared with the ground truth bounding boxes provided in './CalTechFacesDirs/ImageData.mat'. To make the comparison, the Intersection over Union score, also known as IoU score, is used as metric. It evaluates how similar a predicted bounding box is to the ground truth box by comparing the area of the intersection with the area of the union of the two bounding boxes.

$$IoUscore = \frac{area\ of\ overlap}{area\ of\ union} \tag{2}$$

The final score for each face detector is obtained taking the average over the IoU scores for each face.

Face detector	Average IoU score
HAAR Cascade	0.577
HOG-based	0.192
DNN	0.764

Table 5: Average IoU score for different face detection methods

The DNN face detector is the best perfoming as expected. It should also be noticed that even though HAAR Cascade does not recognize faces in more images than the HOG-based, it still performs much better. The average IoU score of the HOG-based face detection method is quite low.

7.2 Implement a classification-based scoring method, using an advanced classifier of your choice. Evaluate this system in an identification and verification scenario. (Hint: Follow steps introduced in section IV. Distance-based and classification-based scoring) (2pt.)

Some classification algorithms return classification scores or soft probabilities based on the likelihood that a picture corresponds to each subject in the dataset. These soft probabilities can be thought as a more advanced classification-based matching score. In this optional task, a classification-based scoring method is implemented and evaluated.

This method is applied to the PCA-based features, because the PCA-based system was one of the worst performing in the previous sections and so it should be easier to notice if there are some improvements.

The chosen classification method is Random Forest, which returns class probabilities through the method predict_proba.

In order to implement the classification-based scoring method, the following procedure is applied:

1. For each individual in the dataset, the first image is left out for the test set, while the others are used to train the classifier. This means that the test set consists of 26 images, the number of individuals in the dataset.
2. The Random Forest classifier is fitted using the train images and then the class probabilities are generate for each test image using the method predict_proba.
3. The 26 by 26 similarity matrix is built. Each row in the matrix corresponds to a test image and each column corresponds to an individual in the dataset.
4. Once the similarity matrix is obtained, the system is tested in a verification and identification scenario using the same metrics of the distance-based scoring system.

Validation as verification system

Again, initially the impostor and genuine score distributions can be plotted to obtain some first insights about this system.

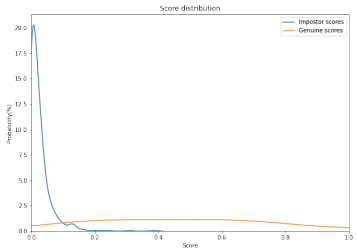

Figure 10: Score distribution of the classification-based scoring method using PCA features

The two distributions overlap by a smaller area than in Figure 3, so the classification-based scoring system is expected to perform better than the distance-based one.

Considering again accuracy and F1-score, the following curves can be obtained by considering different thresholds.

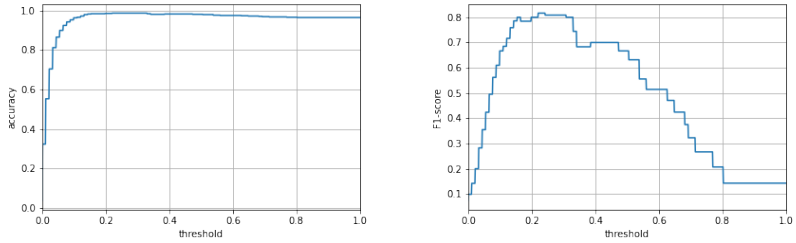

Figure 11: Accuracy and F1-score for different thresholds of the classification-based scoring method using PCA features

Also in this case, an improvement in the performance can be noticed with respect to Figure 4 and 5, for example the maximum F1-score value is higher than 0.8 while the previous system reached a maximum value of ca. 0.6.

Moreover, it should be noticed that the plots are made of many straight lines and not of a continuous curve because less scores are available (676) to determine the curves for this system.

Also in this case, it is better to use rank classifiers to check the performance in different regions.

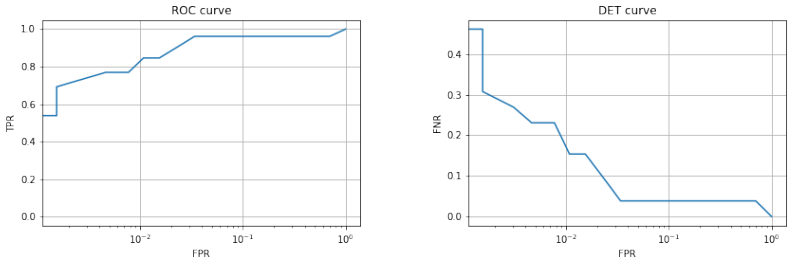

Figure 12: ROC and DET curve of the classification-based scoring method using PCA features

The classification-based system outperforms the distance-based one. For example, it can be observed comparing the ROC curves of the two systems that the first reaches always a higher value of TPR for a given FPR value. Moreover, the EER in this case is of ca. 0.08, which is much lower than the previous value obtained in Figure 6.3.

Finally, the precision-recall curve can be analyzed to consider also class prior dependent information, because again classes are unbalanced.

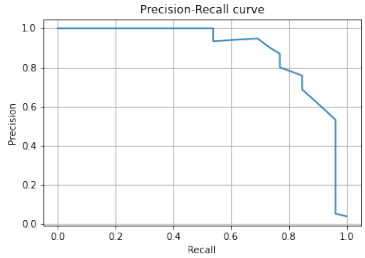

Figure 13: Precision-recall curve of the classification-based scoring method using PCA features

Again the classification-based scoring method outperforms the previous one. The performance improvement can be easily verified also using the AUPR and AP value which are both equal to ca. 0.86.

Validation as identification system

Again, the CMC curve can be plotted to evaluate the classification-based system in an identification scenario.

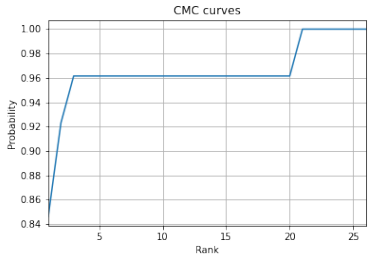

Figure 14: CMC curve of the classification-based scoring method using PCA features

The Rank-1 value for this system is equal to 0.769, which is lower the previously obtained 0.993 for the distance-based scoring method. So, if the correct identification has to be returned has the highest matching score, it is probably better to use the previous system in an identification setting.

7.3 Experiment with the Siamese deep learning model by implementing a different loss function or a different distance calculation layer. (1pt.)

For the siamese deep learning model each image pair can belong to the same individual or to different people. This is a classification problem and since there are only 2 classes, binary cross-entropy can be used.

The obtained performance after 10 epochs are:

	siamese model with constrastive loss	siamese model with binary cross-entropy loss
Training set loss	0.0035	0.0751
Training set accuracy	1.0000	0.9992
Validation set loss	0.0084	0.1088
Validation set accuracy	1.0000	0.9937
Test data accuracy	0.998	0.98

Table 6: Performance comparison between siamese models. The model used in the previous section of the report is trained using contrastive loss (central column), while the second model (right column) is trained using binary cross-entropy loss.

As it can be observed from the above table the siamese model trained with binary-cross entropy loss reaches a similar performance to the previous one which used contrastive loss.

However it should be highlighted that even if binary cross-entropy reaches great performance in this task, the contrastive loss is more suited to train siamese network. The goal of a siamese network is to differentiate between image pairs, not to classify them: indeed contrastive loss evaluates how well the siamese network discerns between impostor and genuine pairs. Even though the difference with the binary cross-entropy idea is subtle, it is very crucial.