Assignment 3

Rojina Kashefi (r1018183)

May 2025

Question 1

Pairwise distances between facial images are computed by extracting feature representations using four different methods: PCA, LDA, LBP, and a Deep Learning-based approach. For Principal Component Analysis (PCA), the dimensionality of the original face dataset is reduced by selecting the number of principal components as the minimum of the number of samples and original features. The data is then projected onto the directions of maximum variance, resulting in a lower-dimensional representation. Each image is then mapped to a embedding. Pairwise distances between these embeddings are computed using the Euclidean distance:

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2}$$

$\mathbf{x}_i$ and $\mathbf{x}_j$ represent the embedding vectors of the i^{th} and j^{th} image each with n dimension. The term x_{ik} refers to the k^{th} element of $\mathbf{x}_i$. For Linear Discriminant Analysis (LDA), dimensionality reduction is performed using class labels to improve the separability of different classes in the new feature space. Unlike PCA, which is unsupervised and only maximizes overall variance, LDA do a projection that maximizes the ratio of inter-class variance to intra-class variance. This leads to embeddings where samples from the same class are grouped closely, and those from different classes are well separated. Pairwise Euclidean distances are again used to measure similarity between the resulting embeddings. For Local Binary Patterns (LBP), handcrafted texture descriptors are extracted from grayscale images. Each image is encoded into a histogram of LBP values computed over a fixed grid. To compare these histograms, we use the Chi-squared distance:

$$\chi^2(\mathbf{P}, \mathbf{Q}) = \frac{1}{2} \sum_{i=1}^{n} \frac{(p_i - q_i)^2}{p_i + q_i + \epsilon}$$

where $\mathbf{P} = [p_1, p_2, \ldots, p_n]$ and $\mathbf{Q} = [q_1, q_2, \ldots, q_n]$ are the LBP histograms of two images.
For the Deep Learning (DL) approach, a Siamese neural network is trained on pairs of face images using contrastive loss. This model learns a feature space where similar faces are close and dissimilar faces are far apart. After training, the encoder is used to extract embeddings from all input images, and pairwise Euclidean distances are computed. Finally, after computing the pairwise distance matrix for each method, we convert distances into similarity or matching scores using:

$$\text{Similarity}(i, j) = 1 - \frac{d(i, j) - d_{\min}}{d_{\max} - d_{\min}}$$

Question 2

To calculate genuine and impostor scores, we compared the labels of each pair of images. Genuine scores correspond to pairs with the same label, and impostor scores come from pairs with different labels. These scores were then normalized and plotted in Figure 1. A clear separation between the two distributions indicates good performance. Ideally, the genuine curve should be shifted to the right, showing higher similarity values, while the impostor curve should be on the left, representing lower similarity. Less overlap between the curves suggests that the method can better distinguish between same and different identities and avoid misclassifications. The height of each curve reflects how often a particular similarity score appears. As we can see, the best is LDA and deep learning.

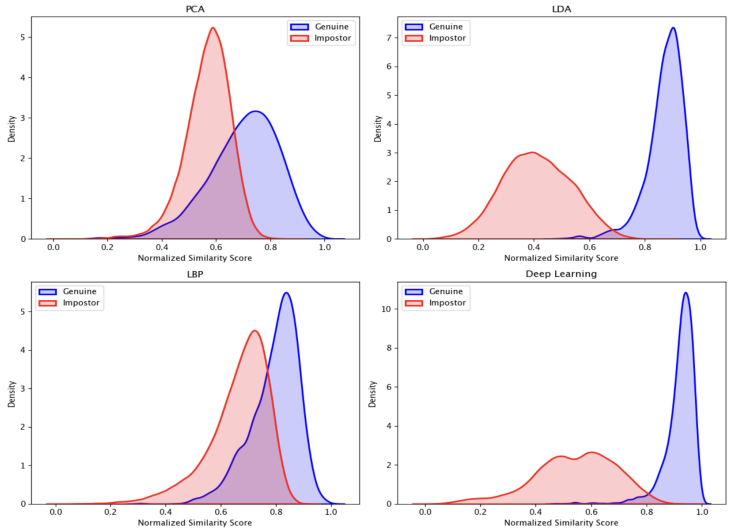
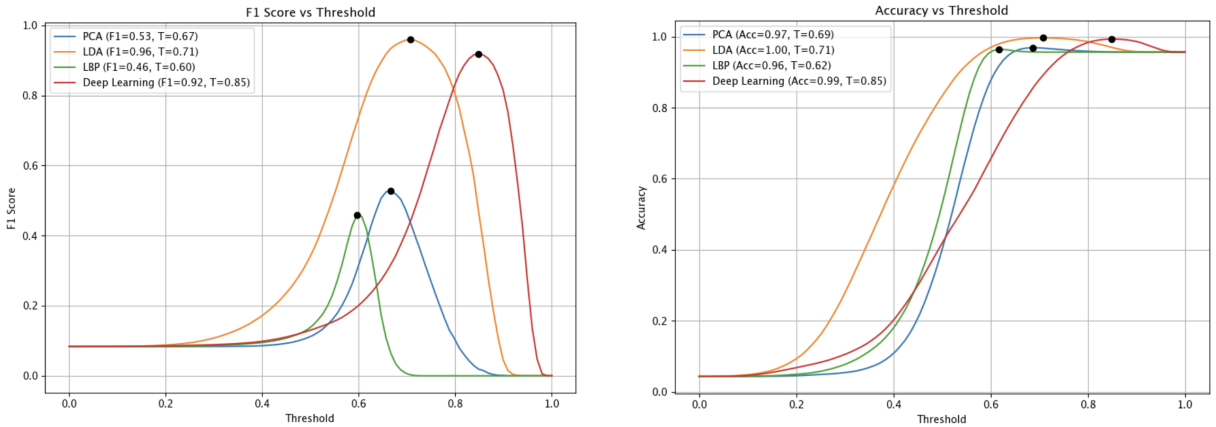


Figure 1: Genuine and impostor distribution.

Question 3

In Figure 2, F1 and accuracy scores over a range of thresholds are shown. For each threshold, we evaluated the classification performance using the ground-truth labels. The black dots indicate the best threshold for each method, and the legend shows the corresponding score and threshold value. LDA performs best overall, achieving an F1 score of 0.96 and an accuracy of 1.00 at threshold of 0.71. This high threshold means that only pairs with very high similarity are classified as the same identity (genuine), reducing false positives. Deep Learning shows the second-best performance, with an F1 score of 0.92 and accuracy of 0.99 at a threshold of 0.85. PCA reaches its highest F1 score of 0.53 and accuracy of 0.97 at a threshold of 0.67. LBP performs the worst, with a maximum F1 score of 0.46 and accuracy of 0.96 at threshold 0.62. This shows that even with high accuracy, a low F1 score means the method struggles to correctly identify genuine pairs. For example, in imbalanced datasets with more impostor pairs, a model can achieve high accuracy by mostly predicting impostors, but this leads to missed genuine matches and low recall, lowering the F1 score.



(a) F1 Score vs Threshold

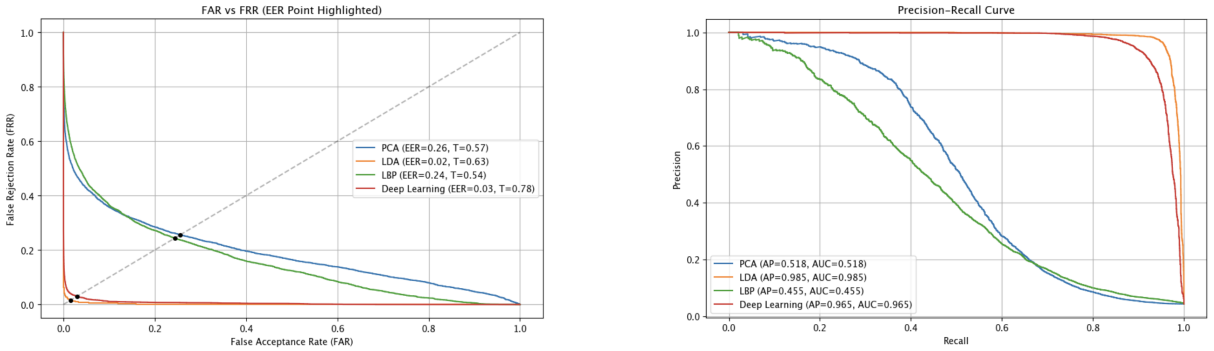
(b) Accuracy vs Threshold

Figure 2: Performance of different methods across thresholds based on F1 score and accuracy.

Question 4

As shown in Figure 3, the False Acceptance Rate (FAR) vs False Rejection Rate (FRR) curves for each method are presented. The point at which the FAR equals FRR is known as the Equal Error Rate (EER). This point shows a trade-off between security and usability. FAR measures how often impostors are incorrectly accepted, while FRR measures how often genuine users are incorrectly rejected. A lower EER indicates better overall system performance. From the plot, LDA achieved the lowest EER, 0.02,

followed closely by Deep Learning, 0.03. This shows that these models are highly reliable in distinguishing between genuine and impostor pairs. In contrast, PCA and LBP showed higher EERs of 0.26 and 0.24. These EER values mean that around 24–26% of the time, the system either incorrectly accepts an impostor or incorrectly rejects a genuine user.
The Precision-Recall (PR) curve shown in Figure 3. Precision shows how many of the predicted genuine matches are actually correct and recall measures how many of the true genuine matches are successfully identified. The Average Precision (AP) is the weighted average of precision across all levels of recall. The Area Under the Curve (AUC) captures the entire PR curve as a single value between 0 and 1. A higher value means that the model achieves high precision across a wide range of recall levels. In the plot, LDA achieves the best performance, with both AP and AUC equal to 0.985. This means it consistently maintains high precision even as recall increases. Deep Learning follows closely with AP and AUC values of 0.965. In contrast, PCA and LBP obtain lower scores, PCA: 0.518 and LBP: 0.455, and their curves decline more sharply. This meaning their precision drops as recall increases. This suggests that while they may initially identify genuine pairs accurately, they increasingly misclassify impostors as genuine when attempting to improve recall. Overall, LDA and Deep Learning show better verification performance.



(a) FRR vs FAR curve

(b) Precision-Recall curve

Figure 3: Performance of different methods on FAR vs FRR curve and PR Curve.

Question 5

To calculate the Cumulative Matching Characteristic (CMC) curve, we compare each query sample to all others and rank them by similarity. We then record the rank where the correct match appears. For each rank r, we compute the percentage of queries where the correct match is found within the top r results. This shows how well the system identifies the right person from a group. A higher CMC curve, especially at lower ranks, means better performance. The Rank-1 accuracy shows how often the correct match is the top result. As seen in Figure 4, LDA (98.4%) and Deep Learning (97.7%) had the best results. LBP (88.9%) performed good, while PCA (82.0%) had the weakest performance.

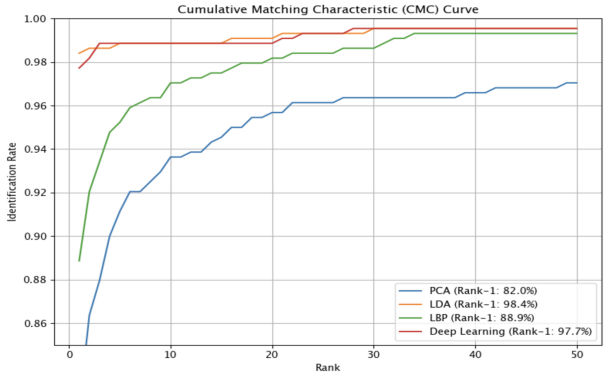


Figure 4: Performance of different methods on CMC curve.

Question 6

To visualize the effect of hyperparameters, I evaluated for the F1-score (for the verification system) and Rank-1 accuracy (for the identification system). As shown in Figure 5, for PCA, I tested different numbers of components and found that the best performance in terms of F1-score was achieved with 20 components. This is because, beyond a certain point, adding more components captures very little useful variance and starts to include noise, which reduces generalization. In contrast, the highest Rank-1 accuracy was achieved with 50 components. This difference arises because the F1-score is more sensitive to both false positives and false negatives, while Rank-1 accuracy only checks whether the correct identity is ranked first. However, the difference in Rank-1 accuracy between 20 and 50 components is small, only about 2%, so I chose 20 components as the best for PCA.

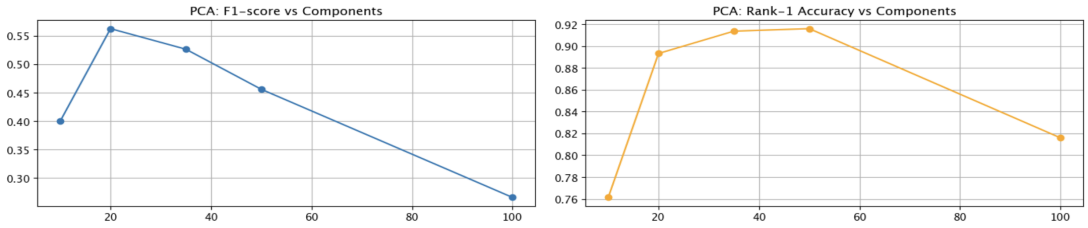


Figure 5: PCA: F1-score and Rank-1 accuracy across different numbers of components.

LDA finds linear combinations of features that maximize class separability. The rank of the between class is at most equal to the number of classes minus one, $C - 1$, components. As shown in Figure 6, shows increasing the number of LDA components leads to better performance. While the first few components capture major separations, later components still contribute smaller but useful distinctions. Unlike PCA, LDA is not sensitive to overfitting when adding more components, because each one is directly optimized for class separation. Based on this, I selected the maximum possible, 24 components for optimal LDA.

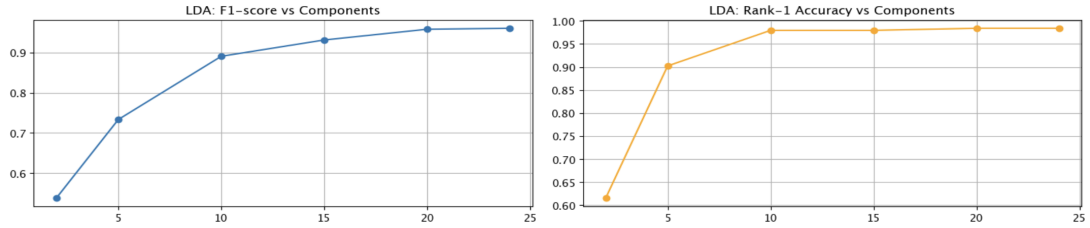


Figure 6: LDA: F1-score and Rank-1 accuracy as the number of components increases.

LBP capture local structures in images by comparing each pixel with its surrounding neighbors. If a neighbor's intensity is greater than or equal to the center pixel, it is assigned a 1; otherwise, it is assigned a 0. These binary values are combined into a pattern, and the image is represented as a histogram of these patterns. The radius parameter in LBP defines how far from the center pixel the neighbors are sampled. As shown in Figure 7, LBP performs best when the radius is set to 3. This is because, in this distance, fine texture details, such as facial edges and small patterns are captures. Increasing the radius leads to the capturing less relevant areas, which blurs useful details and introduces noise. Therefore, I chose radius 3 as the optimal value for LBP.

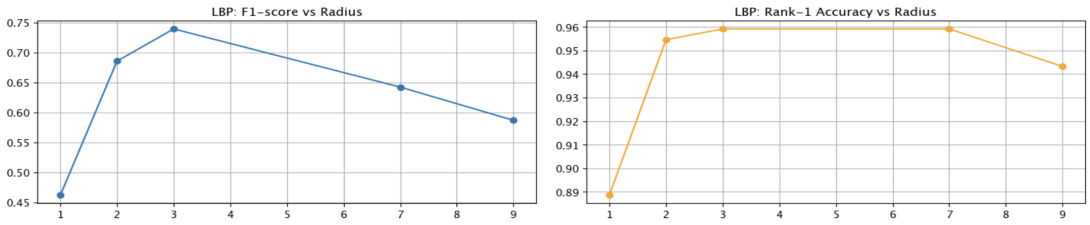


Figure 7: LBP: Effect of radius on F1-score and Rank-1 accuracy.

In the deep learning model, the embedding dimension determines how much information is encoded in the feature space. Since the embedding is the compact representation that carries all distinguishing information, increasing the dimension from 32 to 128 improves both F1-score and Rank-1 accuracy (Figure 8).

This happens because a larger embedding space allows the model to capture more discriminative features. However, increasing the dimension too much may lead to overfitting, where the model memorizes noise instead of general patterns. In this regard, I chose 128 as the best embedding dimension.

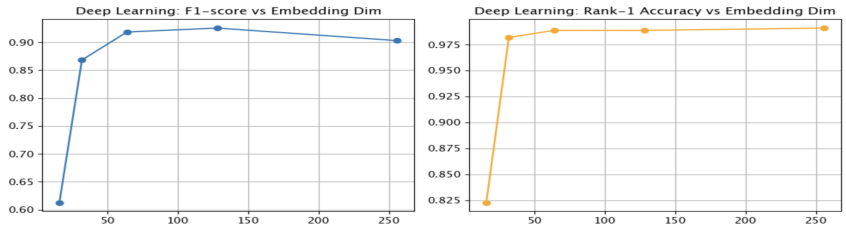


Figure 8: Deep Learning: F1-score and Rank-1 accuracy with different embedding dimensions.

Question 7

For comparison, we used 20 components for PCA, 24 for LDA, a radius of 3 for LBP, and 128 dimensions for deep learning. In Figure 9, we can see that LBP separates genuine and impostor scores more clearly. This shows that LBP works better in this new setting.

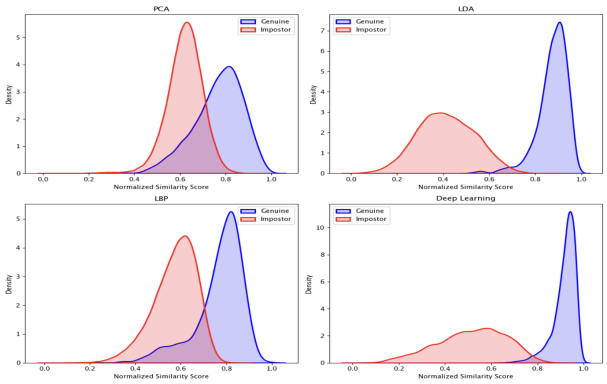
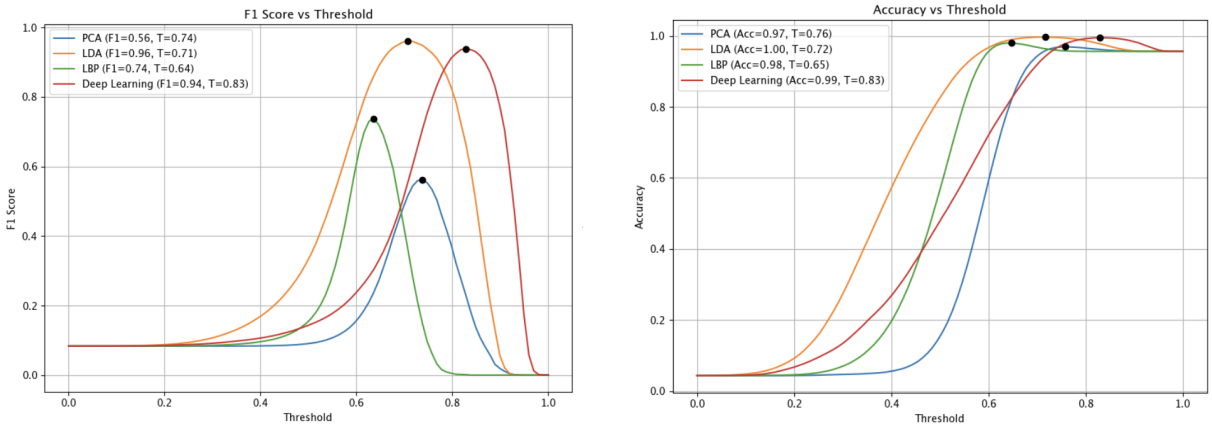


Figure 9: Genuine and impostor score distributions using the best parameters.

Figure 10 shows the F1 score and accuracy for each method. All methods improved except LDA, which stayed the same because we already used 24 components before. Deep learning gives results close to LDA when the embedding size is increased. LBP with radius 3 also does better than PCA, even though PCA was better earlier.



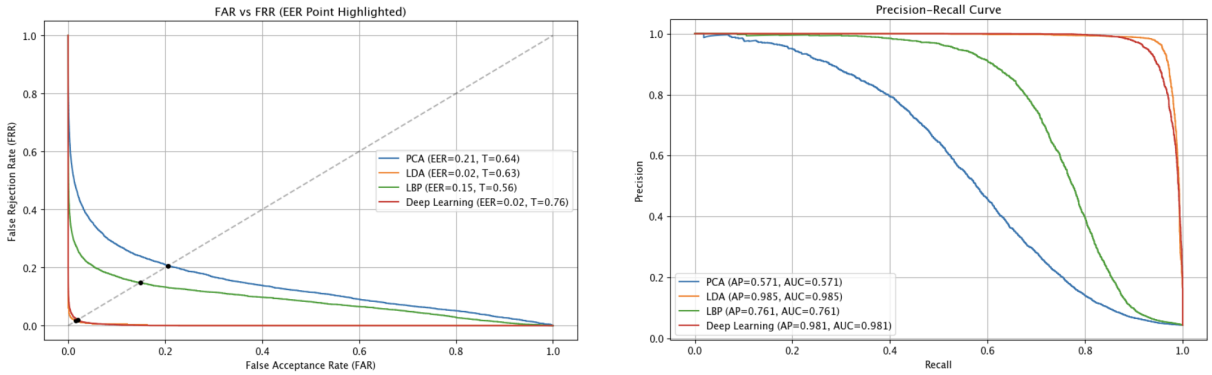
(a) F1 Score vs Threshold

(b) Accuracy vs Threshold

Figure 10: F1 score and accuracy of optimal method across thresholds.

As shown in Figure 11, all models now achieve lower Equal Error Rates (EER) and higher Average Precision (AP) and Area Under the Curve (AUC) scores. Among the methods, LBP outperforms PCA.

For the verification task, LDA and the deep learning model perform best, with LDA slightly leading. LBP follows, while PCA shows the weakest performance.



(a) FRR vs FAR curve

(b) PR Curve

Figure 11: Performance of optimal methods on FAR vs FRR curve and PR Curve.

For identification, we look at the CMC curve in Figure 12. LDA and deep learning have similar results at first, but deep learning does better at higher ranks. LBP also beats PCA in Rank-1. So, the best methods for identification are in order: LDA, Deep Learning, LBP, and PCA.

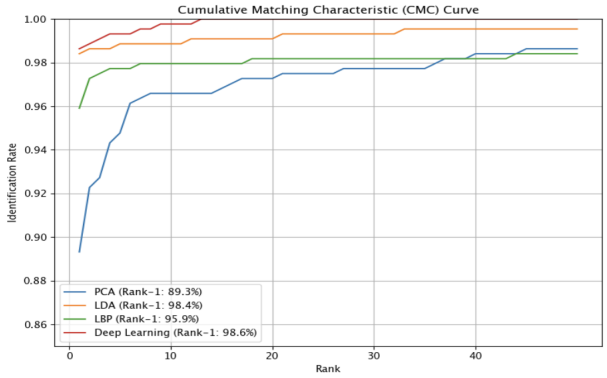


Figure 12: CMC curve of each optimal method works for identification.

Question 8

For Task 6, I built a new deep learning model using a triplet network. Instead of using image pairs, this model learns from triplets, an anchor image, a positive image (same person), and a negative image (different person). I created 5,000 triplets from the CalTech dataset. The model uses a CNN architecture consisting of two convolutional layers with ReLU activation functions, followed by max pooling and dense layers. It maps face images into a 128-dimensional embedding space where images of the same identity are placed closer together. Training was done using triplet loss, which encourages the network to bring the anchor and positive embeddings closer while pushing the negative embedding farther away, with respect to a margin.

$$\mathcal{L}_{\text{triplet}} = \frac{1}{N}\sum_{i=1}^{N} \max\left(\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha,\, 0\right)$$

where $f(x)$ is the embedding of image x, x_i^a, x_i^p, and x_i^n represent the anchor, positive, and negative images respectively, and α is the margin parameter (set to 1.0 in our case).
The results, compared to the Siamese network, are shown in Figure 13, 14 and 15. Although the triplet model did not outperform the Siamese network, it shows promising potential. One reason the Siamese network may have performed better is that it was trained using contrastive loss on image pairs, which is often easier to optimize, especially on smaller datasets. In contrast, triplet loss relies on carefully constructed and diverse triplets to be effective. Moreover, the binary supervision used in the Siamese model of, same or different, provides a simpler and more stable training signal.

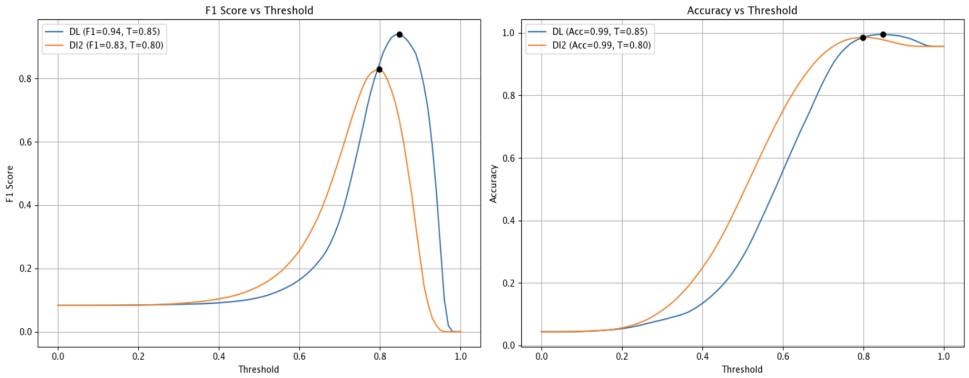
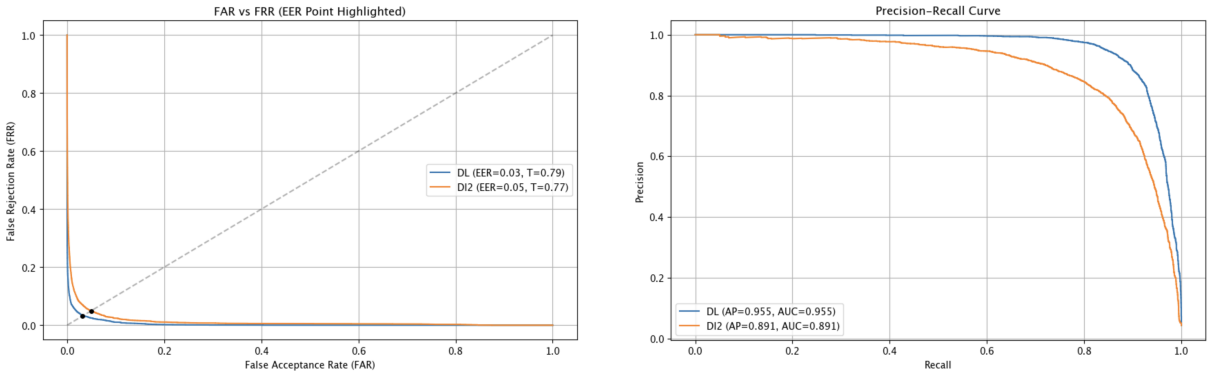


Figure 13: Comparison of Triplet (DL2) and Siamese networks (DL) on F1 score and accuracy.



(a) FRR vs FAR curve

(b) PR Curve

Figure 14: Comparison of Triplet (DL2) and Siamese networks (DL) on FAR vs FRR and PR Curve.

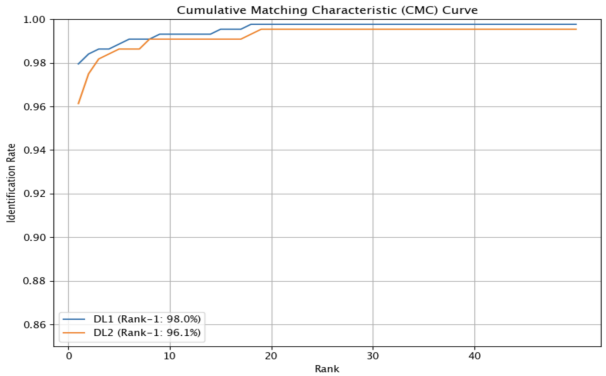


Figure 15: CMC curve of Triplet (DL2) and Siamese networks (DL) for identification.