Biometric System Concepts – Assignment 3 Report

Ali Tolga Dinçer – r0866432

Q1) In this assignment, we were first asked to compare the performance of 4 methods that are used for extracting feature eigenvectors. The mentioned methods were Principal Components Analysis (Eigenfaces), Linear Discriminant Analysis (Fisherfaces), Local Binary Pattern (LBP) and Deep Learning. The first question required us to compute the matching scores (distance scores) matrix using the feature vectors obtained from these methods. We can calculate the distance using different metrics, such as Euclidean distance and Chisquare distance (. Example matrices that use the Euclidean distance from each method are listed below:

	0	1	2	...	437	438	439
0	0.000000	3.666423	6.199222	...	5.587180	6.838945	5.432090
1	3.666423	0.000000	5.098935	...	4.623896	5.177505	4.588192
2	6.199222	5.098935	0.000000	...	6.656280	6.830815	6.316935
..	...	...	...	...	...	...	...
437	5.587180	4.623896	6.656280	...	0.000000	3.246977	3.464389
438	6.838945	5.177505	6.830815	...	3.246977	0.000000	3.974177
439	5.432090	4.588192	6.316935	...	3.464389	3.974177	0.000000

[440 rows x 440 columns]

	0	1	2	...	437	438	439
0	0.000000	6.255456	9.828133	...	18.768981	16.740232	19.382462
1	6.255456	0.000000	8.985383	...	18.172530	15.901287	18.289974
2	9.828133	8.985383	0.000000	...	17.111925	15.270795	17.731951
..	...	...	...	...	...	...	...
437	18.768981	18.172530	17.111925	...	0.000000	6.148688	6.811352
438	16.740232	15.901287	15.270795	...	6.148688	0.000000	5.784223
439	19.382462	18.289974	17.731951	...	6.811352	5.784223	0.000000

[440 rows x 440 columns]

	0	1	2	...	437	438	439
0	0.000000	1.346578	1.720878	...	1.602467	1.570857	1.560013
1	1.346578	0.000000	1.784709	...	1.633466	1.654119	1.635354
2	1.720878	1.784709	0.000000	...	1.946428	1.856337	1.787301
..	...	...	...	...	...	...	...
437	1.602467	1.633466	1.946428	...	0.000000	1.411483	1.467761
438	1.570857	1.654119	1.856337	...	1.411483	0.000000	1.454028
439	1.560013	1.635354	1.787301	...	1.467761	1.454028	0.000000

[440 rows x 440 columns]

	0	1	2	...	437	438	439
0	0.000000	68.986053	66.268250	...	165.131363	157.882523	153.311615
1	68.986053	0.000000	56.089577	...	187.273941	176.532181	170.236893
2	66.268250	56.089577	0.000000	...	156.793289	145.747681	142.063538
..	...	...	...	...	...	... ..	...
437	165.131363	187.273941	156.793289	...	0.000000	22.471422	45.652100
438	157.882523	176.532181	145.747681	...	22.471422	0.000000	25.453356
439	153.311615	170.236893	142.063538	...	45.652100	25.453356	0.000000

[440 rows x 440 columns]
As these represent the distance between the images, lower values indicate more similarity while higher values indicate that the two are not likely the same image/person.

Q2) Here, the question asks us to compare the f1 and accuracy scores for the 4 different methods. I wrote a function to convert the class labels into a binary array for each image, where the array would contain the value of 1 for the images of the same person and 0 otherwise. Then, I generated the predictions using the score, where the prediction is 1 if it's smaller than a threshold (shorter distance). I apply this for a range of thresholds. Finally, I apply this for every image and take the mean of it to obtain the final score for each threshold. The output plots can be seen below:

PCA: Best threshold according to F1 scores is: 0.43434343434343436

PCA: Best threshold according to Accuracies is: 0.4040404040404041

LDA: Best threshold according to F1 scores is: 0.38383838383838387

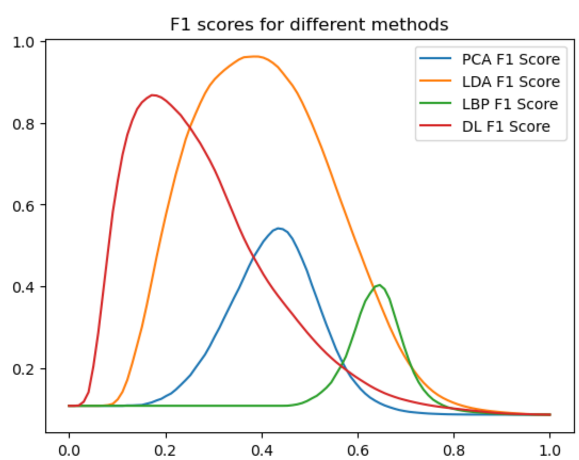
LDA: Best threshold according to Accuracies is: 0.36363636363636365

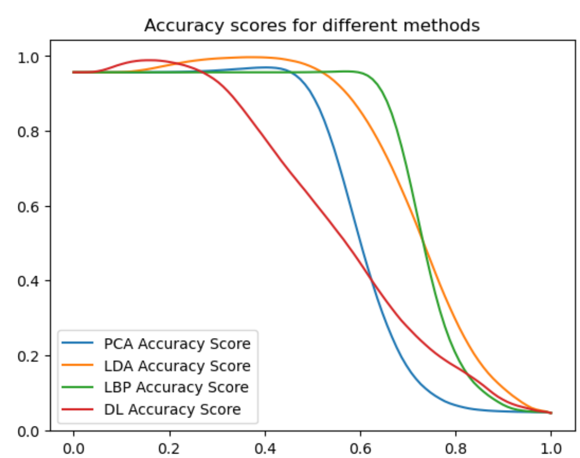
LBP: Best threshold according to F1 scores is: 0.6464646464646465

LBP: Best threshold according to Accuracies is: 0.5656565656565657

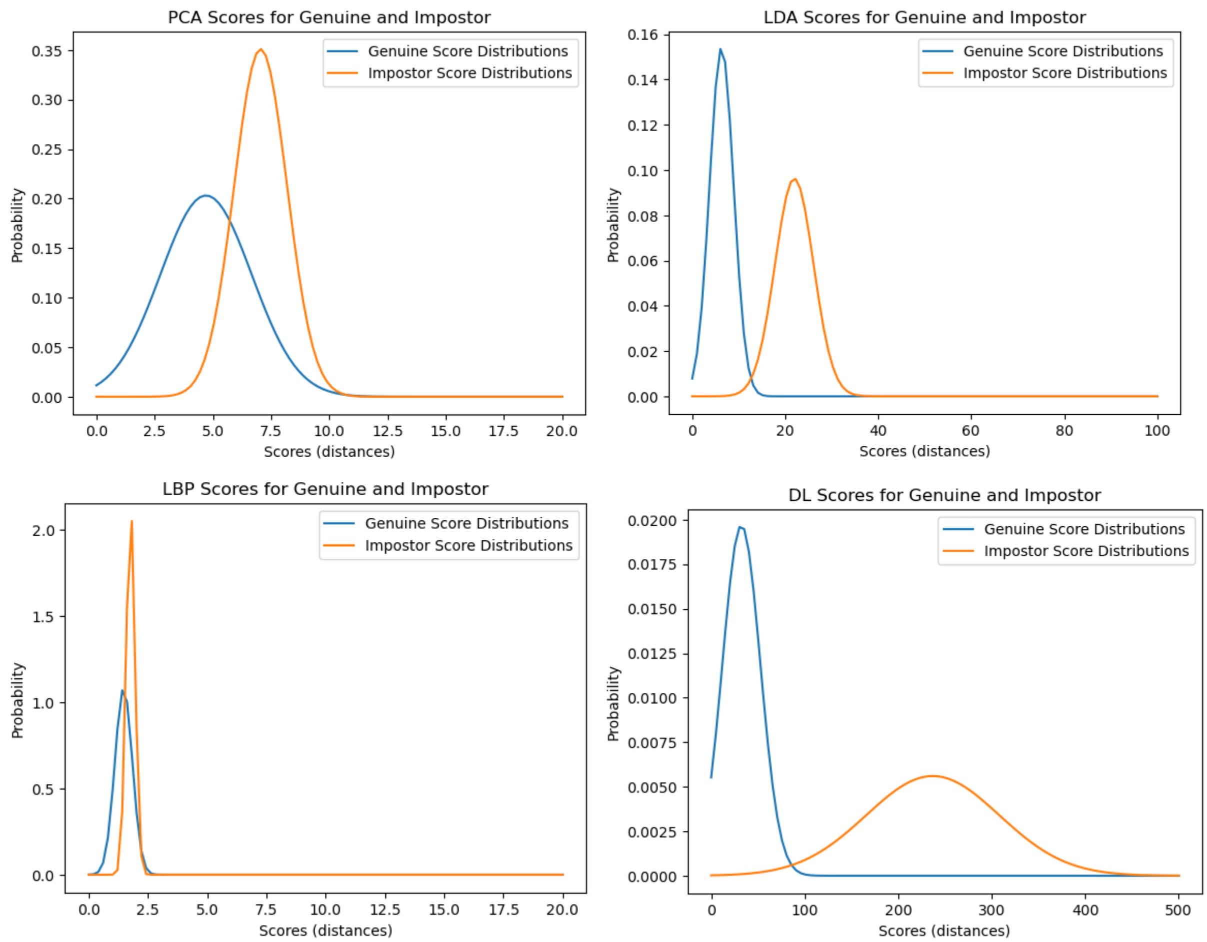
DL: Best threshold according to F1 scores is: 0.17171717171717174

DL: Best threshold according to Accuracies is: 0.16161616161616163



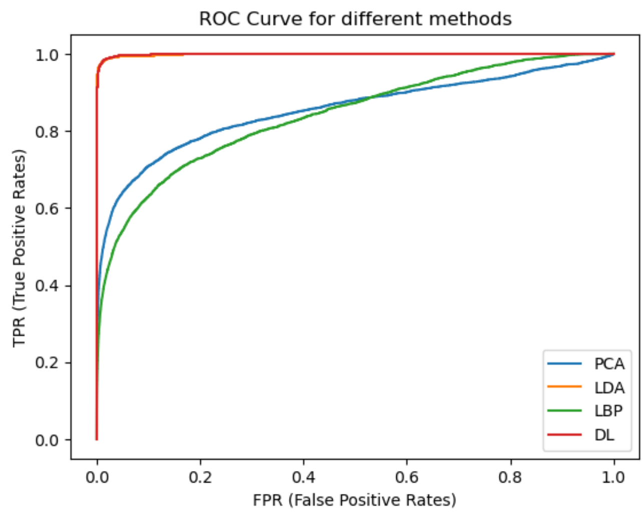


Q3) Here, we were asked to plot the Genuine and Impostor score distributions. The score distributions that are fit in a Gaussian Distribution for each 4 methods can be consulted below:



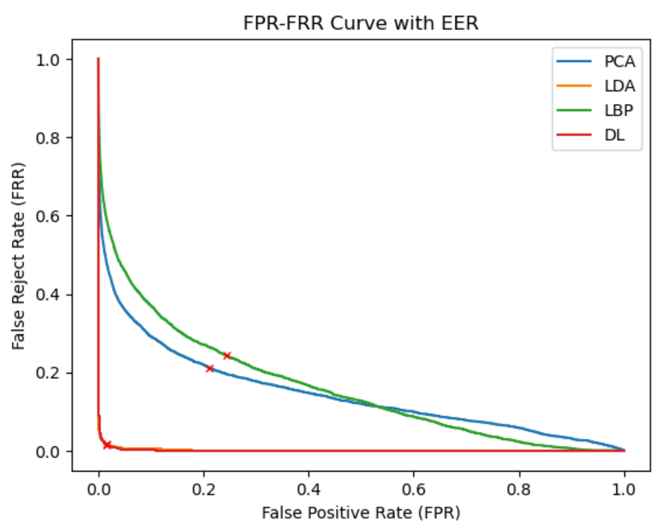
Notice the difference between the x-axis of DL scores and the others. DL provides us with a much higher score gap between genuine and impostor samples. LDA also provides a clear distinction, compared to PCA and LBP. Especially in LBP, we can see a clear overlap between the distributions, and they would be very hard to distinguish and classify accurately.

Q4) For the full assessment, I converted the matrix to an array in the sense that it has the length of (n choose 2). This allowed me to give the true labels and scores (inverse of distance scores) in one array and calculate the f1 and accuracy scores in a faster manner. I computed the scores for all 4 methods. The results are seen below:



As we know, ROC curve is the FPR and TPR for a range of thresholds. The closer to the top left means that our accuracy is quite good while maintaining a low acceptance rate for false positives. As we can see, both LDA and DL have a much higher TPR for lower thresholds (better distinguishing) while the other methods have to increase the threshold, in order to increase their accuracy; at the expense of accepting false samples as well.

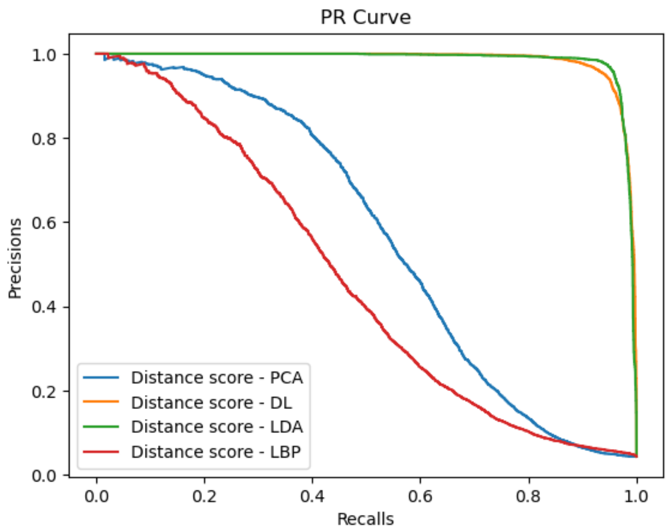
Below are the DET curves of the systems. Likewise, the LDA and DL have a much lower EER value, as they have a better performance. As mentioned in Assignment 1, EER is the point where FAR=FRR; and it is used to measure the performance of the biometrics system.



We also have the Precision-Recall curves, AUC and Average Precision Score.

Precision PCA: 0.5692533518735872
Precision LDA: 0.9853387732625174
Precision LBP: 0.46043081437238187
Precision DL: 0.9851673831022401

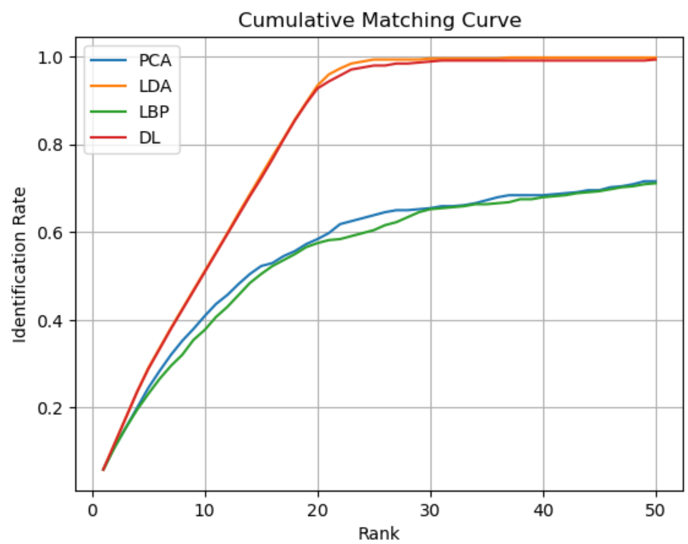
The average precision scores are shown above. These scores summarize the PR curve by taking the mean of precision at various recall levels. As we know, Precision represents the proportion of true positive predictions among the predicted positive instances; while Recall, also known as sensitivity or true positive rate, represents the proportion of true positive predictions among the actual positive instances. The plot of the PR curves can be seen here:



AUC (Area Under the (PR) Curve) can also be used to measure performances:

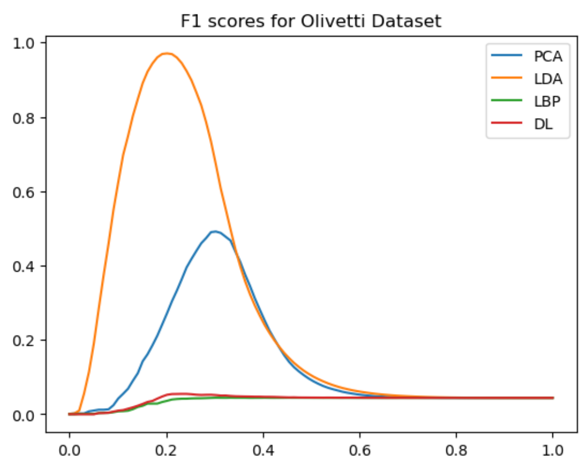
AUC PCA: 0.5692213509801451
AUC LDA: 0.9853376104458218
AUC LBP: 0.4603884886137576
AUC DL: 0.985166031364752

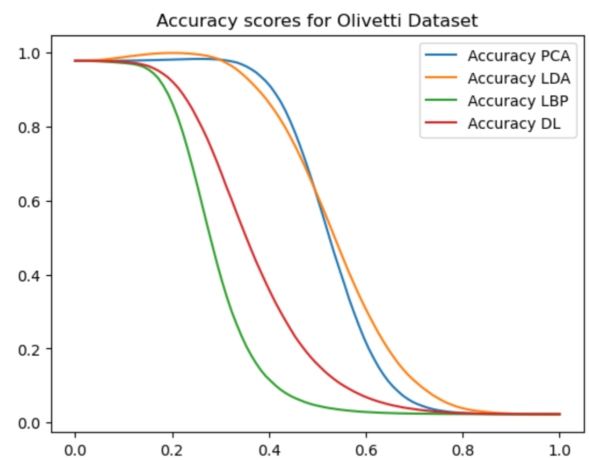
Q5) Here, we are asked to validate the system in an identification scenario. To do this by generating the Cumulative Matching Characteristics curve, we first take a sample image from each class (person). And for each of these images, we check upon the first max_rank number of matches and count the number of correct matches. As we have one image from each class, the maximum number of matches will sum up to the number of samples; as each image can only match with the other images from the same class. We count our total number of matches for each rank and we plot our CMC curve as follows:



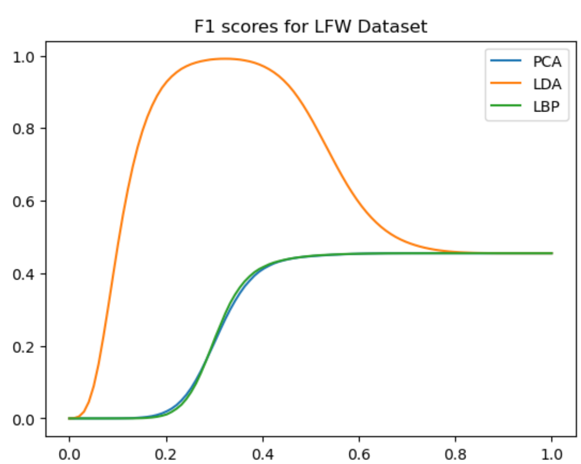
Optional Task 1) We evaluate our systems for different databases, AT&T (Olivetti) and LFW.

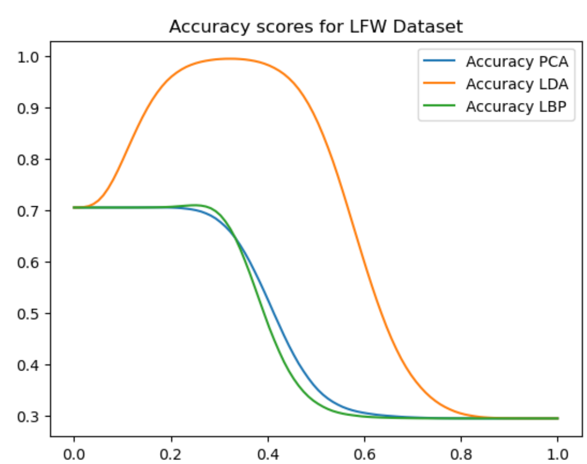
Below are the F1 and Accuracy scores for different methods for Olivetti dataset:





Below are the same plots for LFW:





Due to a last-minute error that I couldn't resolve, I had a shape error with the DL for LFW dataset and couldn't plot it. However, even if the performances are different between datasets, the comparisons are still the same as LDA performing the best, possibly followed by DL and PCA and LBP performing poorly in general.

Optional Task 2) For this task, in addition to the HAAR Cascade Face Detector that was given to us; I have implemented HOG and MMOD face detectors.

HOG (Histogram of Oriented Gradients) divides the image into small cells and computes the histograms of gradient orientations within each cell. These histograms are then concatenated to form a feature vector representation of the image.

MMOD (Max-Margin Object Detection) is a framework for object detection that uses the concept of max-margin learning. It employs a combination of cascades and sliding windows to detect faces at different scales and positions in an image. Afterwards, it trains a classifier to maximize the margin between positive (face) and negative (non-face) samples in order to achieve better discrimination between faces. This classifier uses a Convolutional Neural Network.

For comparison, I used the IOU (Intersection over Union) score, which is calculated by taking the area of intersection between the predicted bounding box and the ground truth bounding box and dividing it to the union of the predicted bounding box and the true bounding box. The results (scores) can be seen in the graph below for each image in our dataset.

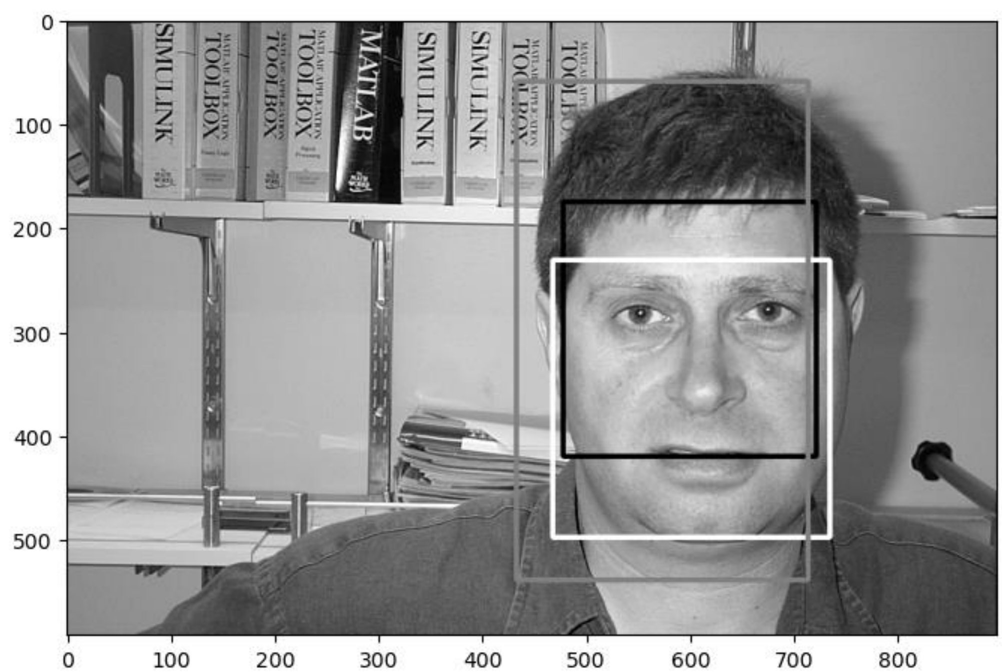


Figure 1 – The output of predictions. Black is MMOD, white is HOG and gray is the ground truth.

IOU for MMOD is 0.35344827586206895

IOU for HOG is 0.42120572366877784

As it consumed too much time to compute bounding boxes using MMOD, I only compared their performances for one image. This can easily be extended to any set of images.

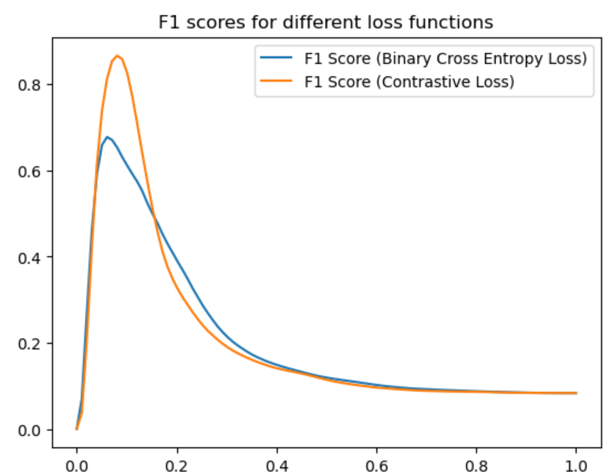
Optional Task 4) For this task, I decided to compare Binary Cross Entropy with Contrastive Loss. BCE is used for binary classification problems, while Contrastive Loss is generally used in Siamese networks. The formula for BCE is:

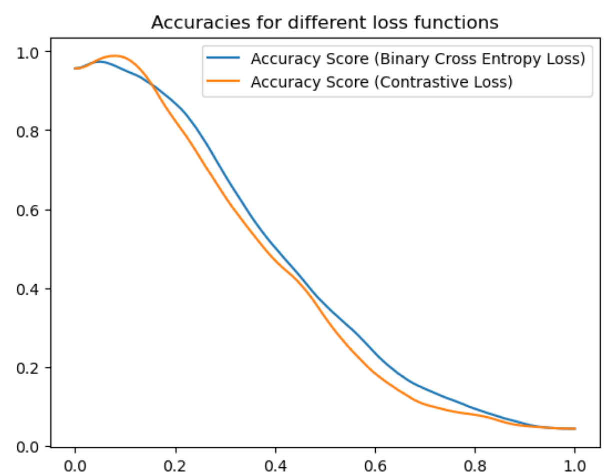
$$Loss = -(y\_true * log(y\_pred) + (1 - y\_true) * log(1 - y\_pred))$$

Where y_pred is the predicted probability (or score). And CL is :

$$Loss = (1 - y\_true) * distance\hat{}2 + y\_true * max(margin - distance, 0)\hat{}2$$

BCE loss aims to output high probabilities for the correct class and low probabilities for the incorrect class, while CL aims to minimize the distance between similar samples and maximize the distance between dissimilar samples. Both can be adapted to be used for our multiclass classification problem. The results are seen below:





The Contrastive Loss fits our model more and seems to be the right choice in the end.