Question 2

CS663 (Digital Image Processing) Assignment 4

Atishay Jain (210050026)
Cheshta Damor (210050040)
Kanad Shende (210050078)

210050026@iitb.ac.in
210050040@iitb.ac.in
210050078@iitb.ac.in

Contents

Question 2

PART I

Problem 1
What will happen if you test your system on images of people which were not part of the training set? (i.e. the last 8 people from the ORL database). What mechanism will you use to report the fact that there is no matching identity? Work this out carefully and explain briefly in your report. Write code to test whatever you propose on all the 32 remaining images (i.e. 8 people times 4 images per person), as also the entire test set containing 6 images each of the first 32 people. How many false positives/negatives did you get? [15 points]

SECTION 1

Testing on Unseen Data

During Face recognition, our model has a set of faces which it knows (the train set), and the task is that when it is given a test face, it has to identify whose face is among the people's faces it knows. For this, it uses a metric to calculate dissimilarity between the eigenfaces (like Euclidian distance), and the person's face whose eigenface had minimum dissimilarity is considered to be the one present in test image.
A limitation of this approach is that any test image will always fall into one of the face classes of our database. So, if the test image is of a person who is actually not a part of our database, that image would also get classified to one of the face classes, but ideally it should not be, because we don't know that person's face. In this way, such a test image acts as a imposter to our system.

For solving this problem, we need to decide a **Threshold** value of the dissimilarity, based on which we can decide if a person is really very dissimilar, and hence is not a valid face from our database.

Subsection 1.1

Finding Threshold

As explained before, we need to determine an appropriate value of the threshold such that,
if(minEuclianDist $\leq$ Threshold) then model classifies the test image
else model declares the test image as "Image with no Matching Identity"
Now there are 4 cases :

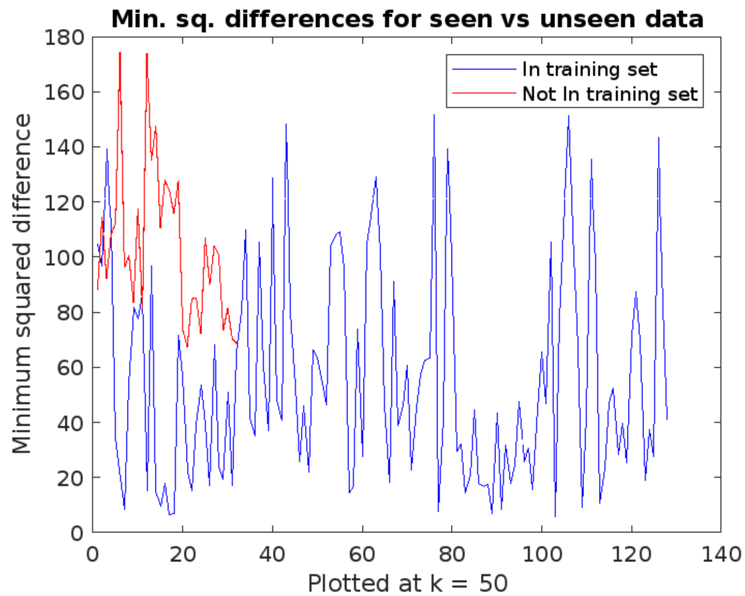
- Model classifies the test image and test image is indeed one of those from our database: True Positive (TP)
- Model classifies the test image but test image is of an imposter : False Positive (FP)
- Model declares test image as "Image with no Matching Identity" and indeed it is an imposter image: True Negative (TN)
- Model declares test image as "Image with no Matching Identity" but it was of a person present in our database : False Negative (FN)

There are several metrics which we can use to find an appropriate threshold from an available set of possible thresholds, based on the above 4 values which we get -

1. Accuracy $= \frac{TP+TN}{TP+TN+FP+FN}$
2. Precision $= \frac{TP}{TP+FP}$
3. Recall/Sensitivity $= \frac{TP}{TP+FN}$
4. Specificity $= \frac{TN}{TN+FP}$
5. F1 Score $= 2\times\left(\frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}\right)$

Among these, F1 Score is a good way to evaluate our model because it is combination of both precision and recall. But, in general, what we want is that false positives and false negatives should be minimum.

So, we added the last 4 images each of the remaining 8 people (resulting in total 32 images of unknown people, and total test dataset size of 160) to our test dataset and evaluated the model for different values of threshold, by maximizing the metrics which are explained above.
We decided to take the value of $k = 50$, because as seen in Question1, the recognition rate was maximized at k = 50.



The above plot shows the minimum dissimilarity for different test images. The Blue ones are for those images in test dataset, whose face were present in train set also. And the red ones are for those test images, which are not part of train set and consists of unknown people's faces. As seen from this plot, we should decide a threshold value in range $[60, 180]$ such that all those images for which minimum dissimilarity is greater than threshold, they are considered as unmatched. So, we iterated over threshold values in this range and found those which maximized different evaluation metrics -

- Maximizing F1 Score : The results we got are
  `Threshold Found : 151.786834`
  `Accuracy : 0.812500`
  `Precision : 0.810127`
  `Recall : 1.000000`
  `Specificity : 0.062500`
  `F1 Score : 0.895105`
  `False Positive : 30.000000`
  `False Negative : 0.000000`
  `True Positive : 128.000000`
  `True Negative : 2.000000`
  So, when F1 score is maximized, we can infer from the plot that at threshold of around 151, most of the unknown faces are classified by model (30 of them) resulting in 30 false negatives, and 2 of them are treated as with no matching identity, resulting in 2 true negative. We also observe that for this threshold, most of the metrics are having decent values, but specificity is very low, because true negatives are less.

- Maximizing specificity : This means that we are now trying that most of the unknown images should be declared as matching with no identity. The results for this are
  `Threshold Found :  67.147335`
  `Accuracy :  0.743750`
  `Precision :  1.000000`
  `Recall :  0.679688`
  `Specificity :  1.000000`
  `F1 Score :  0.809302`
  `False Positive :  0.000000`
  `False Negative :  41.000000`
  `True Positive :  87.000000`
  `True Negative :  32.000000`
  Now, from the plot we see that the best threshold found is such that all the unknown faces are identified with no matching identity, leading to true negative of 32 and precision also 1. But, on other hand, it also increased the false negatives and reduced the true positives also. So, this resulted in unidentification of some of the valid faces too, which would have decreased recognition rate.

- Maximizing area under ROC curve, which depends on ability of threshold to balance between false positives and false negatives. We ran our code with condition of this also, that is, false positives == false negatives, and the result was
  `Threshold Found :  105.141066`
  `Accuracy :  0.775000`
  `Precision :  0.859375`
  `Recall :  0.859375`
  `Specificity :  0.437500`
  `F1 Score :  0.859375`
  `False Positive :  18.000000`
  `False Negative :  18.000000`
  `True Positive :  110.000000`
  `True Negative :  14.000000`
  Now, the best threshold found is somewhere in the middle, where false positives and false negatives are balanced. For this threshold, although 18 known images which are misidentified, but at the same time, there are 14 unknown images which are also misidentified as they should have been. F1 score and other metrics also have a decent enough value, along with specificity also not being too low (as in the first case, when maximizing f1 score).

As seen from the 3 cases we described, we can say that the value of threshold to be chosen also depends on the application for which we are using our model. If the application requires more precision (that is anything which is identified should be correct, no matter even if some correct things also go to unidentified class), then 2nd metrics is better, and if it needs overall more accuracy and F1 score (but with a cost of low specificity), then 1st metrics is better. However, on a general case, we would say that 3rd metric's threshold is good as both the conditions are taken care there with no too much loss of one of them. And also almost all the other metrics are having a decent enough score for threshold of 105.14
The code we are submitting currently is based on 1st metric, but that can be changed (as we did while writing this report) for calculating over 2nd & 3rd cases as well, just by changing the if condition. We have written this in comments as well.