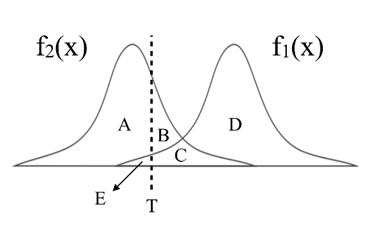
Computer Vision: from Recognition to Geometry

HW2

Name: 毛弘仁 Department: 電機四 Student ID: B04901117

Problem 1

1. Assume 𝑋 is a continuous random variable that denotes the estimated probability of a binary classifier. The instance is classified as positive if 𝑋 > 𝑇 and negative otherwise. When the instance is positive, 𝑋 follows a PDF 𝑓1(𝑥). When the instance is negative, 𝑋 follows a PDF 𝑓2(𝑥). Please specify which regions (A ~ E) represent the cases of *False Positive* and *False Negative*, respectively. Clearly explain why.

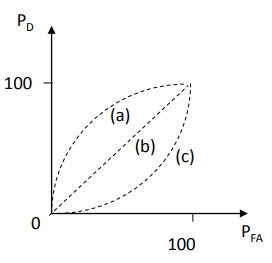


**Ans:**

False Positive: B+C, because they are both to the right of T (classified as positive) and under f2(x) (true negative).

False Negative: E, because it is to the left of T (classified as negative) and under f1(x) (true positive).

1. There are three ROC curves in the plot below. Please specify which ROC curves are considered to have reasonable discriminating ability, and which are not. Also, please answer that under what circumstances will the ROC curve fall on curve (b)?



**Ans:**

We see that (a) has reasonable discriminating ability, because given any point on the curve, PD has a higher value than PFA, which means there is a higher probability of the classifier being right than wrong. The curve of (b) means the classifier has an equal probability of being correct or incorrect, thus it does not perform better than random guessing; the curve of (c) means the classifier is more often wrong than right, so even random guessing would yield better performance. For these reasons, (b) and (c) do not have reasonable discriminating ability, as they are no better than pure guesswork.

A possible reason for the ROC curve falling on curve (b) may be that f1(x) and f2(x) overlap and cannot be differentiated, thus no matter which value of T we choose, performance is always the same as random guessing.

Problem 2

1. PCA

In this task, you need to implement PCA from scratch, which means you cannot call PCA function directly from existing packages.

1. Perform PCA on the training data. Plot the mean face and the first five eigenfaces and show them in the report.





1. Take 𝒑𝒆𝒓𝒔𝒐𝒏8\_ 𝒊𝒎𝒂𝒈𝒆6, and project it onto the above PCA eigenspace. Reconstruct this image using the first n = { 5, 50, 150, all } eigenfaces. For each n, compute the mean square error (MSE) between the reconstructed face image and the original 𝒑𝒆𝒓𝒔𝒐𝒏8\_ 𝒊𝒎𝒂𝒈𝒆6. Plot these reconstructed images with the corresponding MSE values in the report.

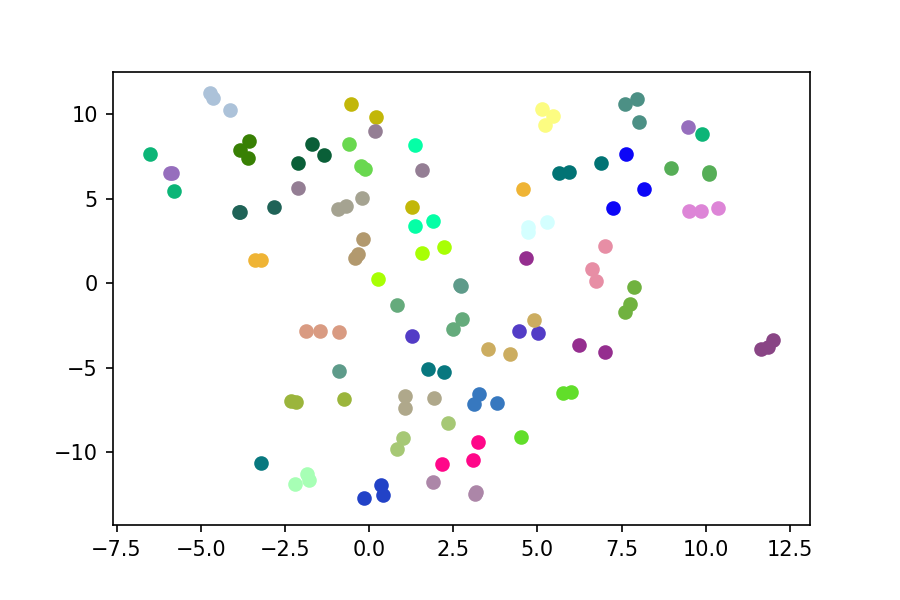
used 5, reconstruction error 693.7021744253194

used 50, reconstruction error 119.20025848187896

used 150, reconstruction error 38.75113372522805

used 279, reconstruction error 1.6014640738105883e-24

1. Reduce the dimension of the image in testing set to dim = 100. Use t-SNE to visualize the distribution of test images.



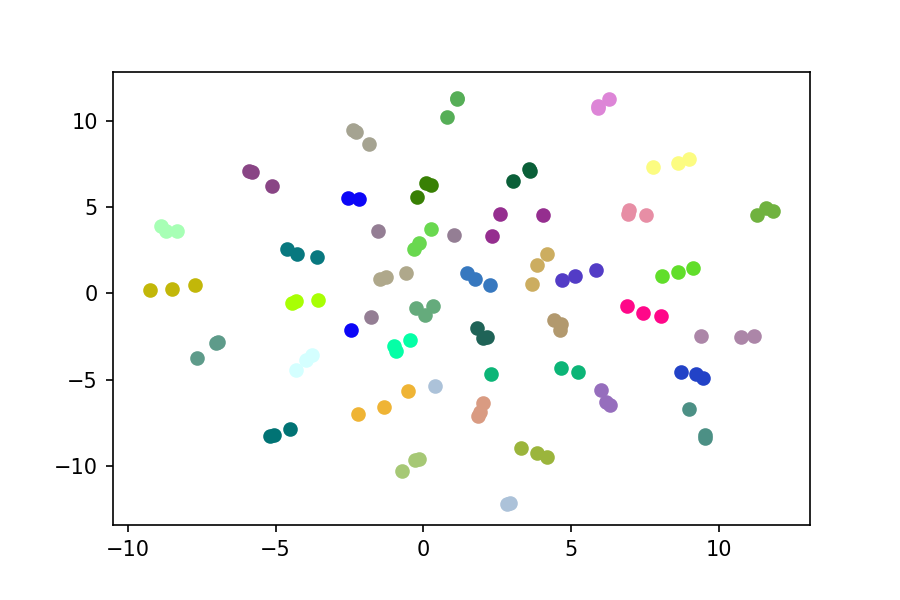
1. LDA

In this task, you need to implement LDA from scratch, which means you cannot call LDA function directly from existing packages.

1. Implement LDA and plot first 5 Fisherfaces.

1. Use t-SNE to visualize the distribution of the projected testing data, which has the dimension of 30.



1. To apply the k-nearest neighbors (k-NN) classifier to recognize the testing set images, please determine the best k and n values by 3-fold cross-validation.

For simplicity, the choices for such hyper-parameters are:

k = {1, 3, 5} and n = {3, 10, 39}.

Please show the cross-validation results and explain your choice for (k, n). Also, show the recognition rate on the testing set using your hyper-parameter choice. Please apply the above comparing method on both PCA and LDA.

Do you observe an improved recognition rate using fisherfaces (compared to eigenfaces obtained by PCA)? If so (or if not), what might be the possible explanation?

PCA

Choose N=39, K=1, validation accuracy=0.9180555555555556

N: 3 K: 1 val\_acc: 0.6888888888888889

N: 3 K: 3 val\_acc: 0.6124999999999999

N: 3 K: 5 val\_acc: 0.5513888888888888

N: 10 K: 1 val\_acc: 0.8888888888888888

N: 10 K: 3 val\_acc: 0.7819444444444446

N: 10 K: 5 val\_acc: 0.7013888888888888

N: 39 K: 1 val\_acc: 0.9180555555555556

N: 39 K: 3 val\_acc: 0.8722222222222222

N: 39 K: 5 val\_acc: 0.7805555555555556

best\_N: 39 best\_K: 1 best\_val\_acc: 0.9180555555555556

test\_acc: 0.9583333333333334

LDA

N: 3 K: 1 val\_acc: 0.41944444444444445

N: 3 K: 3 val\_acc: 0.4152777777777778

N: 3 K: 5 val\_acc: 0.4138888888888889

N: 10 K: 1 val\_acc: 0.7833333333333332

N: 10 K: 3 val\_acc: 0.7833333333333332

N: 10 K: 5 val\_acc: 0.786111111111111

N: 39 K: 1 val\_acc: 0.9236111111111112

N: 39 K: 3 val\_acc: 0.9291666666666667

N: 39 K: 5 val\_acc: 0.9319444444444445

best\_N: 39 best\_K: 5 best\_val\_acc: 0.9319444444444445

test\_acc: 0.9166666666666666

Problem 3

1. Build a CNN model and train it on the given dataset. Show the architecture of your model in the report.

1. Report your training / validation accuracy, and plot the learning curve (loss, accuracy) of the training process.
2. Visualize at least 6 filters on both the first and the last convolutional layers.
3. Visualize high-level and low-level features of 1000 validation data (100 for each class) extracted from different layers, and explain what you have observed from the two t-SNE plots.