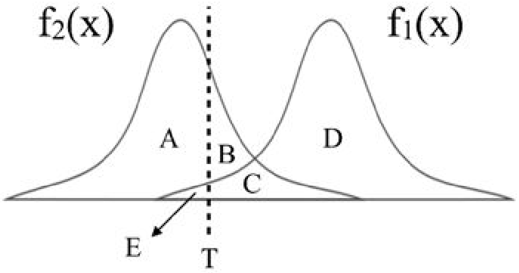
Up Computer Vision: from Recognition to Geometry
HW2

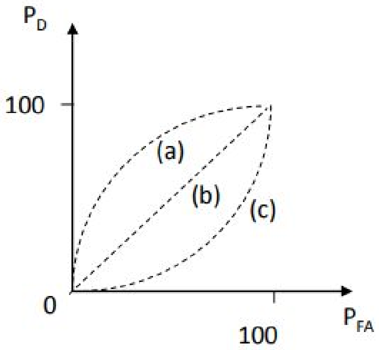
Name: _________ Department: _________ Student ID: _________

Problem 1

(a) 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 $f_1(x)$. When the instance is negative, X follows a PDF $f_2(x)$. Please specify which regions (A ~ E) represent the cases of *False Positive* and *False Negative*, respectively. Clearly explain why. (6%)



(b) 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)? (6%)



Problem 2

(a) Given a convolutional layer which contains:

n kernels with size $k * k * n_{in}$
padding size p,
stride size (s,s).

With input feature size $W * W * n_{in}$, calculate the size of output feature $W_{out} * W_{out} * n_{out}$
.

(i) W_out=? (2%)
(ii) n_out =? (2%)

(b) k=5, s=2, p=1, n=256, W=64, calculate the number of parameters in the convolutional layer (4%)

Problem 3 (report 35% + code 5%)

(a) PCA (15%)

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. (5%)

2. Take $\boldsymbol{person_8\_ image_6}$, 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 $\boldsymbol{person_8\_ image_6}$. Plot these reconstructed images with the corresponding MSE values in the report. (5%)

3. Reduce the dimension of the image in testing set to dim = 100. Use t-SNE to visualize the distribution of test images. (5%)

4. (bonus 5%) Implement the Gram Matrix trick for PCA. Compare the two reconstruction images from standard PCA/Gram Matrix process. If the results are different, please explain the reason. Paste the main code fragment(screenshot) on the report with discussion.

(b) k-NN (10%)

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:

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

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.

(c) K-means (10%)

Reduce the dimension of the images in the first 10 class of training set to dim = 10 using PCA. Implement the k-means clustering method to classify the these images.

1. Please use weighted Euclidean distance to implement the k-means clustering. The weight of the best 10 eigenfaces is

$$[\ 0.6, 0.4, 0.2, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1\ ]$$

2. Please visualize the features and the centroids to 2D space with the first two eigenfaces for each iteration in k-means clustering (up to 5 images). (5%)

3. Compare the results of K-means and ground truth. (5%)

Problem 4 (report 30% + baseline 10%)

(a) MNIST Classification (20%)

1. Build a CNN model and a Fully-Connected Network and train them on the MNIST dataset. Show the architecture of your models and the correspond parameters amount in the report. (5%)

2. Report your training / validation accuracy, and plot the learning curve (loss, accuracy) of the training process. (figure 5%+ baseline 5%)

3. Compare the results of both models and explain the difference. (5%)

(b) Face Recognition (20%)

1. Extract image feature using pytorch pretrained alexnet and train a KNN classifier to perform human face recognition on the given dataset. Report the validation accuracy. (5%)

2. Build your own model and train it on the given dataset to surpass the accuracy of previous stage. Show and explain the architecture of your model and report the validation accuracy. Paste the main code fragment(screenshot). (5%)

3. Plot the learning curve of your training process(training/validation loss/accuracy) in stage 2, explain the result you observed. (5%)

4. Pick the first 10 identities from dataset. Visualize the features of training/validation data you extract from pretrained alexnet and your own model to 2D space with t-distributed stochastic neighbor embedding (t-SNE), compare the results and explain the difference. (5%)