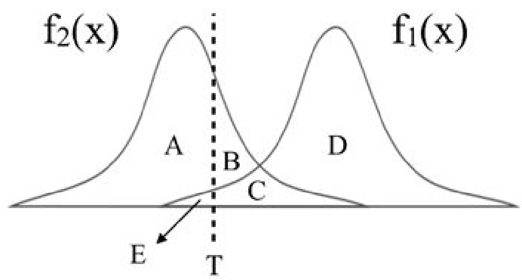
Computer Vision: from Recognition to Geometry
HW2

Name: 林棋祥 Department: 電機四 Student ID: b04505004

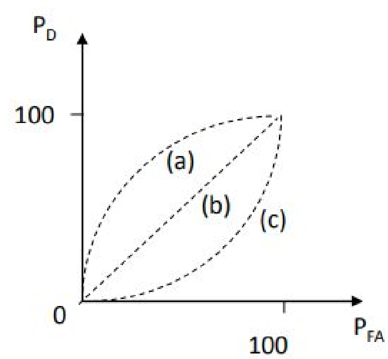
Problem 1

(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.



False Positive: B, C, 曲線f2(x)在x>T的曲線面積即代表把negative instance判斷成positive
False Negative: E, 曲線f1(x)在x<T的曲線面積即代表把positive instance判斷成negative

(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)?



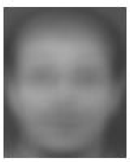
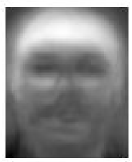
Curve (a) has reasonable discriminating ability and curve (c) doesn't.
當model完全沒學到任何有意義的feature時，model隨機判斷，結果為curve (b)。

Problem 2

(a) PCA

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

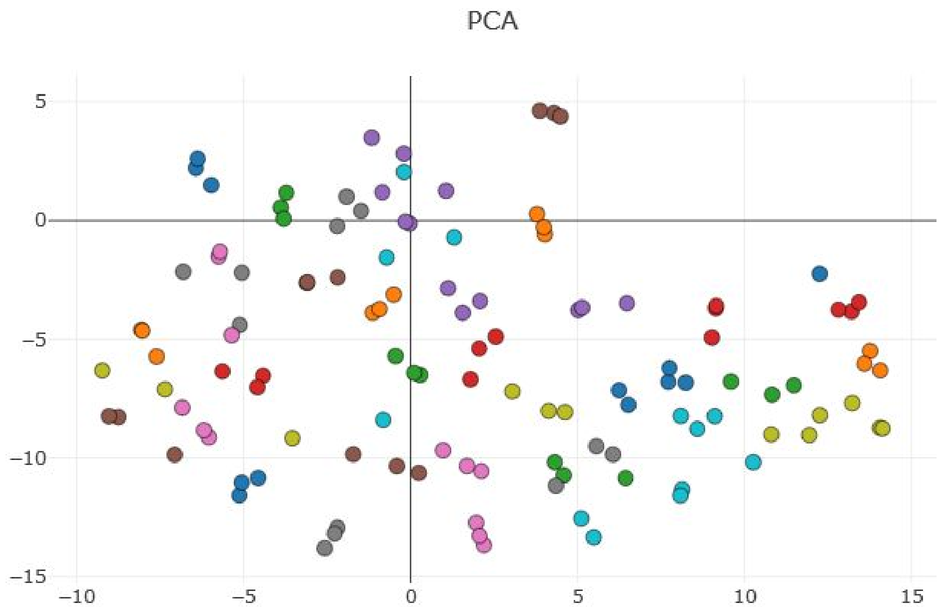
1. Perform PCA on the training data. Plot the mean face and the first five eigenfaces and show them in the report.

mean	eigen1	eigen2	eigen3	eigen4	eigen5
					

2. Take $person_{8}\_\ image_{6}$, 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 $person_{8}\_\ image_{6}$. Plot these reconstructed images with the corresponding MSE values in the report.

origin	5 eigens	50 eigens	150 eigens	279 eigens
 MSE: 0	 MSE: 693	 MSE: 119	 MSE: 40	 MSE: 8.9e-12

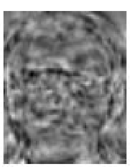
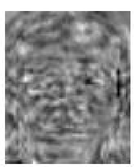
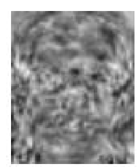
3. Reduce the dimension of the image in testing set to dim = 100. Use t-SNE to visualize the distribution of test images.



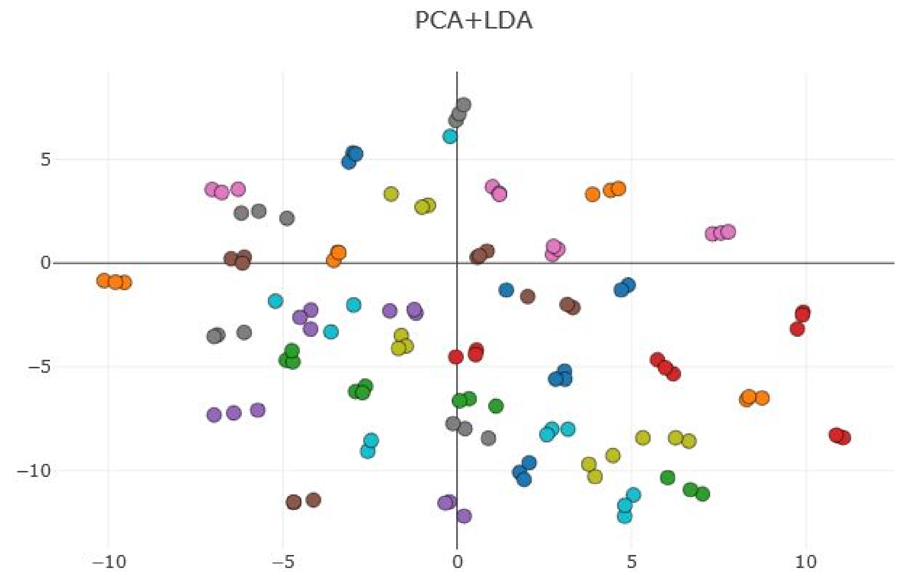
(b) LDA

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.

fisher1	fisher2	fisher3	fisher4	fisher5
				

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



(c) 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.

PCA + KNN

Training Accuracy

k	n	fold1	fold2	fold3	average
1	3	80.0%	72.5%	65.8%	72.8%
1	10	93.8%	95.0%	90.0%	92.9%
1	**39**	**96.3%**	**97.5%**	**91.7%**	**95.1%**
3	3	67.5%	61.3%	60.8%	63.2%
3	10	81.3%	81.3%	76.7%	79.7%
3	39	90.0%	87.5%	80.0%	85.8%
5	3	62.5%	57.5%	55.0%	58.3%
5	10	72.5%	68.8%	66.7%	69.3%
5	39	78.8%	77.5%	73.3%	76.5%

選擇(k, n) = (1, 39)，因為平均正確率最高。Recognition rate on the testing set = 95.8%

PCA + LDA + KNN

Training Accuracy

k	n	fold1	fold2	fold3	average
1	3	51.3%	37.5%	40.0%	42.9%
1	10	85.0%	83.8.0%	81.7%	83.5%
1	39	96.3%	96.3%	90.0%	94.2%
3	3	50.0%	37.5%	40.8%	42.8%
3	10	83.8%	83.8%	81.7%	83.1%
3	39	95.0%	96.3%	91.7%	94.3%
5	3	50.0%	37.5%	40.0%	42.5%
5	10	83.8%	83.8%	81.7%	83.1%
5	**39**	**96.3%**	**97.5%**	**91.7%**	**95.1%**

選擇(k, n) = (5, 39)，因為平均正確率最高。Recognition rate on the testing set = 95.8%

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?
沒有進步，認為原因可能是dataset太小，fisherface的降維效果沒辦法超過eigenface太多，使knn最後的正確率一致。

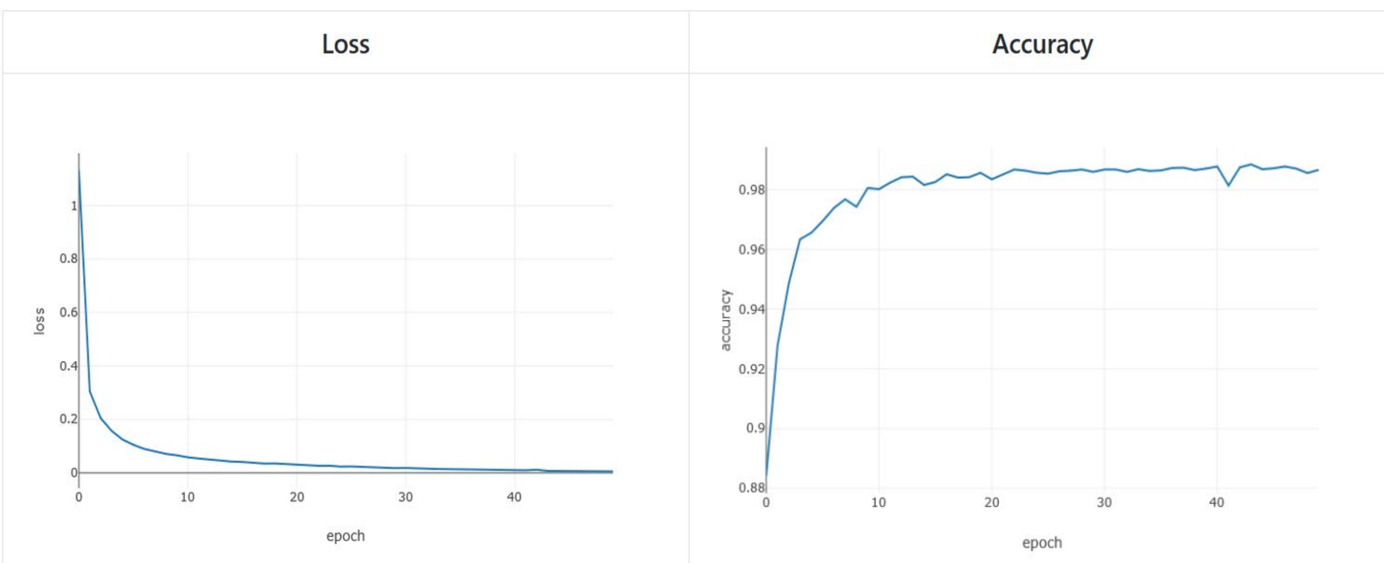
Problem 3

(a) Build a CNN model and train it on the given dataset. Show the architecture of your model in the report.
採用助教建議的Lenet5，使用ReLU，conv層用maxpooling降維。

```
Lenet5(
  (conv1): Conv2d(1, 6, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
  (conv2): Conv2d(6, 16, kernel_size=(5, 5), stride=(1, 1))
  (fc1): Linear(in_features=400, out_features=120, bias=True)
  (fc2): Linear(in_features=120, out_features=84, bias=True)
  (fc3): Linear(in_features=84, out_features=10, bias=True)
)
```

(b) Report your training / validation accuracy, and plot the learning curve (loss, accuracy) of the training process.

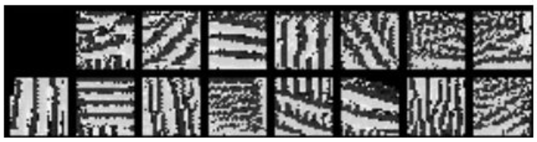
Training	Validation
99.92%	98.67%



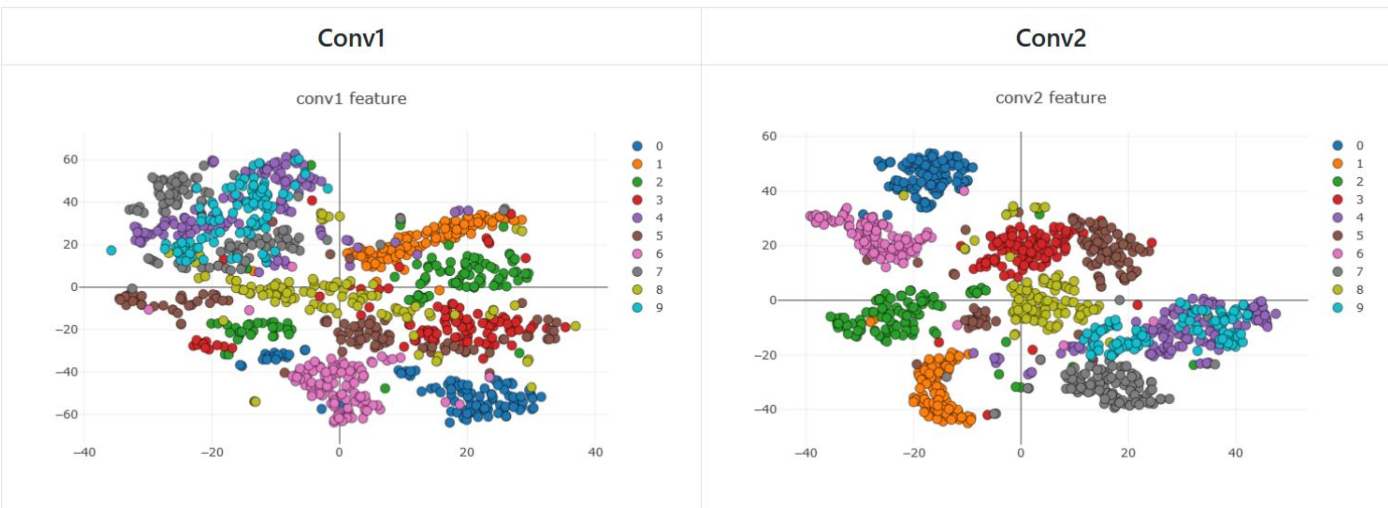
(c) Visualize at least 6 filters on both the first and the last convolutional layers.

Conv1

Conv2



(d) 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.



可以看到Conv2(high-level features)把不同的數字分得比較開，Conv1(low-level features)會把不同的數字重疊。這是由於CNN在前面會先學著判斷形狀、邊界等每個數字都有的特性，後面幾層才會學更高階的特徵，這些更高階的特徵才能幫助fc layer做更好的分類。