HW10 Report

學號：b07901039　系級：電機二　　姓名：劉知穎

1. (2%) 任取一個 baseline model (sample code 裡定義的 fcn，cnn，vae) 與你在 kaggle leaderboard 上表現最好的單純 autoencoder 架構的 model（如果表現最好的 model 就是 sample code 裡定義的 model 的話就再任選一個，e.g. 如果 cnn 最好那就再選 fcn），對各自重建的 testing data 的 image 中選出與原圖 mse 最大的兩張加上最小的兩張並畫出來。（假設有五張圖，每張圖經由 autoencoder A 重建的圖片與原圖的 MSE 分別為 [25.4, 33.6, 15, 39, 54.8]，則 MSE 最大的兩張是圖 4、5 而最小的是圖 1、3）。須同時附上原圖與經 autoencoder 重建的圖片。（圖片總數：(原圖+重建)*(兩顆 model)*(mse 最大兩張+mse 最小兩張) = 16 張）

   (1) kaggle leaderboard 上表現最好的單純 autoencoder 架構 (score: 0.64440)

   模型架構：使用 cnn auto-encoder，在 encoder 和 decoder 中間加對稱的 dense layer。

   Encoder 架構：

   Conv2d(3, 6, 3, 1, 1) / Conv2d(6, 12, 3, 1, 1) / MaxPool(2, 2, 0) / BatchNorm2d(12) / Dropout(0.5) / ReLU()

   Conv2d(12, 24, 3, 1, 1) / MaxPool(2, 2, 0) / BatchNorm2d(24) / Dropout(0.5) / ReLU()

   Conv2d(24, 48, 3, 1, 1) / MaxPool(2, 2, 0) / BatchNorm2d(48) / Dropout(0.5) / ReLU()

   Conv2d(48, 96, 3, 1, 1) / MaxPool(2, 2, 0) / BatchNorm2d(96) / Dropout(0.5) / ReLU()

   Linear(96*2*2, 96) / ReLU() / Linear(96, 32)

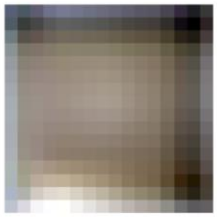
   Decoder 架構

   Linear(96, 32) / ReLU() / Linear(96*2*2, 96)

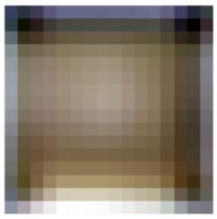
   ConvTranspose2d(96, 48, 3, 1, 1)/ Upsample(2, 'nearest')/BatchNorm2d(48)/ Dropout(0.5)/ ReLU()
   ConvTranspose2d(48, 24, 3, 1, 1)/ Upsample(2, 'nearest')/BatchNorm2d(24)/ Dropout(0.5)/ ReLU()
   ConvTranspose2d(24, 12, 3, 1, 1)/ Upsample(2, 'nearest')/BatchNorm2d(12)/ Dropout(0.5)/ ReLU()
   ConvTranspose2d(12, 6, 3, 1, 1)/ConvTranspose2d(6, 3, 3, 1, 1)/ Upsample(2, 'nearest')/BatchNorm2d(3)/ Dropout(0.5)/ Tanh()

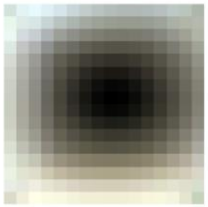
   optimizer 使用 SGD(learning rate = 1e-3, momentum=0.8) train 2000 個 epoch，存下 minibatch loss 最小的 model。

   左邊為原圖，右邊為 reconstructed 的圖，最右側的數字為原圖與 reconstructed 圖的 mse。



6.80


53.34


7.24


51.74

(2) sample code 中 fcn model (score = 0.59329)

模型架構：

Encoder 架構：

Linear(32*32*3, 128)/ ReLU()/ Linear(128, 64)/ ReLU()/ Linear(64, 12)/ ReLU()/ Linear(12, 3)

Decoder 架構：

Linear(3, 12)/ ReLU()/ Linear(12, 64)/ ReLU()/ Linear(64, 128)/ ReLU()/ Linear(128, 32*32*3)/ Tanh()

optimizer 使用 AdamW(learning rate = 1e-3) train 1000 個 epoch，存下 minibatch loss 最小的 model。

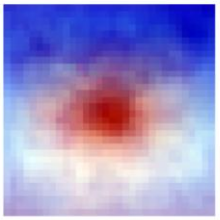
左邊為原圖，右邊為 reconstructed 的圖，最右側的數字為原圖與 reconstructed 圖的 mse。



5.04


39.92


5.66


39.44

結論：

(1) 不論是 cnn 或是 fcn 架構 reconstruct 出來的圖與原圖從肉眼來看都有很大的差距，因為我的 cnn 模型在做完 convolution 後有過幾層 linear layer 將向量壓縮到 32 維，而 fcn 的 bottle neck 則是 3 維的向量，因此會有許多資訊遺失。

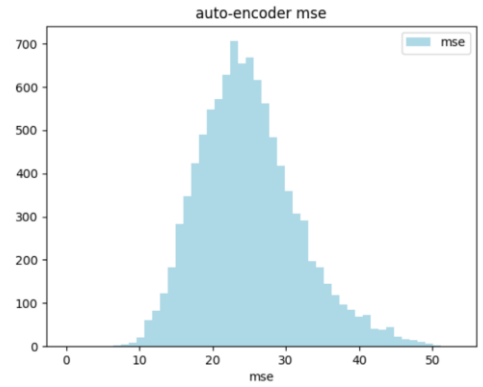
(2) 我發現我的 model 重建 mse 較大的圖大多是白色背景的圖，內容物則不一定。似乎是因為訓練資料大多是有背景的圖片，所以模型認為白色背景的圖是異常的。fcn 模型也會被背景影響，上圖中有背景的青蛙重建後的 mse 為 5.66，白色背景的則是 39.44，且 mse 5.66 的青蛙成功重建的地方大多是背景的綠色，青蛙本體的顏色則完全不同。由此可知這兩個模型在判斷上都會受背景影響。

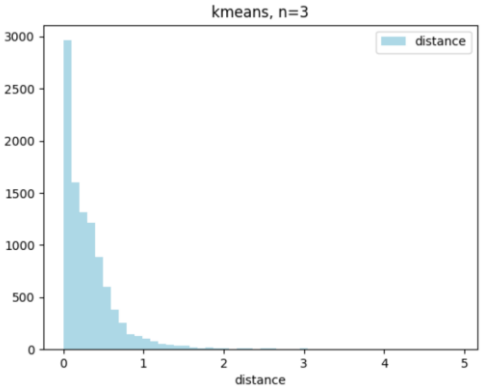
(3) 雖然肉眼上看不太出來，但照理說 cnn 因為用 convolution 比起 fcn 直接將向量攤平更能學到圖片的特徵，所以得到的 auc score 比較高。

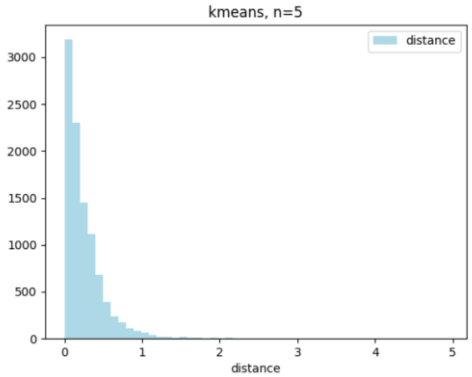
2. (1%) 嘗試把 sample code 中的 K-means 與 PCA 分別做在 autoencoder 的 encoder output 上，並回報兩者的 auc score 以及本來 model 的 auc。autoencoder 不限。不論分數與本來的 model 相比有上升還是下降，請同學簡述原因。

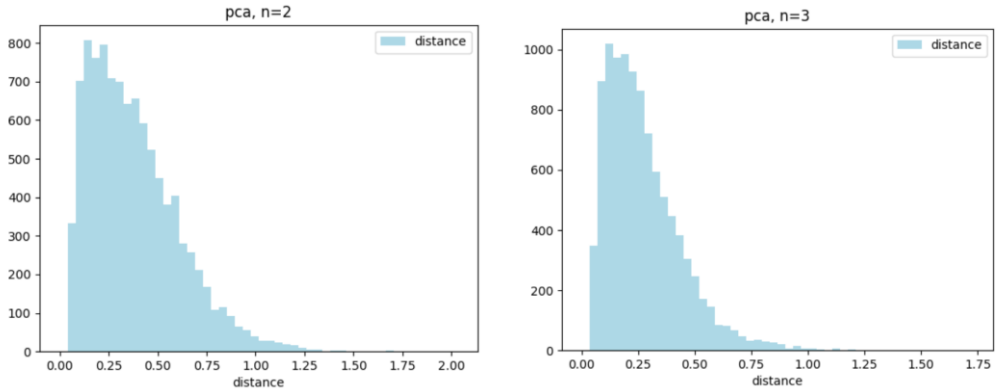
模型架構與訓練方式如第一題第一小題所述。將 encoder output (維度 32 的向量)分別做 K-means 和 PCA。

Evaluation mode	Auc score
Reconstruction mse	0.64440
k-means, num of clusters = 3	0.56195
k-means, num of clusters = 5	0.54320
Pca, num of components = 2	0.59532
Pca, num of components = 3	0.56061

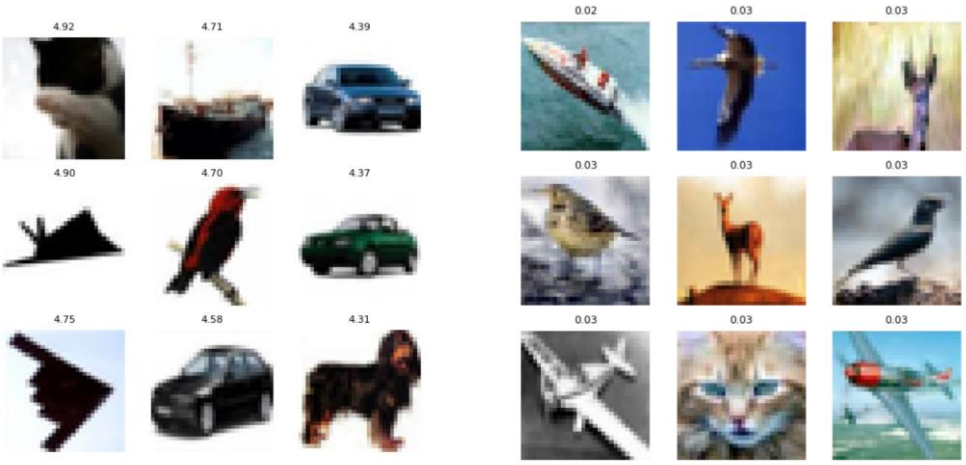








將 k-means 和 pca 算出來的 distance、auto-encoder reconstruction mse 在(0, 距離或 mse 最大值)之間分 50 個區間，作圖每個區間的資料個數。發現 auc score 較高的 evaluation 方法，得到的 mse 或距離分布較廣，例如 auto-encoder 得到的圖，在 mse 較大處個數衰減較其他方法慢。而 k-means 和 pca 大部分資料都集中在 distance 較小處，得到的 auc score 較差。比較同用 k-means 或 pca 但 n 不同的兩組圖，當 n 越大時，資料越集中在距離較小處，得到的 auc score 較差。

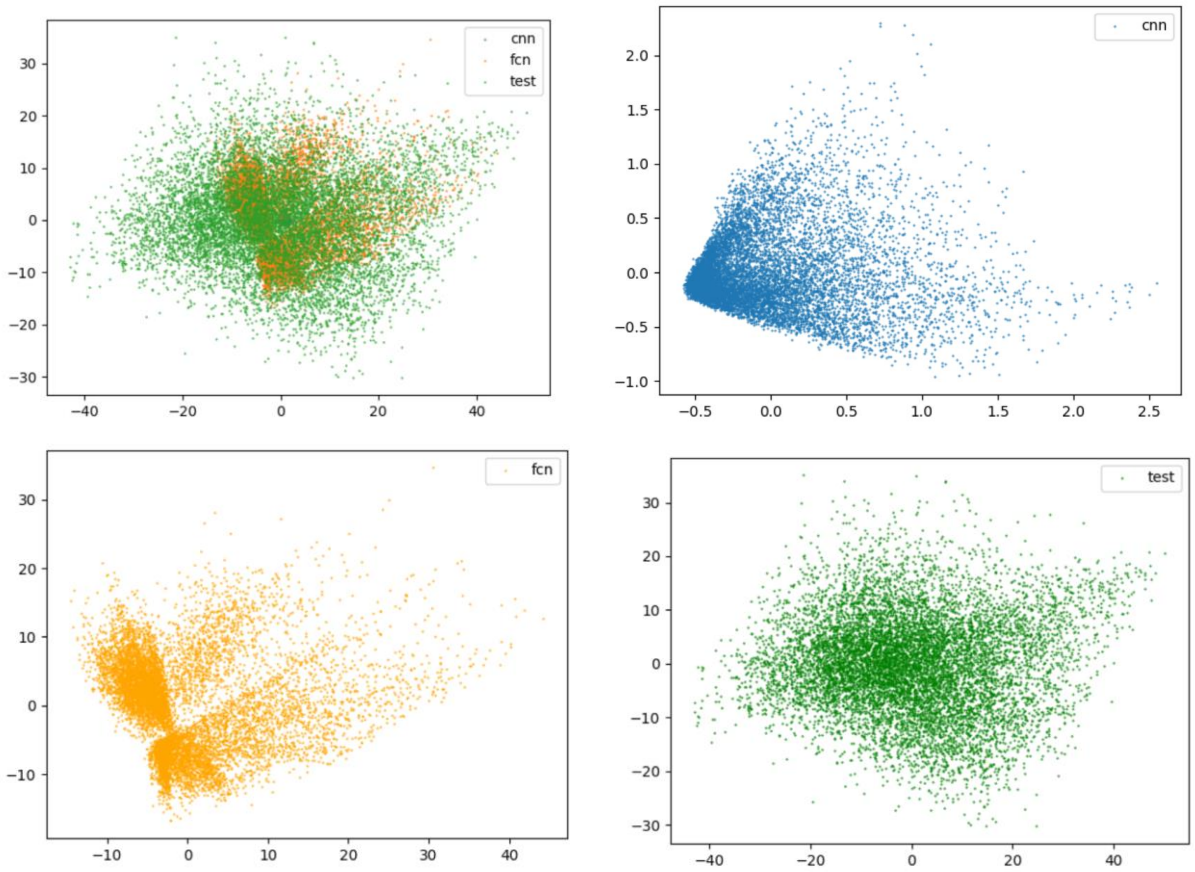


上圖左邊九張圖是經過 k-means, n=3 後 distance 最大的 9 張圖；右邊則是 distance 最小的九張。

和第一題用 auto-encoder reconstruct 的結果相同，用 k-means 做 cluster 時，接近白色背景的圖與大部分有背景的圖相比是異常的。

3. (1%) 如 hw9，使用 PCA 或 T-sne 將 testing data 投影在 2 維平面上，並將 testing data 經第 1 題的兩顆 model 的 encoder 降維後的 output 投影在 2 維平面上，觀察經 encoder 降維後是否分成兩群的情況更明顯。（因未給定 testing label，所以點不須著色）

   使用 PCA 將第一題所述的兩個模型的 encoder output 和 testing data 用 pca 降維投影到二維平面上。



結論：

使用 encoder 降維後資料有集中的趨勢，cnn 模型和 fcn 模型資料集中的方式不一樣。Cnn 模型的 encoder output 資料沒有明顯分成兩群，但密集和稀疏的差距有比原本的 testing data 更明顯。fcn 模型的 encoder output 有兩個集中的區域。

4. (2%) 說明為何使用 auc score 來衡量而非 binary classification 常用的 f1 score。如果使用 f1 score 會有什麼不便之處？

$$sensitivity = \frac{true\ positive}{true\ positive + false\ negative}$$
$$specificity = \frac{true\ negative}{true\ negative + false\ positive}$$
$$precision = \frac{true\ positive}{true\ positive + false\ positive}$$
$$recall = \frac{true\ positive}{true\ positive + false\ negative}$$
$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recal}$$
$$AUC = Area\ under\ ROC\ curve$$

ROC curve 的橫軸是 specificity、縱軸是 sensitivity。Auc score 越高，代表 true positive 和 true negative 越高。

因為 F1 score 沒有考量 true negative，在資料分布極不平均的情況可能會有問題。例如在做 anomaly detection 時，異常的資料在 testing data 裡占比很小。在極端的情況下，若模型只學會將所有看到的資料判斷成正常資料。False positive 相比總資料數是很小的值，false negative 等於 0，true positive 是所有正常資料的數量。用 F1 score 衡量時，會得到趨近於 1 的值，但實際上模型甚麼都沒有學會。若用 AUC score 衡量，因為同時會考量到 true positive 和 true negative，可以衡量模型對於正常與異常資料的綜合表現。

參考資料：

https://stackoverflow.com/questions/44172162/f1-score-vs-roc-auc

https://en.wikipedia.org/wiki/F1_score