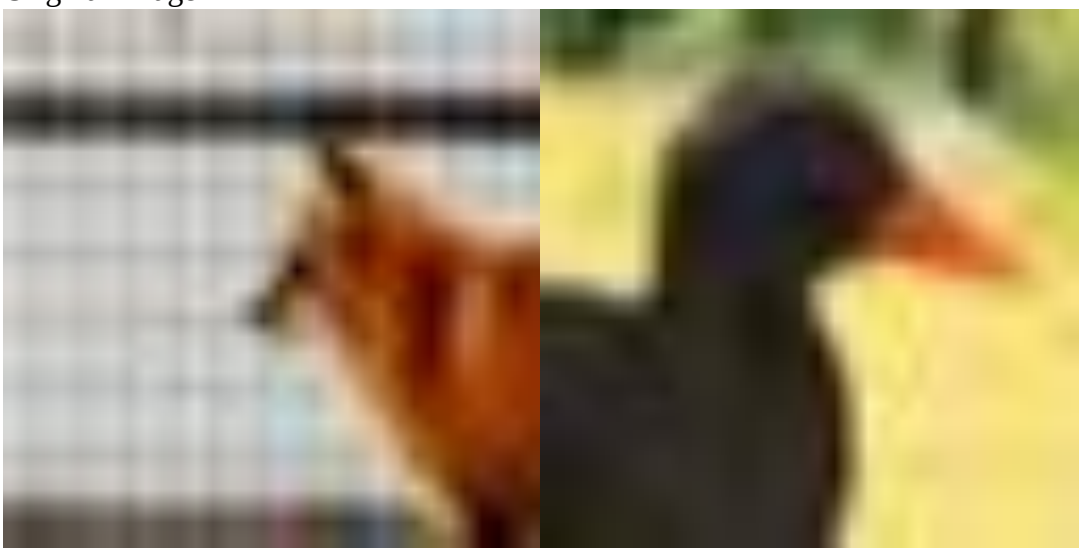
學號：R11921A16　系級：電機所碩一　姓名：何秉學

Kaggle Competition

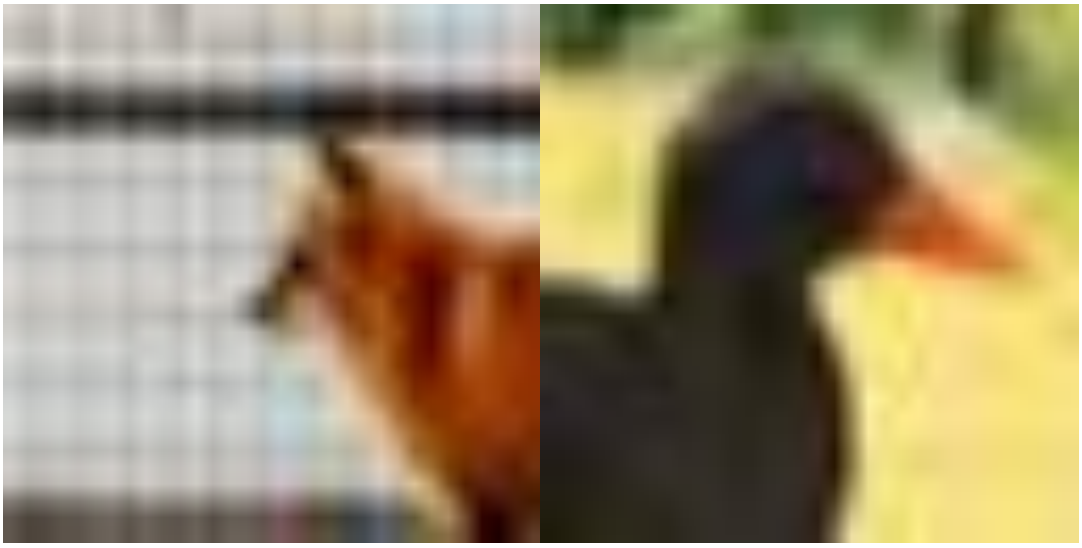
1. (1%) 請附上你在 kaggle 競賽上表現最好的降維以及分群方式，並條列五種不同降維維度的設定對應到的表現(public / private accuracy)
   註 1: auto-encoder 和 PCA 只要任一維度不一樣即可算是一種組合。
   註 2: 不限於以上方法，同學也可以使用任何其他 embedding algorithm 實現降維。

Public Score	Private Score	Reduced Method	Latent Dimension	Reduced Dimension
0.78355	0.77733	PCA	32	8
0.78333	0.77755	tSNE	32	8
0.77822	0.78244	PCA	32	16
0.77800	0.78377	PCA	32	64
0.77222	0.77133	PCA	128	8
0.77044	0.77244	PCA	64	8

2. (1%) 從 train.npy 選出不同類別的 2 張圖，貼上原圖以及你的 autoencoder reconstruct 的圖片。用 Mean Square Error 計算這兩張圖的 reconstruction error, 並說明該 error 與 kaggle score 的關係。
   註 1: 所以一共要貼上 4 張圖片。
   註 2: 原圖請貼上做 augmentation 之前的版本。
   Original Image



   Reconstruct Image



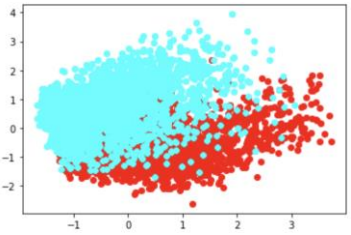
The error loss made by MSELoss is 0.0162 and 0.0055 respectively. And the checkpoint is assigned by the highest score version on Kaggle. When you assigned different checkpoint parameter on Kaggle, you'll find that they have positive correlation.

3. (2%) 請使用 pca 以及 tsne 兩種方法, 將 visualization.npy 的圖片經過 autoencoder 降維後得到之 latent vector，進一步降維至二維平面並作圖。並說明兩張圖之差異。

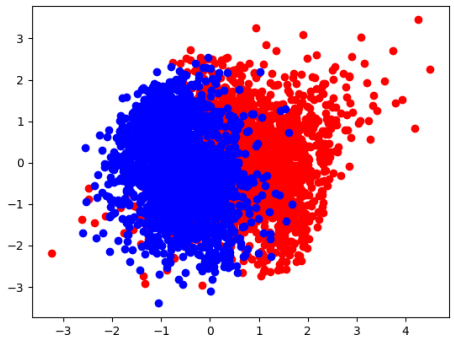
   註 1: visualization.npy 前 2500 張 label 為 0；後 2500 張 label 為 1

   註 2: 一共要貼上 2 張圖片。

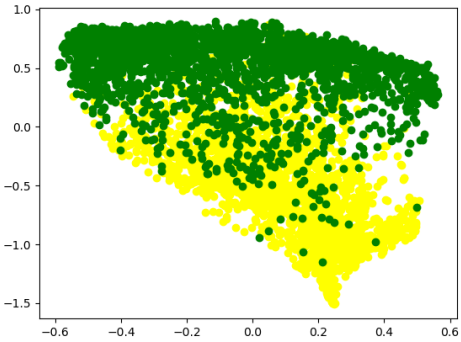
   註 3: 範例圖片如下（顏色、分佈不用完全一樣）



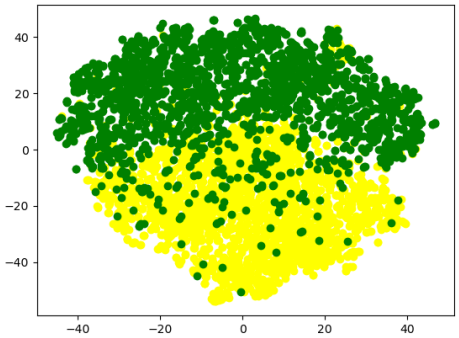
The outcome of the testing is as below. I used PCA with default configuration and tSNE with 250 iterations, 500 iterations and 1000 respectively. And the model is used that same as the previous question, that is, the best score on Kaggle. You can see that the result generated from PCA is crowded that can not split the data clearly. But, tSNE can split the data as much as better when iteration getting higher. You can see that the result from tSNE with 500 iterations, 750 iterations, and 1000 iterations are similar.



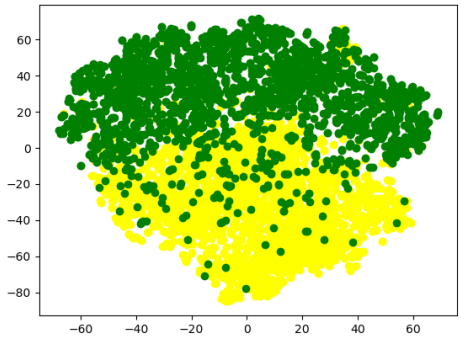
▲PCA



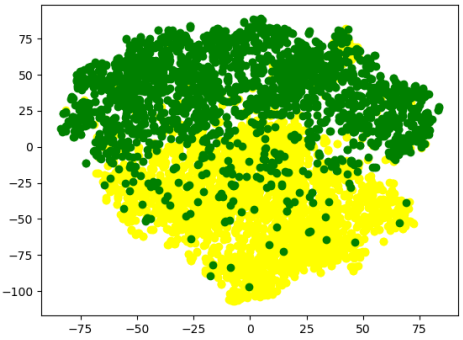
▲tSNE with iter=250



▲tSNE with iter=500



▲tSNE with iter=750



▲tSNE with iter=1000

4. (4%) Refer to math problem：
https://hackmd.io/@g4HRMJCzQL2hzLedRcbVPQ/SyCBoc1qt