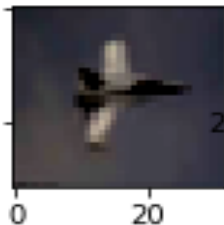
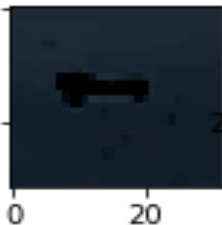
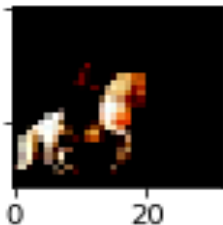
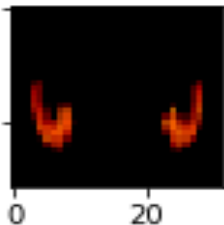
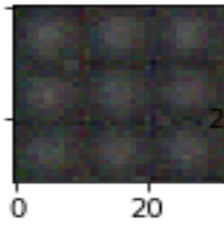
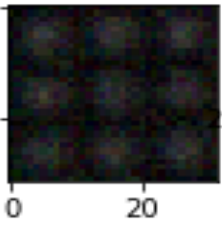
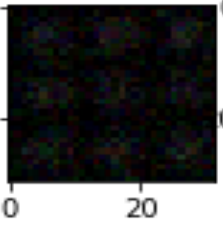
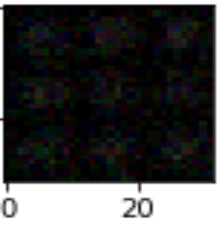
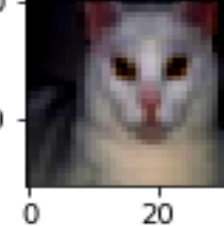
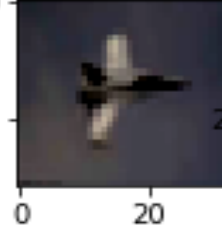
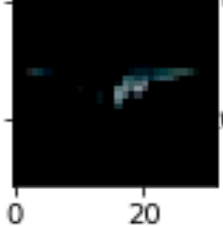
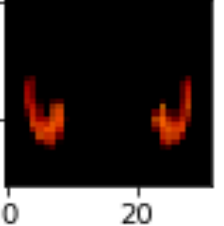
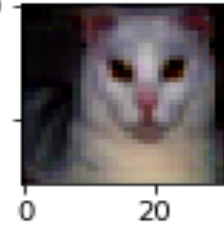
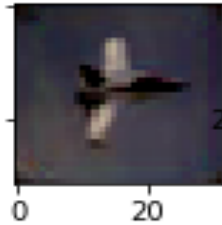
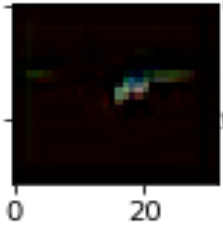
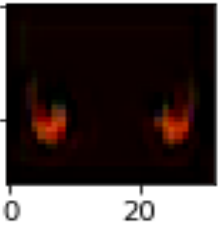


1. (2%) 任取一個 baseline model (sample code 裡定義的 fcn, cnn, vae) 與你在 kaggle leaderboard 上表現最好的單純 autoencoder 架構的 model (如果表現最好的 model 就是 sample code 裡定義的 model 的話就再任選一個, e.g. 如果 cnn 最好那就再選 fcn), 對各自重建的 testing data 的 image 中選出與原圖 mse 最大的兩張加上最小的兩張並畫出來。(假設有五張圖, 每張圖經由 autoencoder A 重建的圖片與原圖的 MSE 分別為 [25.4, 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)

Loss	最小	次小	次大	最大
fcn (best model) 原圖				
fcn (best model) 重建				
cnn 原圖				
cnn 重建				

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。

原本的 source: 0.57142

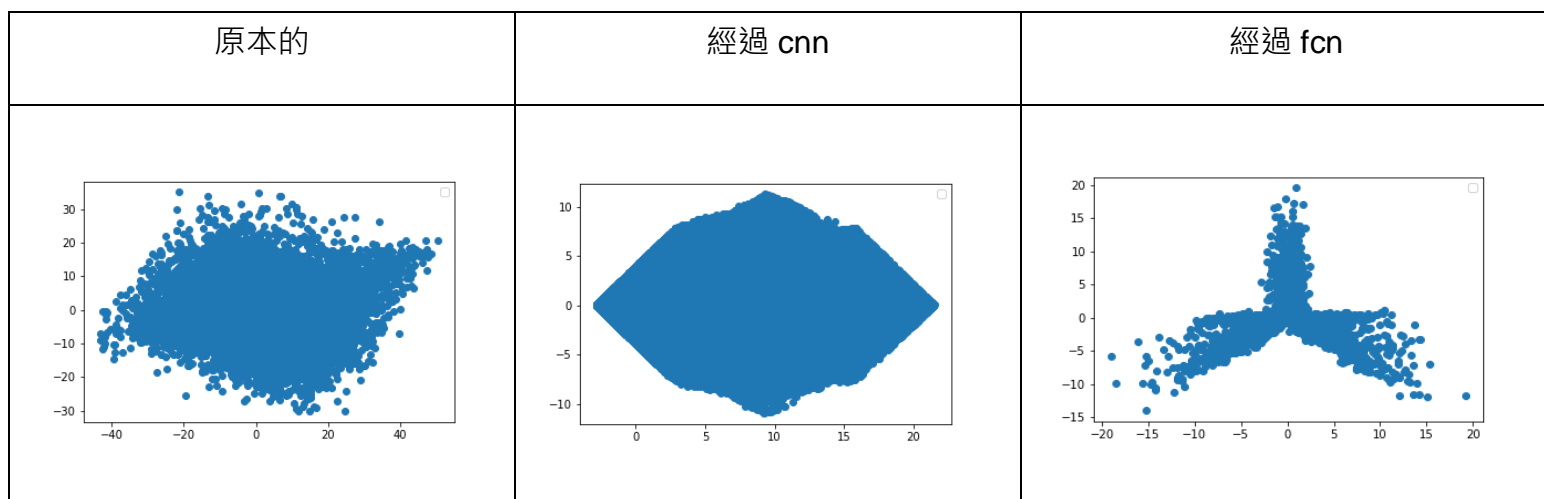
經過 K-means(n= 3) : 0.58742

經過 PCA(n=2): 0.57638

都有略微上升，因為其實 autoencoder 效果不佳(參考下一題)，所以 K-means 或 PCA 會有幫助。BTW，如果直接把 test data 拿去做 PCA 或 K-means 的話，accuracy 就有 0.59 了，所以經過 encoder 完全沒有幫助呢，還不如直接做 PCA 或 K-means。

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)

都使用 pca 降到 2 維



沒有，看起來沒有分得更好，所以 accuracy 也並不佳。但為什麼 autoencoder 效果不佳，我的理解是很難設計出一個 autoencoder 剛好可以抽取特徵(可以看第一題的圖片)，卻不會完全的還原各種圖片。要不是 model 太弱，導致無法還原(fcn)，不然就是強，會 generalize 的還原了各種圖片(cnn)，所以要很好的抽取特徵的 model 要不強也不弱才行。

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？

$$F1 = 2TP/(2TP+FP+FN)$$

因為 **anomaly data** 可能遠少於正常的 **data**，所以即使沒有捉出任何 **anomaly**，**FP** 所占比例還是很低，**f1 score** 還是會很高。而使用 **auc score** 的話，就不會因為資料的不平衡而受影響了。