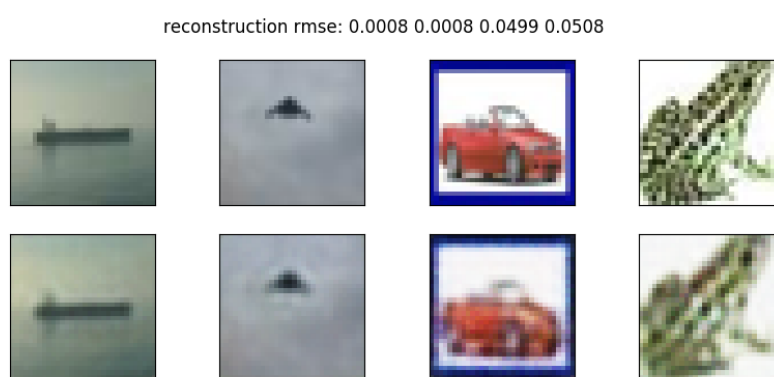
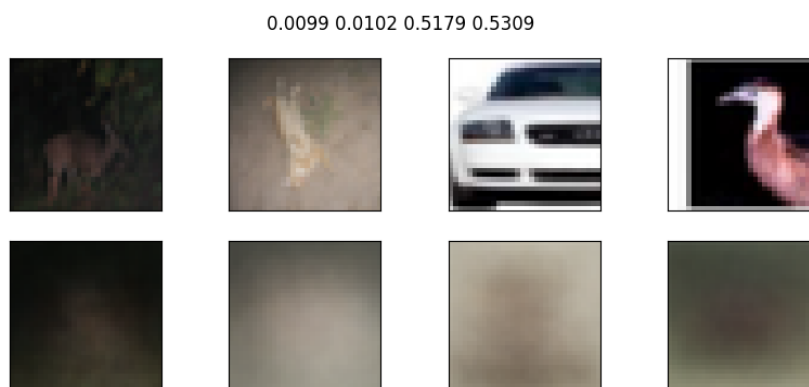


1. (2%) 任取一個 baseline model (sample code 裡定義的 fcn, cnn, vae) 與你在 kaggle leaderboard 上表現最好的 model (如果表現最好的 model 就是 sample code 裡定義的 model 的話就再任選一個, e.g. 如果 cnn 最好那就再選 fcn), 對各自重建的 testing data 的 image 中選出與原圖 mse 最大的兩張加上最小的兩張並畫出來。(假設有五張圖, 每張圖經由 autoencoder A 重建的圖片與原圖的 MSE 分別為 [25.4, 33.6, 15, 39, 54.8], 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)

首先第一個 model 用的是 cnn-based autoencoder, 重建的結果如下圖:

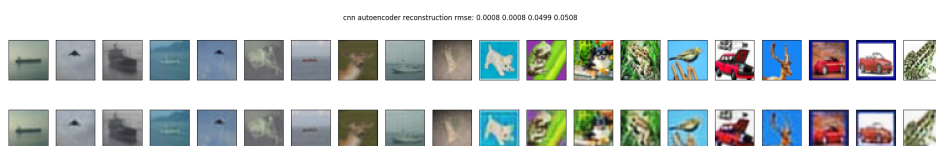


第二個 model 用的是 fcn-based autoencoder, 重建的結果如下圖:



由上面的結果可以得知, cnn-based 的 autoencoder 在圖片上的重建效果還是較 fcn-based 來的好很多, 但是這次的作業考慮的點並非和重建效果完全相關, 如果 model 能夠完美的重建出任何圖片, 那 anomaly 的 reconstruction error 也會很低, 就喪失了鑑別度。所以這個 autoencoder 如果要拿來做 anomaly detection 應該要對正確的類別的 reconstruction error 很低 (在左側), 沒看過的類別的 reconstruction error 很高 (在右側), 而為了看出更多的趨勢, 下圖列出最高和最低 error 的十筆:

cnn-based autoencoder :



fcn-based autoencoder



並觀察的一些 train dataset 裡的圖片，原本有的圖片種類有貓 狗 鳥 飛機 車 青蛙 鹿 馬 船，因此還是看不太出來 anomaly 是指什麼，因為此作業並沒有給任何 label，所以後來就直接丟 kaggle，結果如下一題，用 reconstruction error 來當 score 算 AUC 的結果正確率僅不到六成。

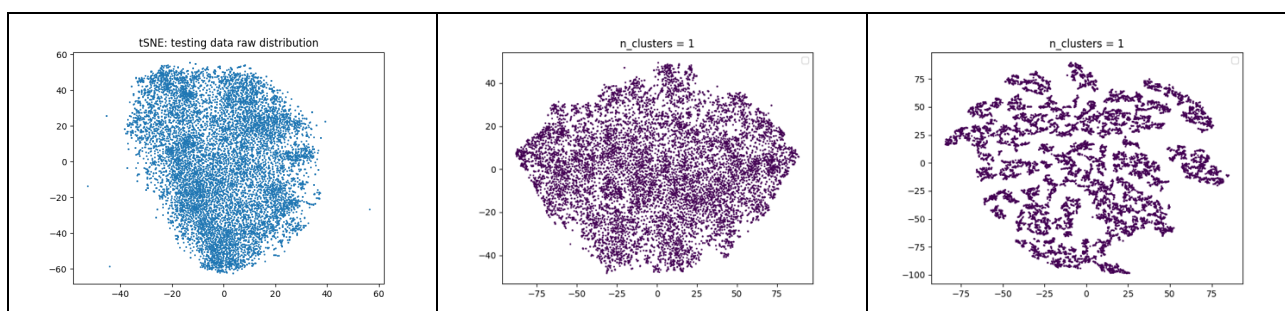
- (1%) 嘗試把 sample code 中的 KNN 與 PCA 做在 autoencoder 的 encoder output 上，並回報兩者的 auc score。

Judging method	AUC score	
	Sample FCN autoencoder	Best CNN autoencoder
Reconstruction error (rmse)	0.59659	0.57497
KNN on latent vector (dist to center)	0.61389	0.62203
PCA on latent vector (recon error)	0.53198	0.49579
Fusion (recon & knn)	0.64277	0.63424

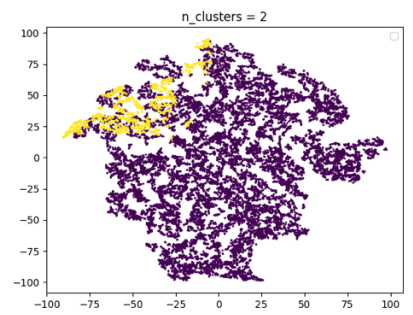
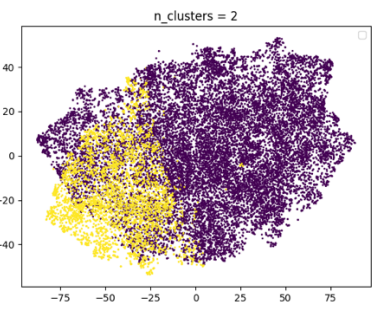
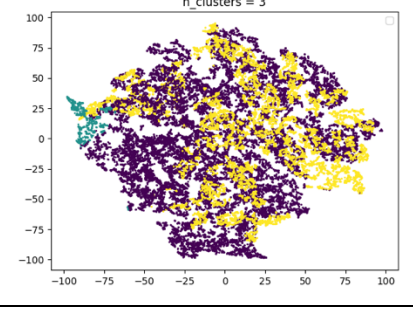
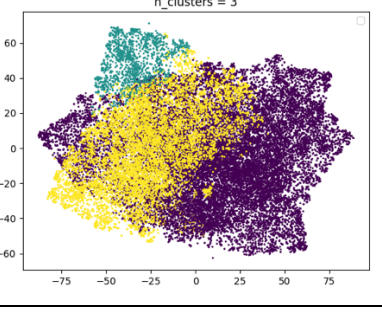
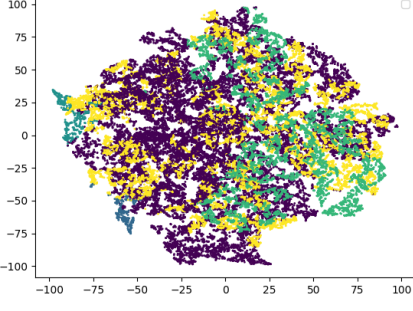
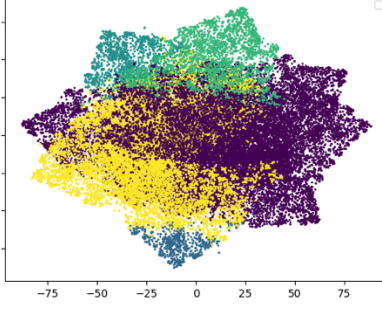
由上表可知靠 reconstruction error 做出來的 AUC 是 FCN model 比較好，這也驗證了第一題的想法，重建效果好並不等於 anomaly detection 的效果好，要必須有針對性才行。而針對 autoencoder 的 latent factor 做再處理，法一是 KNN，對 latent vector 做分群，然後再計算各個點距離群中心的距離當作評斷的 score，以兩種 model 來說都是分成三群(k=3)的效果最好。而法二則是 PCA，對 latent vector 做 PCA 降維後再升回去，一樣看 reconstruction error，這個方法在這兩個 model 上效果不太明顯，不如用原本的 autoencoder 的 output 的 reconstruction error。法三是把 auto encoder output image 的 reconstruction error 和 KNN k=3 距離中心的 distance 的 score 一起根據兩者個 mean 縮放到同一個 scale 後的平均，為最佳結果。

- (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）

tSNE projection		
Testing data (32*32*3)	Latent vector from cnn autoencoder (4*4*96)	Latent vector from fcn autoencoder (3)



由上圖可以看出結果其實非常不明顯，畢竟最終結果的 AUC 也只有六成多，代表這兩個 model 都還是不能很明確的分出兩類。而若使用第二題提到的方式，對 latent vector 用 knn 分群，然後再畫一次 tSNE 投影會如下表：

tSNE projection with knn clustering label		
	Latent vector from fcn_autoencoder	Latent vector from cnn_autoencoder
k=2		
k=3		
k=4		

由上表可知，隨著 k 值越大分群效果越差，且在 k=2 的時候亦不能明確的分出兩類，僅能使用第三題的方法，各個 latent vector 計算自己與 cluster 中心的距離當作 score，距離中心越遠的越不屬於這個 cluster，越有可能為 Anomaly。

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？

因為我們這次作業所輸出的並不是一個已經分完類的結果(0 or 1)，而是一個連續的 score，F1 Score 是以 Recall 和 precision 來計算的，所以勢必得直接先決定一個 threshold 然後根據以下公式算出值：

$F_1 \text{ score} = \frac{1}{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}} \cdot 2$	<p>F1 score is the harmonic mean of precision and sensitivity</p> $F_1 = 2 \cdot \frac{\text{PPV} \cdot \text{TPR}}{\text{PPV} + \text{TPR}} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}$
--	--

但如果用 area under ROC curve (AUC score)的話，就可以針對這個 output (一個連續的 distribution) 在不同的 threshold 底下畫出的以 TPR 和 FPR 為兩軸的 ROC curve，然後計算曲線下面積，這個做法就不受限於 threshold 取得如何，能夠直接 judge 這個 model 所預測出的分佈是否正確。比起直接決定 threshold 得到分類結果再算 F1 score，用 ROC curve 算 AUC score 能更好的判斷輸出不是直接分成兩類而是一個連續 score 的 model 的 binary classification performance.

ROC curve的y軸 True positive rate	ROC curve的x軸 False positive rate
(TPR), Recall, Hit rate	(FPR), Fall-out,
Sensitivity, probability of detection	probability of false alarm
$\frac{\sum \text{True positive}}{\sum \text{Condition positive}}$	$\frac{\sum \text{False positive}}{\sum \text{Condition negative}}$
	False alarm rate 1-specificity