

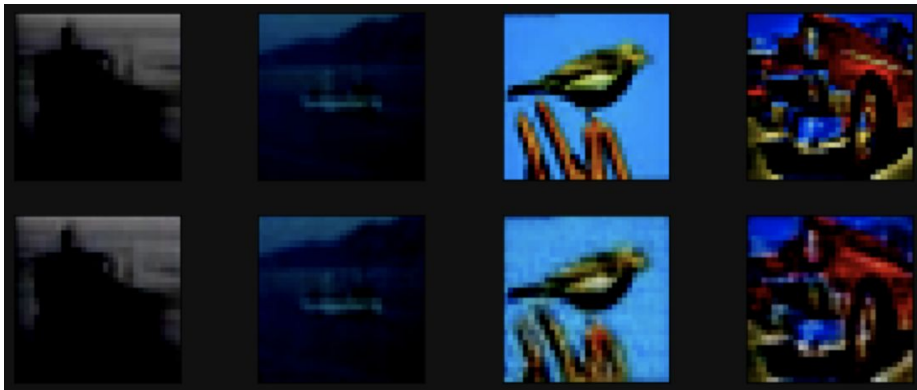
1. (2%) 任取一個baseline model (sample code裡定義的 fcn , cnn , vae) 與你在kaggle leaderboard上表現最好的單純autoencoder架構的model ( 如果表現最好的model就是sample code裡定義的model的話就再任選一個, e.g. 如果cnn最好那就再選fcn ), 對各自重建的testing data的image中選出與原圖mse最大的兩張加上最小的兩張並畫出來。( 假設有五張圖, 每張圖經由autoencoder A重建的圖片與原圖的MSE分別為 [25.4, 33.6, 15, 39, 54.8],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

baseline model :

架構是使用助教提供的 sample code 裡的 cnn , epoch 為 1000、learning rate 為 0.001 , 在 kaggle 的 public score 為 0.55225。MSE 最小為第 7392 張 , 第二小為第 3848 張 ; 最大為第 7892 張 , 第二大為第 67 張 , 以下是它們各自對應到的 MSE (依 list 中的排序)。

```
[7392, 3848, 67, 7892]  
0.8330127534047902  
0.8545794296948108  
7.693290223845474  
7.7633016047532735
```

下圖為第 7392、3848、67、7892 的原圖以及對應到的重建後的圖。



best model :

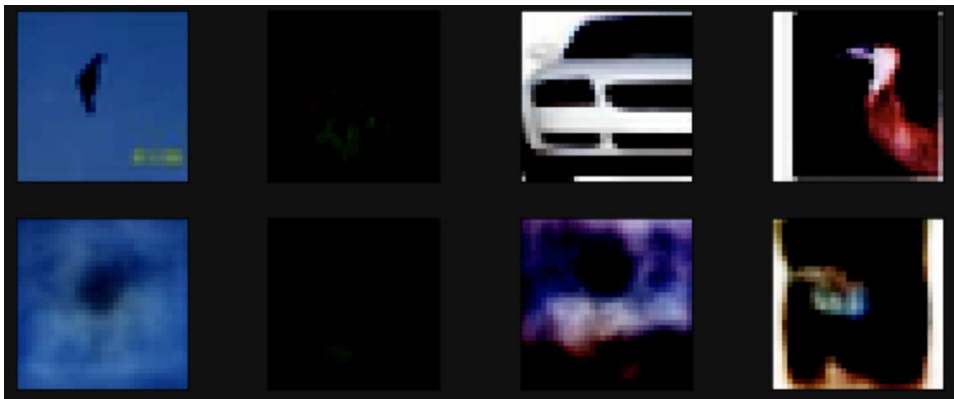
架構是使用 fcn , epoch 為 1000、learning rate 為 0.0001 , 在 kaggle 的 public score 為 0.61018。MSE 最小為第 3429 張 , 第二小為第 9318 張 ; 最大為第 4421 張 , 第二大為第 3485 張 , 以下是它們各自對應到的 MSE (依 list 中的排序)。

```
[3429, 9318, 3485, 4421]  
5.400995212386773  
5.406606144882036  
42.44849636311327  
46.891214747914844
```

fcn 架構：

```
self.encoder = nn.Sequential(
    nn.Linear(32 * 32 * 3, 1024),
    nn.ReLU(True),
    nn.Linear(1024, 512),
    nn.ReLU(True),
    nn.Linear(512, 256),
    nn.ReLU(True),
    nn.Linear(256, 128),
    nn.ReLU(True),
    nn.Linear(128, 7)
)
self.decoder = nn.Sequential(
    nn.Linear(7, 128),
    nn.ReLU(True),
    nn.Linear(128, 256),
    nn.ReLU(True),
    nn.Linear(256, 512),
    nn.ReLU(True),
    nn.Linear(512, 1024),
    nn.ReLU(True),
    nn.Linear(1024, 32 * 32 * 3),
    nn.Tanh()
)
```

下圖為第 3429、9318、3485、4421 的原圖以及對應到的重建後的圖。



2. (1%) 嘗試把 sample code 中的 KNN 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。

sample code 的 KNN 在 kaggle 的 public score 為 0.60185。

sample code 的 PCA 在 kaggle 的 public score 為 0.58829。

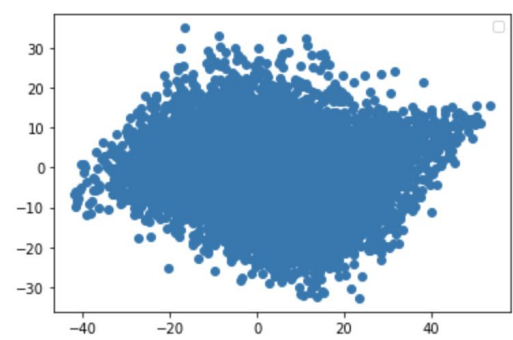
這邊的 model 是使用第一題所提到的 best model (架構和參數都已在上題說明)，在 kaggle 的 public score 為 0.61018。

model + KNN 在 kaggle 的 public score 為 0.55967，表現結果下降猜測可能原因是因為我的 model 在做 encode 的時候最後 output 只有 7 個維度，而由於降維降的過多導致在做分群時並不能有效地區分出 inlier 和 outlier。

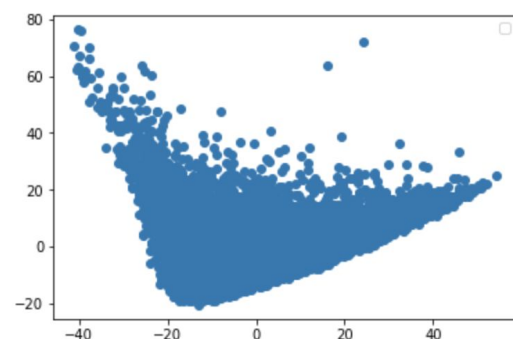
model + PCA 在 kaggle 的 public score 為 0.57730，表現結果下降猜測可能原因是因為我的 model 在做 encode 時已經降維到 7 個維度，再使用 PCA 做降維可能就有點多此一舉。

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）

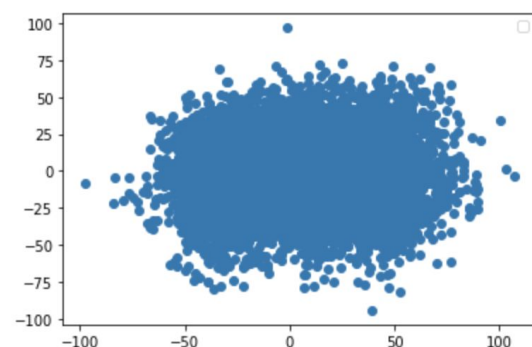
使用 PCA 投影在二維平面。



使用 baseline model + PCA 投影在二維平面。



使用 best model + PCA 投影在二維平面。



觀察結果，感覺經過降維後，outlier 有變得比較明顯，也就是離 inlier 的距離有拉得比較大，但是最後丟到 kaggle 的結果卻是單純使用 PCA 的結果最好。

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？

因為在判斷一個用來辨識 anomaly detection 的 model 的好壞通常會因不同的情境、任務而有不同的 cost table，再來使用 auc score 的話並不需要去決定 threshold，而是依序各個 testing data 給予值來做排序，因此也避免了因為給定的 threshold 值過高或過低而影響判斷結果。