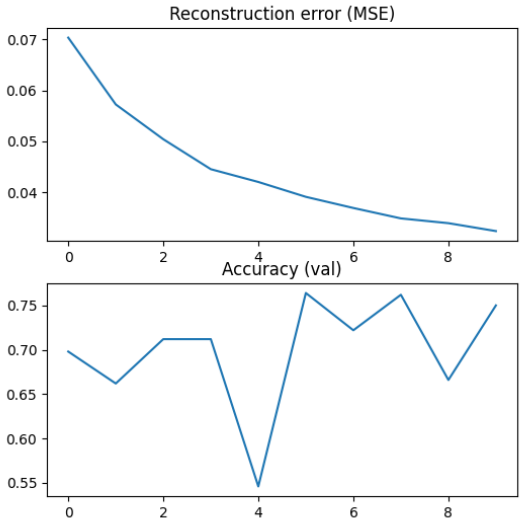
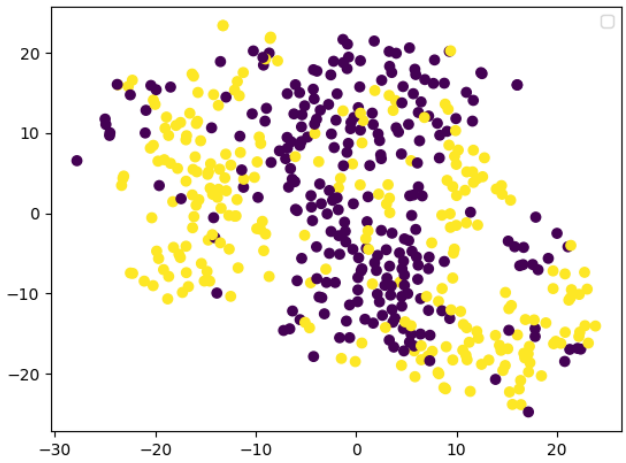
學號：B05901168　　系級：電機四　　姓名：陳冠豪

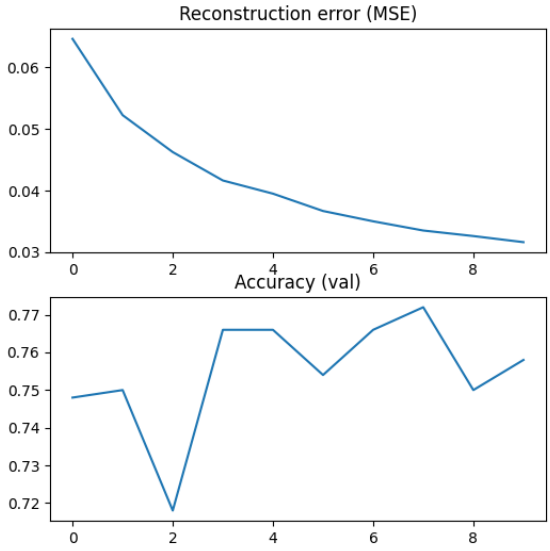
1. (3%) 請至少使用兩種方法 (autoencoder 架構、optimizer、data preprocessing、後續降維方法、clustering 算法等等) 來改進 baseline code 的 accuracy。
   a. 分別記錄改進前、後的 test accuracy 為多少。
   b. 分別使用改進前、後的方法，將 **val data** 的降維結果 (embedding) 與他們對應的 label 畫出來。
   c. 盡量詳細說明你做了哪些改進。

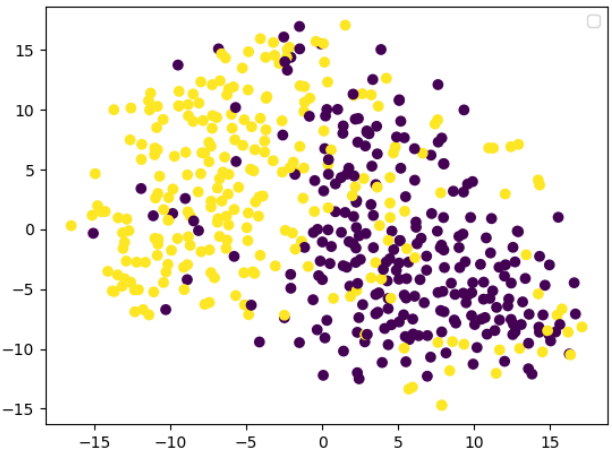
   1 : Baseline model : 使用助教提供之範例, preprocessing 部分為將圖片轉成+-1 之間的 float. 架構為 autoencoder, Encoder 部分為三層 CNN, 輸出為 4096 維的 vector, decoder 部分為三層的 ConvTranspose2d, 使用 Adam optimizer. 後續降維的方法為利用 KernelPCA 先將 encoder 的結果降成 400 維度的 vector, 接著再用 TSNE 降成 2 維的 vector, 最後用 MiniBatchMeans 來做 clustering. 下圖為訓練的結果以及 TSNE 降維後的結果,可以看到正確率大約在 70-75%左右





2: Improved model. preprocessing 部分為將圖片轉成+-1 之間的 float. 架構為 autoencoder, Encoder 部分為四層 CNN, 輸出為 2048 維的 vector, decoder 部分為四層的 ConvTranspose2d, 使用 Adam optimizer. 後續降維的方法為利用 KernelPCA 先將 encoder 的結果降成 400 維度的 vector, 接著再用 TSNE 降成 2 維的 vector, 最後用 KMeans 來做 clustering. 跟 baseline model 的差別為 encoder 多加了一層 CNN layer, 最後 encode 出來的維度是原來的一半, 讓每個維度的重要性上升.而用 KMeans 來做 cluster 效果也較 MiniBatchMeans 好. 下圖為訓練的結果以及 TSNE 降維後的結果, 可以看到正確率大約在 75-77%左右,

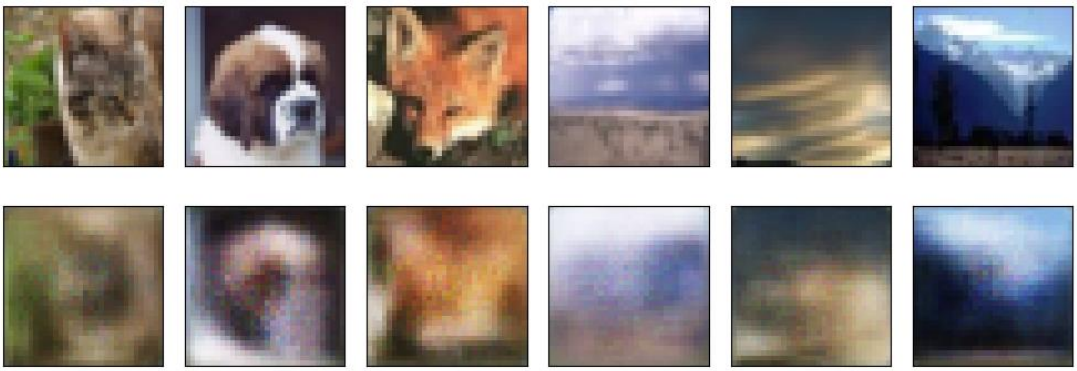




註: Directly PCA : 除了使用 Autoencoder, 我嘗試直接將 preprocess 後的圖片直接做 PCA (preprocess 方法同上), 取前 400 個重要的 component 後拿去做 KMeans 也能得到不差的效果(Kaggle 上分數為 78%). 可能是因為這次的任務只需要將圖片分成兩類,所以不需要使用 NN 也可以得到好的結果,因為不需要訓練所以就沒有附 training curve 了.

2. (1%) 使用你 test accuracy 最高的 autoencoder，從 trainX 中，取出 index 1, 2, 3, 6, 7, 9 這 6 張圖片

a. 畫出他們的原圖以及 reconstruct 之後的圖片。

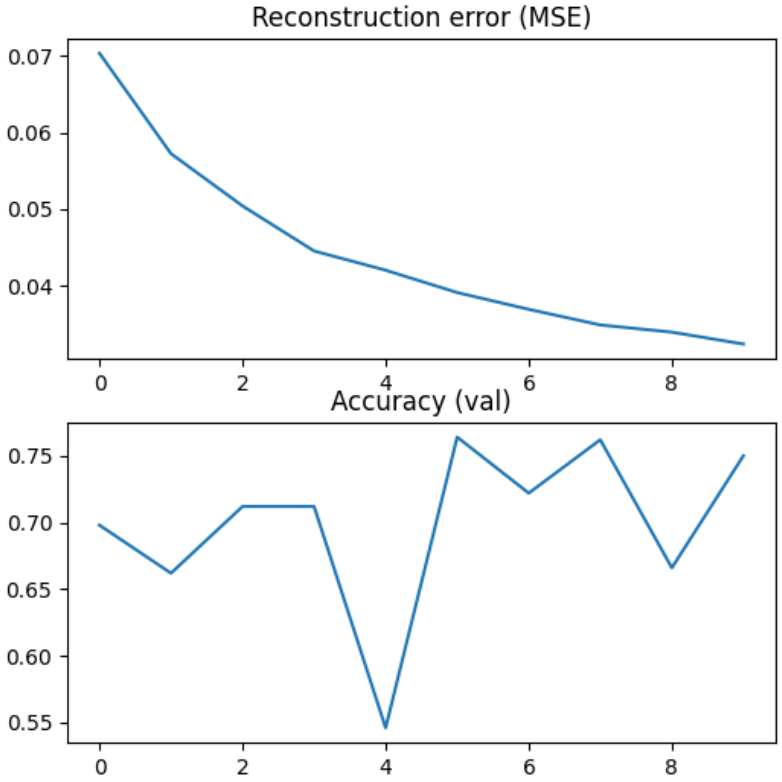


上方為原圖，下方為還原過後的圖片

3. (2%) 在 autoencoder 的訓練過程中，至少挑選 10 個 checkpoints

a. 請用 model 的 train reconstruction error (用所有的 trainX 計算 MSE) 和 **val accuracy** 對那些 checkpoints 作圖。

b. 簡單說明你觀察到的現象。



可以看到因為訓練的目標是降低 MSE loss, 因此可以看到 MSE loss 是隨著訓練次數上升隨之下降的. 但是 val accuracy 卻不一定會隨著 MSE loss 下降而跟著下降, 我覺得是因為即使 MSE loss 降低, 但是 encoder 產生的輸出不一定會是做 clustering 的理想輸出, 有可能是 decoder 的能力太強, 所以可以還原出很接近的圖片, 但是因為做 clustring 的方法沒有這麼強的能力, 所以不一定可以得到好的分類結果.