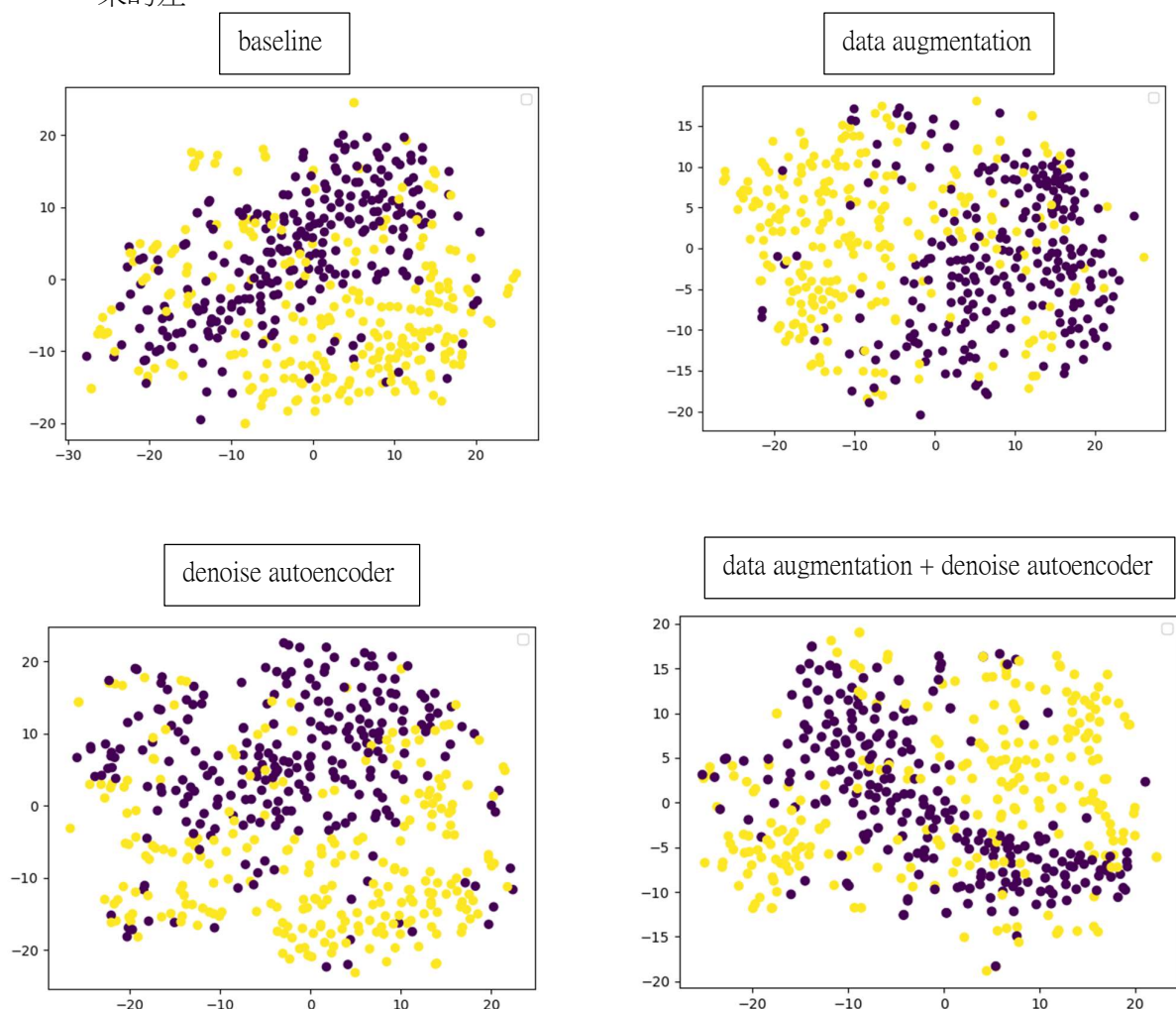


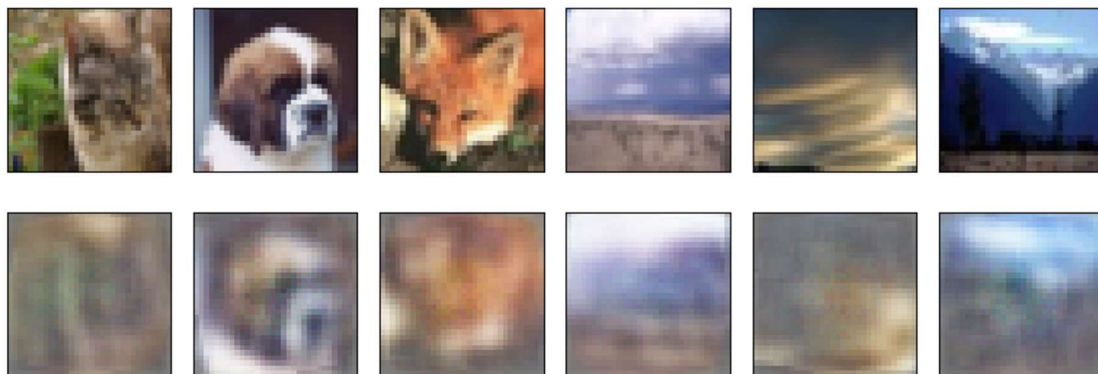
1. (3%) 請至少使用兩種方法 (autoencoder 架構、optimizer、data preprocessing、後續降維方法、clustering 算法等等) 來改進 baseline code 的 accuracy。

- 分別記錄改進前、後的 test accuracy 為多少。
- 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。
- 盡量詳細說明你做了哪些改進。

我所選擇的兩種改進方法分別為data augmentation和denoise autoencoder，改進前在validation set的準確率為0.508，而我data augmentation的實作方法為將trainX的資料量增加為三倍，並且對圖片做horizontal flip, random rotation, color jitter，改進後的準確率進步到0.77，而denoise autoencoder的實作方法為在圖片上加入一個範圍是[0, 1]的random tensor，並且將加入noise後的圖片放進model中做預測，但是計算loss時是要以原始的image為target，改進後的準確率進步到0.69，但是當我將這兩個方法同時實作時，準確率會變成0.572，比起個別使用兩種方法得到的結果來的差



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。



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。



(x軸每一單位為30個epochs)

由於我總共進行了300個epochs，因此我在繪製圖片時使用的10個checkpoints為每30個epochs記錄一次，而我觀察到的現象為隨著epoch的次數越多，基本上reconstruction error是越來越低，但是val accuracy到後面的epoch時已經沒有明顯的提升了，反而是在一直在波動，原因大概是在training結束前就已經接近收斂，因此準確率不會再有明顯的提升