學號：R07922163 系級： 資工所二 姓名：徐浩翔

1.(1%) 請使用不同的Autoencoder model，以及不同的降維方式(降到不同維度)，討論其reconstruction loss & public / private accuracy。 (因此模型需要兩種，降維方法也需要兩種，但clustrering不用兩種。)

Model 1 VAE(Variational Autoencoder)

```
VAE(
  (conv1): Sequential(
    (0): Conv2d(3, 16, kernel_size=(3, 3), stride=(2, 2))
    (1): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): LeakyReLU(negative_slope=0.01, inplace)
  )
  (conv2): Sequential(
    (0): Conv2d(16, 32, kernel_size=(3, 3), stride=(2, 2))
    (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): LeakyReLU(negative_slope=0.01, inplace)
  )
  (conv3): Sequential(
    (0): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1))
    (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): LeakyReLU(negative_slope=0.01, inplace)
  )
  (fc1): Linear(in_features=1600, out_features=512, bias=True)
  (fc2): Linear(in_features=1600, out_features=512, bias=True)
  (decoder1): Sequential(
    (0): ConvTranspose2d(32, 16, kernel_size=(4, 4), stride=(2, 2), dilation=(2, 2))
    (1): LeakyReLU(negative_slope=0.01, inplace)
  )
  (decoder2): Sequential(
    (0): ConvTranspose2d(16, 8, kernel_size=(4, 4), stride=(2, 2))
    (1): LeakyReLU(negative_slope=0.01, inplace)
  )
  (decoder3): Sequential(
    (0): ConvTranspose2d(8, 3, kernel_size=(5, 5), stride=(1, 1))
    (1): Sigmoid()
  )
)
```

Moedl 2 Autoencoder

```
Autoencoder(
  (encoder): Sequential(
    (0): Conv2d(3, 4, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))
    (1): Conv2d(4, 8, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))
    (2): Conv2d(8, 16, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))
  )
  (decoder): Sequential(
    (0): ConvTranspose2d(16, 8, kernel_size=(2, 2), stride=(2, 2))
    (1): ConvTranspose2d(8, 3, kernel_size=(2, 2), stride=(2, 2))
    (2): Tanh()
  )
)
```

使用的降維方法有PCA與T-SEN

	PCA	T-SNE
AE Reconstruction erro: 0.00004	Public: 0.77873 Private: 0.79037	Public: 0.63920 Private: 0.63666

VAE Reconstruction error: 0.000034	Public: 0.74148 Private:0.74285	Publice:0.78148 Private:0.77571

其中reconstruction error這麼小的原因是一開始input都已經先經過處理了所以圖片中每個pixel都除以255那可以看出來VAE做出來的reconstruction error比AE小一點

而PCA與 T-SNE的表現來看在VAE中用tsne會高一點但是在AE卻遠不及PCA我覺得他們是各有優缺 PCA 我都將latent降成32維，但是tsne一定要降到2~3維所以圖片的資訊可能會流失太多，所以我做得最好的model是先用pca 降成32 再使用 tsne，對同一個model(train好的)

他可以在private/public正確率提升至0.83492/0.84000 比上表的0.74 0.78高出好幾個百分點

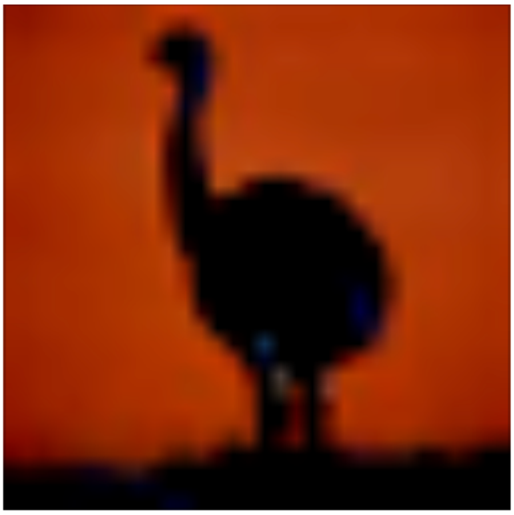
最後的clustering我使用的皆為Kmeans

2.(1%) 從dataset選出2張圖，並貼上原圖以及經過autoencoder後reconstruct的圖片。

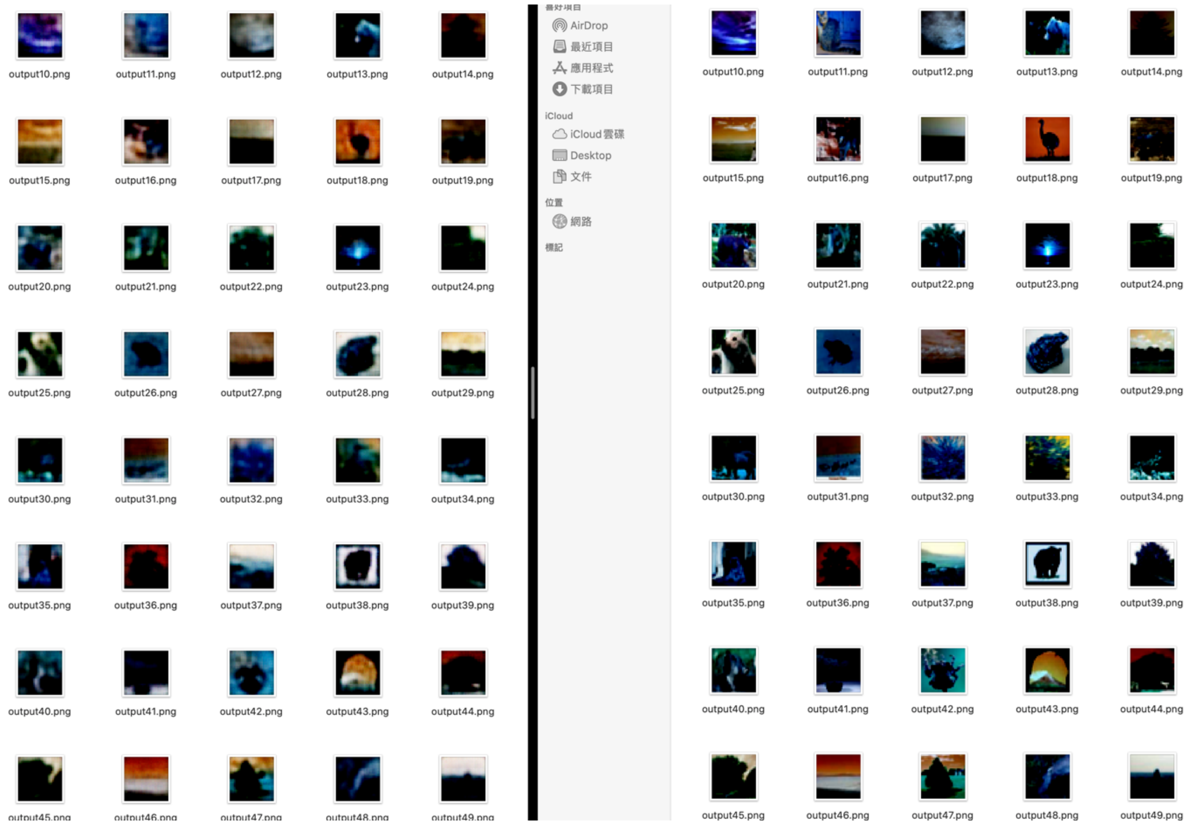
原圖(狗)

reconstruct

原圖(鴕鳥？)

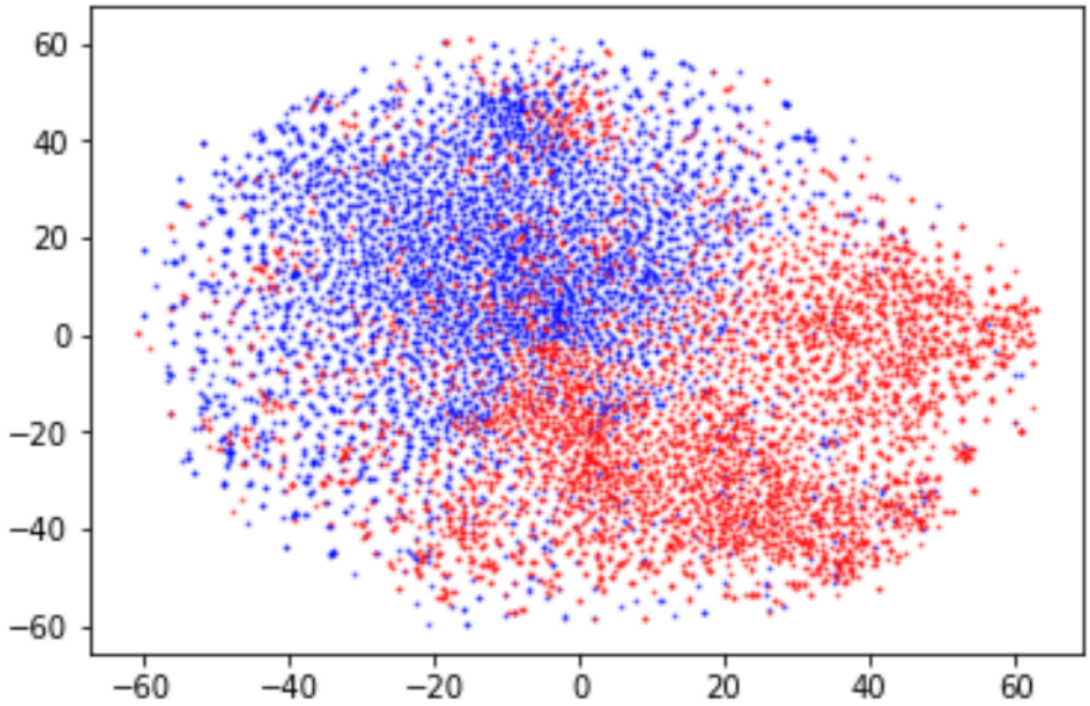


reconstruct



左邊為reconstruct右邊為原圖基本上都有還原出圖片的輪廓只是稍微有些模糊

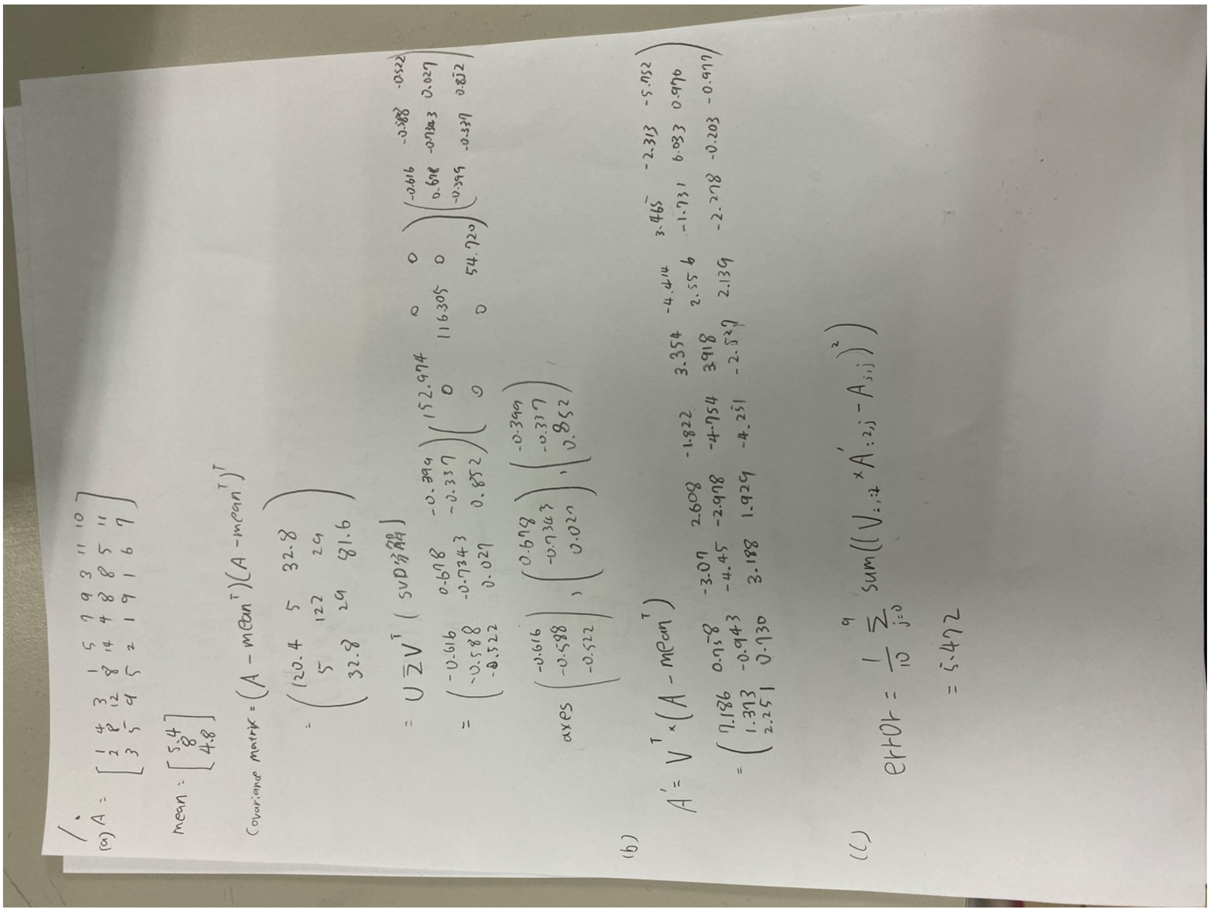
3.(1%) 在之後我們會給你dataset的label。請在二維平面上視覺化label的分佈。



Latent經過t-sne降成二維後大致可看出左上為一塊右下為一塊但沒有分得很好在kaggle public/private上經過kmeans分群後正確率為0.83 / 0.84

4.(3%)Refer to math problem

https://drive.google.com/file/d/1e_IDAV2yv0YEhIuVWpDdaH4Pzz5s1p2P/view?fbclid=IwAR0tO9NRxK9JZeUDNdawNuSbGTvqI7niuMX3Kkk9arauC8O6p6iJc7oMz84



1.

(a) $A = \begin{bmatrix} 1 & 4 & 3 & 1 & 5 & 7 & 9 & 3 & 11 & 10 \\ 2 & 8 & 12 & 8 & 14 & 4 & 8 & 8 & 5 & 11 \\ 3 & 5 & 9 & 5 & 2 & 1 & 9 & 1 & 6 & 7 \end{bmatrix}$

mean $= \begin{bmatrix} 5.4 \\ 8 \\ 4.8 \end{bmatrix}$

Covariance matrix $= (A - mean^T)(A - mean^T)^T$

$= \begin{pmatrix} 120.4 & 5 & 32.8 \\ 5 & 122 & 29 \\ 32.8 & 29 & 81.6 \end{pmatrix}$

$= U \Sigma V^T$ (SVD分解)

$= \begin{pmatrix} -0.616 & 0.678 & -0.399 \\ -0.588 & -0.7343 & -0.337 \\ -0.522 & 0.027 & 0.852 \end{pmatrix} \begin{pmatrix} 152.974 & 0 & 0 \\ 0 & 116.305 & 0 \\ 0 & 0 & 54.720 \end{pmatrix} \begin{pmatrix} -0.616 & -0.588 & -0.522 \\ 0.678 & -0.7343 & 0.027 \\ -0.399 & -0.337 & 0.852 \end{pmatrix}$

axes $\begin{pmatrix} -0.616 \\ -0.588 \\ -0.522 \end{pmatrix}, \begin{pmatrix} 0.678 \\ -0.7343 \\ 0.027 \end{pmatrix}, \begin{pmatrix} -0.399 \\ -0.337 \\ 0.852 \end{pmatrix}$

(b)

$A' = V^T \times (A - mean^T)$

$= \begin{pmatrix} 7.186 & 0.758 & -3.07 & 2.608 & -1.822 & 3.354 & -4.414 & 3.465 & -2.313 & -5.752 \\ 1.373 & -0.943 & -4.45 & -2.978 & -4.754 & 3.918 & 2.556 & -1.731 & 6.033 & 0.976 \\ 2.251 & 0.730 & 3.188 & 1.929 & -4.251 & -2.527 & 2.139 & -2.278 & -0.203 & -0.977 \end{pmatrix}$

(c)

$\text{error} = \frac{1}{10} \sum_{j=0}^{9} \text{sum}\left((V_{:,:2} \times A'_{:2,j} - A_{:,j})^2 \right)$

$= 5.472$

b. by SVD 分解

$$\Sigma = U \Sigma' V^T$$

$$= (v_0\ v_1\ \cdots\ v_{m-2}\ v_{m-1}) \begin{bmatrix} \lambda_0 & & & 0 \\ & \lambda_1 & & \vdots \\ & & \ddots & \\ 0 & & \lambda_{m-2} & \lambda_{m-1} \end{bmatrix} \begin{pmatrix} v_0 \\ \vdots \\ v_{m-2} \\ v_{m-1} \end{pmatrix}$$

where U is orthonormal, and $\lambda_i \geq 0\ \forall i$

$$= (v_0\ v_1\ \cdots\ v_{m-2}\ v_{m-1}) \begin{bmatrix} \sqrt{\lambda_0} & & 0 \\ & \sqrt{\lambda_1} & \\ 0 & & \sqrt{\lambda_{m-1}} \end{bmatrix} \begin{bmatrix} \sqrt{\lambda_0} & & 0 \\ & \sqrt{\lambda_1} & \\ 0 & & \sqrt{\lambda_{m-1}} \end{bmatrix} \begin{bmatrix} v_0 \\ \vdots \\ v_{m-1} \end{bmatrix}$$

$$= (\sqrt{\lambda_0} v_0, \sqrt{\lambda_1} v_1 \cdots \sqrt{\lambda_{m-1}} v_{m-1}) \begin{bmatrix} \sqrt{\lambda_0} v_0 \\ \vdots \\ \sqrt{\lambda_{m-1}} v_{m-1} \end{bmatrix} = AA^T$$

$\lambda_i = i$th column of A $m=0$

c. $\phi^T\phi = I_k \Rightarrow \phi$ 是一組 m-dim orthonormal basis 中的 k 個

Let $\Sigma = \frac{1}{N} XX^T \Rightarrow \phi^T \Sigma \phi = \frac{1}{N} \phi^T X X^T \phi = \frac{1}{N}(X^T\phi)^2$

minimize $\mathrm{Trace}(\Phi^T \Sigma \Phi) = \min \mathrm{Trace}((X^T\phi)^2) = \min \|X^T\phi\|_F^2$ 因為 ϕ 是一組標準正交基底,我們可以把 $X^T\phi$ 看作是投影到 ϕ span 出來的 hyperplane,因此我們是要最小化投影後長度的平方,而這個長度(的平方)相當於是投影到這組標準正交基底 ϕ 之外的 m-k 的基底所 span 出的 hyperplane 上的 Reconstruction (Square) Error 最小化投影到 m-k 維的 Reconstruction Error 的 opt 便是 PCA 後 k 個 basis (eigenvalue 最小的那 k 個 eigenvector)

2.

(a) proof symetric

$$(AA^T)^T = (A^T)^T A^T = AA^T$$

$$(A^TA)^T = A^T(A^T)^T = A^TA$$

proof positive semi-definite

$$\forall x \in R^n$$
$$\|Ax\|^2 \geq 0$$
$$(Ax)^T Ax \geq 0$$
$$x^T A^T A x \geq 0$$

A^TA is positive semi-definite

$$\forall x \in R^m$$
$$\|A^Tx\|^2 \geq 0$$
$$(A^Tx)^T A^T x \geq 0$$
$$x^T A A^T x \geq 0$$

AA^T is positive semi-definite

proof share the same non-zero eigen values

suppose λ_i, V_i are pair of eigen value and eigenvector of AA^T, $\lambda_i > 0$

$$AA^T v_i = \lambda_i V_i$$
$$A^TAA^T v_i = A^T \lambda_i v_i$$
$$A^TAA^T v_i = A^T \lambda_i v_i$$
$$A^TA(A^Tv_i) = \lambda_i (A^Tv_i)$$

that is λ_i is also a non-zero eigen value of A^TA

3.

Multiclass AdaBoost

let $y_k = \begin{cases} 1, & \text{if } \hat{y}_i = k \\ -\frac{1}{k-1}, & \text{if } \hat{y}_i \neq k \end{cases}$ $y_{\hat{y}} = [y_1 \cdots y_k]$, $g_t = [g_t^1 \cdots g_t^k]$

$= L(g_t) = \sum_{i=1}^n \exp(-y_{\hat{y}_i} g_t)$

1. Initialize $g_0(x) = 0$
2. For $t = 1$ to T:

(a) Minimize $\sum_{i=1}^n \exp(y_{\hat{y}_i}^T (g_t(x_i) + \alpha f(x_i)))$

$= \sum_{i=1}^n w_i \exp(\alpha y_{\hat{y}_i}^T (\alpha y_{\hat{y}_i}^T f(x_i))$, where $w_i = \exp(y_{\hat{y}_i}^T g_t(x_i))$

Note that there is a one-to-one corresponding $T(x)$ for $f(x)$ in the following way: $T(x) = k$ if $f_k(x) = 1$

$T_t(x) = \arg\min \sum_{i=1}^n w_i \mathbb{I}(\hat{y}_i \neq T(x_i))$

Proof: $\sum_{i=1}^n w_i \exp(\alpha y_{\hat{y}_i}^T f(x_i))$

$= \sum_{\hat{y}_i = T(x_i)} w_i \exp\left(-\frac{\alpha}{k-1}\right) + \sum_{\hat{y}_i \neq T(x_i)} w_i \exp\left(\frac{\alpha}{(k-1)^2}\right)$

$= \exp\left(-\frac{\alpha}{k-1}\right) \sum_i w_i + \left(\exp\left(\frac{\alpha}{(k-1)^2}\right) - \exp\left(-\frac{\alpha}{k-1}\right)\right) \sum_i w_i \mathbb{I}(\hat{y}_i \neq T(x_i))$

Since only the last sum depends on $T(x)$, we have:

$T_t(x) = \operatorname{argmin} \sum_{i=1}^n w_i \mathbb{I}(\hat{y}_i \neq T(x_i))$

Now, plug the result into the formula, we get

$\alpha_t = (k-1)^2 \left(\log \frac{1 - err_t}{err_t} + \log(k-1)\right)$

where $err_t = \sum_{i=1}^n w \mathbb{I}(\hat{y}_i \neq T(x_i))$

(b) Update $g_t(x) = g_{t-1}(x) + \alpha_t f_t(x)$