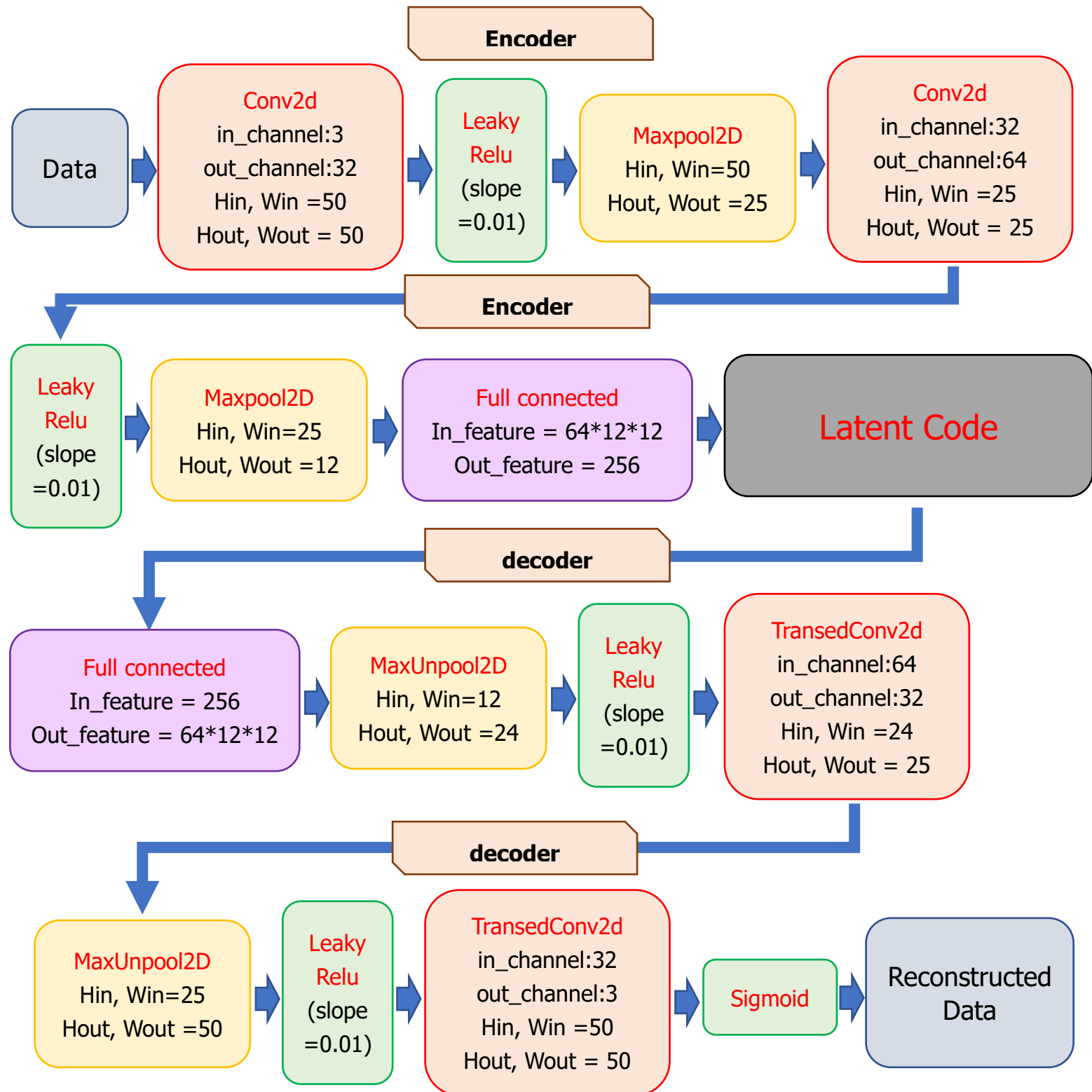
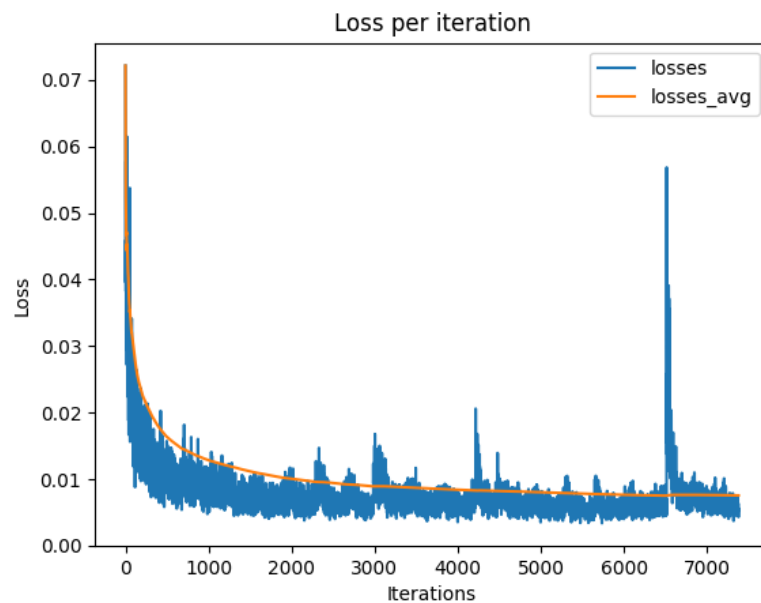


Problem 1:

(1)AE model architecture



AE model training Loss(MSELoss):



AE model discussion:

1.condition:

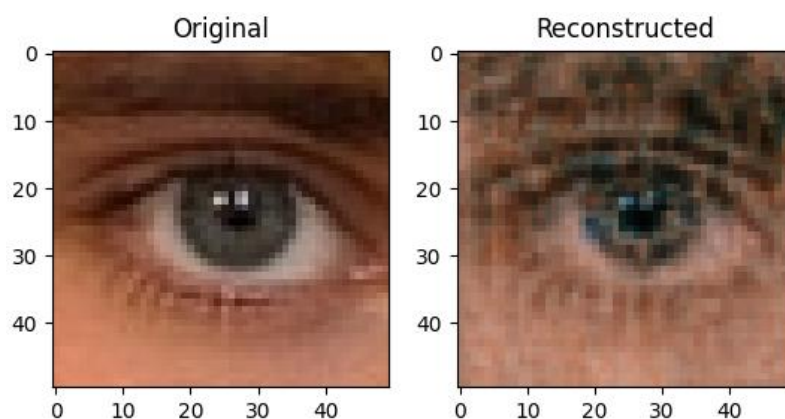
Epoch =20, batch_size=4, learning_rate=0.005

2.result:

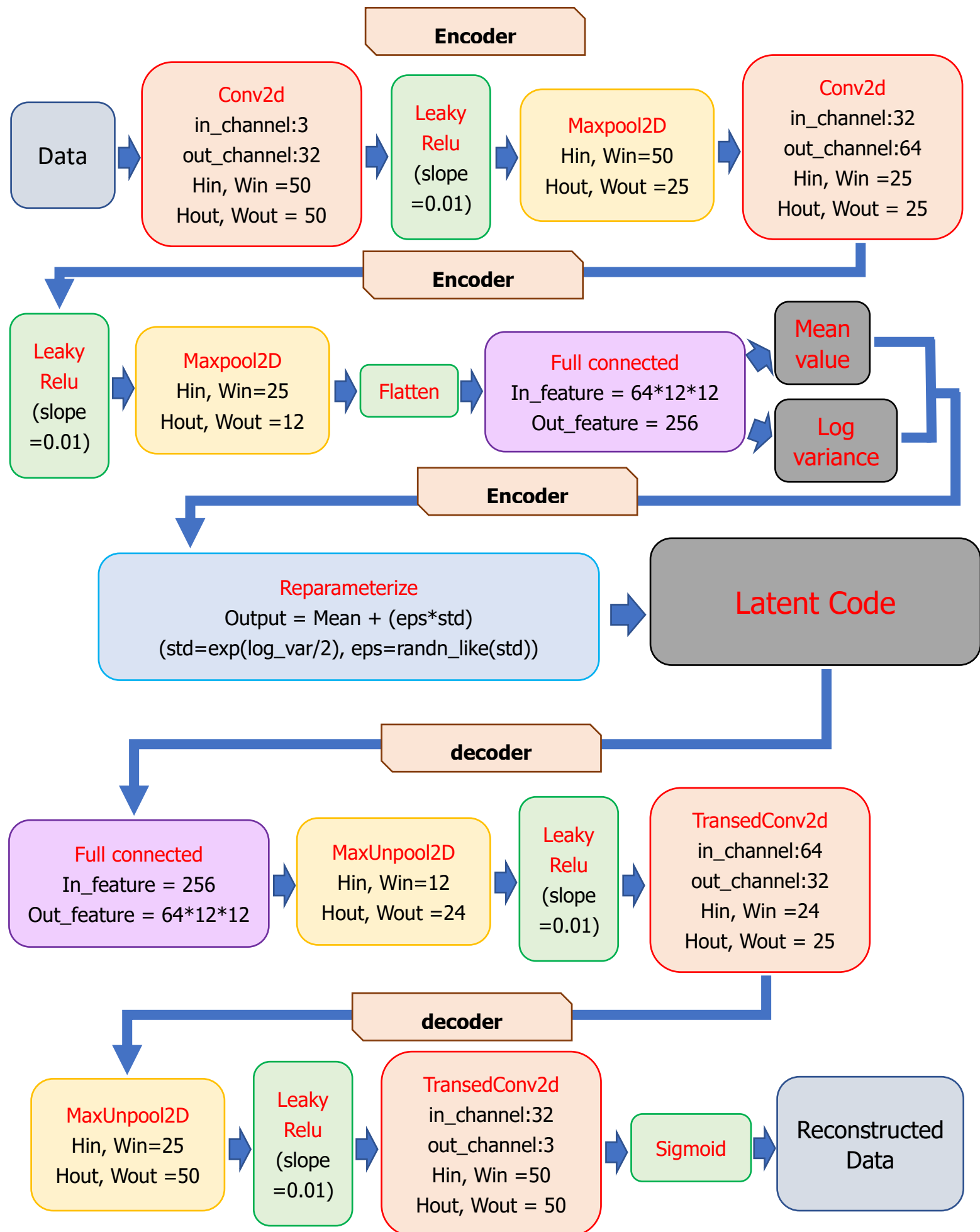
PSNR=21.9617, SSIM=0.7218 , reconstruct Loss: 0.0075

3.disscussion:

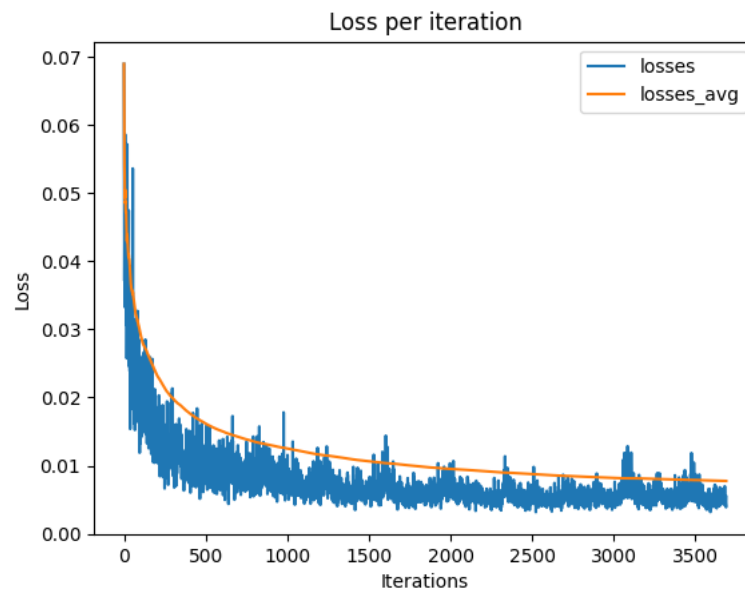
會使用 conv2D 以及 MaxPool2D 是因為針對圖片的學習中，這兩種 hidden layers 比起 full connected，能夠更好的去 fit 住圖片中更有意義的特徵，藉由保留圖片中每個特徵與特徵之間的空間及距離，因此在連續的 hidden layer 計算中，能夠更精準地傳達圖片特徵至下一層連接層。但由於 conv2D 消耗的運算資源相對比較大，所以在 encoder 最後使用 full connected 來編成最後的 latent code，這樣做的目的除了可以減少運算資源外，主要是測試 latent code 能夠壓縮至何種程度，卻仍然能夠有效地重建出原始圖片。經由實驗過後，可以發現將原來為每張 3*50*50 的 RGB 圖片，編成 1*256 的 latent code 之後，重建出來的眼睛與原圖片的眼睛雖然較為模糊，但是顏色上以及輪廓上的表現都能夠人眼能夠看出的範圍下表現出來。



(1)VAE model architecture



VAE model training Loss(MSELoss):



VAE model discussion:

1.condition:

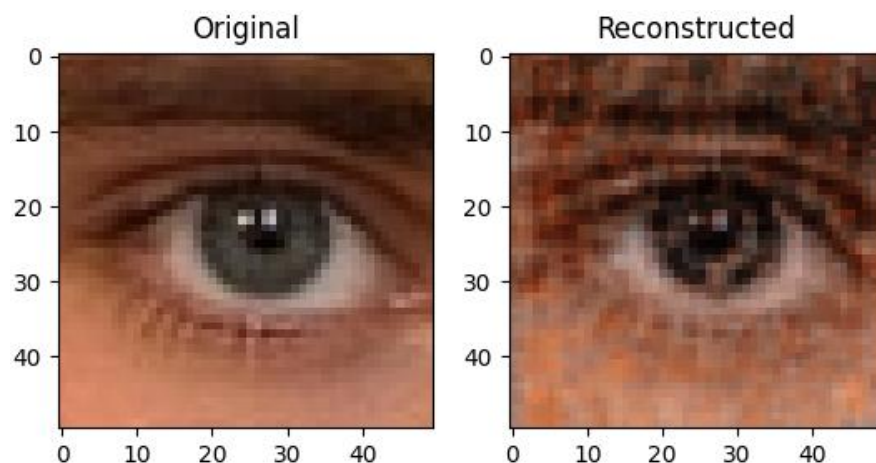
Epoch =10, batch_size=4, learning_rate=0.005

2.result:

PSNR=24.09, SSIM=0.7297 , reconstruct Loss: 0.0073

3.disscussion:

在 VAE 的架構中，多了 Reparameterize 的過程，函式包括兩個參數:1. mean of encoder's latent space encoding, 2. the encoder's latent space，在這裡使用的是 simple variational autoencoder，由於標準的 AE 將圖片的 feature 壓縮成 latent code 之後，在還原的過程中，只能還原出那些已經訓練過的特徵，無法產生出新的圖片，相反的 VAE 壓縮成的 latent vector space 具有連續性，能夠幫助生產出新圖片。由數據可以看出，VAE 的表現確實比標準 AE 表現來的優異，從 reconstructed 的圖片也可以看出，兩者雖然都能夠劃出原圖片的眼睛輪廓，但是舉眉毛以及睫毛的部分來說，可以發現 VAE 相較於 AE，更能重塑出圖片的顏色以及細節。



Problem 2: construct the new sample with gaussian noise(VAE)

(1)

依題意選擇要求的圖片，並且重新 group 出一個新的 choose_loader，方便處理後續資料。

```
break
#////////////////////////////////choose img to reconstruct////////////////////////////////
choose_id = [1,2,3,4,5,226,227,228,229,230,841,842,843,844,845,1471,1472,1473,1474,1475]
indices = torch.tensor([i-1 for i in choose_id])
choose_tensor = torch.index_select(data, 0, indices)
choose_label = torch.index_select(label, 0, indices)
choose_tensor_gpu, choose_label_gpu = choose_tensor.to(device), choose_label.to(device)
choose_dataset = TensorDataset(choose_tensor_gpu, choose_label_gpu)
choose_loader = DataLoader(choose_dataset, batch_size=4)
```

(2)

batch_size=4, 共 5 個 batches, 每個 batch generate 5 次，所以共有
4*5*5=100 張圖片，最後將一張一張圖片存成 ndarray 類別，並存成.npy
file。

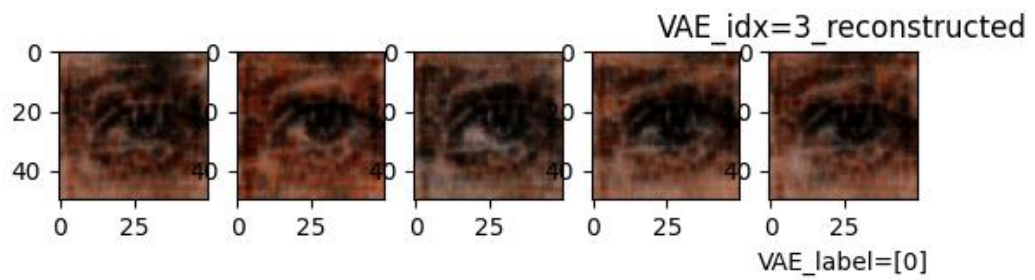
```
choose_reco = []
choose_label = []
def gnerate_with_noise(img_batch, model, label):
    reco_result = []
    gen_label_result = []
    for _ in range(5):
        each_result = model.add_gaussian(img_batch)
        reco_result.append(each_result)
        gen_label_result.append(label)
    return reco_result, gen_label_result

for img_batch, label in choose_loader:
    reco_result, gen_label_result = gnerate_with_noise(img_batch, model, label)
    for i in range(len(reco_result)):
        choose_reco.append(reco_result[i])
        choose_label.append(gen_label_result[i])

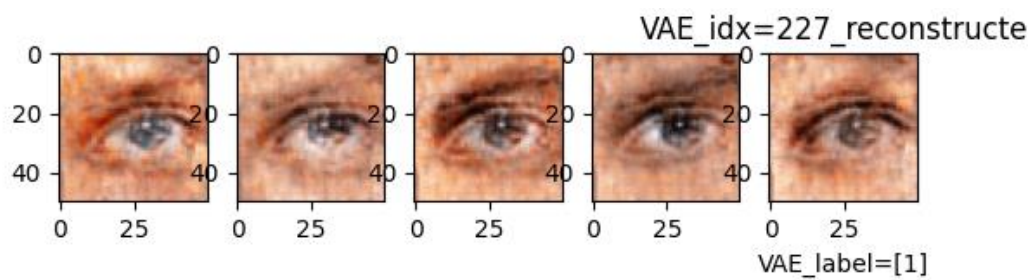
gen_data = choose_reco[0]
gen_label = choose_label[0]
for i in range(1, len(choose_reco)):
    merge_data = choose_reco[i]
    merge_label = choose_label[i]
    gen_data = torch.cat((gen_data, merge_data), 0)
    gen_label = torch.cat((gen_label, merge_label), 0)
gen_data = gen_data.cpu().detach().numpy()
gen_label = gen_label.cpu().detach().numpy()
np.save(r"C:\Users\User\Desktop\碩一上修課資料\深度學習_林嘉文\HW2\VAE_gen_data.npy", gen_data)
np.save(r"C:\Users\User\Desktop\碩一上修課資料\深度學習_林嘉文\HW2\VAE_gen_label.npy", gen_label)
```

(3)constructed image with different gaussian noise

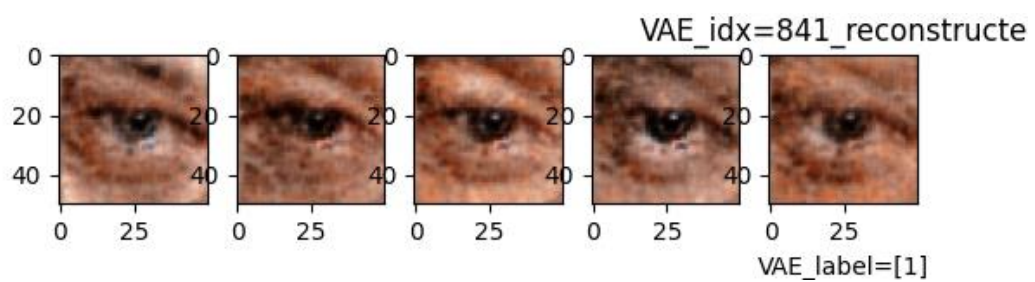
第 3 張



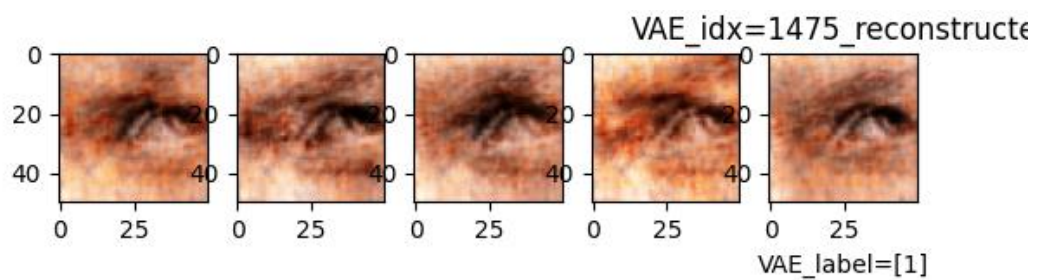
第 227 張



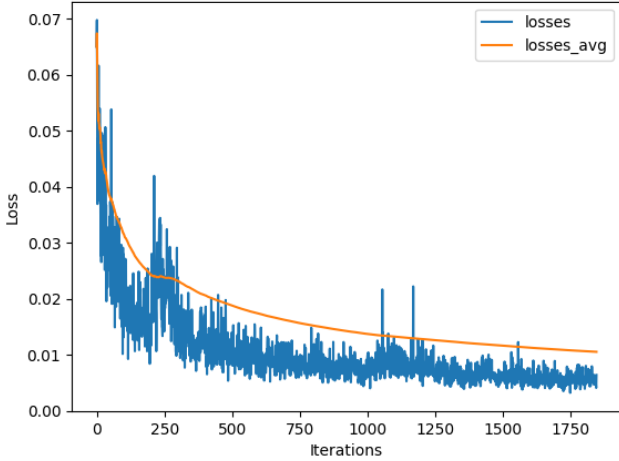
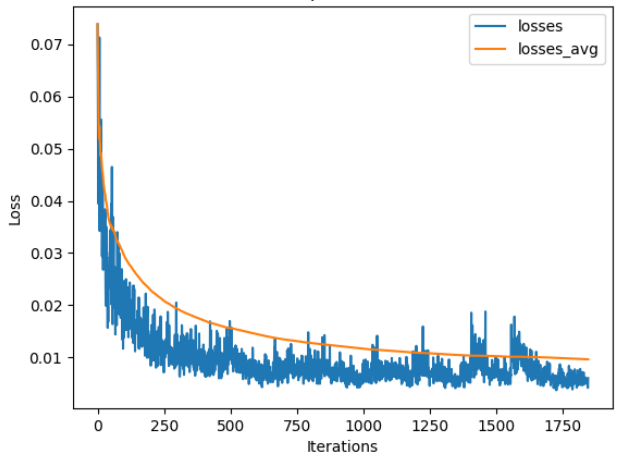
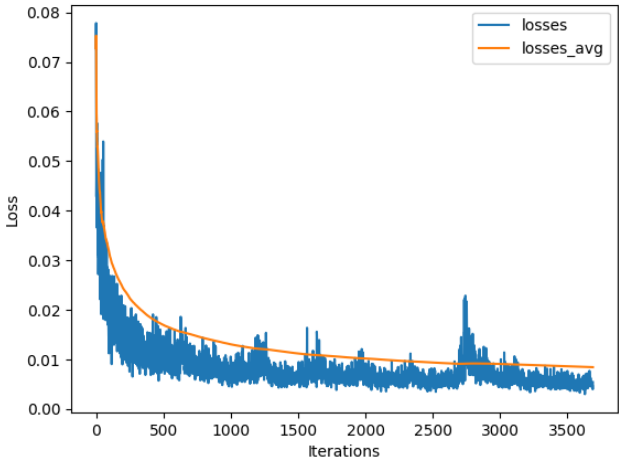
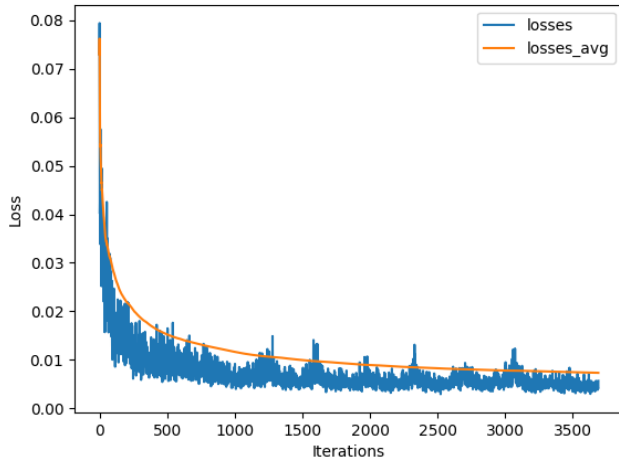
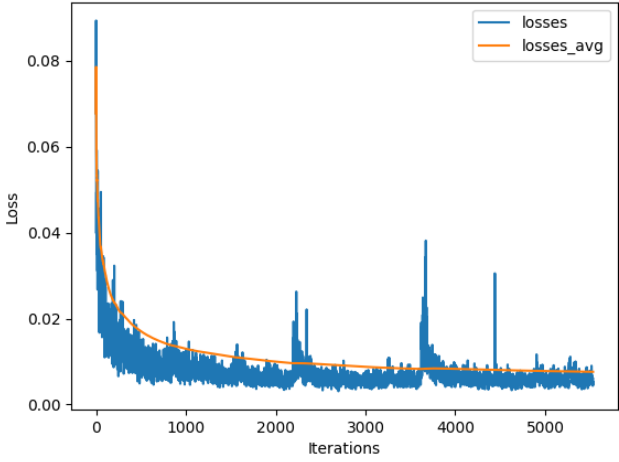
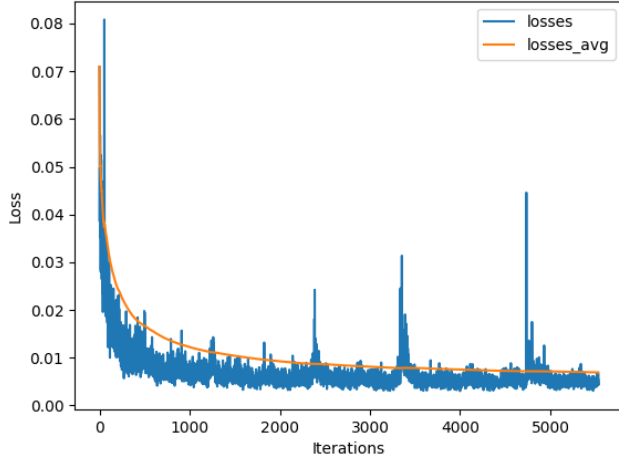
第 841 張



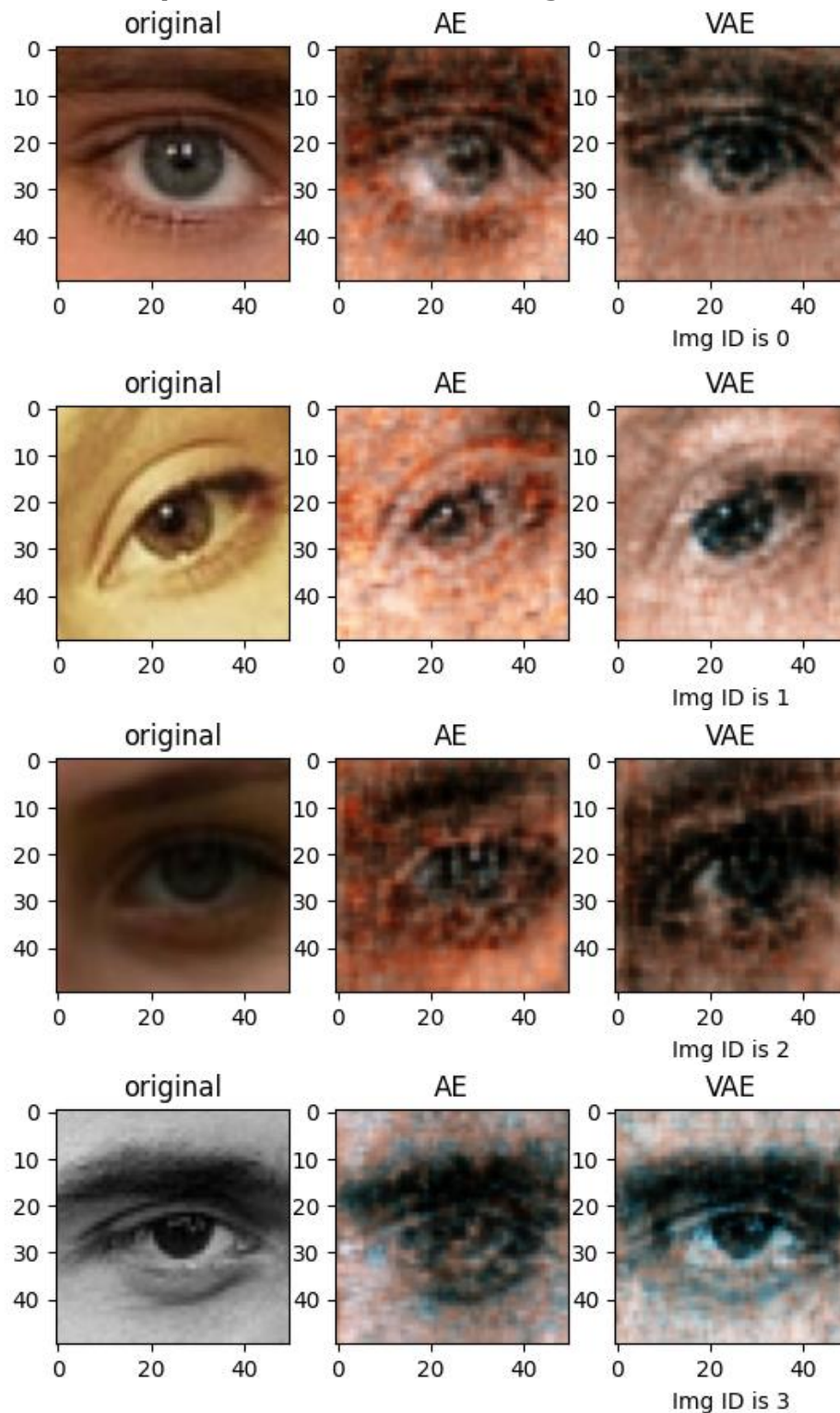
第 1475 張



Problem 3: hyperparameters adjusting (batch_size are all 4)

Epochs	AE	VAE
5	<p>Avg_loss = 0.106 , PSNR=22.99</p> <p>Loss per iteration</p> 	<p>Avg_loss = 0.0096 , PSNR=23.01</p> <p>Loss per iteration</p> 
10	<p>Avg_loss = 0.0085, PSNR=22.84</p> <p>Loss per iteration</p> 	<p>Avg_loss = 0.0073 , PSNR=24.09</p> <p>Loss per iteration</p> 
15	<p>Avg_loss = 0.0076 , PSNR=22.89</p> <p>Loss per iteration</p> 	<p>Avg_loss = 0.0069, PSNR=23.53</p> <p>Loss per iteration</p> 

Problem 4: compare AE & VAE with same gaussian noise

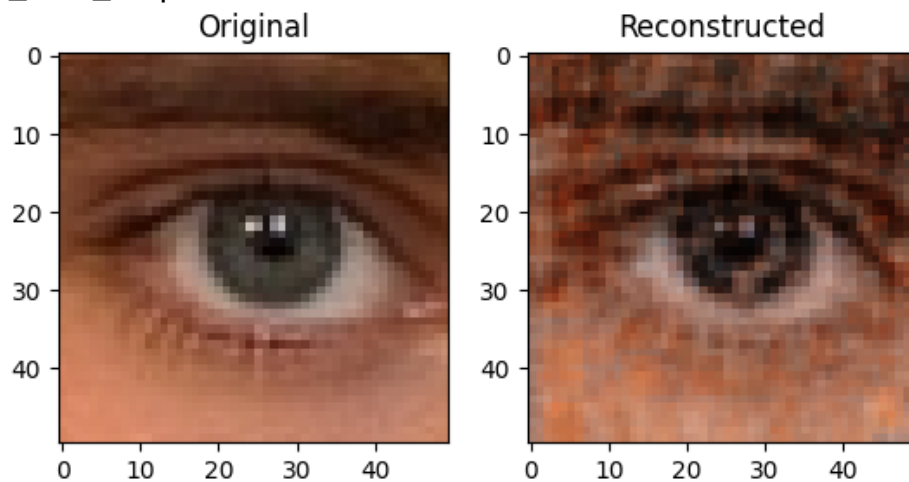


從以上的 4 個 cases 來比較，可以發現 AE 跟 VAE 最大的差異在於眼睛輪廓以及顏色上的表現，VAE 能將原圖的眼睛形狀特徵表達得比較好，AE 有時會過於表達某些方面的特徵(眼白、陰影)，反而讓生成的圖片與原圖片出現較大的差異，進而使損失提升。

Problem 5-1:

因為降維的關係，AE&VAE 通常使用比較小的 latent 維度，大部分輸入圖片的資訊以及特徵很難通過 bottleneck 的地方，除此之外，model 在每個 batch 迭代時，會試著去將低 loss，因此 AE&VAE 只能獲得較模糊的圖片。但藉由藉由提高 latent 的維度，可以得到非常好的重建效果，如下比較圖所示，但也因此失去降維的目的。

Latent_code_shape = 1 * 256:



Latent_code_shape = 64 * 48 * 48:

