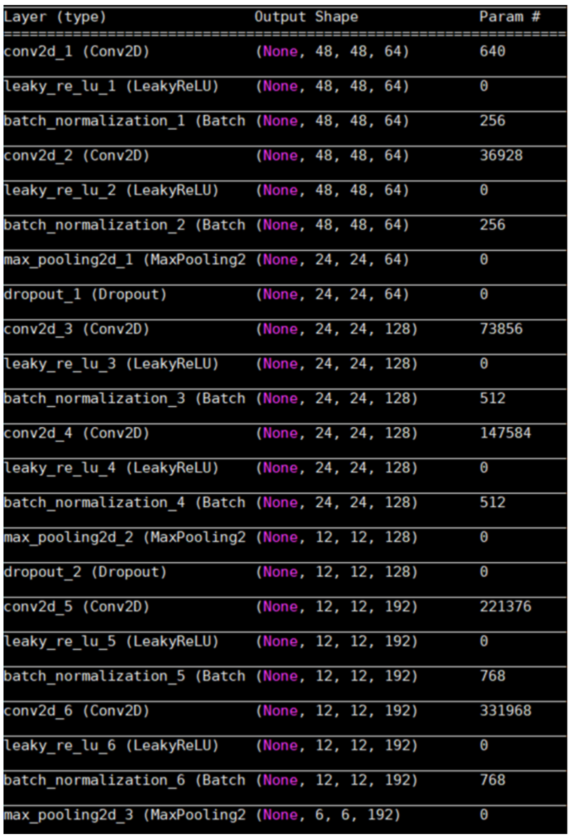
Homework3 Report Template

Professor Pei-Yuan Wu
EE5184 - Machine Learning

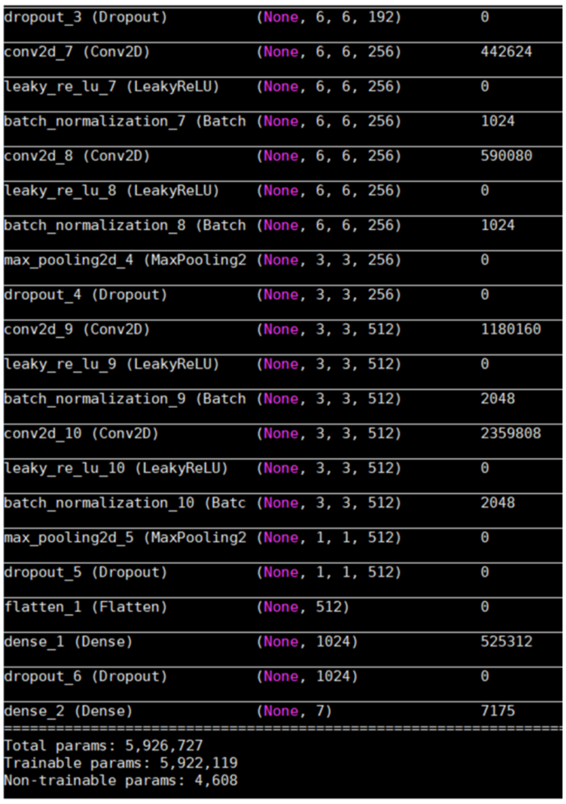
姓名：馬咏治
學號：R07943123

Note:1~3 題建議不要超過三頁

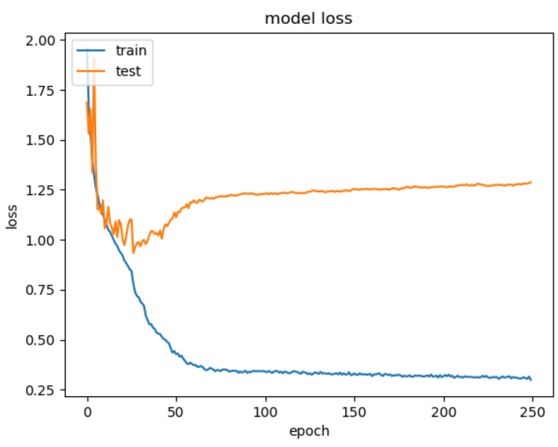
1. (1%) 請說明你實作的 CNN model，其模型架構、訓練過程和準確率為何？

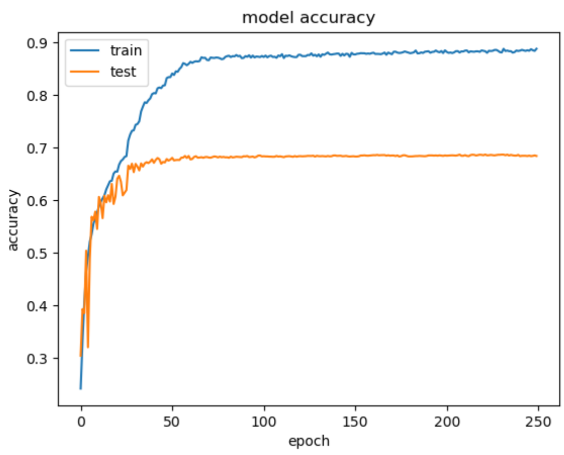


```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 48, 48, 64)        640
leaky_re_lu_1 (LeakyReLU)    (None, 48, 48, 64)        0
batch_normalization_1 (Batch (None, 48, 48, 64)        256
conv2d_2 (Conv2D)            (None, 48, 48, 64)        36928
leaky_re_lu_2 (LeakyReLU)    (None, 48, 48, 64)        0
batch_normalization_2 (Batch (None, 48, 48, 64)        256
max_pooling2d_1 (MaxPooling2 (None, 24, 24, 64)        0
dropout_1 (Dropout)          (None, 24, 24, 64)        0
conv2d_3 (Conv2D)            (None, 24, 24, 128)       73856
leaky_re_lu_3 (LeakyReLU)    (None, 24, 24, 128)       0
batch_normalization_3 (Batch (None, 24, 24, 128)       512
conv2d_4 (Conv2D)            (None, 24, 24, 128)       147584
leaky_re_lu_4 (LeakyReLU)    (None, 24, 24, 128)       0
batch_normalization_4 (Batch (None, 24, 24, 128)       512
max_pooling2d_2 (MaxPooling2 (None, 12, 12, 128)       0
dropout_2 (Dropout)          (None, 12, 12, 128)       0
conv2d_5 (Conv2D)            (None, 12, 12, 192)       221376
leaky_re_lu_5 (LeakyReLU)    (None, 12, 12, 192)       0
batch_normalization_5 (Batch (None, 12, 12, 192)       768
conv2d_6 (Conv2D)            (None, 12, 12, 192)       331968
leaky_re_lu_6 (LeakyReLU)    (None, 12, 12, 192)       0
batch_normalization_6 (Batch (None, 12, 12, 192)       768
max_pooling2d_3 (MaxPooling2 (None, 6, 6, 192)         0
```



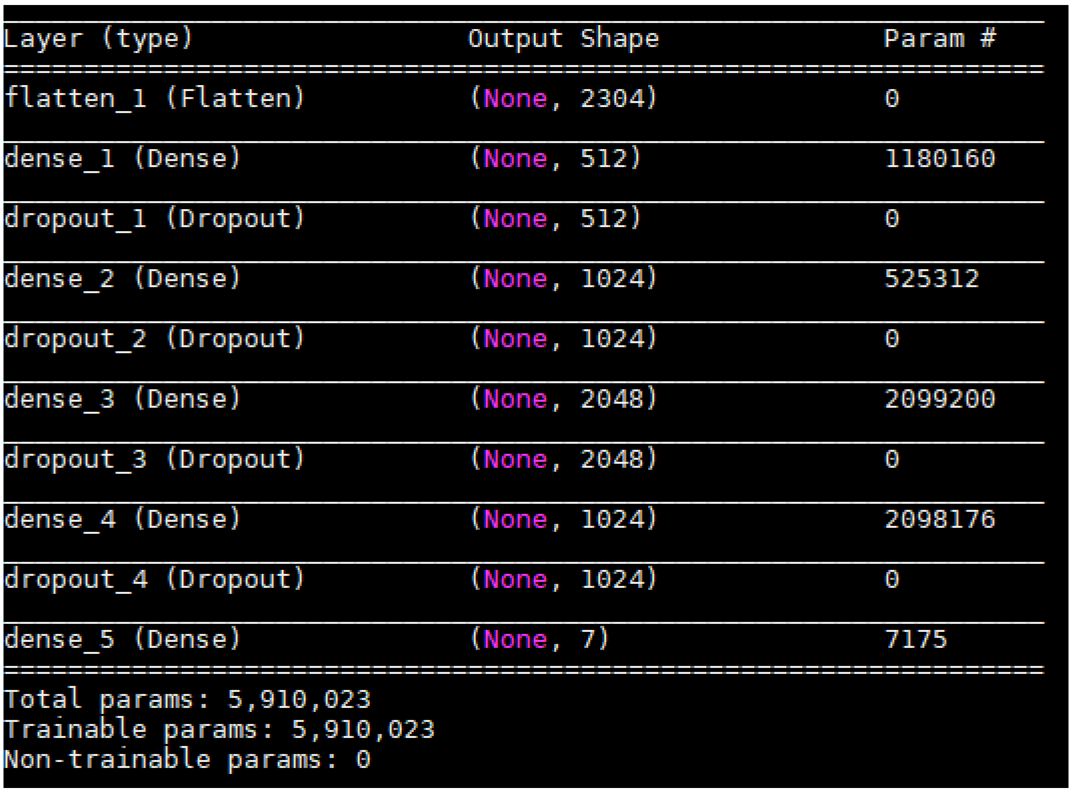
```
dropout_3 (Dropout)          (None, 6, 6, 192)         0
conv2d_7 (Conv2D)            (None, 6, 6, 256)         442624
leaky_re_lu_7 (LeakyReLU)    (None, 6, 6, 256)         0
batch_normalization_7 (Batch (None, 6, 6, 256)         1024
conv2d_8 (Conv2D)            (None, 6, 6, 256)         590080
leaky_re_lu_8 (LeakyReLU)    (None, 6, 6, 256)         0
batch_normalization_8 (Batch (None, 6, 6, 256)         1024
max_pooling2d_4 (MaxPooling2 (None, 3, 3, 256)         0
dropout_4 (Dropout)          (None, 3, 3, 256)         0
conv2d_9 (Conv2D)            (None, 3, 3, 512)         1180160
leaky_re_lu_9 (LeakyReLU)    (None, 3, 3, 512)         0
batch_normalization_9 (Batch (None, 3, 3, 512)         2048
conv2d_10 (Conv2D)           (None, 3, 3, 512)         2359808
leaky_re_lu_10 (LeakyReLU)   (None, 3, 3, 512)         0
batch_normalization_10 (Batc (None, 3, 3, 512)         2048
max_pooling2d_5 (MaxPooling2 (None, 1, 1, 512)         0
dropout_5 (Dropout)          (None, 1, 1, 512)         0
flatten_1 (Flatten)          (None, 512)               0
dense_1 (Dense)              (None, 1024)              525312
dropout_6 (Dropout)          (None, 1024)              0
dense_2 (Dense)              (None, 7)                 7175
=================================================================
Total params: 5,926,727
Trainable params: 5,922,119
Non-trainable params: 4,608
```





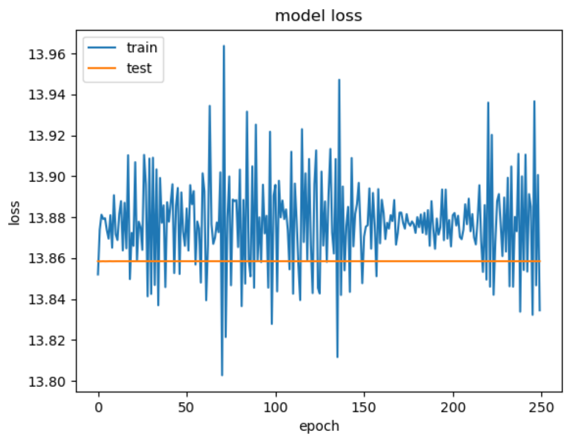
訓練過程:
首先我把 80%的 data 來做 training，20%的 data 來做 validation，batch size 調 128， epoch 調 100. 在訓練過程中會使用 image generator 對 data 進行旋轉和 shift，以圖更好的效果。在過程中，如果 validation accuracy 在 5 個 epoch 後依然沒有上升的跡象，我就會調低 learning rate。最後會依照 validation accuracy 去選擇最佳的 model。

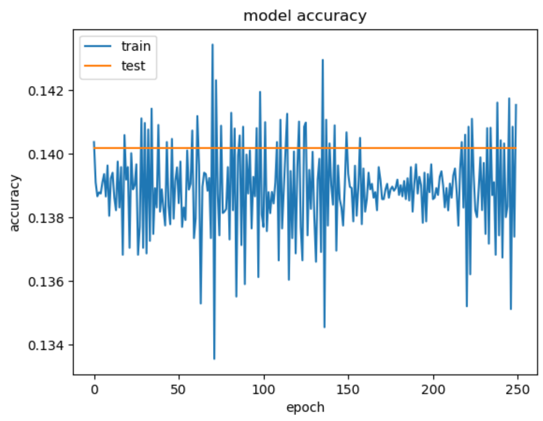
2. (1%) 承上題，請用與上述 CNN 接近的參數量，實做簡單的 DNN model，其模型架構、訓練過程和準確率為何？試與上題結果做比較，並說明你觀察到了什麼？



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
flatten_1 (Flatten)          (None, 2304)              0
_________________________________________________________________
dense_1 (Dense)              (None, 512)               1180160
_________________________________________________________________
dropout_1 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_2 (Dense)              (None, 1024)              525312
_________________________________________________________________
dropout_2 (Dropout)          (None, 1024)              0
_________________________________________________________________
dense_3 (Dense)              (None, 2048)              2099200
_________________________________________________________________
dropout_3 (Dropout)          (None, 2048)              0
_________________________________________________________________
dense_4 (Dense)              (None, 1024)              2098176
_________________________________________________________________
dropout_4 (Dropout)          (None, 1024)              0
_________________________________________________________________
dense_5 (Dense)              (None, 7)                 7175
=================================================================
Total params: 5,910,023
Trainable params: 5,910,023
Non-trainable params: 0
```

訓練過程:　　　　　　　　　　　　準確率: 14.1%

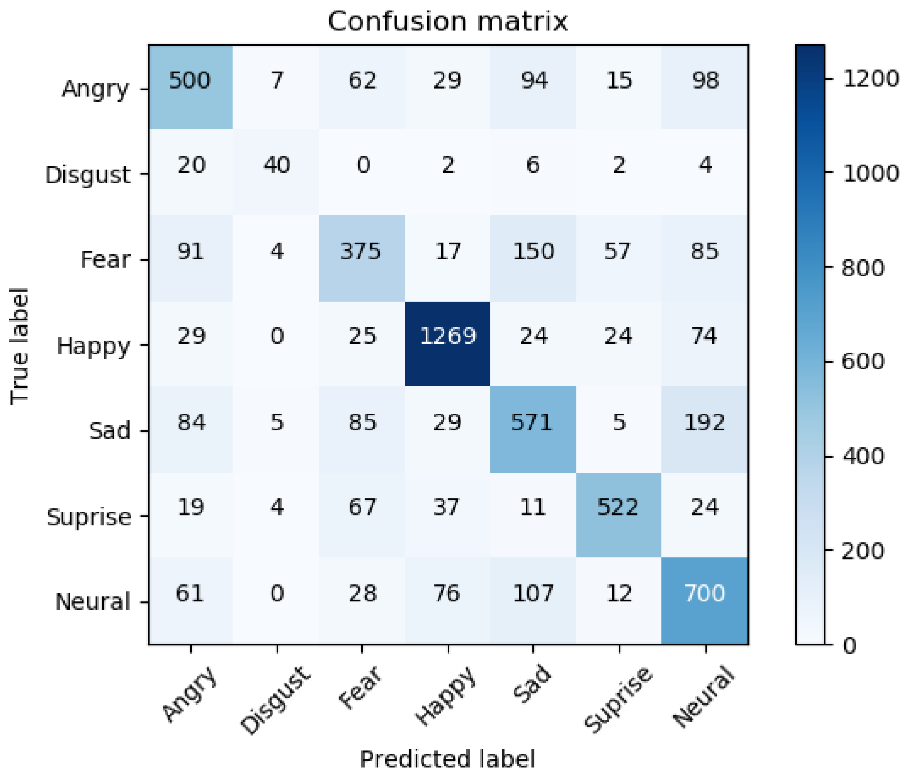




使用 dnn 來 training 可以發現到 loss 和 accuracy 都一直在震盪，testing accuracy 一直保持在 14.1%，代表 nn 裡面的 weight 基本上已經穩定不會有大變

動了。我想可能 dnn 在把圖片拉平後，讓圖片失去了一些局部的特性，是的這個被拉平的向量不具有太多的意義。

3. (1%) 觀察答錯的圖片中，哪些 class 彼此間容易用混？並說明你觀察到了什麼？[繪出 confusion matrix 分析]

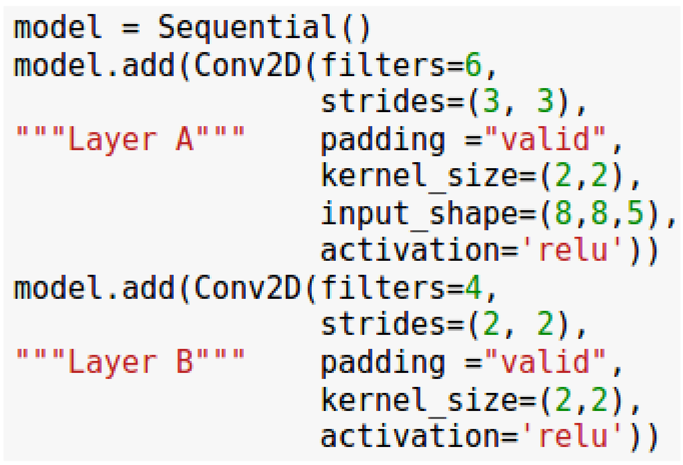


根據我的觀察，Fear 和 Sad，Disgust 和 Angry 兩個 class 蠻容易混在一起的。在 779 個 Fear 照片中有 150 誤認為是 Sad，佔了 19%，而在 74 個 Disgust 照片中，有 20 張誤認為是 Angry。
會產生他們弄混的原因，我想是因為一般人很難單純以臉部標表情來判斷這個人的 emotion status，更何況是 Fear，Sad 和 Disgust，Angry 這幾個 class 的照片相似度這麼高，一個原因是 label 可能會標錯，或者是他們這幾個 class 之間並不好分辨，儘管是讓人類來分辨的話

----------------Handwritten question----------------

4. (1.5%,each 0.5%)CNN time/space complexity:

For a. b. Given a CNN model as



```
model = Sequential()
model.add(Conv2D(filters=6,
                 strides=(3, 3),
"""Layer A"""    padding ="valid",
                 kernel_size=(2,2),
                 input_shape=(8,8,5),
                 activation='relu'))
model.add(Conv2D(filters=4,
                 strides=(2, 2),
"""Layer B"""    padding ="valid",
                 kernel_size=(2,2),
                 activation='relu'))
```

And for the c. given the parameter as:
kernel size = (k,k);
channel size = c;
filter size = f;
input shape = (n,n);
padding = 1;
strides = (s,s);

a. How many parameters are there in each layer(Hint: you may consider whether the number of parameter is related with)
Layer A: 126
Layer B: 100

b. How many multiplications/additions are needed for a forward pass(each layer).
Multiplication = kernel size*kernel size*input channel *output size*filter no
Addition = kernel size*kernel size*input channel *output size*filter no
Layer A:
    Multiplication :2*2*5*9*6 = 1080
    Addition :(2*2*5-1)*9*6 = 1026
Layer B:
    Multiplication : 2*2*6*1*4 = 96
    Addition : (2*2*6-1)*1*4 = 92

c. What is the time complexity of convolutional neural networks?(note: you must use big-O upper bound, and there are l layer, you can use □ $_{□}$,□ $_{□-1}$as lth and l-1th layer)

$$C_1: k_1^2 * \mathrm{c} * f_1 * \left(\frac{n}{s_1}\right)^2$$

$$C_2: k_2^2 * f_1 f_2 * \left(\frac{n}{s_1 s_2}\right)^2$$

$$\vdots$$

$$C_l: k_l^2 * f_{l-1} f_l * \left(\frac{n}{s_1 s_2 \dots s_l}\right)^2$$

$$\text{Complexity} = C_1 + C_2 + \cdots + C_l = O\left(\sum_{i=1}^{l} C_l\right)$$

$$= O(\sum_{i=1}^{l} k_i^2 f_{i-1} f_i (\frac{n}{\prod_{k=1}^{i} s_k})^2)$$

5. (1.5%,each 0.5%)PCA practice:Problem statement: Given 10 samples in 3D space.(1,2,3),(4,8,5),(3,12,9),(1,8,5),(5,14,2),(7,4,1),(9,8,9),(3,8,1),(11,5,6),(10,11,7)

a. (1) What are the principal axes?

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

$$\Sigma = \frac{1}{10}\sum_{i=1}^{10}(x_i - \mu)(x_i - \mu)^T$$

$$= \begin{bmatrix} 12.04 & 0.5 & 3.28 \\ 0.5 & 12.2 & 2.9 \\ 3.28 & 2.9 & 8.16 \end{bmatrix}$$

After Calculated Covariance Matrix, find the three eigenvector, which also is the principal axes

$$v_1 = \begin{bmatrix} -0.616595 \\ -0.588816 \\ -0.522596 \end{bmatrix} \quad v_2 = \begin{bmatrix} 0.678179 \\ -0.73439 \\ 0.0272856 \end{bmatrix} \quad v_3 = \begin{bmatrix} -0.399855 \\ -0.337589 \\ 0.852144 \end{bmatrix}$$

$$\lambda_1 = 15.2974 \,, \lambda_2 = 11.6305 \,, \lambda_3 = 5.47203$$

b. (2) Compute the principal components for each sample.

After Mapping to the principal axes (v_1, v_2, v_3) , the 10 sample are:

$$a_1 = \begin{bmatrix}-3.36\\-0.709\\1.481\end{bmatrix}, a_2 = \begin{bmatrix}-9.79\\-3.026\\-0.039\end{bmatrix}, a_3 = \begin{bmatrix}-13.62\\-6.53\\2.42\end{bmatrix}, a_4 = \begin{bmatrix}-7.94\\-5.06\\1.16\end{bmatrix},$$

$$a_5 = \begin{bmatrix}-12.37\\-6.836\\-5.02\end{bmatrix}, a_6 = \begin{bmatrix}-7.19\\1.837\\-3.30\end{bmatrix}, a_7 = \begin{bmatrix}-14.96\\0.474\\1.370\end{bmatrix}$$

$$a_8 = \begin{bmatrix}-7.083\\-3.813\\-3.048\end{bmatrix}, a_9 = \begin{bmatrix}-12.86\\3.952\\-0.973\end{bmatrix}, a_{10} = \begin{bmatrix}-16.301\\-1.106\\-1.747\end{bmatrix}$$

c. (3) Reconstruction error if reduced to 2D. (Calculate the L2-norm)

We reduce these 10 sample to v_1, v_2 since they have highest eigenvalue

After reconstruction,

$$a_1 = \begin{bmatrix}1.591\\2.499\\1.737\end{bmatrix}, a_2 = \begin{bmatrix}3.984\\7.987\\5.034\end{bmatrix}, a_3 = \begin{bmatrix}3.970\\12.815\\6.940\end{bmatrix}, a_4 = \begin{bmatrix}1.464\\8.391\\4.011\end{bmatrix},$$

$$a_5 = \begin{bmatrix}2.991\\12.304\\6.278\end{bmatrix}, a_6 = \begin{bmatrix}5.680\\2.885\\3.808\end{bmatrix}, a_7 = \begin{bmatrix}9.546\\8.461\\7.831\end{bmatrix}$$

$$a_8 = \begin{bmatrix}1.781\\6.971\\3.598\end{bmatrix}, a_9 = \begin{bmatrix}10.610\\4.670\\6.828\end{bmatrix}, a_{10} = \begin{bmatrix}9.301\\10.411\\8.489\end{bmatrix}$$

Next, we calculate the L2 Norm

$$\mathrm{L2}(a_1) = |\Delta a_1|^2 = 2.1945$$
$$\mathrm{L2}(a_2) = |\Delta a_2|^2 = 0.0016$$
$$\mathrm{L2}(a_3) = |\Delta a_3|^2 = 5.8498$$
$$\mathrm{L2}(a_4) = |\Delta a_4|^2 = 1.3459$$
$$\mathrm{L2}(a_5) = |\Delta a_5|^2 = 25.2129$$
$$\mathrm{L2}(a_6) = |\Delta a_6|^2 = 10.8715$$
$$\mathrm{L2}(a_7) = |\Delta a_7|^2 = 1.8766$$
$$\mathrm{L2}(a_8) = |\Delta a_8|^2 = 9.2911$$
$$\mathrm{L2}(a_9) = |\Delta a_9|^2 = 0.9477$$
$$\mathrm{L2}(a_{10}) = |\Delta a_{10}|^2 = 3.0521$$

Total Reconstruction Error ≈ 60.644