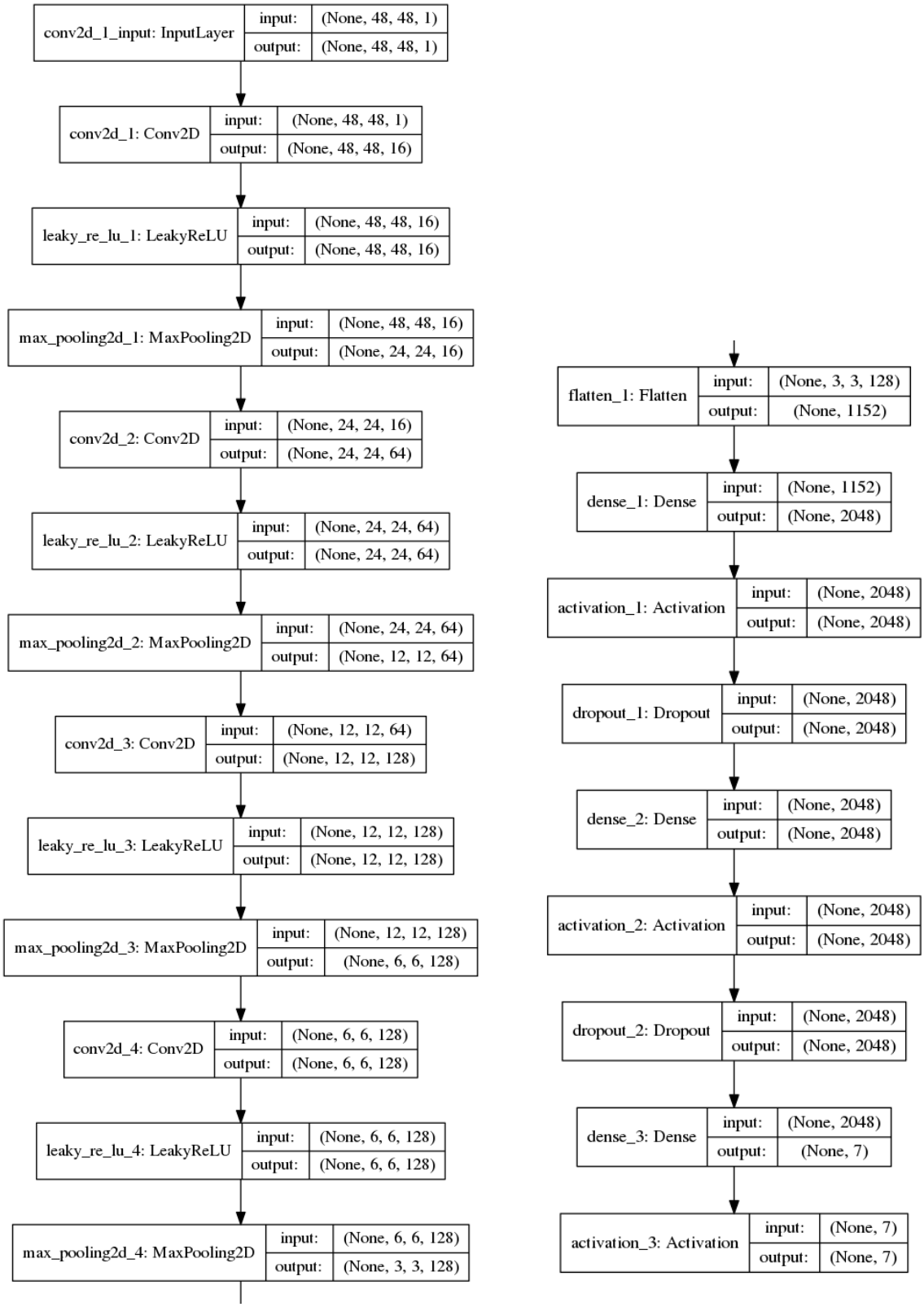
學號：B05901189 系級： 電機一 姓名：吳祥叡

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?

答：

(1)模型架構：



Number of Parameters:7212775

(2) 預處理：使用 ImageDataGenerator

featurewise_center=False,
featurewise_std_normalization=False,
rotation_range=10,
width_shift_range=0.1,
height_shift_range=0.1,
horizontal_flip=True,data_format='channels_last'

原本沒有用 ImageDataGenerator 都會 overfitting
用了以後發現雖然 training set 上的 performance 進步很慢但
不太會 overfitting 了

(3) 訓練過程：

opt=Adam()
loss=’catagorical_crossentropy
model.fit_generator(
datagen.flow(
x_train, y_train,batch_size=100),steps_per_epoch=len(x_train), epochs=1)

總共跑 2 個 epoch，大概 16 小時

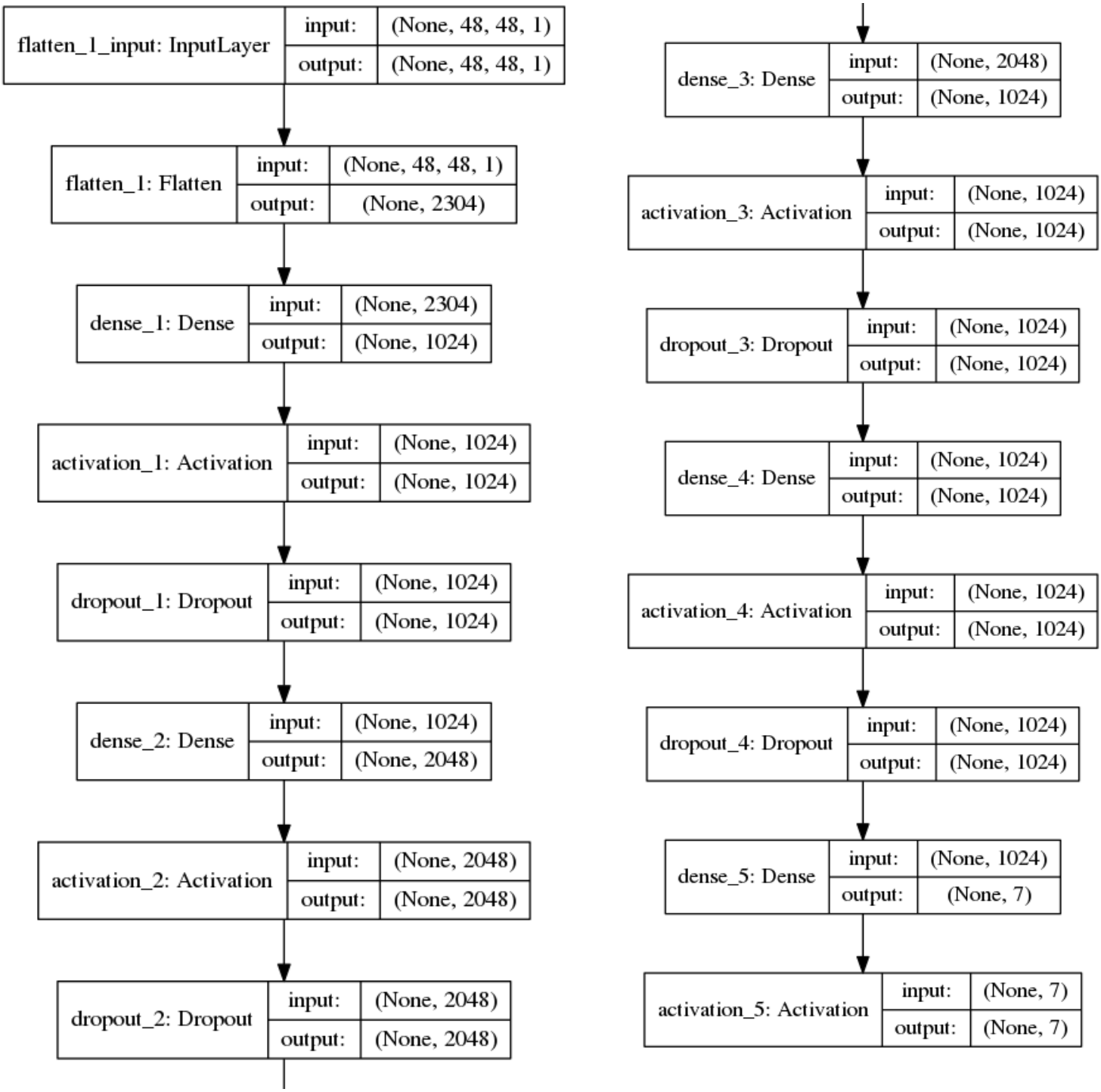
(4) Performance:

68% on Training set
65% on Kaggle Public test.
` 有一點點 overfitting

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？

答：

(1) 模型架構



(2) 預處理：同 CNN.

(3) 訓練過程: 同 CNN 但只訓練 1 個 epoch 大約 6 小時. 有嘗試訓練第二個 epoch 但沒太大進步所以只用訓練一次的結果.

(4) performance:

25% on training set.
24% on Public test.
表現均不如 CNN.

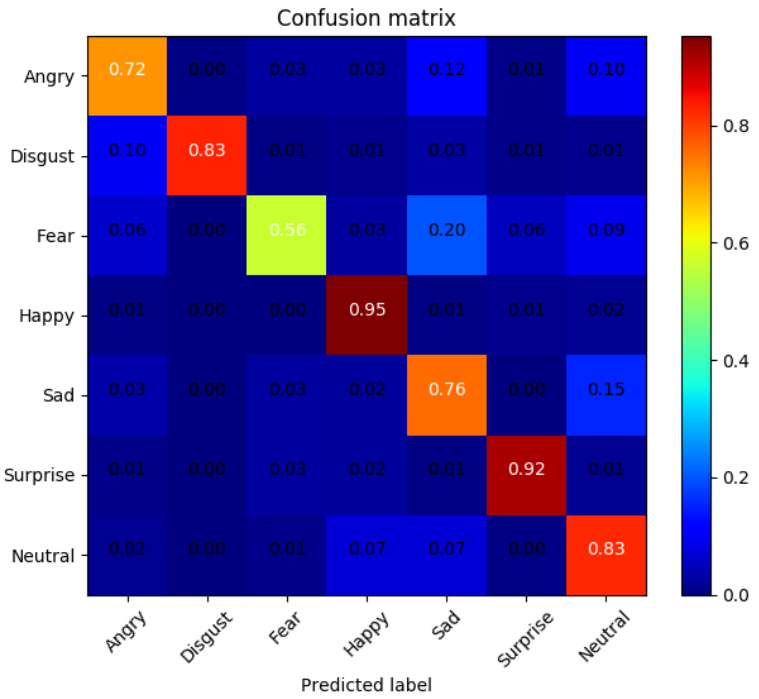
3. (1%) 觀察答錯的圖片中，哪些 class 彼此間容易用混？[繪出 confusion matrix 分析]

答：

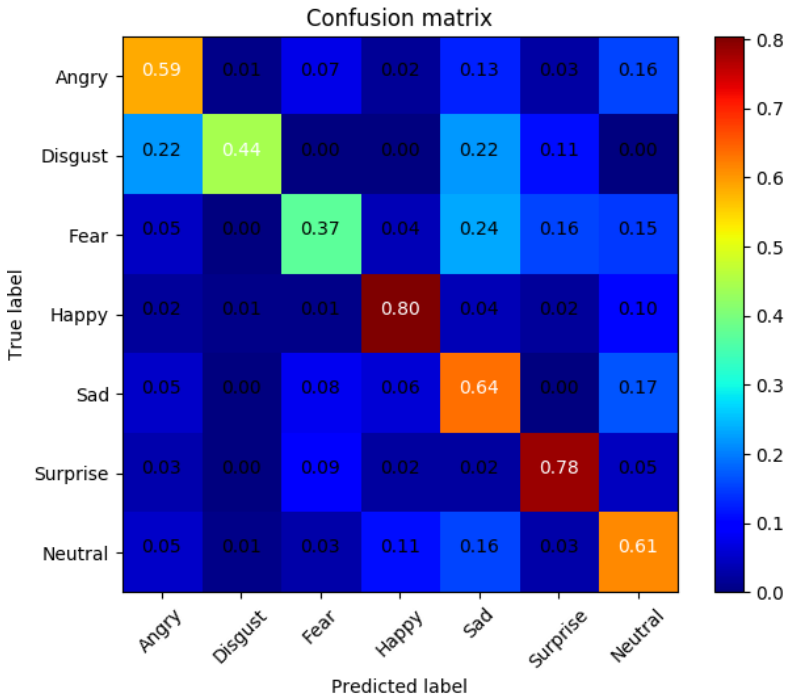
training set 為 train.csv 前 28000 張圖

validation set 為剩下 709 張

(1) Confusion Matrix of training set:




(2) Confusion Matrix of validation set:



(3)分析

在 training set 上原本就容易混淆（>0.2 的）的有：

將 Fear 判斷為 Sad

而在 validation set 上

將 Fear 判斷為 Sad

將 Disgust 判斷為 Angry

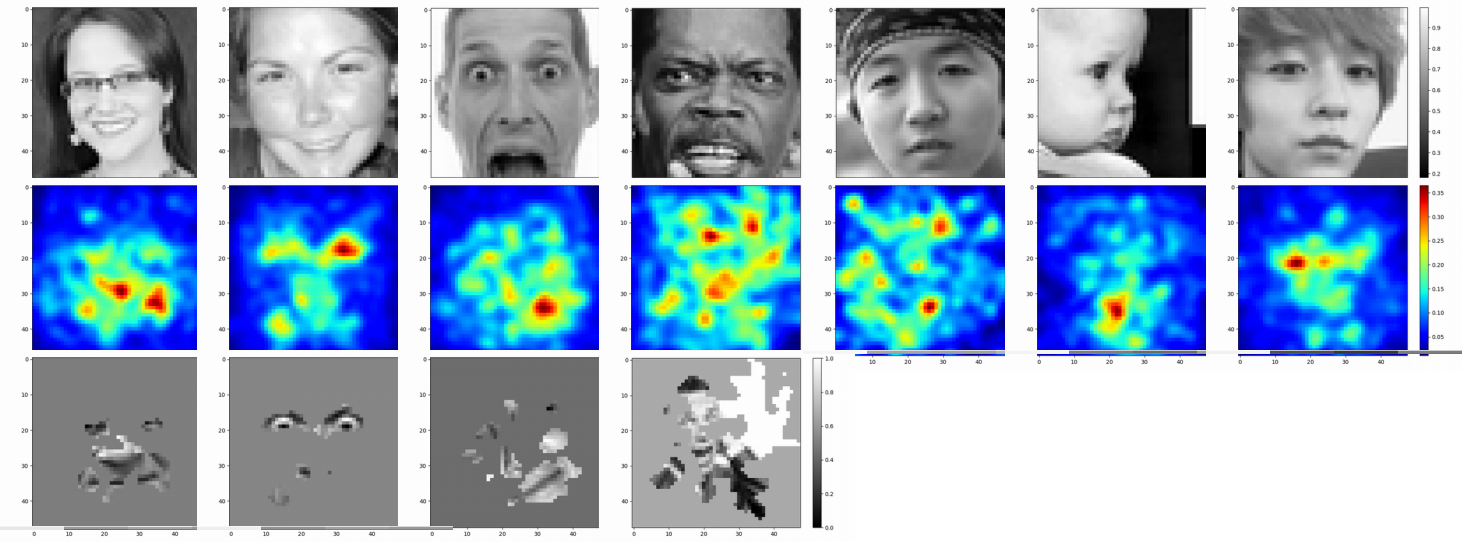
將 Disgust 判斷為 Sad

4. (1%) 從(1)(2)可以發現，使用 CNN 的確有些好處，試繪出其 saliency maps，觀察模型在做 classification 時，是 focus 在圖片的哪些部份?

答：

取 training data 的前 21 張圖做 saliency map,並繪出 heat 大於 threshold 的區域

發現這些區域集中在五官上,可以推論機器是用五官來判斷表情.



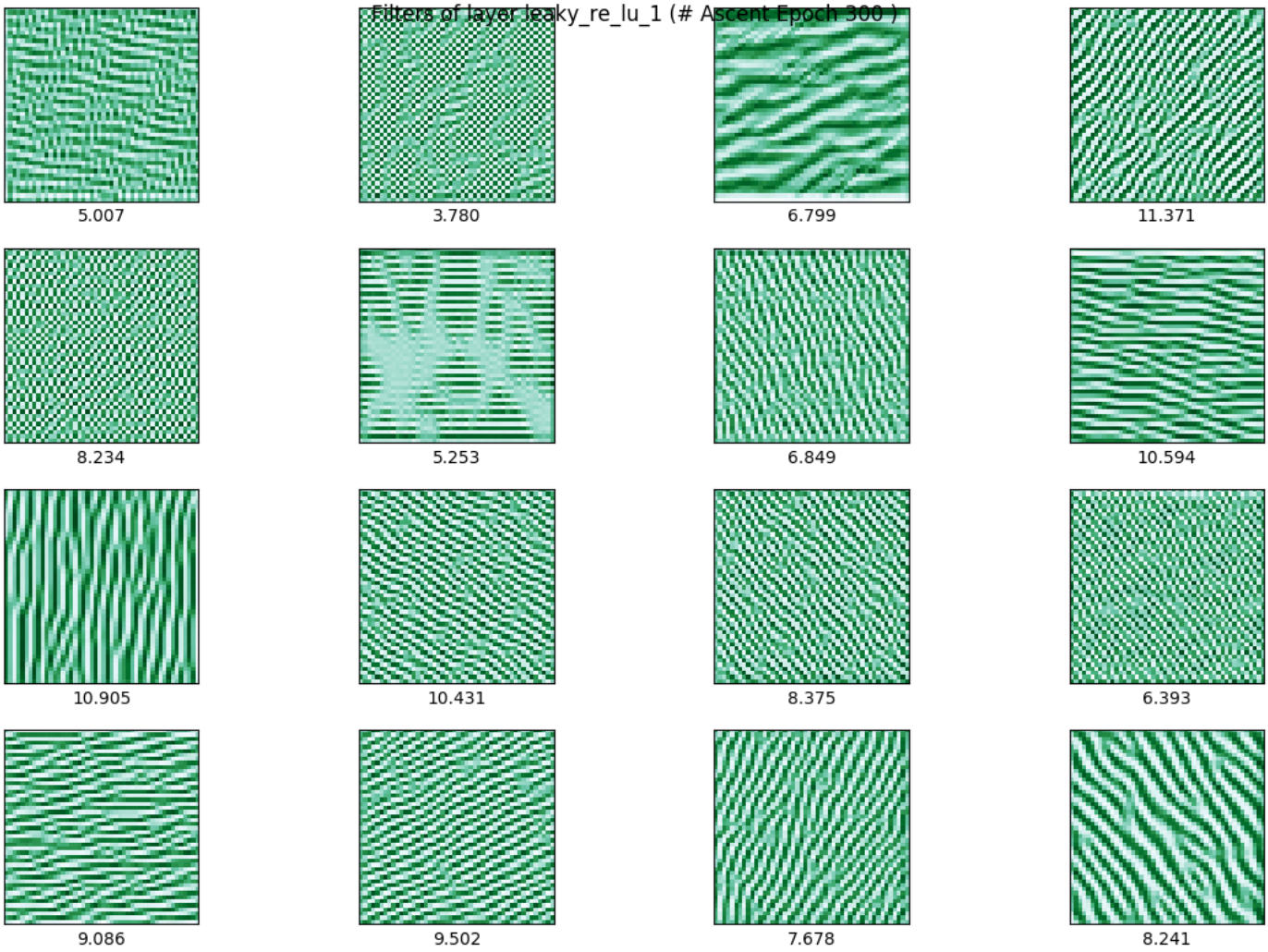
5. (1%) 承(1)(2)，利用上課所提到的 gradient ascent 方法，觀察特定層的 filter


最容易被哪種圖片 activate。

答：

(1) 從 white noise 起始, 觀察第一層 activation function 最容易被以下圖片 activate.

看起來像一些紋路


Filters of layer leaky_re_lu_1 (# Ascent Epoch 300 )
5.007
3.780
6.799
11.371
8.234
5.253
6.849
10.594
10.905
10.431
8.375
6.393
9.086
9.502
7.678
8.241