**Machine Learning Fall 2020 ——— Homework 3**

學號：B07902037 系級： 資工三 姓名：蔡沛勳

**1. (1%) 請說明這次使用的model架構，包含各層維度及連接方式。**

我的model使用了三層CNN，具體結構如下：

self.cnn = nn.Sequential(

# Convolution 1 (1 \* 48 \* 48) -> (64 \* 23 \* 23)

nn.Conv2d(in\_channels = 1, out\_channels = 64, kernel\_size = 3, stride = 1, ),

nn.BatchNorm2d(num\_features = 64),

nn.LeakyReLU(negative\_slope = 0.1),

nn.MaxPool2d(kernel\_size = 2, stride = 2),

# Convolution 2 (64 \* 23 \* 23) -> (128 \* 9 \* 9)

nn.Conv2d(in\_channels = 64, out\_channels = 128, kernel\_size = 6, stride = 1, ),

nn.BatchNorm2d(num\_features = 128),

nn.LeakyReLU(negative\_slope = 0.1),

nn.MaxPool2d(kernel\_size = 2, stride = 2),

# Convolution 3 (128 \* 9 \* 9) -> (256 \* 3 \* 3)

nn.Conv2d(in\_channels = 128, out\_channels = 256, kernel\_size = 4, stride = 1, ),

nn.BatchNorm2d(num\_features = 256),

nn.LeakyReLU(negative\_slope = 0.1),

nn.MaxPool2d(kernel\_size = 2, stride = 2),

)

#Fully connection (256 \* 3 \* 3) -> (1024) -> (128) -> (7)

self.fc = nn.Sequential(

nn.Linear(in\_features = 256 \* 3 \* 3, out\_features = 1024),

nn.LeakyReLU(negative\_slope = 0.1),

nn.Dropout(p = 0.5),

nn.Linear(in\_features = 1024, out\_features = 128),

nn.LeakyReLU(negative\_slope = 0.1),

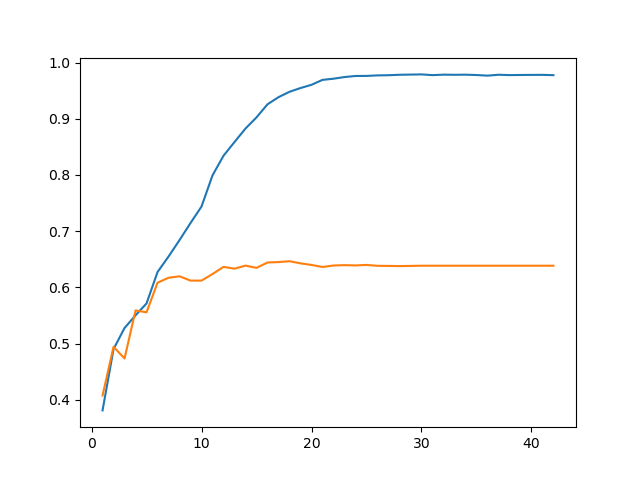
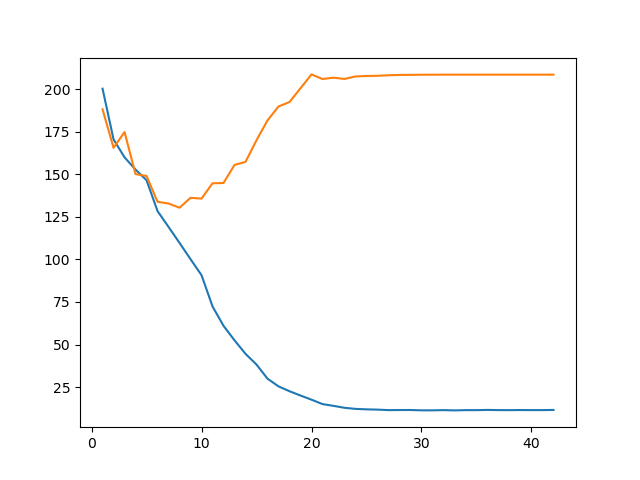
nn.Dropout(p = 0.5),

nn.Linear(in\_features = 128, out\_features = 7)

)

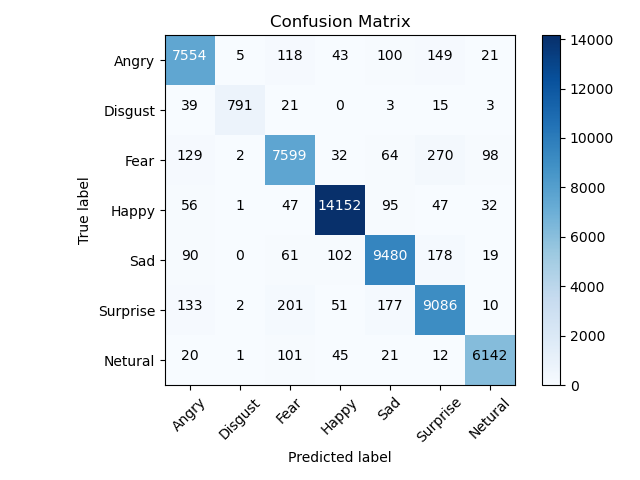
**2.** **(1%) 請附上model的training/validation history (loss and accuracy)。**

我先將所有資料做左右翻轉得到兩倍資料後，再對全部資料的90%拿來做training，10%做validation，在batch size為128時做40次epoch所得到結果如下。其中左圖為training/validation 的accuracy history，右圖為training/validation的loss history。 (藍線為training，橘線為validation)

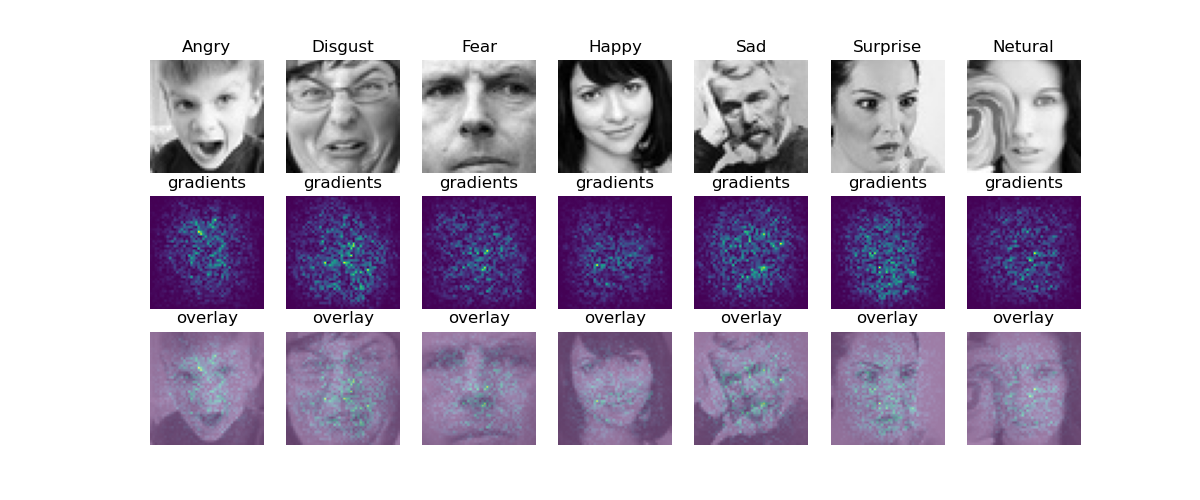
**3.** **(1%) 畫出confusion matrix分析哪些類別的圖片容易使model搞混，並簡單說明。**

下圖為原先資料加上左右翻轉的資料合計57418筆資料產生的confusion matrix。



可以觀察出Fear跟Surprise彼此最容易使model搞混。推測原因為人類對這兩種情緒的反應較為相似(ex. 睜大眼睛、嘴巴)，導致CNN產生的結果也較為接近。

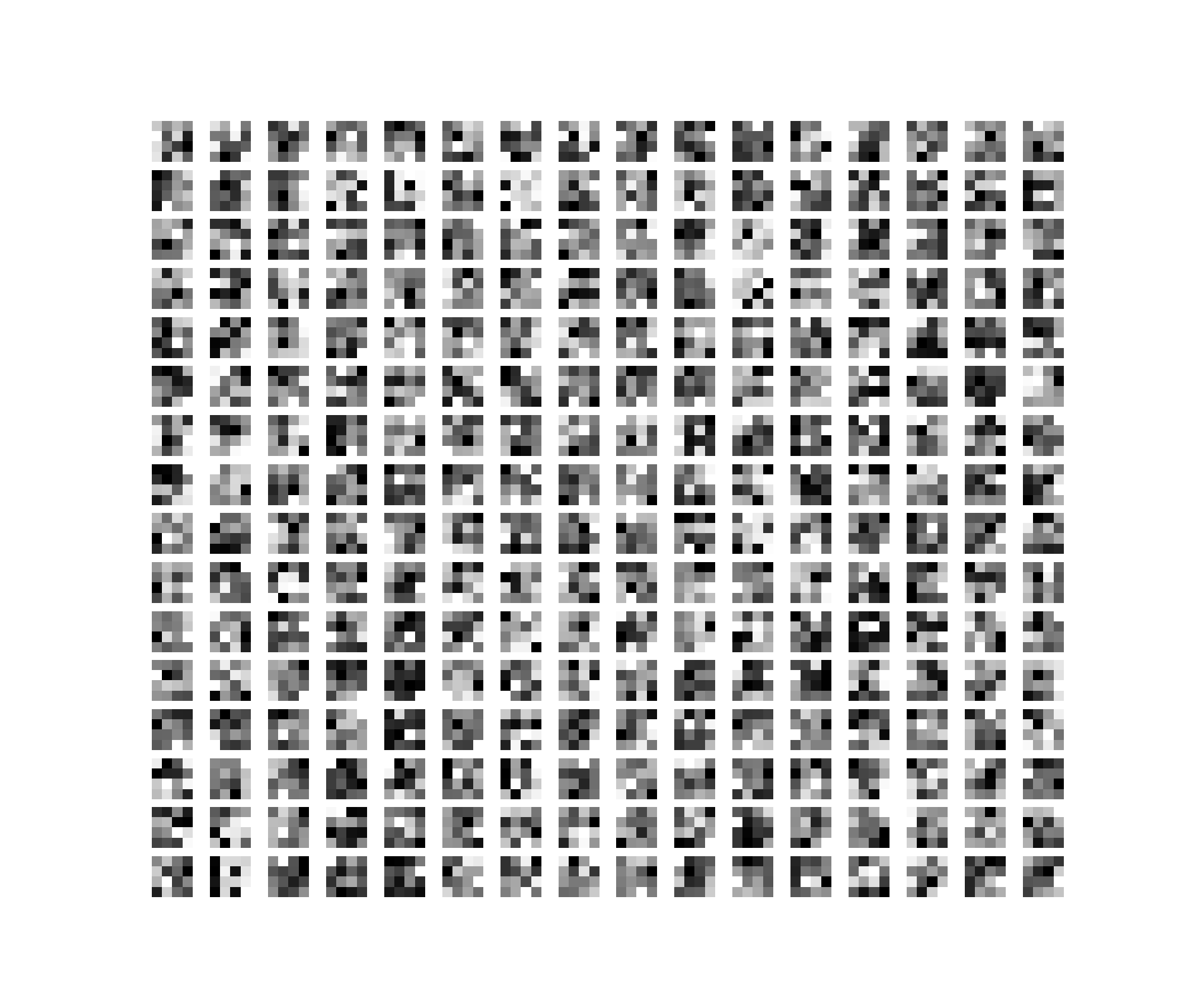
**4. (1%) 畫出CNN model的saliency map，並簡單討論其現象。**



可以看出gradient再眼睛、鼻子及嘴巴等處數值較大，代表model主要是利用人的五官來辨識照片中人物的情緒。

**5. (1%) 畫出最後一層的filters最容易被哪些feature activate。**

以下為我的model中最後一層的256筆4 \* 4 的filters。



**6. (3%) Refer to math problem**

1.

設變化後大小變為。

2.

由題目得到

可推得

:

*:*

*:*

*:*

*:*

*:*

3.

已知

在時，故

在時，故