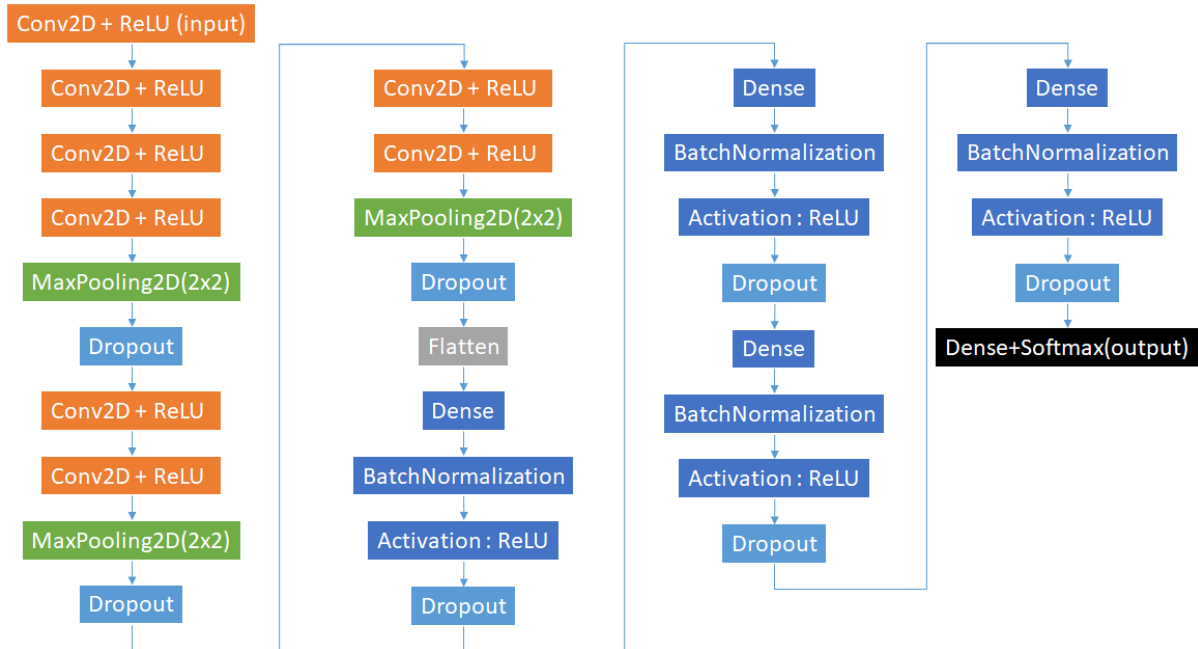


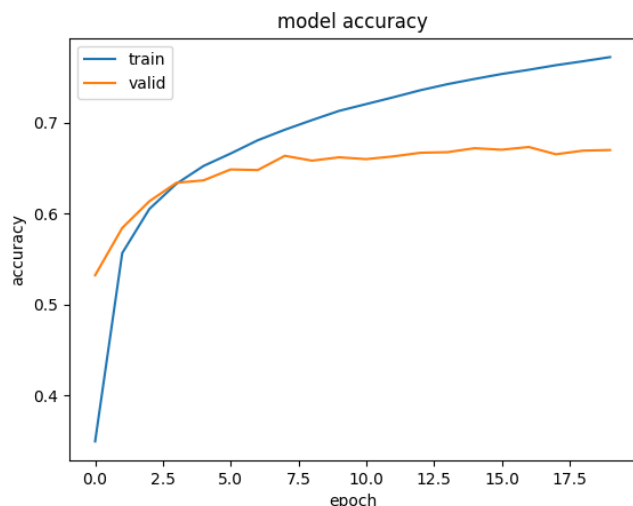
1. (1%) 請說明你實作的 CNN model，其模型架構、訓練過程和準確率為何？
(Collaborators: 之前修課的陳柏瑋、林函毅學長教我可以利用 ImageDataGenerator 來產生更多的 training data 而不用手動做這件事情。同實驗室一起修課、討論的李明庭、游彥勝)

答：我的 CNN model 如下圖：



在這個模型中，加上 input/output 總共有八層 Conv2D、三層 MaxPooling2D、五層 Dense。除了最後一層輸出層以外，使用的 Activation Function 皆為 ReLU。最後使用的 Optimizer 為 Adamax。

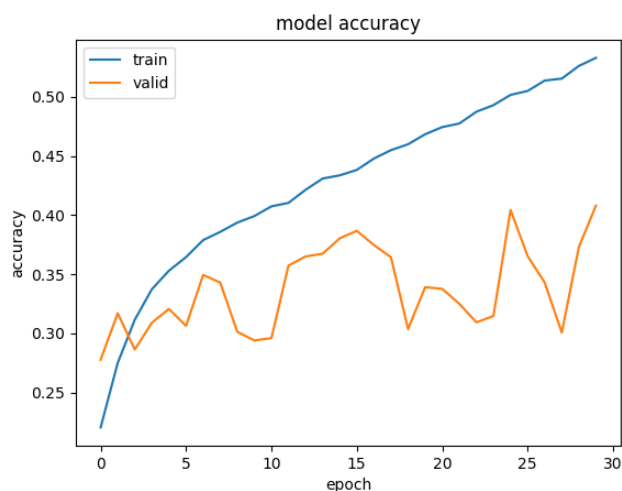
訓練過程如右圖，可以看到當 epoch 到達 5 左右模型的 validation accuracy 便不再增長。此模型的訓練結果在 Kaggle Public Score 得到 **6.8459** 分。



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？

(Collaborators: 李明庭、游彥勝)

答：第一題中所使用的 CNN model 中 total params = 3,812,071。我取出最後五層 Dense，也就是 MLP 的部分來做實驗，第一層在處理資料的時候就存為 1*2304 的 array 而不是 48*48，因此跳過了 flatten 層。Model's summary 如下：



此模型取得的參數數量為 3,951,111，和第一題的 CNN model 相近。最後在 Kaggle Public Score 取得的分數為 **0.24547**。從上圖的趨勢可以看到 validation 幾乎沒有收斂，我認為在做圖像辨識的時候，若使用 DNN，等於是把數據都變成單純的數據再做預測，相較 CNN 使用了 convolution 來取得圖像的特徵，準確率自然較差。因此要使用正確的 model 來處理不同的問題。

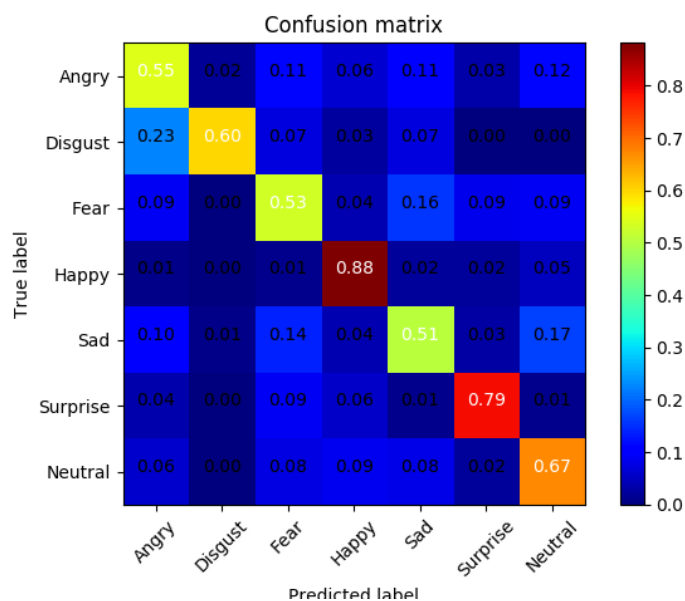
Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 1024)	2360320
batch_normalization_1 (Batch Normalization)	(None, 1024)	4096
activation_1 (Activation)	(None, 1024)	0
dropout_1 (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 512)	524800
batch_normalization_2 (Batch Normalization)	(None, 512)	2048
activation_2 (Activation)	(None, 512)	0
dropout_2 (Dropout)	(None, 512)	0
dense_3 (Dense)	(None, 1024)	525312
batch_normalization_3 (Batch Normalization)	(None, 1024)	4096
activation_3 (Activation)	(None, 1024)	0
dropout_3 (Dropout)	(None, 1024)	0
dense_4 (Dense)	(None, 512)	524800
batch_normalization_4 (Batch Normalization)	(None, 512)	2048
activation_4 (Activation)	(None, 512)	0
dropout_4 (Dropout)	(None, 512)	0
dense_5 (Dense)	(None, 7)	3591
Total params: 3,951,111		
Trainable params: 3,944,967		
Non-trainable params: 6,144		

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]

(Collaborators: 李明庭、游彥勝)

答：使用和 Training 的時候相同的 Validation Data 所繪製的

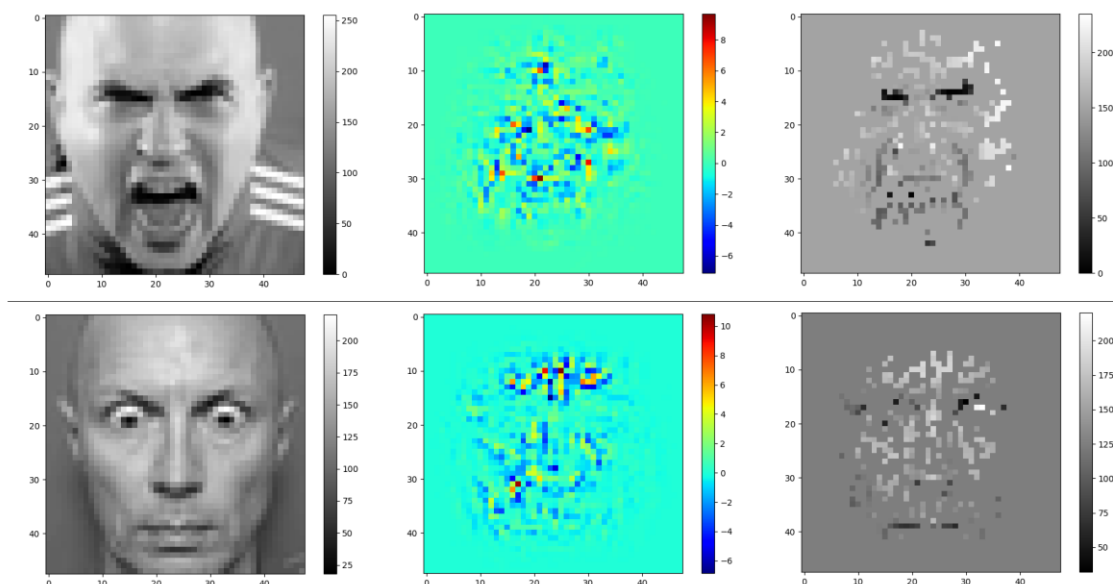
Confusion matrix 如右圖。從右圖中可以看出預測錯誤當中，第一名的是 Disgust 被誤判為 Angry；第二名是 Sad 誤判為 Neutral；第三名是 Fear 誤判為 Sad。以此模型來說，七種表情當中最容易被誤判的是 Sad，正確率只有 0.51。



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？

(Collaborators: 李明庭、游彥勝)

答：如下圖，我們可以觀察出在分類的時候幾乎都是集中在**五官**上(眼睛、鼻子、嘴巴)，跟人類判斷對方表情的方式好像差不多。

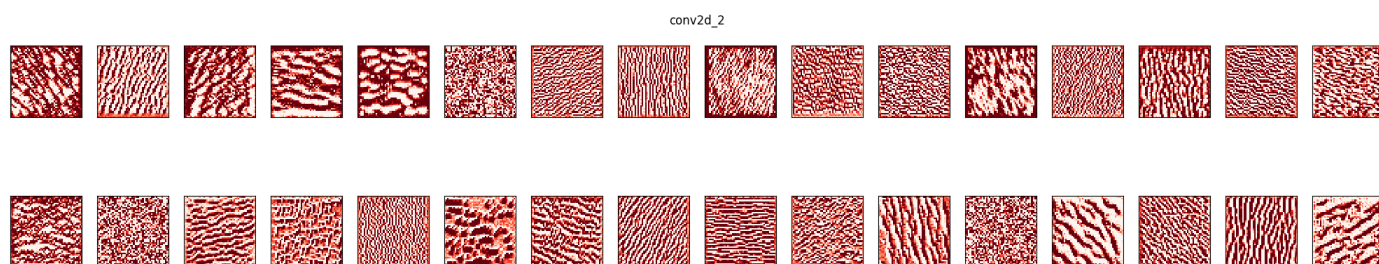


5. (1%) 承(1)(2)，利用上課所提到的 **gradient ascent** 方法，觀察特定層的 **filter** 最容易被哪種圖片 **activate**。

(Collaborators: 李明庭、游彥勝)

答：

5-1. 下圖為模型中第二層，Conv2D(32)的 32 個 filters，並輸入 **white noise** 的圖形來觀察 **filters** 的結果。可以看到每個 **filter** 都抓到不同的特徵。



5-2. 下圖為隨機抽取一筆 **data**，經過 Conv2D(32)的時候通過 32 個 **filters** 所抓取到的結果。

