學號：R06922117 系級：資工碩一 姓名：李岳庭

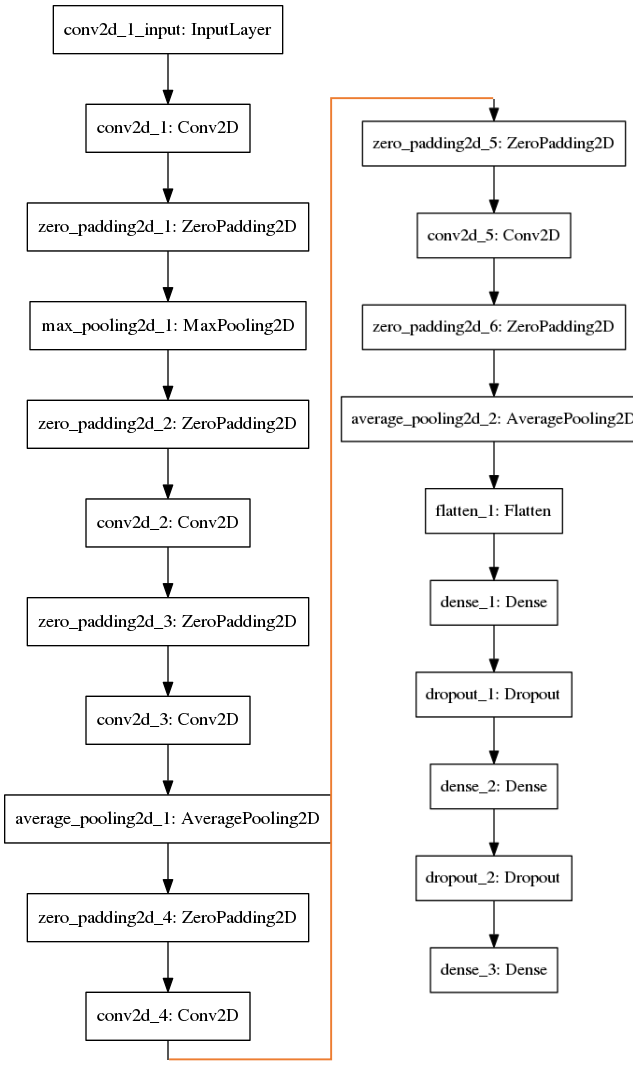
1. (1%) 請說明你實作的 CNN model，其模型架構、訓練過程和準確率為何？

答：

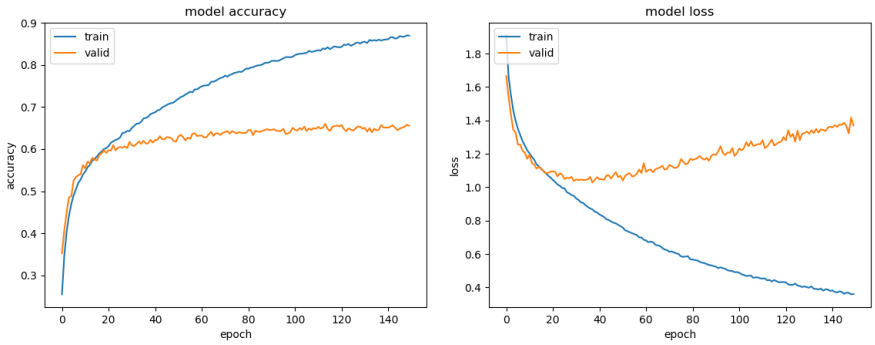
我使用的 CNN 架構與手把手教學的相同，其 summary 如下：

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 44, 44, 64)	1664
zero_padding2d_1 (ZeroPaddin	(None, 48, 48, 64)	0
max_pooling2d_1 (MaxPooling2	(None, 22, 22, 64)	0
zero_padding2d_2 (ZeroPaddin	(None, 24, 24, 64)	0
conv2d_2 (Conv2D)	(None, 22, 22, 64)	36928
zero_padding2d_3 (ZeroPaddin	(None, 24, 24, 64)	0
conv2d_3 (Conv2D)	(None, 22, 22, 64)	36928
average_pooling2d_1 (Average	(None, 10, 10, 64)	0
zero_padding2d_4 (ZeroPaddin	(None, 12, 12, 64)	0
conv2d_4 (Conv2D)	(None, 10, 10, 128)	73856
zero_padding2d_5 (ZeroPaddin	(None, 12, 12, 128)	0
conv2d_5 (Conv2D)	(None, 10, 10, 128)	147584
zero_padding2d_6 (ZeroPaddin	(None, 12, 12, 128)	0
average_pooling2d_2 (Average	(None, 5, 5, 128)	0
flatten_1 (Flatten)	(None, 3200)	0
dense_1 (Dense)	(None, 1024)	3277824
dropout_1 (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 1024)	1049600
dropout_2 (Dropout)	(None, 1024)	0
dense_3 (Dense)	(None, 7)	7175

Total params: 4,631,559
Trainable params: 4,631,559
Non-trainable params: 0



一開始用手把手的架構，準確率大概 60%左右，我有嘗試過其它的模型，但都沒有顯著提升，加了 image generator 後，準確率才提升到 66%~67%。

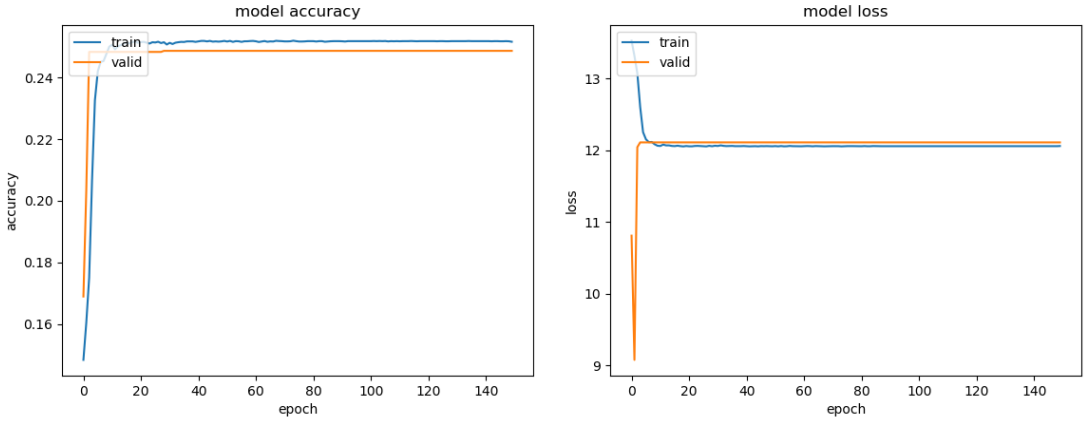


特別的地方是，由上圖的 accuracy 和 loss 的趨勢圖可看出，雖然 val_loss 到後期越來越高，但是 val_acc 也緩慢上升，雖然趨勢圖看起來有 overfitting 的跡象，但 test 的結果丟上 kaggle 的準確率居然達到 67%，若設 early stoping 的結果則是 64%，所以我的推論是這次的 data 及 CNN 架構適合 train 越多 epoch 越準。

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：
將 DNN 架構建構成跟 CNN 的架構有差不多的參數(Total params)，大概 400 多萬個，其準確率只有 24%，而且 loss 很難下降，效果比 CNN 差很多，可見對於影像分類的問題，CNN 是非常重要的技術。

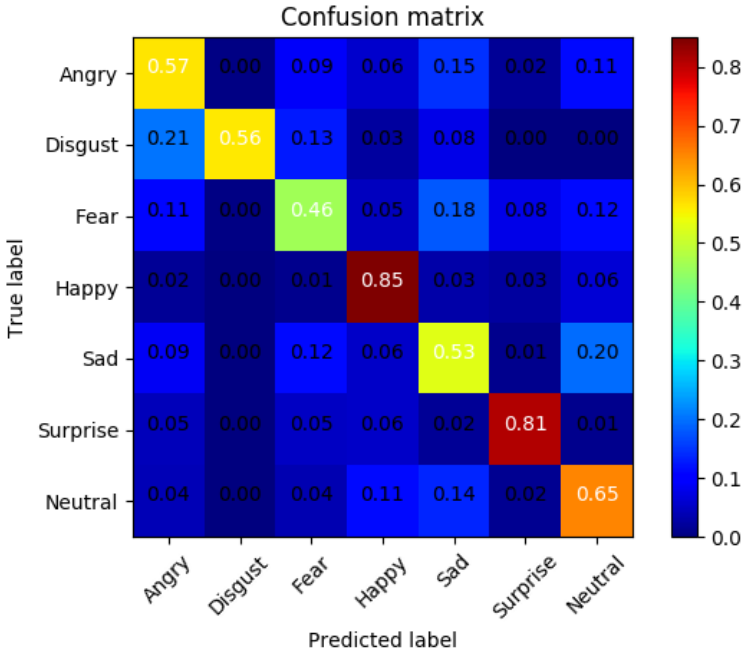
Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 256)	590080
dropout_1 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 256)	65792
dropout_2 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 256)	65792
dropout_3 (Dropout)	(None, 256)	0
dense_4 (Dense)	(None, 1024)	263168
dropout_4 (Dropout)	(None, 1024)	0
dense_5 (Dense)	(None, 1024)	1049600
dropout_5 (Dropout)	(None, 1024)	0
dense_6 (Dense)	(None, 1024)	1049600
dropout_6 (Dropout)	(None, 1024)	0
dense_7 (Dense)	(None, 1024)	1049600
dropout_7 (Dropout)	(None, 1024)	0
dense_8 (Dense)	(None, 7)	7175

Total params: 4,140,807
Trainable params: 4,140,807
Non-trainable params: 0



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：



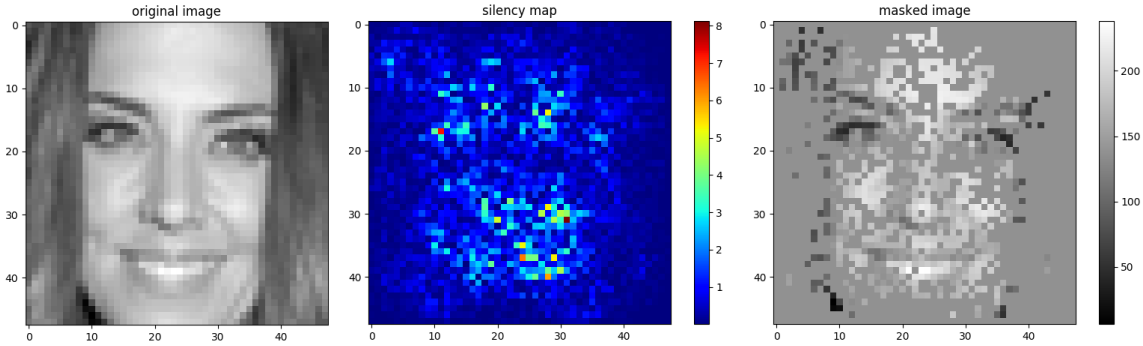
我使用的 validation set 是 training set 的前 1/10 筆資料，其 label 分佈是

Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral
423	39	431	711	466	307	493

其中 Disgust 數量偏少，因此不討論他，預測準確率最低的是 Fear，只有 46%，被預測錯的類別分別是 Sad(18%)、Neutral(12%)、Angry(11%)。

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：

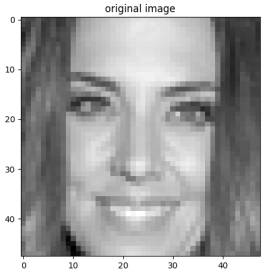


我使用的圖片 idx=9487，由 silency map 可看出 heat 高的部分都在五官處，與人類直覺判斷的方式蠻符合的。

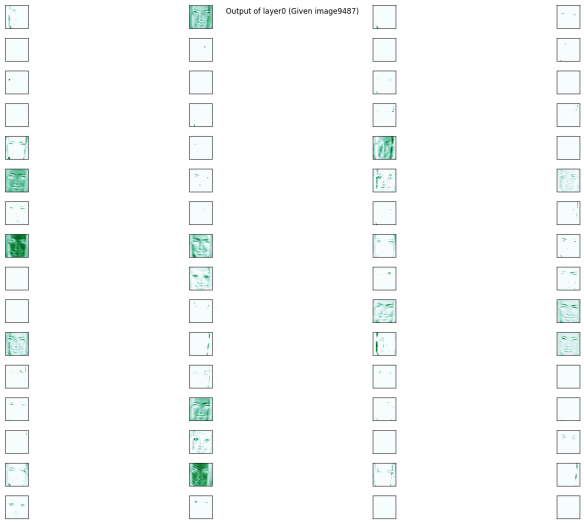
5. (1%) 承(1)(2)，利用上課所提到的 gradient ascent 方法，觀察特定層的 filter 最容易被哪種圖片 activate。

答：

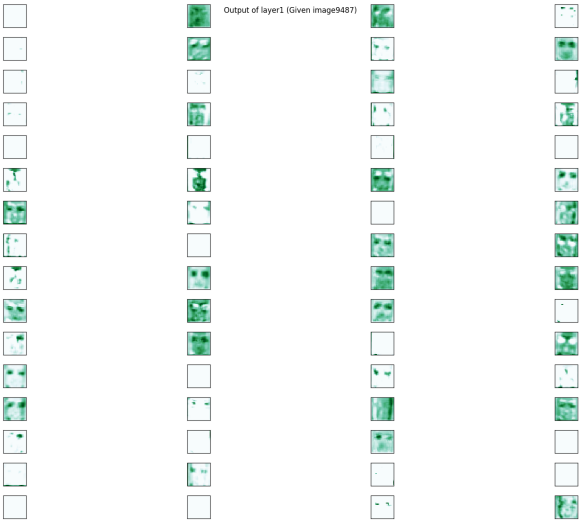
input image:



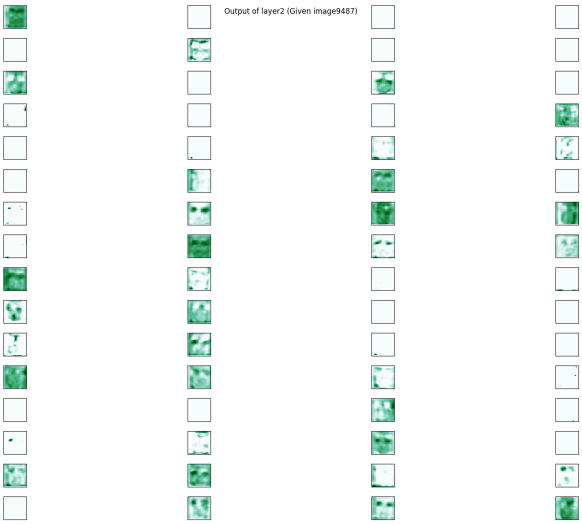
layer 0:



layer 1:



layer 2:



filters: