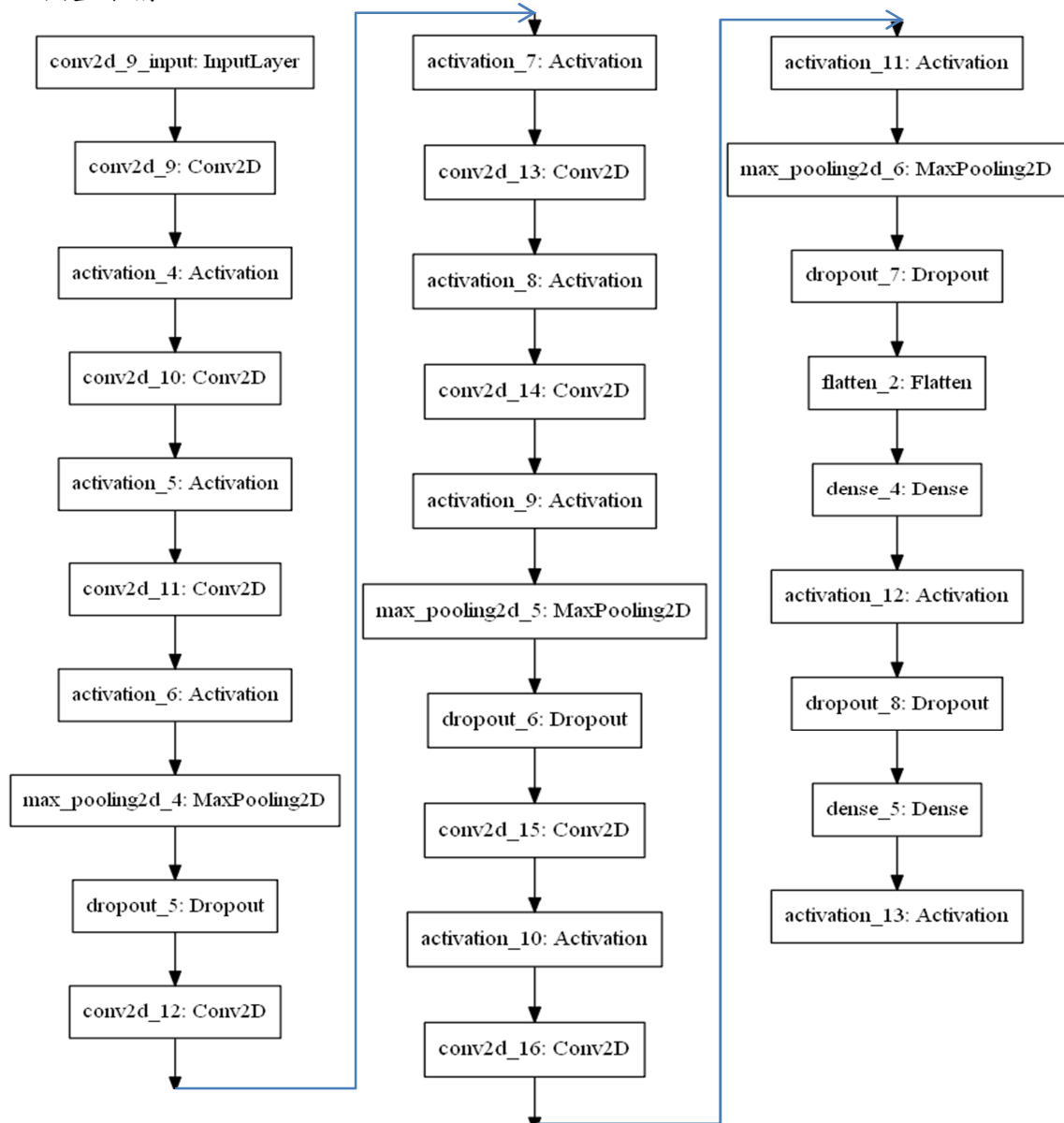


1. (1%) 請說明你實作的 CNN model，其模型架構、訓練過程和準確率為何？

答：

模型架構：



```
model = Sequential()
model.add(Convolution2D(64, 3, padding='same', input_shape=(48,48,1)))
model.add(Activation('relu'))
model.add(Convolution2D(64, 3, padding='same'))
model.add(Activation('relu'))
model.add(Convolution2D(64, 3, padding='same'))
model.add(Activation('relu'))
model.add(MaxPooling2D((2,2)))
model.add(Dropout(0.5))

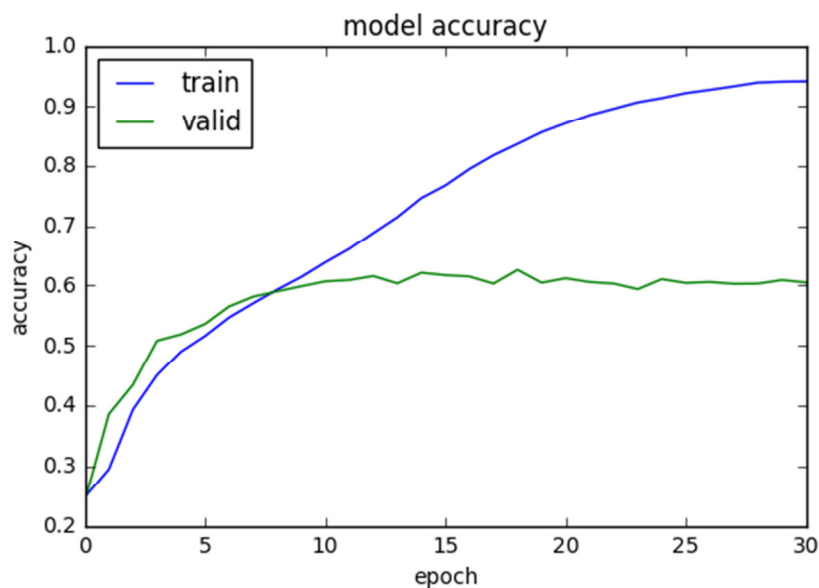
model.add(Convolution2D(64, 3, padding='same'))
model.add(Activation('relu'))
model.add(Convolution2D(64, 3, padding='same'))
model.add(Activation('relu'))
model.add(Convolution2D(64, 3, padding='same'))
model.add(Activation('relu'))
model.add(MaxPooling2D((2,2)))
model.add(Dropout(0.5))

model.add(Convolution2D(128, 3, padding='same'))
model.add(Activation('relu'))
model.add(Convolution2D(128, 3, padding='same'))
model.add(Activation('relu'))
model.add(MaxPooling2D((2,2)))
model.add(Dropout(0.5))

model.add(Flatten())
model.add(Dense(256))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(7))
model.add(Activation('softmax'))
```

Total params: 1,588,423
Trainable params: 1,588,423
Non-trainable params: 0

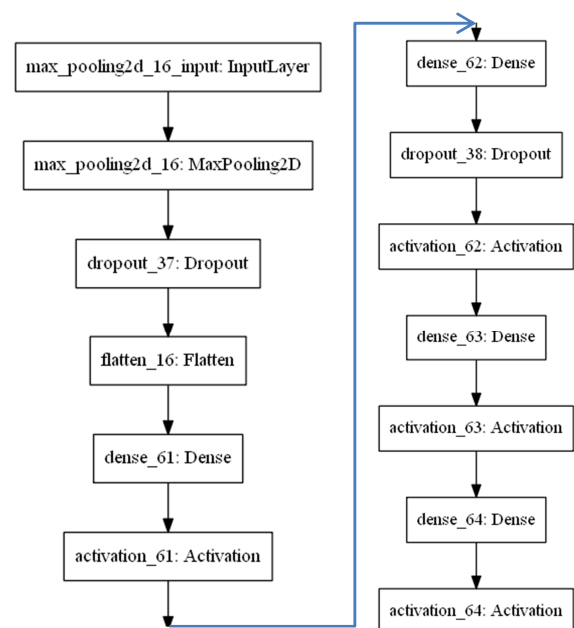
訓練過程與準確率：



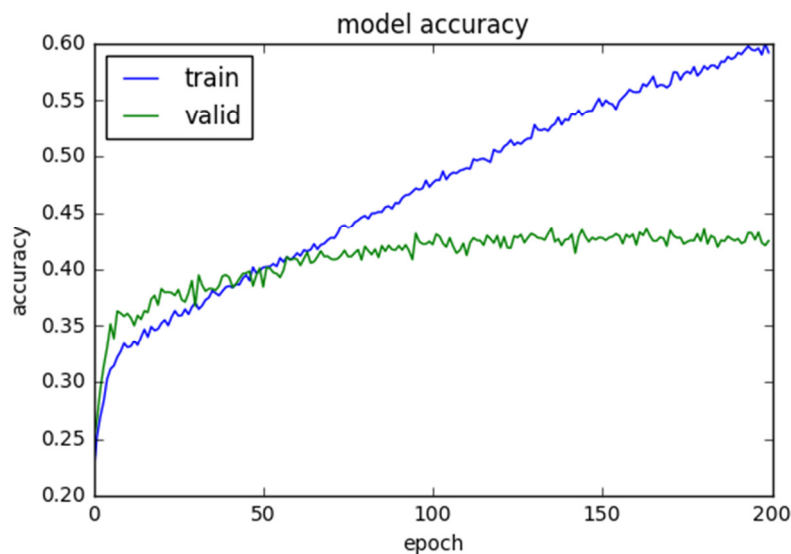
使用最後 20% 的 training data 作為 valid data，訓練過程前期 training accuracy 和 valid accuracy 皆呈現增加，在準確率達約 0.6 後，valid accuracy 便不隨 train accuracy 繼續增加，而在 0.6 上下浮動，應該是防止 overfitting 的部分(dropout 等等)沒有做好，或是模型結構的設計還有待改善。

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：

Layer (type)	Output Shape	Param #
=====		
max_pooling2d_21 (MaxPooling (None, 48, 48, 1))		0
=====		
dropout_47 (Dropout)	(None, 48, 48, 1)	0
=====		
flatten_21 (Flatten)	(None, 2304)	0
=====		
dense_81 (Dense)	(None, 512)	1180160
=====		
activation_81 (Activation)	(None, 512)	0
=====		
dense_82 (Dense)	(None, 512)	262656
=====		
dropout_48 (Dropout)	(None, 512)	0
=====		
activation_82 (Activation)	(None, 512)	0
=====		
dense_83 (Dense)	(None, 512)	262656
=====		
activation_83 (Activation)	(None, 512)	0
=====		
dense_84 (Dense)	(None, 7)	3591
=====		
activation_84 (Activation)	(None, 7)	0
=====		
Total params: 1,709,063		
Trainable params: 1,709,063		
Non-trainable params: 0		



訓練過程與準確率：



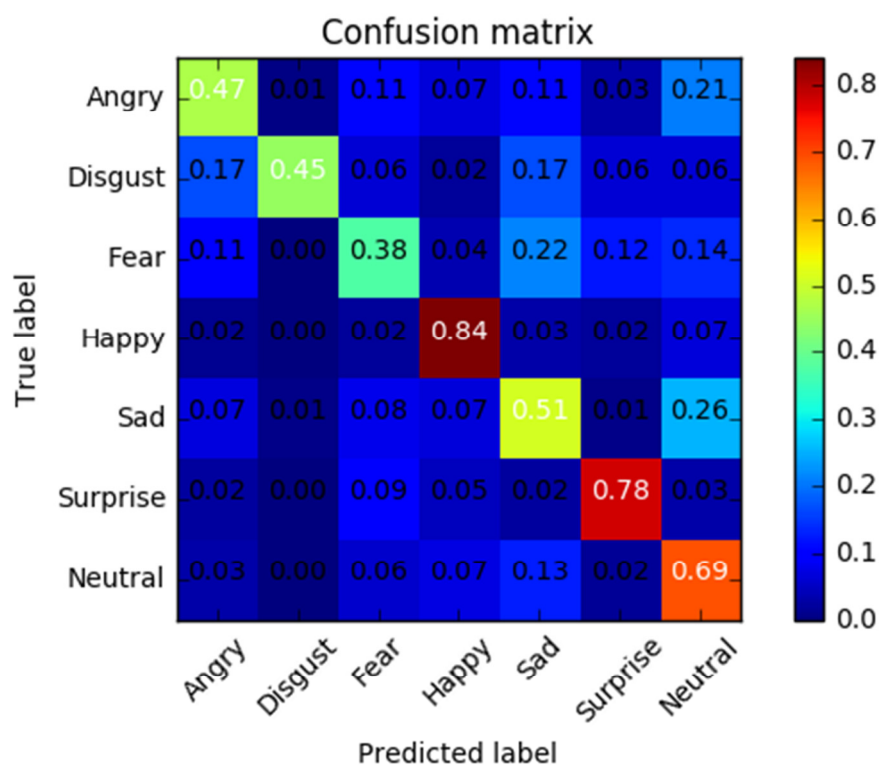
在使用參數量與 CNN 模型差不多的 DNN 模型作訓練時，一開始 training accuracy 和 valid accuracy 會些微增加，但在達到約 0.4 後，valid accuracy 就停止增加，而在 0.42 左右上下浮動。

訓練過程和 CNN 模型比較之下，完成一次 epoch 的速度雖然快上許多，但每次增加的準確率卻很小，並且最後收斂到的 valid accuracy 也比 CNN 模型來得差，顯示在這次圖片表情辨識的工作中，DNN 模型的效果不如 CNN 模型，可能需要特殊設計的模型結構，才能達到較好的效果。

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：

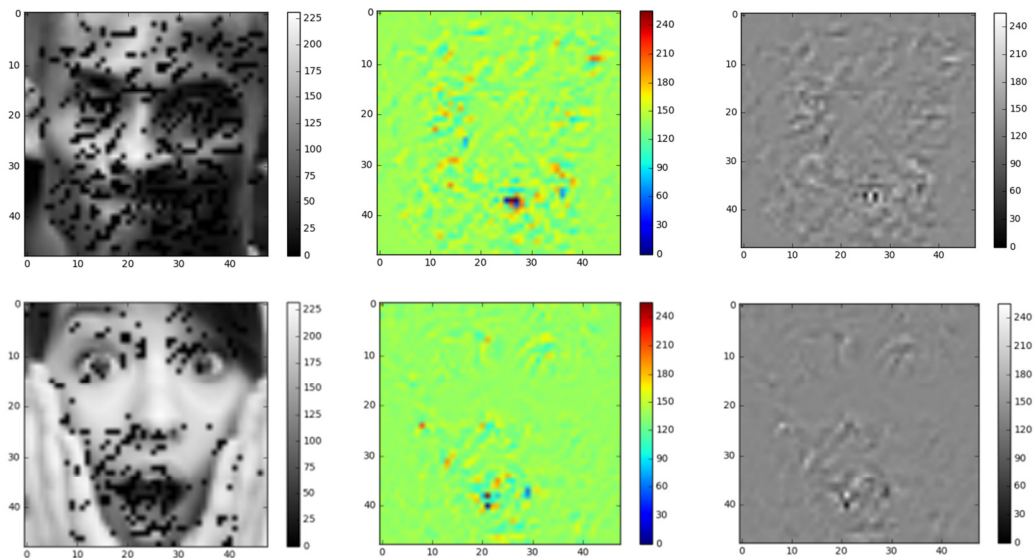
confusion matrix：



由 confusion matrix 觀察可以了解，我訓練出來的模型對於 Angry、Disgust 和 Fear 這三種 class 的表情分辨能力較差，都只有不到 0.5 的準確率，其中 Angry 較容易被誤認為 Neutral，Disgust 較易被誤認為 Angry 或 Sad，Fear 較易被誤認為 Sad，而 Sad 則容易被認成 Neutral。

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：



我在處理 Saliency Map 的部分沒有做好，所以結果不太明顯，推測應該是會 focus 在嘴巴還有眉毛的部分，這部分也與情緒表現有直接的相關性。

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：

[Bonus] (1%) 從 training data 中移除部份 label，實做 semi-supervised learning

[Bonus] (1%) 在 Problem 5 中，提供了 3 個 hint，可以嘗試實作及觀察（但也可以不限於 hint 所提到的方向，也可以自己去研究更多關於 CNN 細節的資料），並說明你做了些什麼？ [完成 1 個：+0.4%，完成 2 個：+0.7%，完成 3 個：+1%]