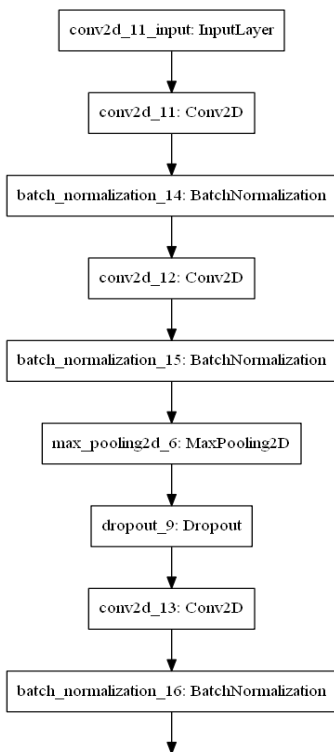


1. (1%) 請說明你實作的 CNN model，其模型架構、訓練過程和準確率為何？

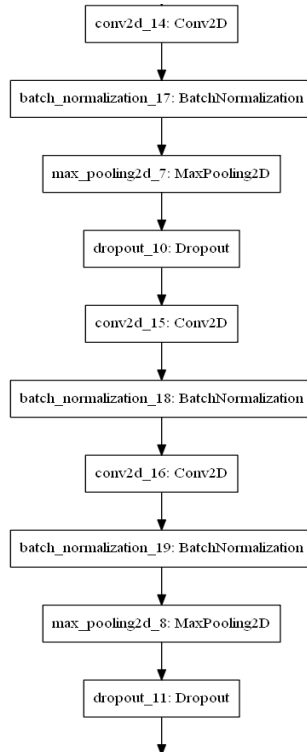
(Collaborators: R06942010 蘇建翰)

答：

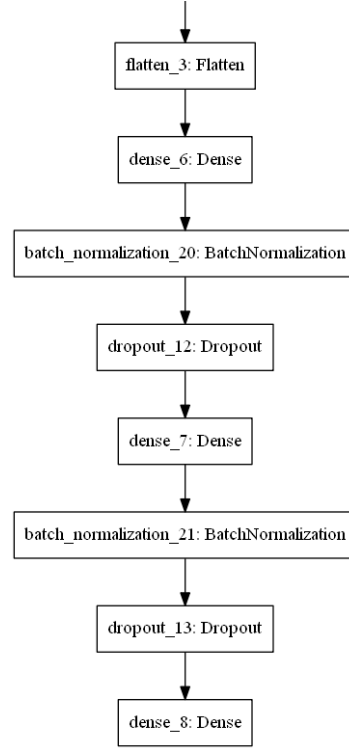
CNN model



(1)



(2)



(3)

我所實作的 CNN 包括了六層 convolution layer 和兩層 hidden layer，最後才是 output layer。(filter 數目分別是 32, 32, 64, 64, 128, 128，kernel size 為 3*3，兩層 hidden layer 分別使用 512 和 256 個 neuron)。我嘗試過 Maxpooling 和 Dropout 似乎不要每層都加效果比較好，而 BatchNormalization 則是每層都有加入。我訓練的 batch 為 32，讓他跑了 200 個 epoch(沒有 early stop)。另外，也有加上 Imagedatagenerator，提高資料量也提高了訓練準確率。

Training Procedure: Kaggle 上為 68.877% (最高 70.103% 為兩個類似的 model 作 ensemble)



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: R06942010 蘇建翰)

答：

DNN model:

Layer (type)	Output Shape	Param #
=====	=====	=====
dense_109 (Dense)	(None, 32)	73760
batch_normalization_181 (Bat	(None, 32)	128
dropout_134 (Dropout)	(None, 32)	0
dense_110 (Dense)	(None, 32)	1056
batch_normalization_182 (Bat	(None, 32)	128
dropout_135 (Dropout)	(None, 32)	0
dense_111 (Dense)	(None, 64)	2112
batch_normalization_183 (Bat	(None, 64)	256
dropout_136 (Dropout)	(None, 64)	0
dense_112 (Dense)	(None, 64)	4160
batch_normalization_184 (Bat	(None, 64)	256
dropout_137 (Dropout)	(None, 64)	0

(1)

dense_113 (Dense)	(None, 128)	8320
batch_normalization_185 (Bat	(None, 128)	512
dropout_138 (Dropout)	(None, 128)	0
dense_114 (Dense)	(None, 128)	16512
batch_normalization_186 (Bat	(None, 128)	512
dropout_139 (Dropout)	(None, 128)	0
dense_115 (Dense)	(None, 512)	66048
batch_normalization_187 (Bat	(None, 512)	2048
dropout_140 (Dropout)	(None, 512)	0
dense_116 (Dense)	(None, 256)	131328
batch_normalization_188 (Bat	(None, 256)	1024
dropout_141 (Dropout)	(None, 256)	0
dense_117 (Dense)	(None, 7)	1799
=====	=====	=====
Total params:	309,959	
Trainable params:	307,527	
Non-trainable params:	2,432	

(2)

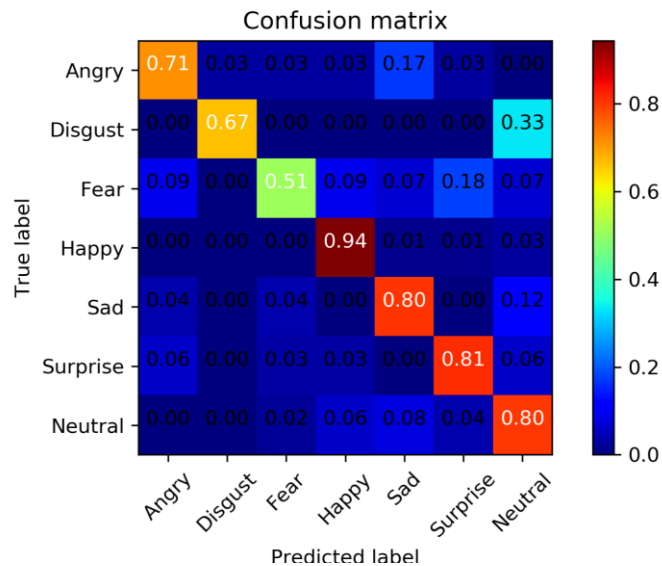
Training Procedure: Validation 上的準確率大概在 30%~40% 中間，Kaggle 分數為 36.712%



將 CNN 換成 DNN，training set 上的正確率成長非常緩慢，而 validation 上的準確率則是一直劇烈跳動(30%~40% 中間)。與 CNN 相近參數的 model，DNN 似乎 train 不太起來。

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: R06942010 蘇建翰)

答：



觀察 Confusion matrix 可以發現，預測模型在某些表現上面比較差。

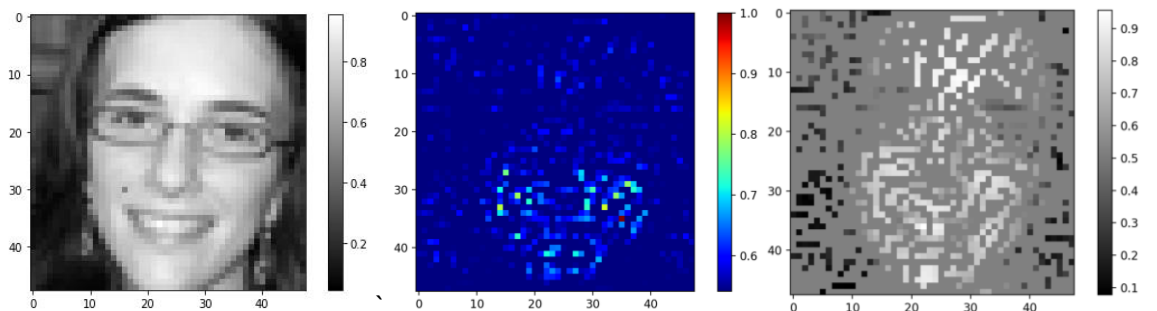
例如，Disgust 的表情容易被誤歸納成 Neutral。

Fear 的圖片預測準確率較低，容易被歸納成 Surprise。

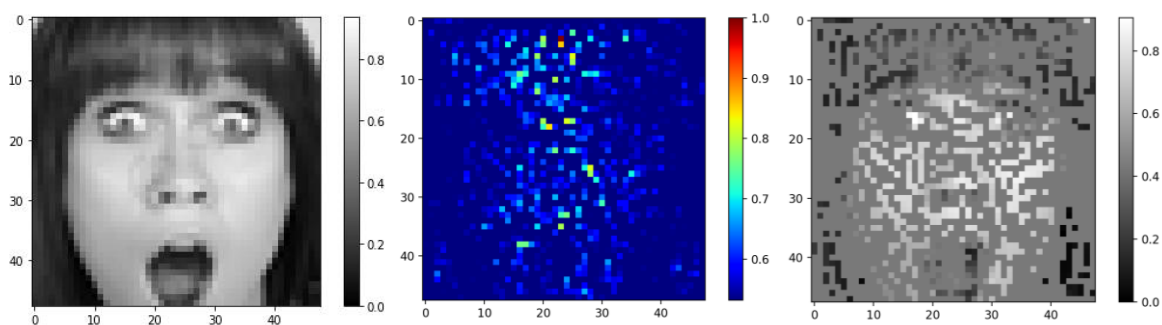
4.(1%) 從(1)(2)可以發現，使用 CNN 的確有些好處，試繪出其 saliency maps，觀察模型在做 classification 時，是 focus 在圖片的哪些部份？(Collaborators: R06942010 蘇建翰)

答：

Train_set: #145，主要可以看到模型抓取比較深邃的地方(彎曲的眼睛，嘴巴的弧)



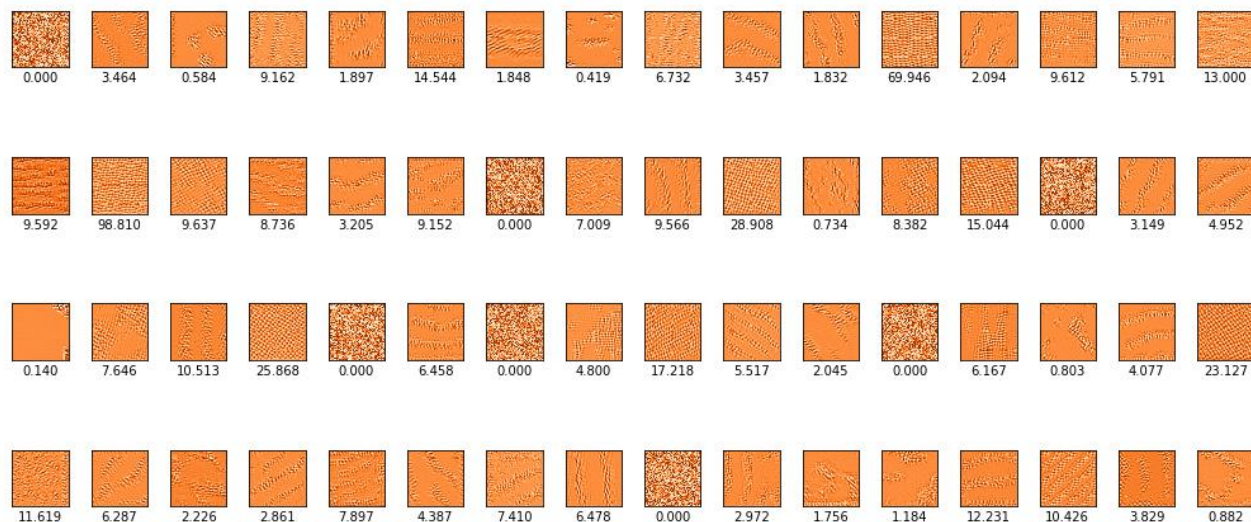
Train_set: #192，主要辨認出眼睛，鼻子和張大的嘴巴



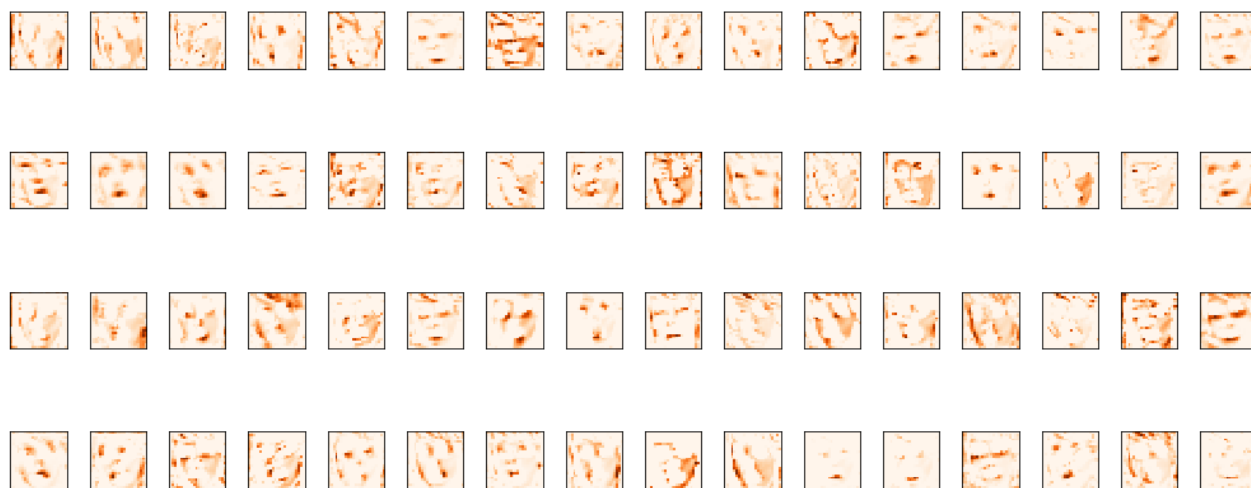
5.(1%) 承(1)(2)，利用上課所提到的 gradient ascent 方法，觀察特定層的 filter 最容易被哪種圖片 activate。
(Collaborators: R06942010 蘇建翰)

答：(不清楚為何編號是 85)，讀取第 11111 張照片。

Filters of layer conv2d_85



Output of layerconv2d_85 (Given image11111)



觀察第二列第二排被 activate 數值為 98.810 的，對應到的圖片便是很清晰的人臉驚慌的圖片，相對其他比較粗糙的圖形 activate 數值為 0 的，圖形就比較扭曲，看不太出形狀。