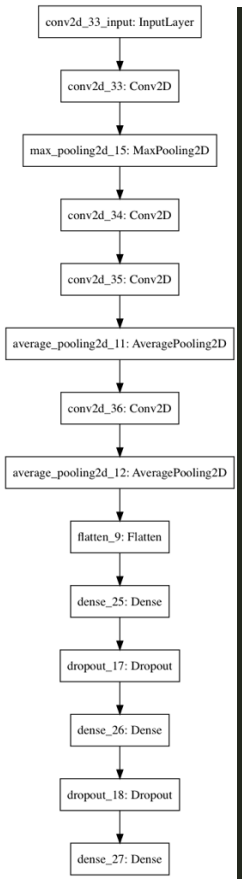
學號：R05922096 系級： 資工碩一 姓名：李哲安

1. Build Convolution Neural Network

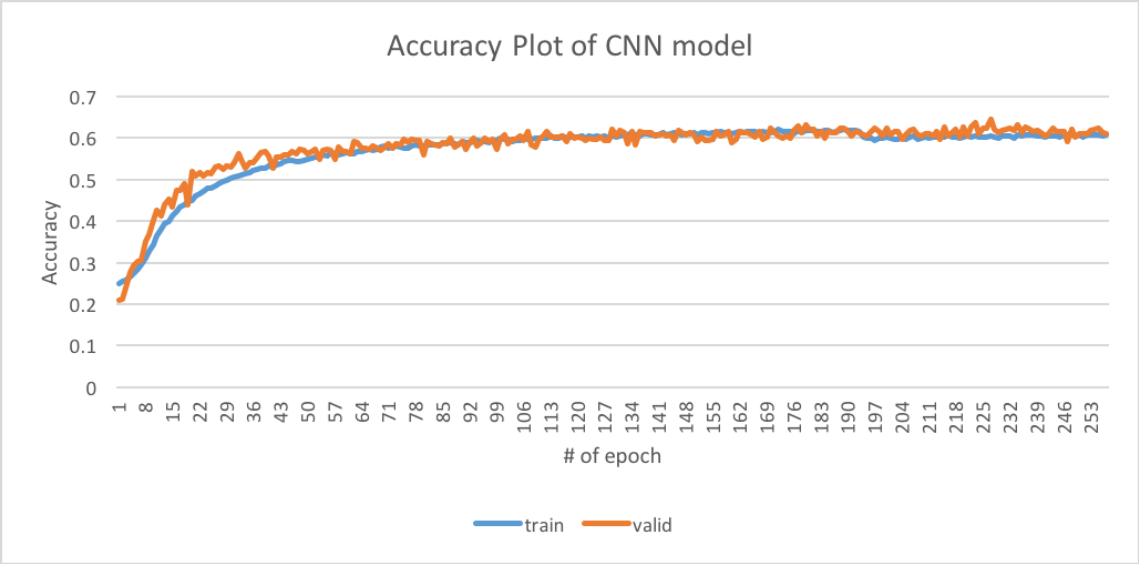
- tune performance
  Accuracy 0.638 on kaggle.
  在使用了 keras 的 ImageDataGenerator 來增加 training data 的數量後準確率有大幅的提升(0.59->0.63)。
- record model structure



Layer (type)	Output Shape	Param #
conv2d_33 (Conv2D)	(None, 48, 48, 64)	1664
max_pooling2d_15 (MaxPooling	(None, 24, 24, 64)	0
conv2d_34 (Conv2D)	(None, 24, 24, 64)	36928
conv2d_35 (Conv2D)	(None, 24, 24, 64)	36928
average_pooling2d_11 (Averag	(None, 12, 12, 64)	0
conv2d_36 (Conv2D)	(None, 12, 12, 128)	73856
average_pooling2d_12 (Averag	(None, 6, 6, 128)	0
flatten_9 (Flatten)	(None, 4608)	0
dense_25 (Dense)	(None, 256)	1179904
dropout_17 (Dropout)	(None, 256)	0
dense_26 (Dense)	(None, 256)	65792
dropout_18 (Dropout)	(None, 256)	0
dense_27 (Dense)	(None, 7)	1799

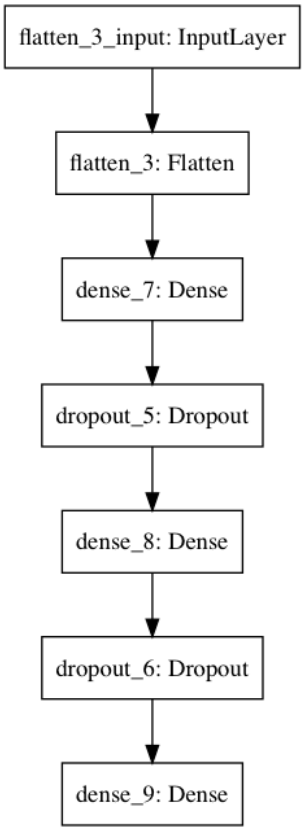
Total params: 1,396,871.0
Trainable params: 1,396,871.0
Non-trainable params: 0.0

- record training procedure



2. Build Deep Neural Network

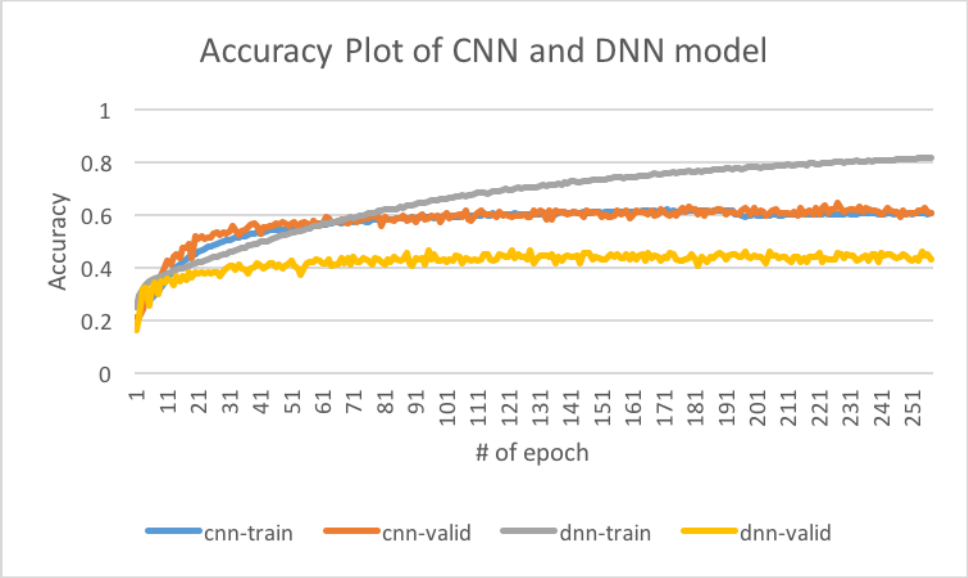
- use the same number of parameters as above CNN, build DNN model to do the same task
  CNN parsms:1,396,871 v.s. DNN parsms:1,446,407
- record model structure



Layer (type)	Output Shape	Param #
flatten_3 (Flatten)	(None, 2304)	0
dense_7 (Dense)	(None, 512)	1180160
dropout_5 (Dropout)	(None, 512)	0
dense_8 (Dense)	(None, 512)	262656
dropout_6 (Dropout)	(None, 512)	0
dense_9 (Dense)	(None, 7)	3591

Total params: 1,446,407.0
Trainable params: 1,446,407.0
Non-trainable params: 0.0

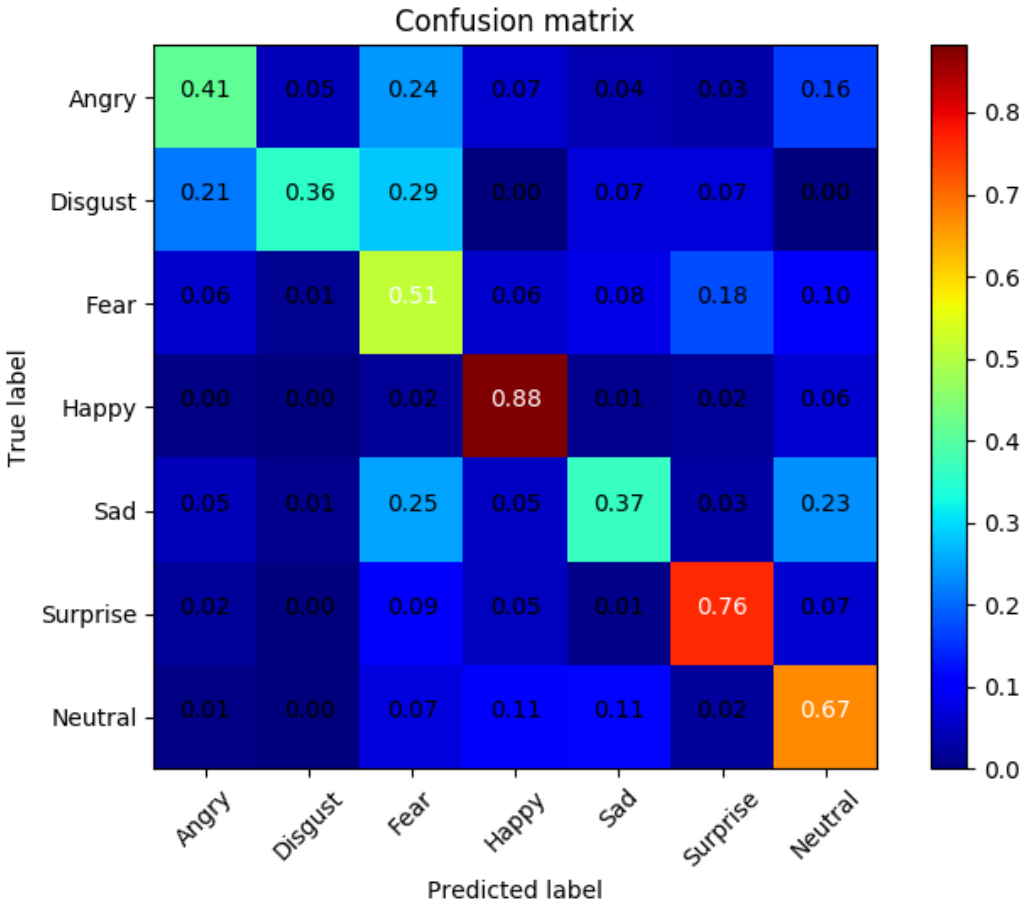
- record training procedure



- describe what you observed
  在差不多參數量且 epoch 一樣時，CNN 相較於 DNN 在 validation set 上有較好的 performance。

3. Analyze the Model by Confusion Matrix

- put the prediction and true label in confusion matrix of your split validation data

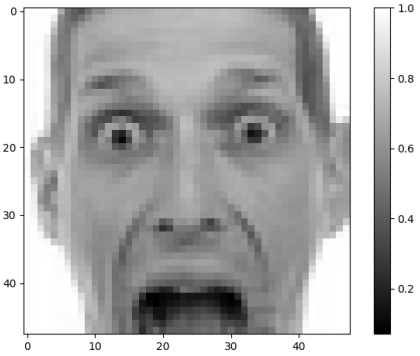


- describe what you observed

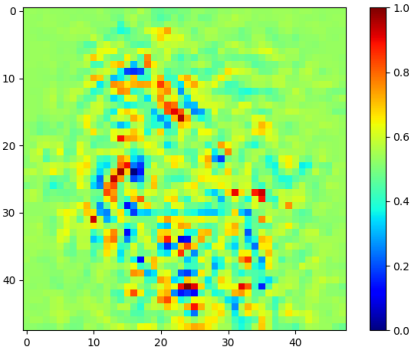
  由 Confusion Matrix 中可以發現 Happy 是最容易辨識且不易混淆的，Disgust 則很容易跟 Fear 和 Angry 混淆。

4. Analyze the Model by Plotting the Saliency Map

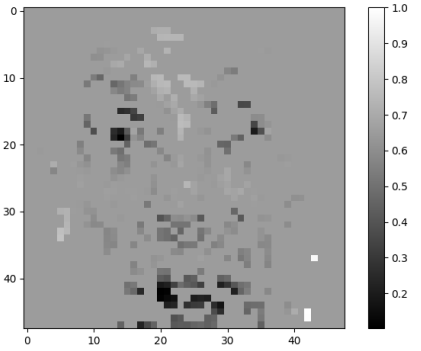
- Given an image and its corresponding class, we would like to rank the pixels of original image based on their influence on the distribution of final



<<< 原圖 (Class = Fear)



<<< Saliency Map

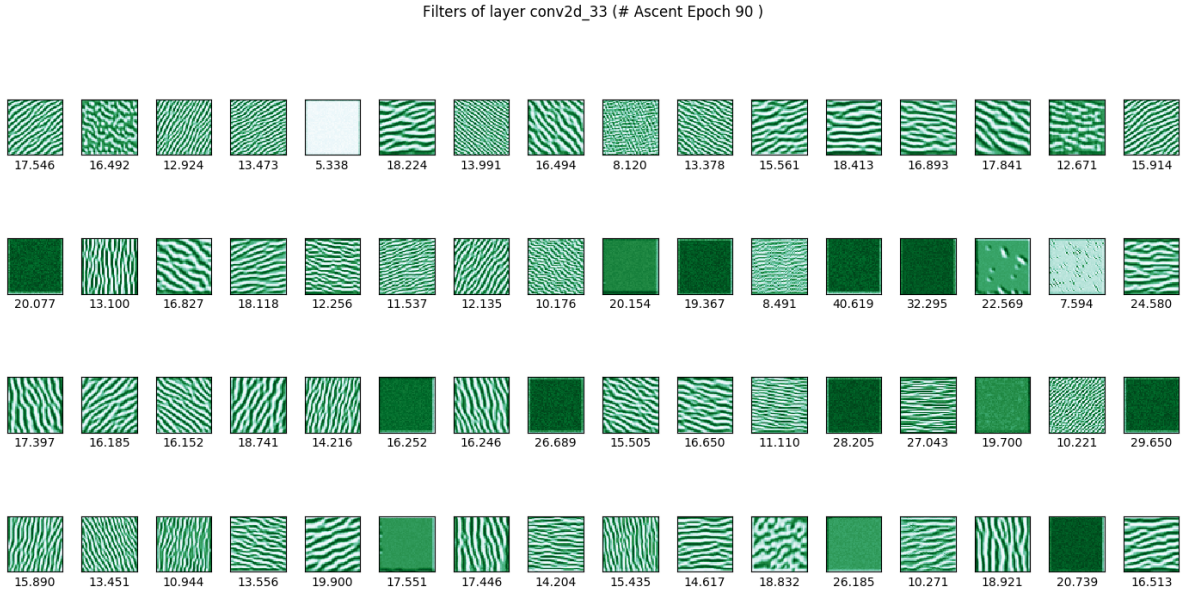


<<< Mask (Threshold = 0.6)

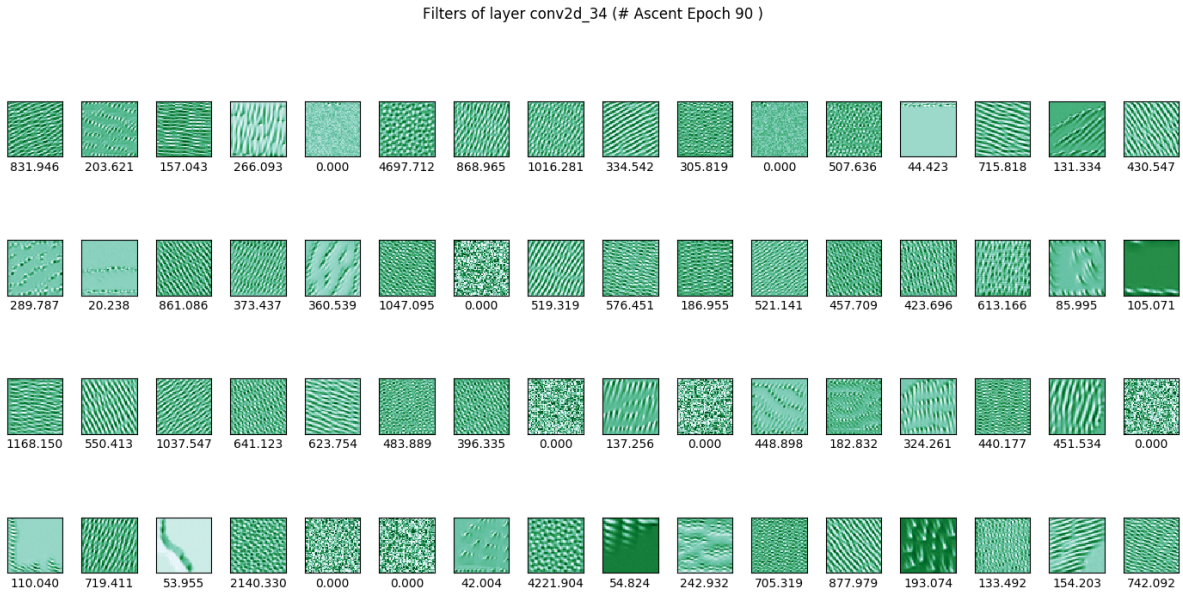
由 saliency map 中可以觀察到 activate 最強的部分在人臉部的五官部份，尤其是眼睛、嘴巴、臉頰的部分最為明顯，為 model 判斷表情最重要的區域。

5. Analyze the Model by Visualizing Filters

- use Gradient Ascent method mentioned in class to find the image that activates the selected filter the most and plot them (start from white noise).

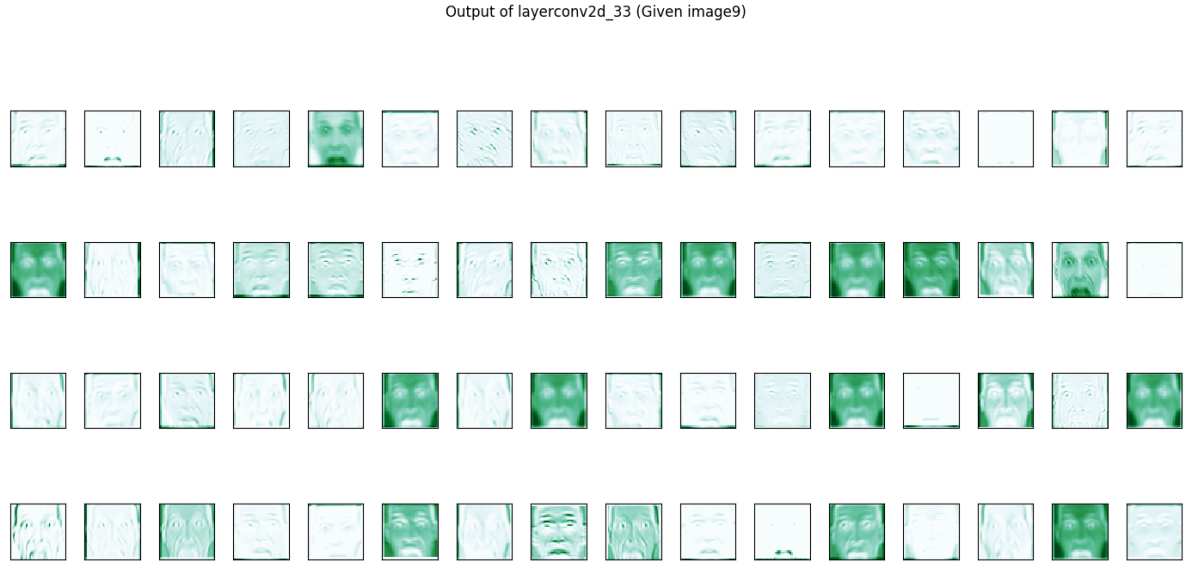


上圖為第一層 conv 的 64 個 filter 分別做 Gradient Ascent (epoch = 90)後找出的圖片。

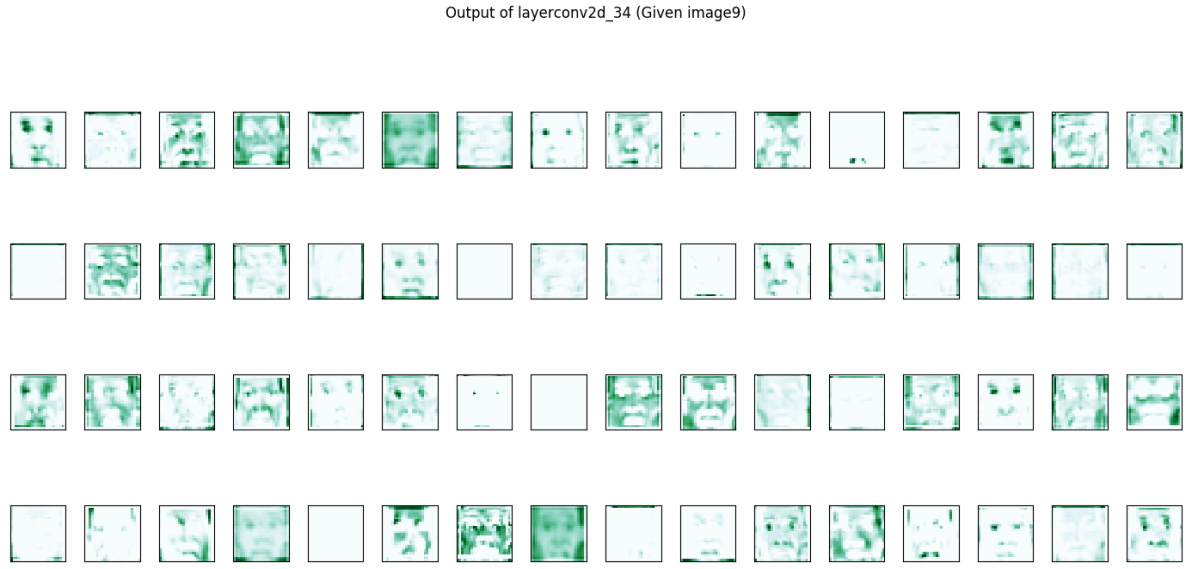


上圖為第二層 conv 的 64 個 filter 分別做 Gradient Ascent (epoch = 90)後找出的圖片。

- Feed an image in your validation set to the model, and plot the output of that filter



上圖為第 9 張(index from 0,同上題圖)圖片對於第一層 conv 的 64 個 filter 的 output。



上圖為第 9 張(index from 0,同上題圖)圖片對於第二層 conv 的 64 個 filter 的 output。

- Describe what you observed, and explain it
  由 gradient ascent 後的 filter 圖可以找出能讓每一個 filter 分別最 activate 的圖片來，也就是 filter 的特徵。而將圖片丟入後可則可以由 filter 的 output 來分析出每張圖片的哪些部位對於不同的 filter 重要程度的分布情況。