

學號：B05902043 系級：資工二 姓名：劉鴻慶

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

### 資料預處理:

1.data normalization : 實作見 Q2

2.splitting validation set :

隨機將 10%的資料分割出來當作 validation set，剩下的 90%為 training set

3.data augmentation : 實作見 Q2

### 模型架構:

如右圖，首先有四層 convolution，一層的架構如下

1.Conv2D() : filter size=(3,3), activation=linear, padding=same

2.LeakyReLU() : alpha=0.05

3.BatchNormalization() :

4.MaxPooling2D() : pool size=(2,2), padding=same

5.Dropout() : 四層的 dropout rate 皆為 0.2

接上攤平層

21.Flatten():攤平

接上 Dense 層

22.Dense() : activation function = linear

23.LeakyReLU() : alpha=0.05

24.BatchNormalization() :

25.Dropout(): rate = 0.2

26.Dense() : activation function = softmax

model 使用的 loss function 為 cross entropy,

optimizer 為 Adam

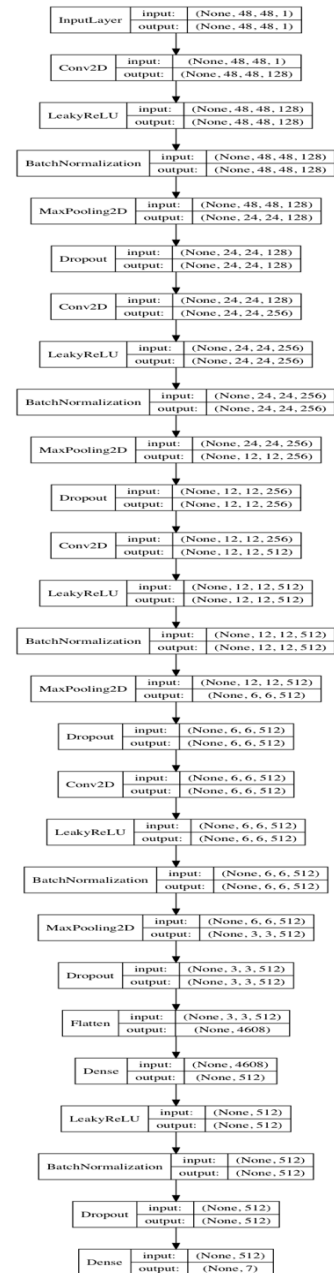
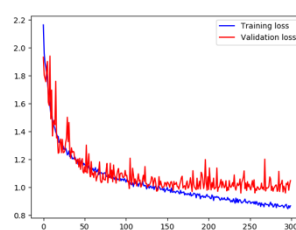
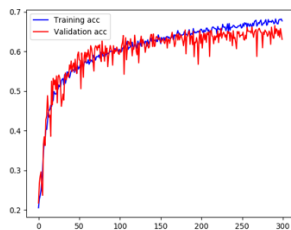
收斂過程: (如下圖)

在 300 個 epoch 後，validation acc 大概收斂到

0.64，上傳後的 public kaggle score:0.68124 我覺得要

過 strong baseline 不夠穩，用 ensemble 的方法來穩

定 model 的表現



### Ensemble model:

kaggle 上過 strong baseline 的 model 是使用 ensemble 的方法，我用上面的架構訓練出三個類似的 model，利用 kaggle 的 model ensemble 和 average 將三個 model 的 output 平均，並選出七個 label 中最高機率的當作 output 的 label

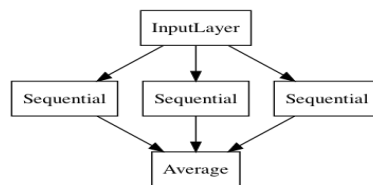


fig. ensemble model

準確率： Final kaggle score: 0.69601/ 0.68932 (pubic/private)

2. (1%) 請嘗試 **data normalization**, **data augmentation**,說明實行方法並且說明對準確率有什麼樣的影響？

**data normalization:**

實作方法：

將在所有圖片裡的特定位置的單個 pixel 取 mean 和 std，讓每個 pixel = pixel / std - mean 來達到 normalization

準確率：

|              | Before data normalization | After data normalization |
|--------------|---------------------------|--------------------------|
| Kaggle score | 0.66425/0.64446           | 0.68124/0.68013          |

**data augmentation:**

實作方法:

透過 keras 的 image generator 來實作 data augmentation，將圖片旋轉，平移，縮放，錯切變換，水平翻轉，來達到增廣 training set 的結果，由於人的表情有方向性所以沒有考慮垂直翻轉。(Generator 的參數如下)

(rotation\_range=30, width\_shift\_range=0.2, height\_shift\_range=0.2, zoom\_range=[0.8, 1.2], shear\_range=0.2, horizontal\_flip=True)

再使用 fit\_generator 來 fit model，產生原本 training set 5 倍的 input

準確率：

|              | Before data augmentation | After data augmentation |
|--------------|--------------------------|-------------------------|
| Kaggle score | 0.61465/0.61604          | 0.68124 / 0.68013       |

由上面兩個結果可知 data normalization 和 data augmentation 都是可以讓 model 的預測結果更加進步的 preprocessing 方法

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：

利用 validation set 來繪製 confusion matrix，可以發現我的 model 再預測 fear 的準確率最差，只有 57% 的圖片會預測正確，其中又以錯分到 sad, angry 為大宗。Sad 也很容易和 neutral 搞混，有 18% 的 sad 被分到 neutral。細想倒也合理，眾多負面情緒通常會混在一起出現連人都難以區別。

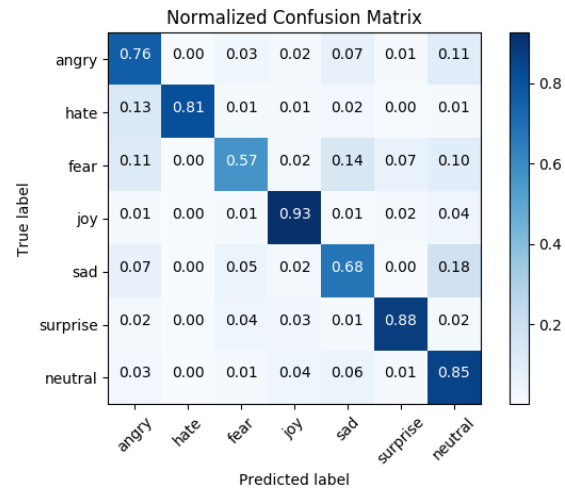
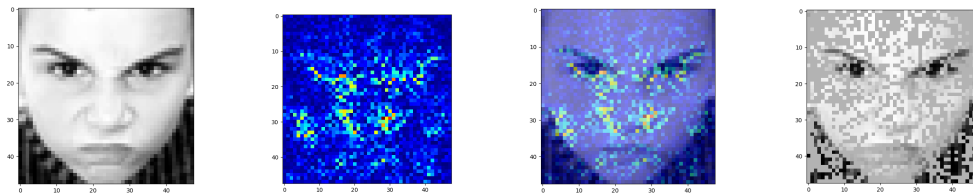


fig. confusion matrix

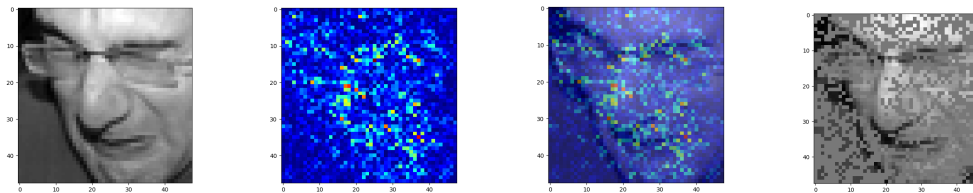
4. (1%) 使用 CNN 的確有些好處，試繪出其 **saliency maps**，觀察模型在做 **classification** 時，是 **focus** 在圖片的哪些部份？

從 validation set 選出每個 class 預測正確的圖片，利用 keras-vis 的 visualize saliency 來觀察每個圖片的 saliency map，由左到用依序為原圖、heat map、overlayed 和 masked

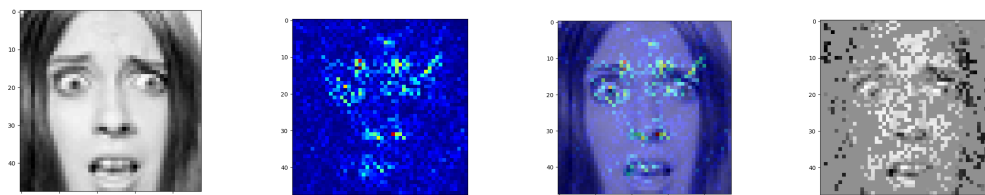
[angry]



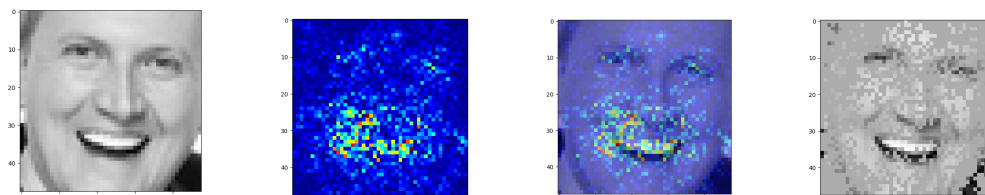
[hate]



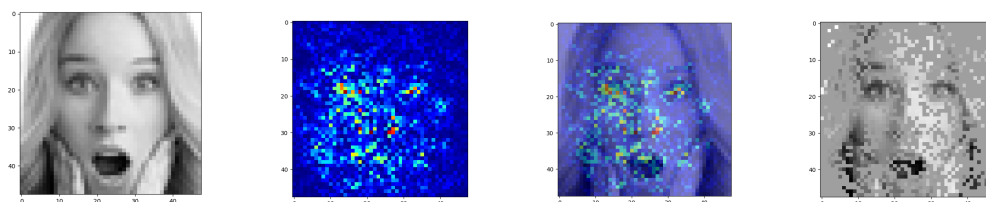
[fear]



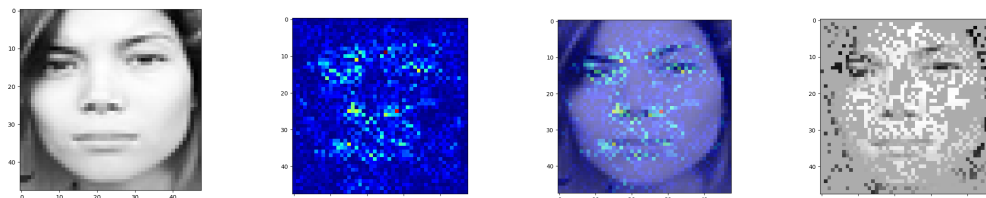
[joy]



[surprise]



[neutral]



由上面幾個圖可以觀察到 **activation** 主要集中在眼睛（眉毛）、鼻子、嘴巴等五官部位，利於判斷情緒。比較特別的是觀察 joy 的圖片，**activation** 有集中在牙齒的狀況，其他 labeled joy 的圖片也有類似的現象，model 可能學到將牙齒得出現作為判斷 joy label 的依據

5. (1%)承(4) 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** 與觀察 **filter** 的 **output**。

答：

以下左邊第一層到第三層 convolutional layer 過 activation function 的一些 **filters**，透過 **gradient ascent** 的方法畫出最大化 **filter output** 的圖片，可以發現越高層的 **filters** 有越複雜的紋理，在第三層甚至可以看到一些很像嘴巴的弧形，像牙齒的方格，也可以看到一些沒有收斂的雜訊。而右邊則是我將上面那張 fear class 的圖經過 **filters** 得到的 **output**，**filters** 已經可以抓到圖片的五官(眉毛，眼睛，鼻子，嘴巴等)辨認情緒的重點部位。

