

Homework 3 Report - Image Sentiment Classification

學號：b05902013 系級：資工二 姓名：吳宗翰

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？

模型架構

```
### model 1 -> 6 CNN + 2 Dense ###

Input (48, 48, 1)

Conv(16, 3, 3) + BN + LeakyRelu
Conv(32, 3, 3) + Gaussian noise(0.1) + BN + LeakyRelu
Conv(64, 3, 3) + BN + LeakyRelu + Maxpooling(2, 2) + Dropout(0.1)
Conv(128, 3, 3) + BN + LeakyRelu + Maxpooling(2, 2) + Dropout(0.2)
Conv(256, 3, 3) + BN + LeakyRelu + Maxpooling(2, 2) + Dropout(0.2)
Conv(512, 3, 3) + BN + LeakyRelu + Maxpooling(2, 2) + Dropout(0.2)

Flatten()
Dense(512) + BN + LeakyRelu + Dropout(0.5)
Dense(256) + LeakyRelu

Output (7)

### model 2 ###
修改model 1, 拿掉Gaussian noise

### model 3 ###
修改model 1, Dense第二層改成512個neuron

### model 4 ###
修改model 1, activation function全部換成selu, 拿掉BN

### Ensemble ###
average 全部的y再做argmax即得到答案
```

訓練參數與預處理

hyper parameter

1. Optimizer: Adam, learning rate = 10^{-3} , decay: 5×10^{-6}
2. iteration: 幾個model從200至900不等 (但是其實200以後就差不多收斂了)
3. batch size: 128
4. 其他參數：BN momentum 0.5，其他均使用keras default

data preprocessing

1. feature scaling (standardization)
2. data augmentation
 - 每100個epoch做一次augmentation
 - 隨機旋轉30度、上下&左右shift0.2、zoom:0.1、shear_range:0.1
 - 有隨機水平翻轉，但是沒有垂直翻轉

準確率

Public Score	Private Score
0.71858	0.70855

備註：單用model 1其實就已經讓accuracy達到0.69以上了，但是不可否認的是ensemble真的幫助不少

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？

實作過程

1. Augmentation :
 - 每100個epoch做一次augmentation
 - 隨機旋轉30度、上下&左右shift0.2、zoom:0.1、shear_range:0.1
 - 有隨機水平翻轉，但是沒有垂直翻轉
2. Normalization : 將training data做standardize

Result

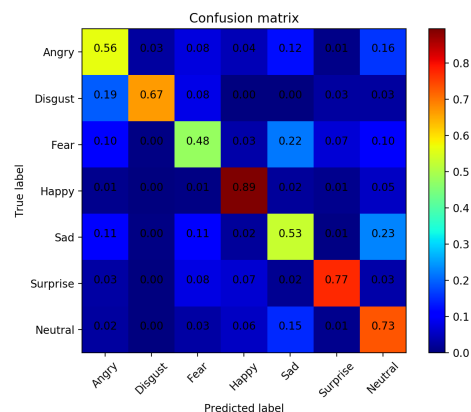
method	Public Score	Private Score
None	0.62524	0.61186
augmentation	0.70381	0.67929
normalization	0.62830	0.62607
augmentation + normalization	0.69378	0.68514

ps. 實驗相關參數如下： epoch: 300, optimizer: Adam(learning rate= 10^{-3} , decay= 5×10^{-6})

結論

1. 有沒有normalize似乎對結果沒差多少
2. 有沒有data augmentation對結果差很多，這是因為augmentation可以避免在training data上overfitting(在training的過程中就有發現 Acc_{train} 已經接近1了但是 Acc_{val} 卻升不下來)。因此在這次作業中我也採取了多個epoch就augment一次的策略。

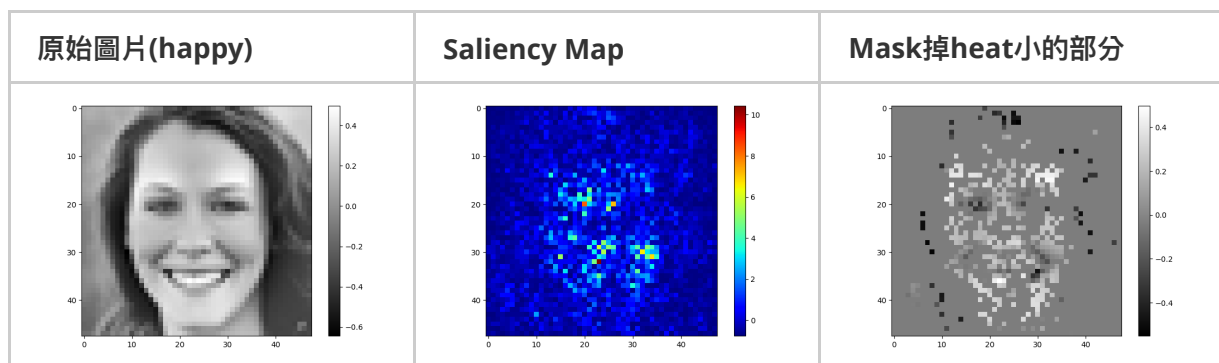
3. (1%) 觀察答錯的圖片中，哪些 class 彼此間容易用混？[繪出 confusion matrix 分析]



觀察結果

1. Sad和Fear之間最容易弄混，而的確在人類認知中兩者都是偏負向的有點難區分
2. 觀察到Happy最不容易搞混，除了本身跟另外幾個表情差異很大以外，我覺得跟dataset中happy的比例特別高也有關係

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？

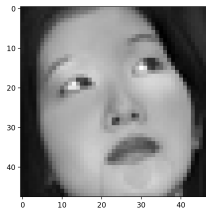


觀察結果

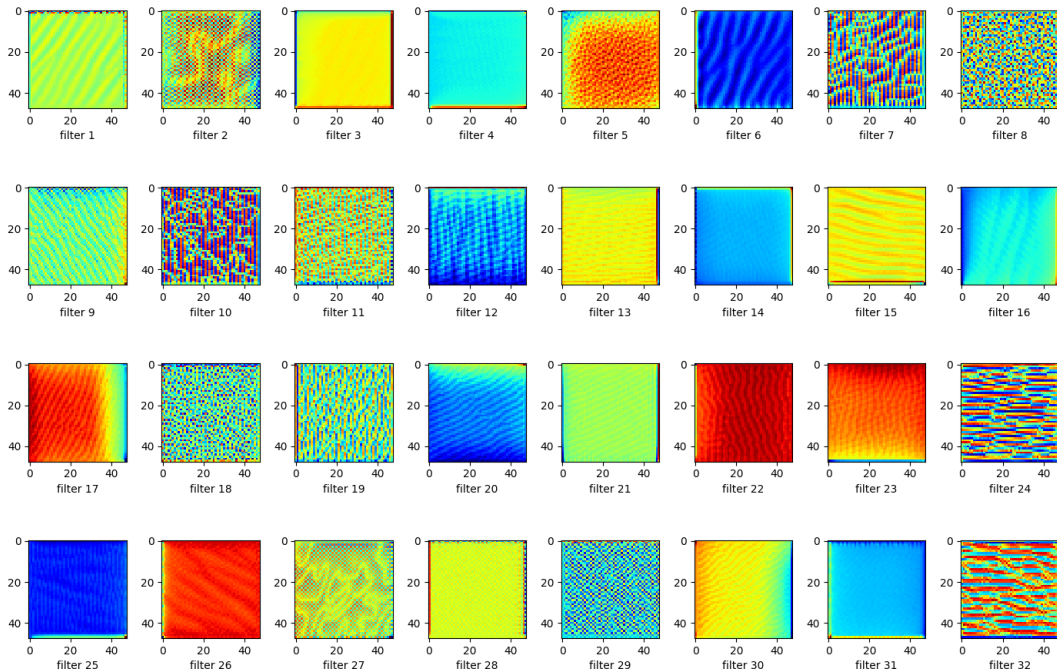
CNN都focus在眼睛、鼻子以及嘴巴處，而這也蠻符合直覺得，因為平常我們人類在觀察一個人的表情的時候確實大多只會關注五官，而不會是臉部肌膚

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。

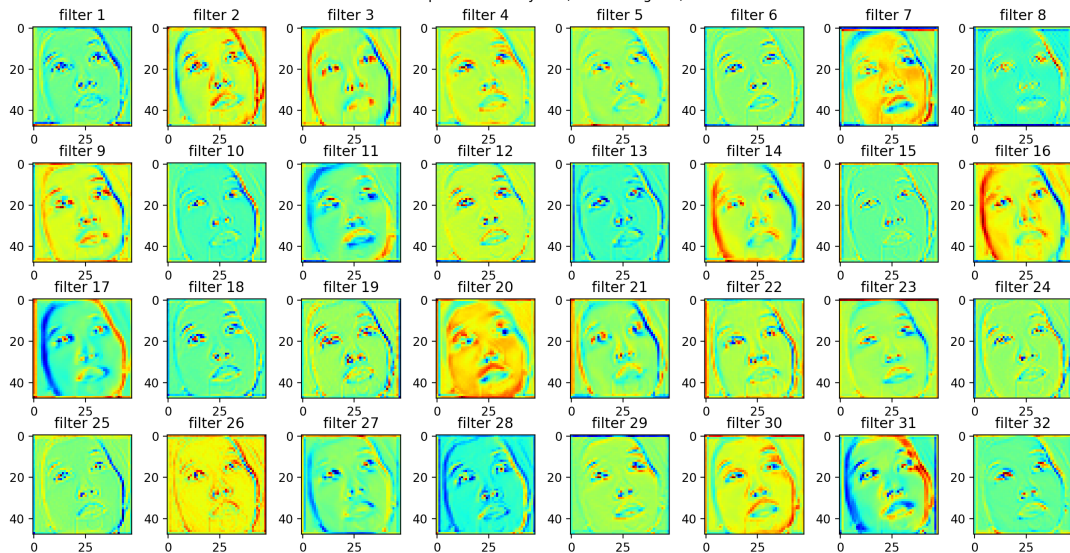
在這題中，我們將會visualize在第一題提到model 1的第二層Convolution Layer，並且使用training data的第17張image，觀察其output的結果。



Filters of Conv layer 2 (asscent 50 epoch)



Output of Conv layer2 (Given image17)



觀察結果

1. 看起來Layer 2的filter蠻粗糙的，也就只是特定某種線條或者是某種材質而已
2. 從Output我們也可以看到顏色變化比較大的部分都在臉的輪廓，跟第一點claim的算是有呼應到