Homework 3 - Image Sentiment Classification

學號：b05901033 系級： 電機二 姓名：莊永松

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 使用類VGG架構，考量到此次圖片大小只有48*48，不需要像真的VGG一樣多層

Conv層有使用padding，所有filter大小皆為3*3，filter數量64-64-128-128-128-128

在大部分的Conv與Activation之間有做Batch Normalization(除了第二個block外，原因:try過發現會比較高分&GPU記憶體太小，全部都塞BN算到一半很容易chuck，拿掉兩層比較安全)

flatten後架三層full-connected層，也有做Batch Normalization

另外在maxpooling完，以及後面full-connected層，都有加上dropout避免overfitting

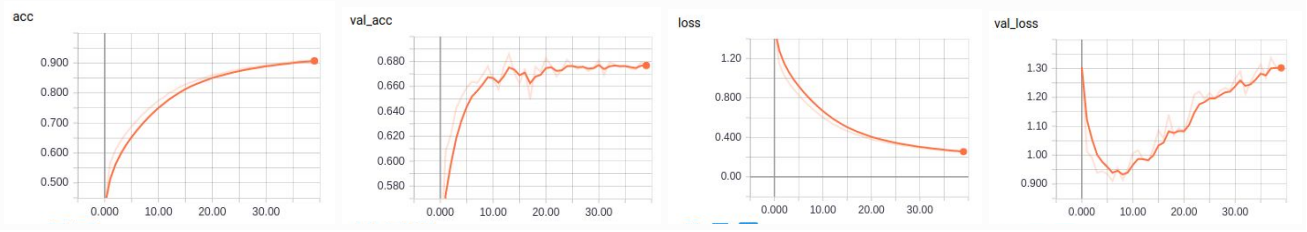
[Input Layer]
Conv2D(64, (3, 3))
[BatchNormalization]+Activation('relu')
Conv2D(64, (3, 3))
[BatchNormalization]+Activation('relu')
MaxPooling2D(pool_size=(2, 2))
Dropout(0.25)
Conv2D(128, (3, 3))+Activation('relu')
Conv2D(128, (3, 3))+Activation('relu')
MaxPooling2D(pool_size=(2, 2))
Dropout(0.25)
Conv2D(128, (3, 3))
[BatchNormalization]+Activation('relu')
Conv2D(128, (3, 3))
[BatchNormalization]+Activation('relu')
MaxPooling2D(pool_size=(2, 2))
Dropout(0.25)
[Flatten]
Dense(512)
[BatchNormalization]+Activation('relu')
Dropout(0.333)
Dense(512)
[BatchNormalization]+Activation('relu')
Dropout(0.333)
Dense(7)+Activation('softmax')

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 48, 48, 64)	640
batch_normalization_1 (Batch	(None, 48, 48, 64)	256
activation_1 (Activation)	(None, 48, 48, 64)	0
conv2d_2 (Conv2D)	(None, 46, 46, 64)	36928
batch_normalization_2 (Batch	(None, 46, 46, 64)	256
activation_2 (Activation)	(None, 46, 46, 64)	0
max_pooling2d_1 (MaxPooling2	(None, 23, 23, 64)	0
dropout_1 (Dropout)	(None, 23, 23, 64)	0
conv2d_3 (Conv2D)	(None, 23, 23, 128)	73856
activation_3 (Activation)	(None, 23, 23, 128)	0
conv2d_4 (Conv2D)	(None, 21, 21, 128)	147584
activation_4 (Activation)	(None, 21, 21, 128)	0
max_pooling2d_2 (MaxPooling2	(None, 10, 10, 128)	0
dropout_2 (Dropout)	(None, 10, 10, 128)	0
conv2d_5 (Conv2D)	(None, 10, 10, 128)	147584
batch_normalization_3 (Batch	(None, 10, 10, 128)	512
activation_5 (Activation)	(None, 10, 10, 128)	0
conv2d_6 (Conv2D)	(None, 8, 8, 128)	147584
batch_normalization_4 (Batch	(None, 8, 8, 128)	512
activation_6 (Activation)	(None, 8, 8, 128)	0
max_pooling2d_3 (MaxPooling2	(None, 4, 4, 128)	0
dropout_3 (Dropout)	(None, 4, 4, 128)	0
flatten_1 (Flatten)	(None, 2048)	0
dense_1 (Dense)	(None, 512)	1049088
batch_normalization_5 (Batch	(None, 512)	2048
activation_7 (Activation)	(None, 512)	0
dropout_4 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 512)	262656
batch_normalization_6 (Batch	(None, 512)	2048
activation_8 (Activation)	(None, 512)	0
dropout_5 (Dropout)	(None, 512)	0
dense_3 (Dense)	(None, 7)	3591
activation_9 (Activation)	(None, 7)	0

Total params: 1,875,143
Trainable params: 1,872,327
Non-trainable params: 2,816

訓練參數 初始化使用Xavier uniform initializer，batch size設256，epoch數設40，另有使用Early Stopping，訓練過程如下(此圖為了完整呈現40 epoch，先關掉Early Stopping)*淺色為沒平滑化的曲線*

training acc　　validation acc　　training loss　　validation loss



可看到val_loss後期雖然有上升，但val_acc也有增長且趨穩定，推測是有做augmentation的緣故

準確率 在kaggle上public score為0.68013，private score為0.67818，後來將我所訓練出最好的三個model做ensemble後，在kaggle上public score為0.71301，private score最佳則為0.70911(是另一組ensemble組合)

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 針對dataset中所有image算出統一的mean及variance，做標準化。對準確率有不錯的幫助，提升了1~2%的準確率。另外我在未做normalize之下訓練結果並沒有下降的很嚴重，推測原因是我的model結構有batch normalize層，所以挽救到一些沒做normalize的缺陷。

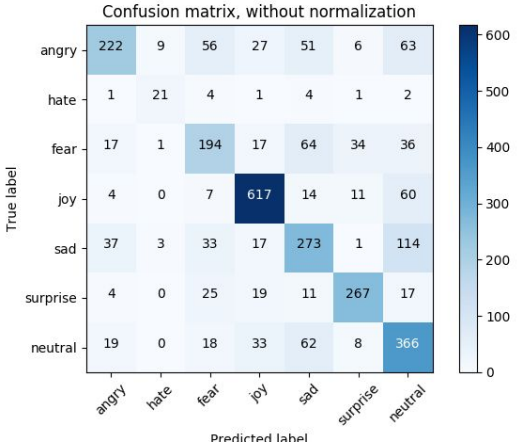
在kaggle上成績如下：	public score	private score
有做 normalization	0.65923	0.65059
未做 normalization	0.63889	0.64781

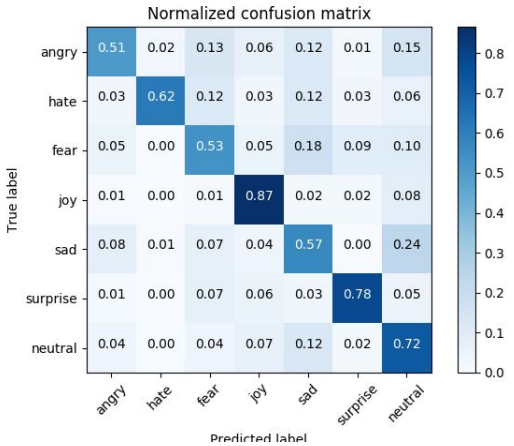
Data augmentation 我是使用keras內建的ImageDataGenerator，有作rotation、shear、zoom、horizontal_flip，沒做shift(因為CNN的結構感覺並不需要shifted的資料)，我將每一張圖generate出五張新的圖，再丟進去訓練。另外這次測試使用的normalization是統一使用0-1 mapping。可以看到做了augmentation後，準確率大幅提升了5%左右，效果十分顯著。

在kaggle上成績如下：	public score	private score
有做 augmentation	0.67483	0.67288
未做 augmentation	0.63388	0.62803

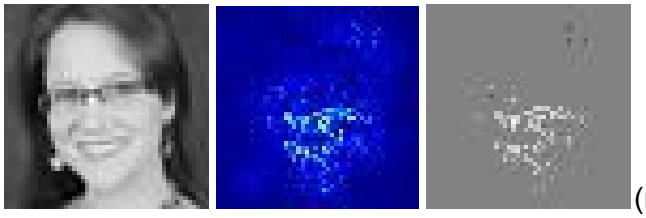
3. (1%) 觀察答錯的圖片中，哪些 class 彼此間容易用混？[繪出 confusion matrix 分析]

可觀察到最容易互相弄混的class是sad跟neutual(sad有24%被分到neutual，neutual有12%被分到sad)。其次fear跟sad也有互相搞混的現象(18%<=>7%)，另外angry的錯誤率最高，容易被分到fear、sad、neutral等class中，而joy則是分類效果最好的class(因data中數量最多)。



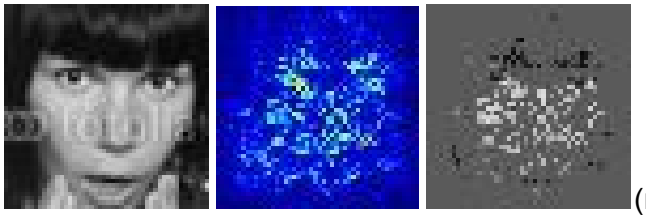


4. (1%) 從(1)(2)可以發現，使用 CNN 的確有些好處，試繪出其 saliency maps，觀察模型在做 classification 時，是 focus 在圖片的哪些部份？



(mask的threshold=0.7)

這張圖是joy，機器便比較focus在嘴型上揚的部分，heatmap在嘴巴附近有較高的值。



(mask的threshold=0.7)

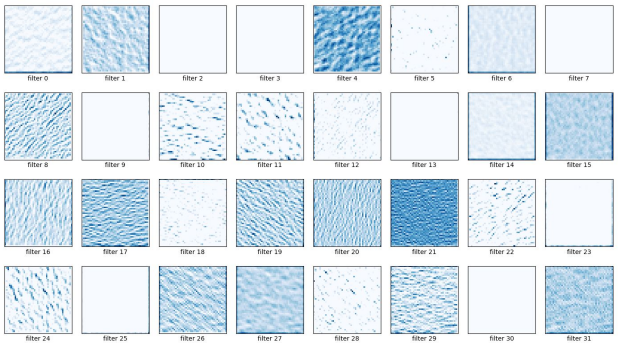
這張是surprise，眼睛張大非常明顯易判斷，heatmap在雙眼附近有較高的值。

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。

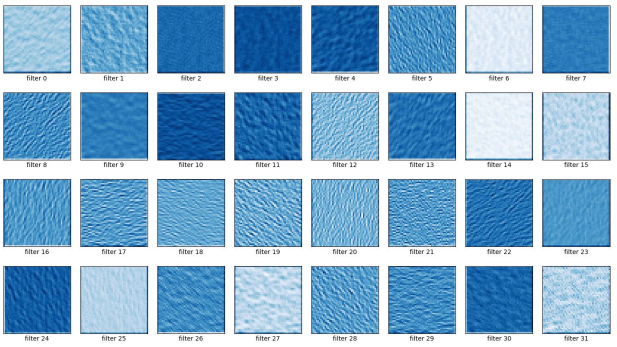
Input 放 white noise => `img = np.random.random((1, 48, 48, 1)) * 20 + 128`

，觀察第二層relu層及第二層conv層的 gradient ascent 結果(放前32張filter)。

可以觀察到大部分filter是佈滿曲線紋路，較可能是用來判斷臉部的輪廓。其中也有些filter是有佈滿著"坑洞"，較可能是用來判斷人臉的五官，如鼻孔眼睛耳朵等。



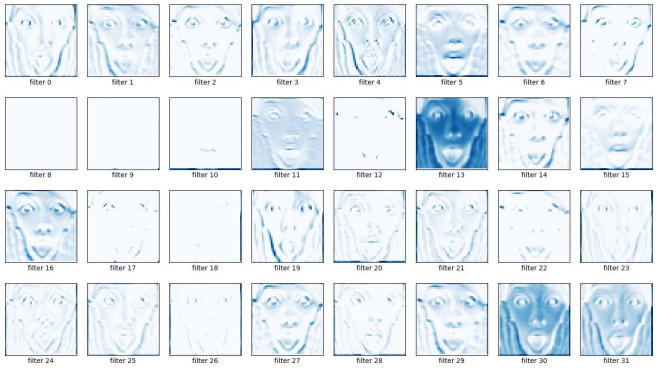
activation_2(ReLU)



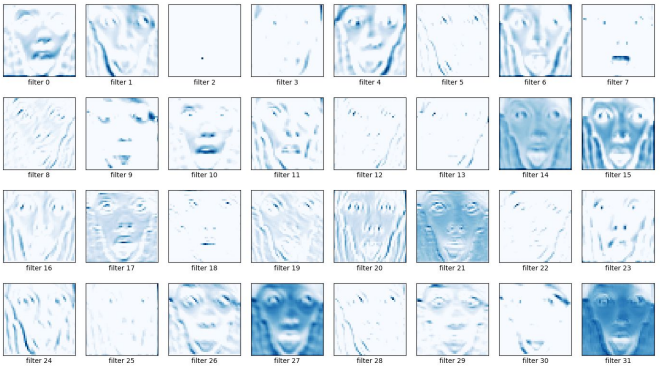
Conv_2

Input 放真實照片(如右圖)，觀察前兩層relu層及conv層的 gradient ascent 結果(取前32個filter)。

不管是conv層或relu層的結果，都可以觀察到filter有針對臉部的特徵做了一些轉化，像是將臉部的邊緣加強，有的則有強調眼睛的部分(感覺眼睛形狀被改變)，也有少數filter是被activate成幾乎一片空白，推測可能是filter數量太多，因此該filter沒有學到東西，或者是針對這張input的圖沒有可以activate的東西。

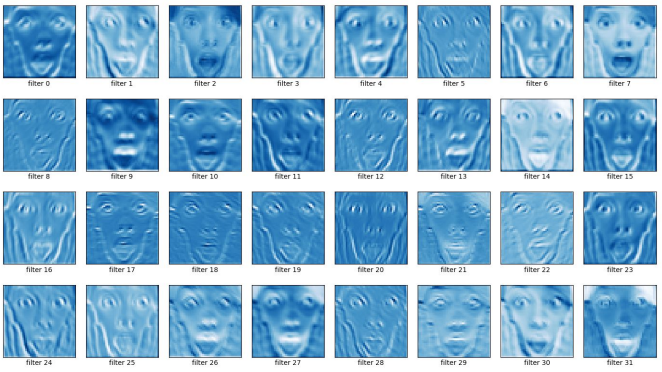


activation_1(ReLU)



activation_2(ReLU)

conv2d_1



conv2d_2