

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？

(Collaborators:)

模型架構:

採用一個挺大的架構，基本上分成四組，每組用 3 層 convolutin2d 再加上 batch normalization，接上 maxpooling，並用 relu 作為 actiovation。每組固定使用一個數量的 filter，第一組 64，第二組 128，第三組 256，第四組 512。Kernalize size 皆使用(3,3)，最後在接上兩個 dense(2048)以及最後的 output dense(7)。

。loss 採用的是 cross entropy，而 optimizer 則是 adam。Epoch 設為 256

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 48, 48, 64)	640
batch_normalization_1 (Batch Normalization)	(None, 48, 48, 64)	256
activation_1 (Activation)	(None, 48, 48, 64)	0
conv2d_2 (Conv2D)	(None, 48, 48, 64)	36928
batch_normalization_2 (Batch Normalization)	(None, 48, 48, 64)	256
activation_2 (Activation)	(None, 48, 48, 64)	0
conv2d_3 (Conv2D)	(None, 48, 48, 64)	36928
batch_normalization_3 (Batch Normalization)	(None, 48, 48, 64)	256
activation_3 (Activation)	(None, 48, 48, 64)	0
max_pooling2d_1 (MaxPooling2D)	(None, 24, 24, 64)	0
conv2d_4 (Conv2D)	(None, 24, 24, 128)	73856
batch_normalization_4 (Batch Normalization)	(None, 24, 24, 128)	512
activation_4 (Activation)	(None, 24, 24, 128)	0
conv2d_5 (Conv2D)	(None, 24, 24, 128)	147584
batch_normalization_5 (Batch Normalization)	(None, 24, 24, 128)	512
activation_5 (Activation)	(None, 24, 24, 128)	0
conv2d_6 (Conv2D)	(None, 24, 24, 128)	147584

batch_normalization_6	(Batch Normalization)	(None, 24, 24, 128)	512
activation_6	(Activation)	(None, 24, 24, 128)	0
max_pooling2d_2	(MaxPooling2D)	(None, 12, 12, 128)	0
conv2d_7	(Conv2D)	(None, 12, 12, 256)	295168
batch_normalization_7	(Batch Normalization)	(None, 12, 12, 256)	1024
activation_7	(Activation)	(None, 12, 12, 256)	0
conv2d_8	(Conv2D)	(None, 12, 12, 256)	590080
batch_normalization_8	(Batch Normalization)	(None, 12, 12, 256)	1024
activation_8	(Activation)	(None, 12, 12, 256)	0
batch_normalization_9	(Batch Normalization)	(None, 12, 12, 256)	1024
conv2d_9	(Conv2D)	(None, 12, 12, 256)	590080
activation_9	(Activation)	(None, 12, 12, 256)	0
max_pooling2d_3	(MaxPooling2D)	(None, 6, 6, 256)	0
conv2d_10	(Conv2D)	(None, 6, 6, 512)	1180160
batch_normalization_10	(Batch Normalization)	(None, 6, 6, 512)	2048
activation_10	(Activation)	(None, 6, 6, 512)	0
conv2d_11	(Conv2D)	(None, 6, 6, 512)	2359808
batch_normalization_11	(Batch Normalization)	(None, 6, 6, 512)	2048
activation_12	(Activation)	(None, 6, 6, 512)	0
max_pooling2d_4	(MaxPooling2D)	(None, 3, 3, 512)	0
flatten_1	(Flatten)	(None, 4608)	0
dense_1	(Dense)	(None, 2048)	9439232
batch_normalization_13	(Batch Normalization)	(None, 2048)	8192
dense_2	(Dense)	(None, 2048)	4196352
batch_normalization_14	(Batch Normalization)	(None, 2048)	8192
dense_3	(Dense)	(None, 7)	14343
activation_13	(Activation)	(None, 7)	0

Total params: 21,496,455

訓練準確度:valid_acc= 0.68

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？

(Collaborators:)

答：寫出實作 data normalization 過程、與實作前、實作後準確率。 -> 0.5 分

寫出實作 data augmentation 過程、與實作前、實作後準確率。 -> 0.5 分

data normalization:真的每個 pixel 進行 normalization，將 data 減去平均再除以標準差。
使用相通的 model structure，Not normalization 只將每筆 data 除以 255

	Kaggle		Valid_acc
	Public	Private	
Normalize	0.67344	0.66285	0.6713
Not normalize	0.68403	0.68960	0.6725

從 valid test 以及 kaggle 上看似乎差異不太大，。且 Not normalization 已經有稍微處理過 data，將所有資料除以 255 壓到 1~0 之間，資料變得更密集，效果看起來甚至比複雜的 normalizrtion 還要好。

data augmentation:

使用 keras 的 ImageDataGenerator，對資料進行平移（shift）、旋轉（rotate :30 度）、縮放（zoom :0.8~1.2）、翻轉（flip），以及推移（shear:0.2）

	Kaggle		Valid_acc
	Public	Private	
Augmentation	0.68626	0.6713	0.67
Not augmentation	0.64056	0.6525	0.65

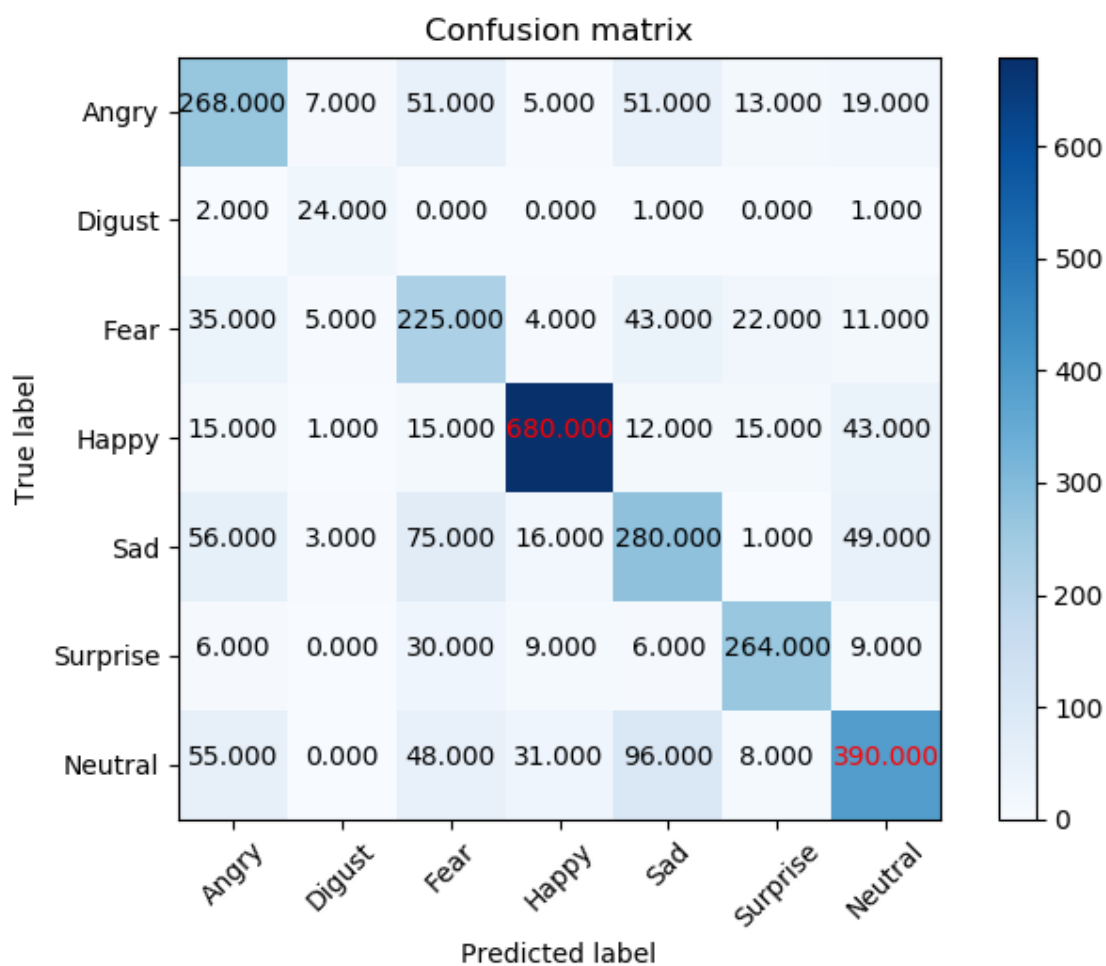
很明顯加上 data augmentation 後整個 model 的準確率大大的上升，抗噪度提高。

Not augmentation 在 valid 上達到 0.65，到 kaggle 上有小小的下降。使用 data augmentation 在 valid 上有所提升，到 kaggle 上表現也沒有下降，顯示出模型的穩固度上升

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]

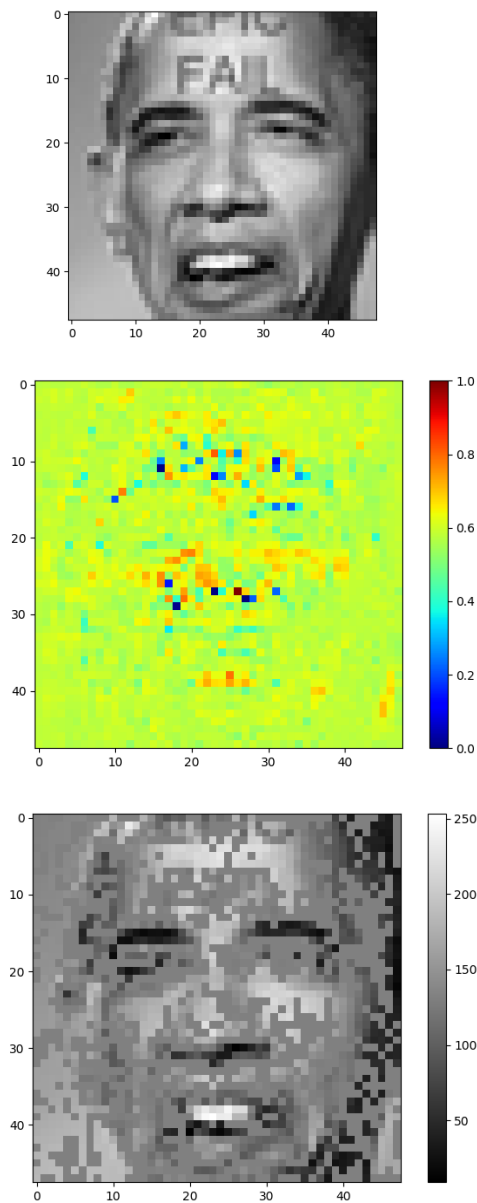
(Collaborators:)

答：



整體來說 sad 和 neutral 錯誤率相較其他來說較高，neutral 挺容易辨認承 sad，sad 則是會與 fear 混淆

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？



由圖片來看，鼻子周圍與臉頰肌肉還有額頭肌肉，是圖片的重點區域，嘴巴附近也是判斷的重點，眼睛反而不是個熱區。而下圖眼睛很明顯的 mask 掉，嘴巴臉頰卻保留了許多，推測這張圖臉頰與嘴巴是重要的 feature

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。(Collaborators:)

答：

(1)Activation2

(2) Activation2 out

