

Report

學號：B05902002 系級：資工二 姓名：李栢淵

(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？

Layer (type)	Output Shape	Connected to
input_1 (InputLayer)	(None, 1, 48, 48)	
conv2d_1 (Conv2D)	(None, 64, 24, 24)	input_1
conv2d_2 (Conv2D)	(None, 16, 24, 24)	conv2d_1
conv2d_3 (Conv2D)	(None, 16, 24, 24)	conv2d_2
conv2d_4 (Conv2D)	(None, 64, 24, 24)	conv2d_3
add_1 (Add)	(None, 64, 24, 24)	conv2d_4 conv2d_1
activation_1 (ReLu)	(None, 64, 24, 24)	add_1
conv2d_5 (Conv2D)	(None, 128, 12, 12)	activation_1
conv2d_6 (Conv2D)	(None, 128, 12, 12)	activation_1
conv2d_7 (Conv2D)	(None, 128, 12, 12)	conv2d_6
add_2 (Add)	(None, 128, 12, 12)	conv2d_5 conv2d_7
conv2d_8 (Conv2D)	(None, 32, 12, 12)	add_2
conv2d_9 (Conv2D)	(None, 32, 12, 12)	conv2d_8
conv2d_10 (Conv2D)	(None, 128, 12, 12)	conv2d_9
add_3 (Add)	(None, 128, 12, 12)	add_2 conv2d_10
activation_2 (ReLu)	(None, 128, 12, 12)	add_3
conv2d_12 (Conv2D)	(None, 256, 6, 6)	activation_2
conv2d_13 (Conv2D)	(None, 256, 6, 6)	conv2d_12
conv2d_11 (Conv2D)	(None, 256, 6, 6)	activation_2
add_4 (Add)	(None, 256, 6, 6)	conv2d_11 conv2d_13
conv2d_14 (Conv2D)	(None, 64, 6, 6)	add_4
conv2d_15 (Conv2D)	(None, 64, 6, 6)	conv2d_14
conv2d_16 (Conv2D)	(None, 256, 6, 6)	conv2d_15
add_5 (Add)	(None, 256, 6, 6)	add_4 conv2d_16
activation_3 (ReLu)	(None, 256, 6, 6)	add_5
average_pooling2d_1 (AveragePool)	(None, 256, 5, 5)	activation_3
flatten_1 (Flatten)	(None, 6400)	average_pooling2d_1
dense_1 (Dense)	(None, 1024)	flatten_1
dropout_1 (Dropout) (0.3)	(None, 1024)	dense_1
dense_2 (Dense)	(None, 256)	dropout_1
dense_3 (Dense)	(None, 7)	dense_2
activation_4 (softmax)	(None, 7)	dense_3

我實作 Residual neural network，過程中將某幾層Add 在一層。架構如上圖，optimizer 用 adam，learning rate 為 10^{-4} 。結果的 Public accuracy 為 0.66843，Private accuracy 為 0.65617。最後我相同的參數 train 四次再 ensemble，結果的 Public accuracy 為 0.69155，Private accuracy 為 0.68069。

(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：

在讀資料進來的時候，直接對每一張圖片(array)減去平均(array.mean())除以標準差(array.std())。

[test_gen.csv](#)

a minute ago by Bai-Yuan Lee

[add submission details](#)

0.65617

0.66843



[test_gen_normal.csv](#)

a few seconds ago by Bai-Yuan Lee

[add submission details](#)

0.66007

0.67316



有 normalization 的結果稍微好一點，只是個人覺得這一點大概是初始值誤差，另外在 training 的時候收斂的速度有快一點點。

data augmentation：

使用 Keras 的 ImageDataGenerator，主要設定為：rotation_range=30, width_shift_range=0.3, height_shift_range=0.3, shear_range=0.3, zoom_range=0.3, horizontal_flip=True。

[test_nogen.csv](#)

2 minutes ago by Bai-Yuan Lee

[add submission details](#)

0.52020

0.52047



[test_gen.csv](#)

a minute ago by Bai-Yuan Lee

[add submission details](#)

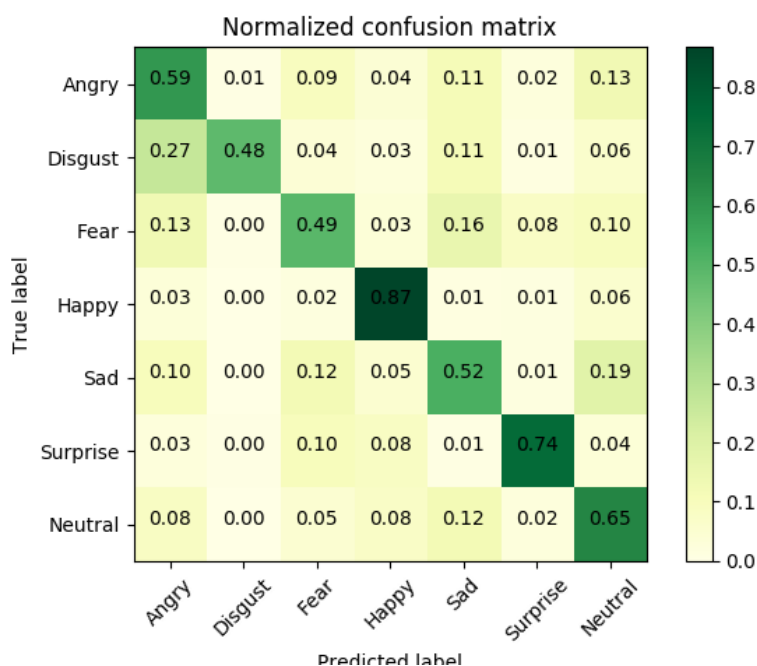
0.65617

0.66843



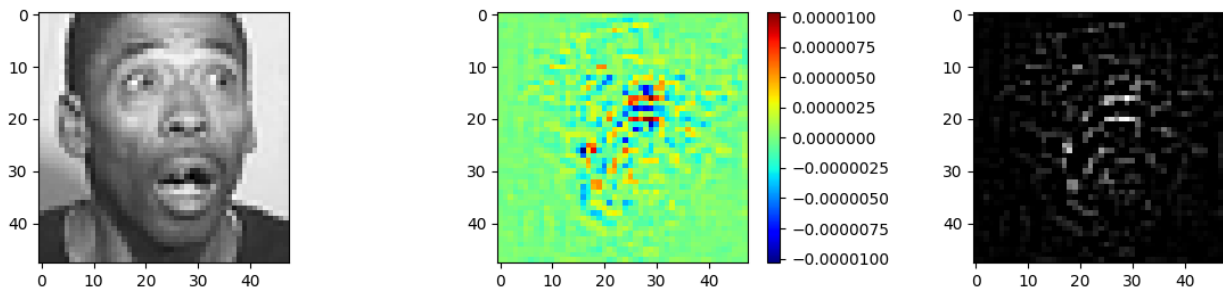
沒有做 data augmentation (no generator) 很容易 overfit，使得最後準確率很低。

(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



如圖，感覺「生氣」跟「厭惡」很容易弄混，其他相比之下還好。

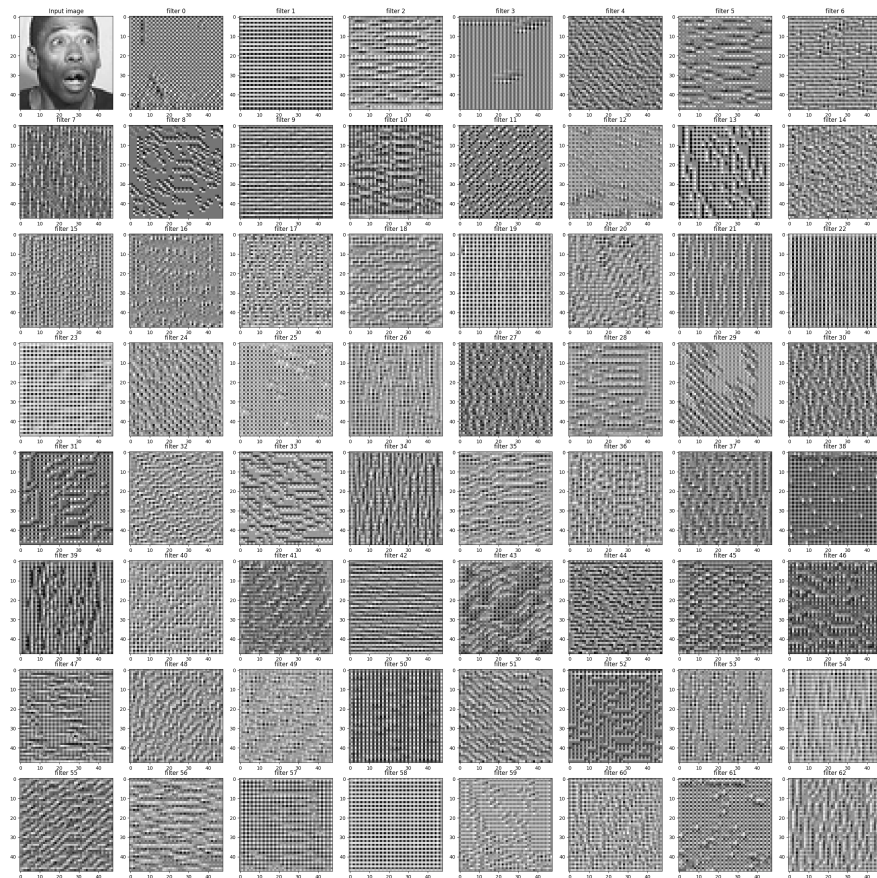
(1%) 從(1)(2)可以發現，使用 CNN 的確有些好處，試繪出其 saliency maps，觀察模型在做 classification 時，是 focus 在圖片的哪些部份？



模型主要應該是觀察到他的臉型輪廓，以及較明顯的五官，如：嘴巴、鼻子等。

(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。

我觀察第一次 Rest 結束的混合層：層數為64，這一層的 filter 比較容易被一些線條、紋狀或是有一點點臉的形狀的圖片 activate。



特定層的filter最容易被哪種圖片 activate



filter 的输出