

學號：B05902019 系級：資工二 姓名：蔡青邑

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？

Kaggle

Score 0.64558/0.64502

0.61799/0.63081

0.65477/0.66508

Model

Layer (type)	Output Shape	Param #
conv2d_10 (Conv2D)	(None, 48, 48, 32)	832
conv2d_11 (Conv2D)	(None, 48, 48, 32)	25632
conv2d_12 (Conv2D)	(None, 48, 48, 32)	25632
max_pooling2d_4 (MaxPooling2D)	(None, 24, 24, 32)	0
dropout_3 (Dropout)	(None, 24, 24, 32)	0
conv2d_13 (Conv2D)	(None, 24, 24, 64)	18496
conv2d_14 (Conv2D)	(None, 24, 24, 64)	36928
conv2d_15 (Conv2D)	(None, 24, 24, 64)	36928
max_pooling2d_5 (MaxPooling2D)	(None, 12, 12, 64)	0
dropout_4 (Dropout)	(None, 12, 12, 64)	0
conv2d_16 (Conv2D)	(None, 12, 12, 128)	73856
conv2d_17 (Conv2D)	(None, 12, 12, 128)	147584
conv2d_18 (Conv2D)	(None, 12, 12, 128)	147584
max_pooling2d_6 (MaxPooling2D)	(None, 6, 6, 128)	0
dropout_7 (Dropout)	(None, 6, 6, 128)	0
flatten_2 (Flatten)	(None, 4608)	0
dense_3 (Dense)	(None, 128)	589952
dropout_8 (Dropout)	(None, 128)	0
dense_4 (Dense)	(None, 7)	903
Total params: 1,104,327		
Trainable params: 1,104,327		

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 48, 48, 32)	832
max_pooling2d_1 (MaxPooling2D)	(None, 24, 24, 32)	0
dropout_1 (Dropout)	(None, 24, 24, 32)	0
conv2d_2 (Conv2D)	(None, 24, 24, 64)	18496
max_pooling2d_2 (MaxPooling2D)	(None, 12, 12, 64)	0
dropout_2 (Dropout)	(None, 12, 12, 64)	0
conv2d_3 (Conv2D)	(None, 12, 12, 128)	73856
max_pooling2d_3 (MaxPooling2D)	(None, 6, 6, 128)	0
dropout_3 (Dropout)	(None, 6, 6, 128)	0
flatten_1 (Flatten)	(None, 4608)	0
dense_1 (Dense)	(None, 128)	589952
dropout_4 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 7)	903
Total params: 684,039		
Trainable params: 684,039		
Non-trainable params: 0		

Layer (type)	Output Shape	Param #
input_3 (InputLayer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
fc1 (Dense)	(None, 4096)	102764544
fc2 (Dense)	(None, 4096)	16781312
predictions (Dense)	(None, 1000)	4097000
Total params: 138,357,544		

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？

Normalization的部分我主要是先減掉RGB三層的平均(但我們只有一層所以這布略過)再除以圖片的max_value(255)

	Normalization	No Normalization
Kaggle Score	0.63360/0.63917	0.61505/0.60477

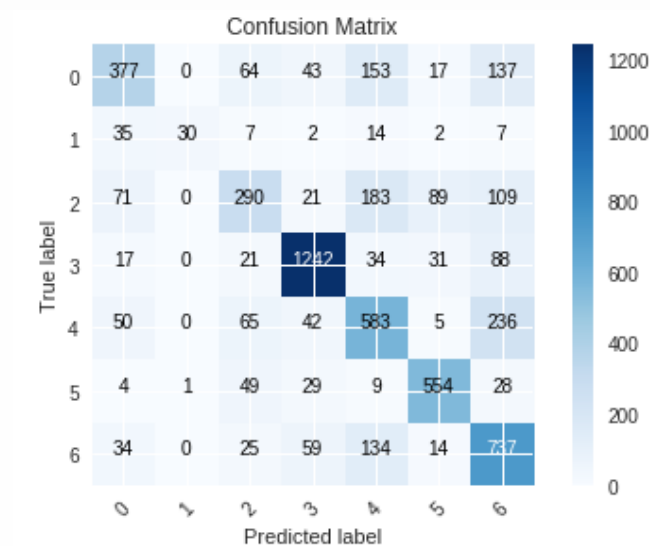
Augmentation 的部分我則用 keras.preprocessing.image 的 ImageDataGenerator 對做旋轉用的 generator 再用gen 去fit我們的training data，最後放到fit_generator裡一起train

	Augmentation	No Augmentation
Kaggle Score	0.63360/0.63917	0.59960/0.58985

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]

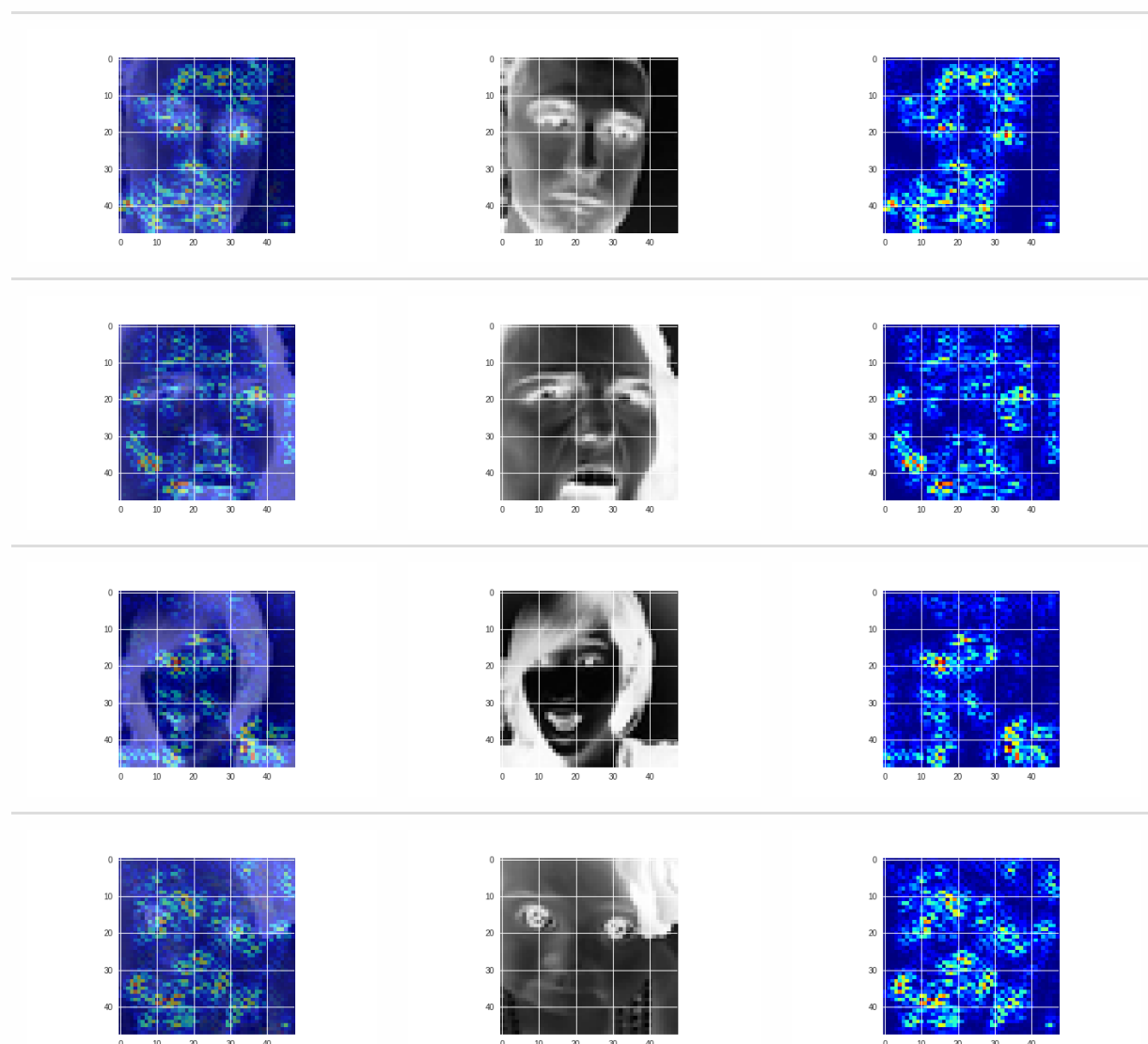
0：生氣, 1：厭惡, 2：恐懼, 3：高興, 4：難過, 5：驚訝, 6：中立(難以區分為前六種的表情))

Conclusion: 4(難過)很容易跟2(恐懼)或6(中立)搞混



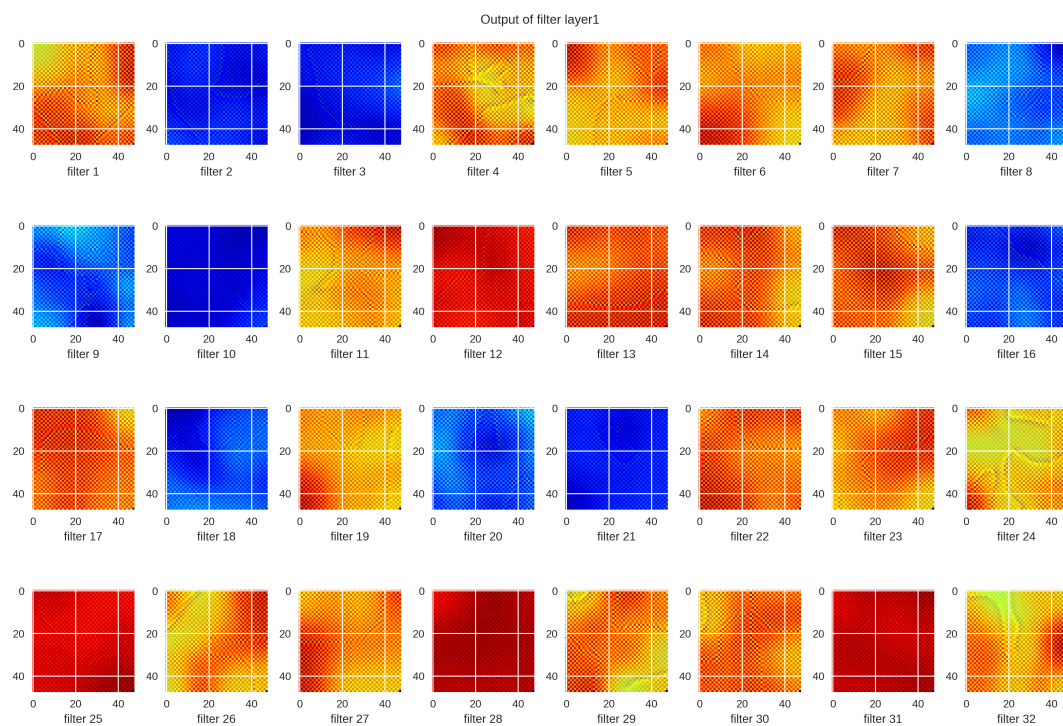
4. (1%) 從(1)(2)可以發現，使用 CNN 的確有些好處，試繪出其 saliency maps，觀察模型在做 classification 時，是 focus 在圖片的哪些部份？

主要是focus在五官(尤其是眼睛)，和眉毛的上緣。



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。

以下是第一層"relu"後觀察到的結果，雖然從filter的圖無法確定著重的意義，但從第20張圖放進去的結果可以知道，output主要activate的部分是眼睛



Output of Conv layer1 (Given image20)

