學號：B0591041 系級： 電機二 姓名：蘇家軒

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: b05902127)

(1) Layer (type)	Output Shape	Param #
batch_normalization_1 (Batch	(None, 48, 48, 1)	4
conv2d_1 (Conv2D)	(None, 48, 48, 32)	160
batch_normalization_2 (Batch	(None, 48, 48, 32)	128
conv2d_2 (Conv2D)	(None, 48, 48, 32)	4128
batch_normalization_3 (Batch	(None, 48, 48, 32)	128
max_pooling2d_1 (MaxPooling2	(None, 24, 24, 32)	0
batch_normalization_4 (Batch	(None, 24, 24, 32)	128
conv2d_3 (Conv2D)	(None, 24, 24, 32)	4128
batch_normalization_5 (Batch	(None, 24, 24, 32)	128
conv2d_4 (Conv2D)	(None, 24, 24, 32)	4128
batch_normalization_6 (Batch	(None, 24, 24, 32)	128
max_pooling2d_2 (MaxPooling2	(None, 12, 12, 32)	0
batch_normalization_7 (Batch	(None, 12, 12, 32)	128
flatten_1 (Flatten)	(None, 4608)	0
batch_normalization_8 (Batch	(None, 4608)	18432
dropout_1 (Dropout)	(None, 4608)	0
dense_1 (Dense)	(None, 512)	2359808
dense_2 (Dense)	(None, 512)	262656

batch_normalization_9 (Batch (None, 512) 2048

dense_3 (Dense) (None, 7) 3591

accuracy: 0.60936

(2) batch_normalization_1 (Batch (None, 48, 48, 1) 4

conv2d_1 (Conv2D) (None, 48, 48, 63) 630

batch_normalization_2 (Batch (None, 48, 48, 63) 252

conv2d_2 (Conv2D) (None, 48, 48, 127) 72136

batch_normalization_3 (Batch (None, 48, 48, 127) 508

max_pooling2d_1 (MaxPooling2 (None, 24, 24, 127) 0

batch_normalization_4 (Batch (None, 24, 24, 127) 508

conv2d_3 (Conv2D) (None, 24, 24, 255) 291720

batch_normalization_5 (Batch (None, 24, 24, 255) 1020

conv2d_4 (Conv2D) (None, 24, 24, 511) 1173256

batch_normalization_6 (Batch (None, 24, 24, 511) 2044

max_pooling2d_2 (MaxPooling2 (None, 12, 12, 511) 0

batch_normalization_7 (Batch (None, 12, 12, 511) 2044

flatten_1 (Flatten) (None, 73584) 0

batch_normalization_8 (Batch (None, 73584) 294336

dropout_1 (Dropout) (None, 73584) 0

dense_1 (Dense) (None, 512) 37675520

dense_2 (Dense) (None, 512) 262656

batch_normalization_9 (Batch (None, 512) 2048

dropout_2 (Dropout) (None, 512) 0

dense_3 (Dense) (None, 7) 3591

accuracy: 0.64140

(3)input_1 (InputLayer) (None, 48, 48, 1) 0

batch_normalization_1 (Batch (None, 48, 48, 1) 4

conv2d_1 (Conv2D) (None, 48, 48, 64) 640

batch_normalization_2 (Batch (None, 48, 48, 64) 256

conv2d_2 (Conv2D) (None, 48, 48, 64) 36928

batch_normalization_3 (Batch (None, 48, 48, 64) 256

max_pooling2d_1 (MaxPooling2 (None, 24, 24, 64) 0

batch_normalization_4 (Batch (None, 24, 24, 64) 256

conv2d_3 (Conv2D) (None, 24, 24, 128) 73856

batch_normalization_5 (Batch (None, 24, 24, 128) 512

conv2d_4 (Conv2D) (None, 24, 24, 128) 147584

batch_normalization_6 (Batch (None, 24, 24, 128) 512

max_pooling2d_2 (MaxPooling2 (None, 12, 12, 128) 0

batch_normalization_7 (Batch (None, 12, 12, 128) 512

conv2d_5 (Conv2D) (None, 12, 12, 256) 295168

batch_normalization_8 (Batch (None, 12, 12, 256) 1024

conv2d_6 (Conv2D) (None, 12, 12, 256) 590080

batch_normalization_9 (Batch	(None, 12, 12, 256)	1024
conv2d_7 (Conv2D)	(None, 12, 12, 256)	590080
batch_normalization_10 (Batc	(None, 12, 12, 256)	1024
max_pooling2d_3 (MaxPooling2	(None, 6, 6, 256)	0
batch_normalization_11 (Batc	(None, 6, 6, 256)	1024
conv2d_8 (Conv2D)	(None, 6, 6, 512)	1180160
batch_normalization_12 (Batc	(None, 6, 6, 512)	2048
conv2d_9 (Conv2D)	(None, 6, 6, 512)	2359808
batch_normalization_13 (Batc	(None, 6, 6, 512)	2048
conv2d_10 (Conv2D)	(None, 6, 6, 512)	2359808
batch_normalization_14 (Batc	(None, 6, 6, 512)	2048
max_pooling2d_4 (MaxPooling2	(None, 3, 3, 512)	0
batch_normalization_15 (Batc	(None, 3, 3, 512)	2048
conv2d_11 (Conv2D)	(None, 3, 3, 512)	2359808
batch_normalization_16 (Batc	(None, 3, 3, 512)	2048
conv2d_12 (Conv2D)	(None, 3, 3, 512)	2359808
batch_normalization_17 (Batc	(None, 3, 3, 512)	2048
conv2d_13 (Conv2D)	(None, 3, 3, 512)	2359808
batch_normalization_18 (Batc	(None, 3, 3, 512)	2048
max_pooling2d_5 (MaxPooling2	(None, 2, 2, 512)	0
batch_normalization_19 (Batc	(None, 2, 2, 512)	2048

flatten_1 (Flatten)	(None, 2048)	0
batch_normalization_20 (Batc	(None, 2048)	8192
dropout_1 (Dropout)	(None, 2048)	0
dense_1 (Dense)	(None, 512)	1049088
dense_2 (Dense)	(None, 512)	262656
batch_normalization_21 (Batc	(None, 512)	2048
dense_3 (Dense)	(None, 512)	262656
batch_normalization_22 (Batc	(None, 512)	2048
dense_4 (Dense)	(None, 7)	3591

accuracy: 0.65254

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？

藉由添加 batch_normalization 層完成 data normalization，準確率由 56%上升至 60%; from keras.preprocessing.image import ImageDataGenerator 來處理 data augmentation，準確率由 60%上升至 65%。

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]

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？

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。