

學號：R06942082 系級： 電信碩一 姓名：黃釋平

1. (1%) 請說明你實作的 CNN model，其模型架構、訓練過程和準確率為何？

(Collaborators:鄭立晟、黃梓鳴、姚嘉昇)

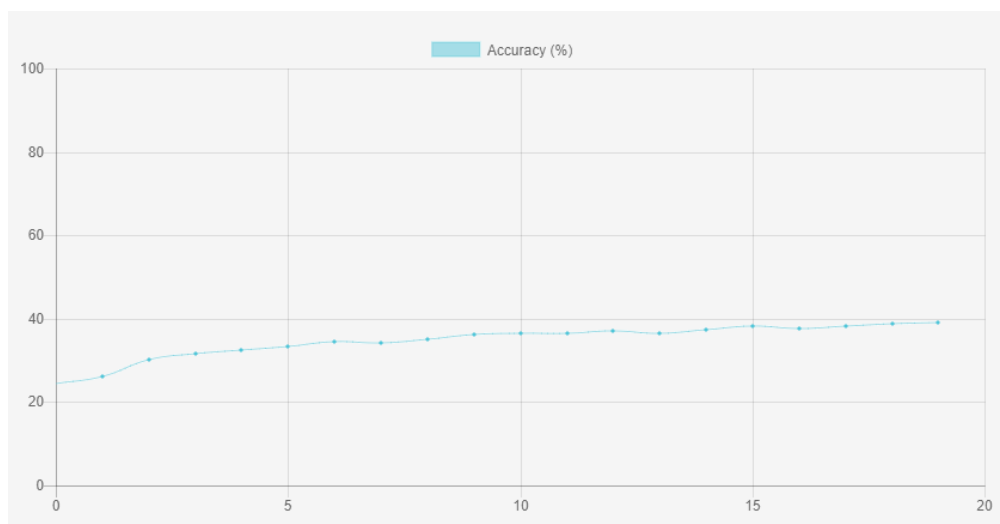
答

### CNN Model Structure:

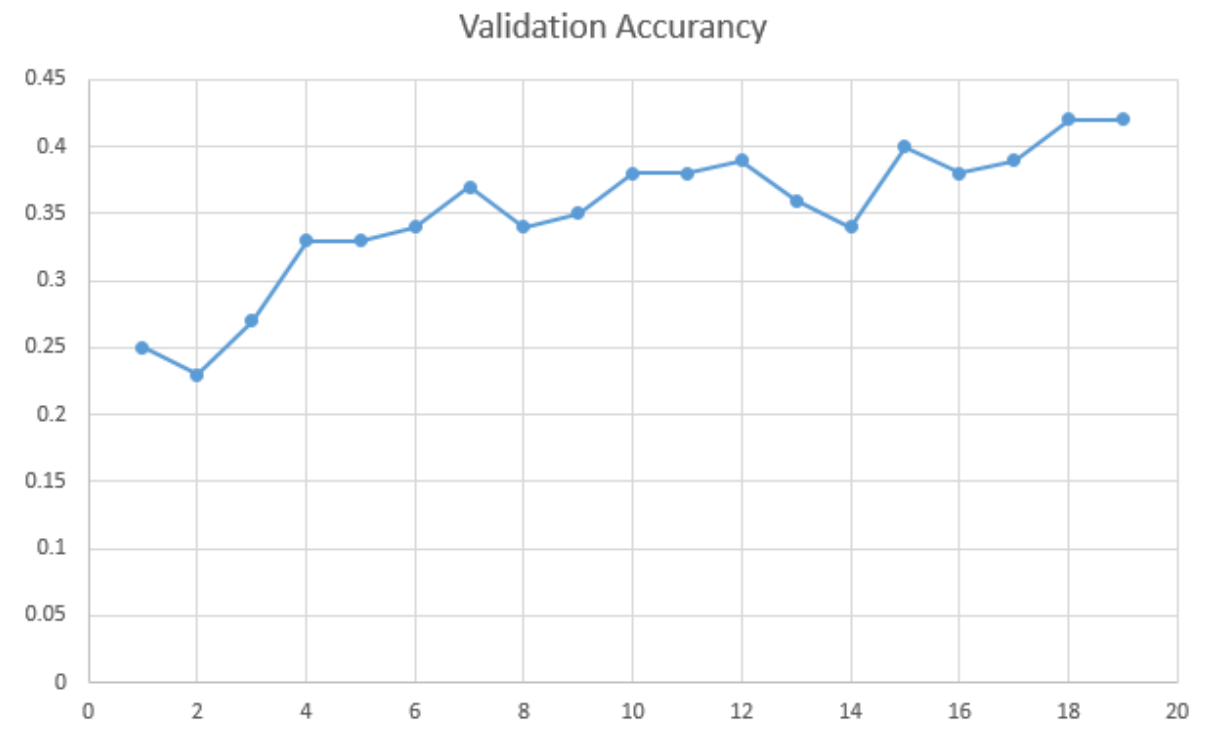
Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 44, 44, 16)	416
activation_1 (Activation)	(None, 44, 44, 16)	0
max_pooling2d_1 (MaxPooling2D)	(None, 22, 22, 16)	0
conv2d_2 (Conv2D)	(None, 18, 18, 128)	51328
activation_2 (Activation)	(None, 18, 18, 128)	0
max_pooling2d_2 (MaxPooling2D)	(None, 9, 9, 128)	0
conv2d_3 (Conv2D)	(None, 7, 7, 32)	36896
batch_normalization_1 (Batch Normalization)	(None, 7, 7, 32)	128
activation_3 (Activation)	(None, 7, 7, 32)	0
max_pooling2d_3 (MaxPooling2D)	(None, 3, 3, 32)	0
flatten_1 (Flatten)	(None, 288)	0
dense_1 (Dense)	(None, 128)	36992
batch_normalization_2 (Batch Normalization)	(None, 128)	512
activation_4 (Activation)	(None, 128)	0
dropout_1 (Dropout)	(None, 128)	0

dense_2 (Dense)	(None, 512)	66048
batch_normalization_3 (Batch Normalization)	(None, 512)	2048
activation_5 (Activation)	(None, 512)	0
dropout_2 (Dropout)	(None, 512)	0
dense_3 (Dense)	(None, 7)	3591
Total params: 197,959		
Trainable params: 196,615		
Non-trainable params: 1,344		

### CNN Training Accuracy:



## CNN Validation Accuracy:



尋找過程試過很多種架構，包含三層 CNN、單層 DNN、調 kernal\_size、調 padding... 等，這篇 report 中是其中一個嘗試過的，想要在 20 個 epoch 之內達到 40% accuracy。也許架構設計不良，導致卡在 40% 上不去，不過觀察 validation accuracy 比 training accuracy 還高，應該不是 overfitting，所以接下來試著增加複雜度。

之後也跟同學討論過，稍微參考他們的做法，像是連續做 CNN、圖片 normalize，最新的上傳能達到 Public score = 0.635，尚有待改進。

2. (1%) 承上題，請用與上述 CNN 接近的參數量，實做簡單的 DNN model。其模型架構、訓練過程和準確率為何？試與上題結果做比較，並說明你觀察到了什麼？

(Collaborators: )

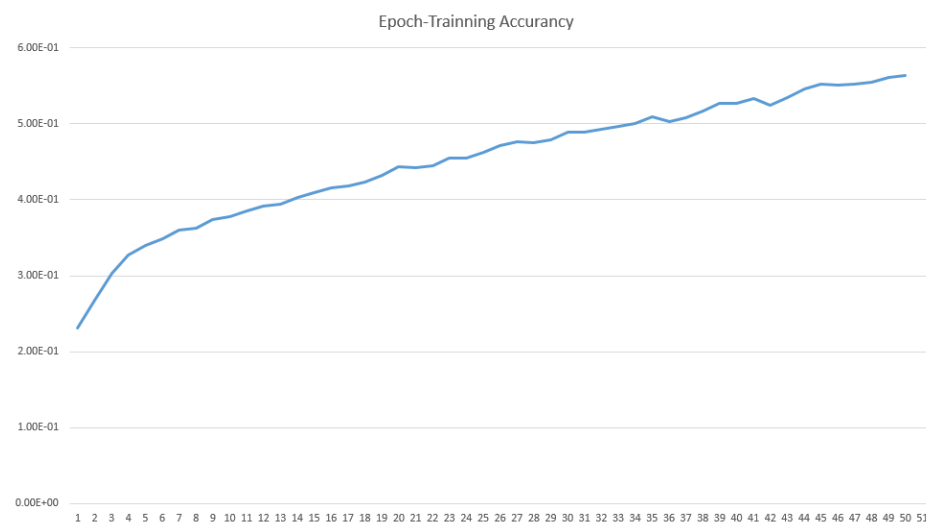
答：

## DNN Model Structure:

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 96)	221280
batch_normalization_1 (Batch Normalization)	(None, 96)	384
activation_1 (Activation)	(None, 96)	0
dense_2 (Dense)	(None, 64)	6208
batch_normalization_2 (Batch Normalization)	(None, 64)	256
activation_2 (Activation)	(None, 64)	0
dropout_1 (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 7)	455
Total params: 228,583		
Trainable params: 228,263		
Non-trainable params: 320		

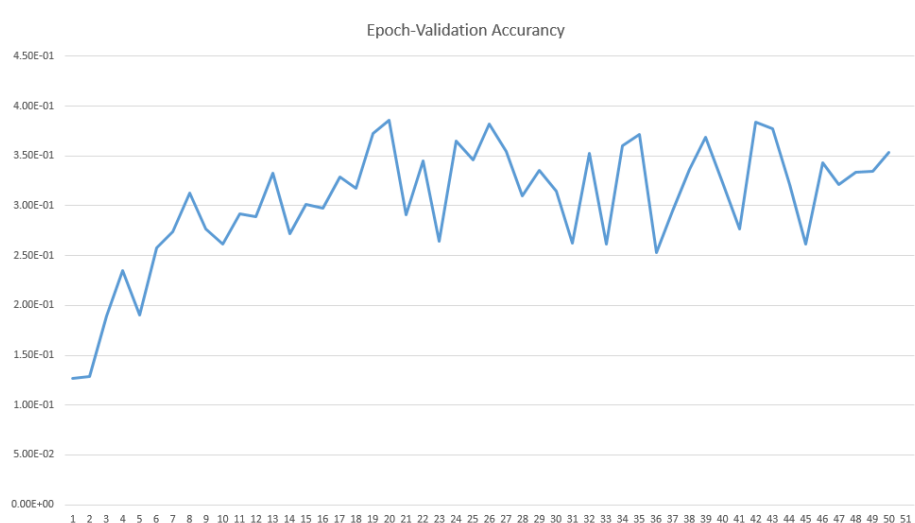
使用兩層 DNN，Params 數量略比 CNN 的多一點，並加一層 dropout=0.5。

## DNN Training Accuracy:



跑 50 個 epoch，Training accuracy 達到 0.55 左右，epoch 設定為 50 的目的是觀察 overfitting 現象。

## DNN Validation Accuracy:



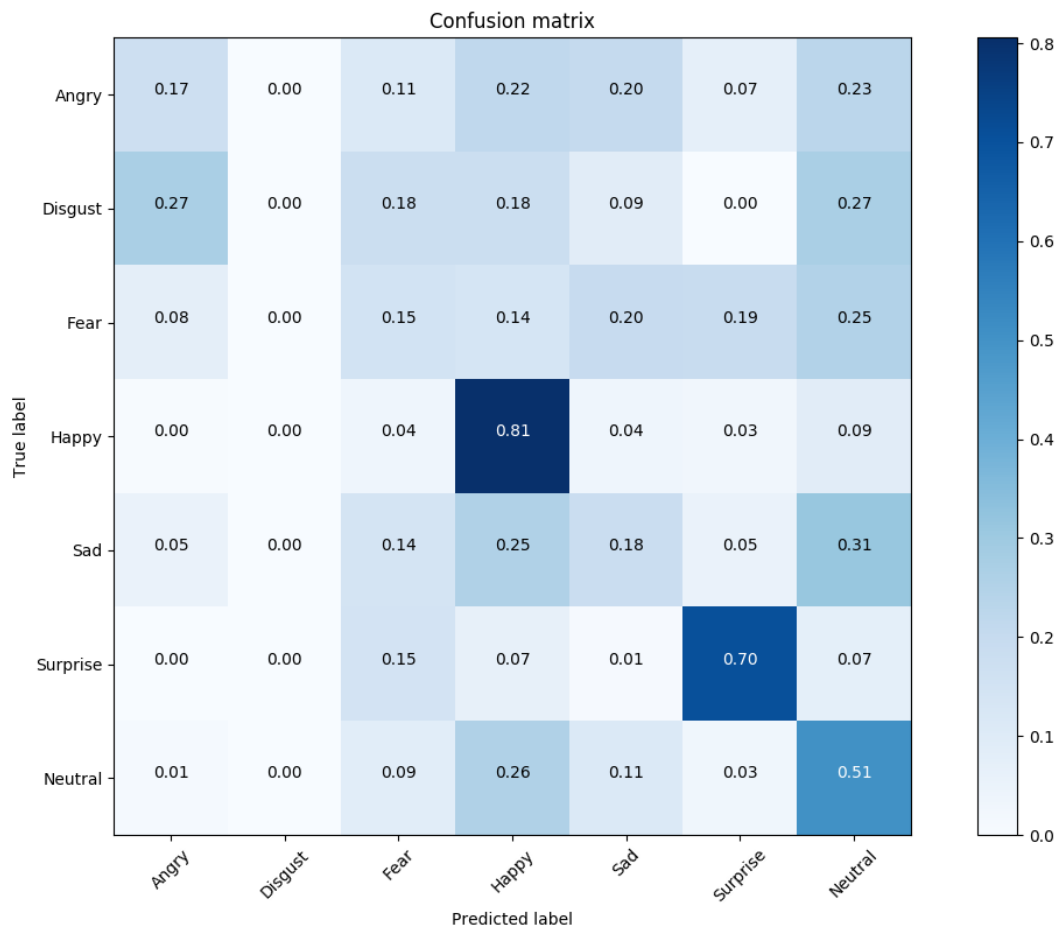
在 epoch=19 時就已經達到最好的 validation accuracy，之後開始震盪，而同時 training accuracy 卻在上升，所以應該是 overfitting。

設定的 unit 數量不多，但已經有如此龐大的 params，而效果跟 CNN 差不多，甚至更為不穩定，可以看出來 CNN 在這部分的确比 DNN 來的有效率。

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：



觀察到非常有趣的事，最好的 confusion matrix 應該在對角線上都是 1.0，但在「Disgust」那格卻是 0，甚至沒有其他資料被判成 Disgust；而「Happy」部分雖然精準度很高，但也許並不是判斷的好，而是大部分的東西都被當作「Happy」了，本身當然包括在內，從「Happy」的直排數字都偏高可以觀察出來。此外，「Surprise」準確率也相當不錯，也正好也是「Happy」中比例最低 (0.07) 的一項，也就是說，這兩類在這 model 中不易搞混。

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？

(Collaborators: )

答：

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。

(Collaborators: )

答：