學號：R07942115 系級： 電信碩一 姓名：謝硯澤

1. (2%) 請說明你實作的 CNN model，其模型架構、訓練參數和準確率為何？並請用與上述 CNN 接近的參數量，實做簡單的 DNN model，同時也說明其模型架構、訓練參數和準確率為何？並說明你觀察到了什麼？
(Collaborators:)

答：

(a)CNN模型架構

Layer (type)	Output Shape	Param #
conv2d_151 (Conv2D)	(None, 44, 44, 32)	832
conv2d_152 (Conv2D)	(None, 44, 44, 32)	25632
conv2d_153 (Conv2D)	(None, 44, 44, 32)	25632
batch_normalization_51 (Batc	(None, 44, 44, 32)	128
max_pooling2d_51 (MaxPooling	(None, 21, 21, 32)	0
conv2d_154 (Conv2D)	(None, 21, 21, 64)	51264
conv2d_155 (Conv2D)	(None, 21, 21, 64)	102464
conv2d_156 (Conv2D)	(None, 21, 21, 64)	102464
batch_normalization_52 (Batc	(None, 21, 21, 64)	256
max_pooling2d_52 (MaxPooling	(None, 10, 10, 64)	0
conv2d_157 (Conv2D)	(None, 10, 10, 96)	55392
conv2d_158 (Conv2D)	(None, 10, 10, 96)	83040
conv2d_159 (Conv2D)	(None, 10, 10, 96)	83040
batch_normalization_53 (Batc	(None, 10, 10, 96)	384
conv2d_160 (Conv2D)	(None, 5, 5, 128)	110720
conv2d_161 (Conv2D)	(None, 5, 5, 128)	147584
conv2d_162 (Conv2D)	(None, 5, 5, 128)	147584
batch_normalization_54 (Batc	(None, 5, 5, 128)	512
max_pooling2d_54 (MaxPooling	(None, 2, 2, 128)	0
conv2d_163 (Conv2D)	(None, 2, 2, 160)	184480
conv2d_164 (Conv2D)	(None, 2, 2, 160)	230560
conv2d_165 (Conv2D)	(None, 2, 2, 160)	230560
batch_normalization_55 (Batc	(None, 2, 2, 160)	640
max_pooling2d_55 (MaxPooling	(None, 1, 1, 160)	0
Flatten (Flatten)	(None, 160)	0
dropout_61 (Dropout)	(None, 160)	0
dense_61 (Dense)	(None, 1024)	164864
dropout_62 (Dropout)	(None, 1024)	0
dense_62 (Dense)	(None, 512)	524800
dropout_63 (Dropout)	(None, 512)	0
dense_63 (Dense)	(None, 256)	131328
dropout_64 (Dropout)	(None, 256)	0
dense_64 (Dense)	(None, 128)	32896
dropout_65 (Dropout)	(None, 128)	0
dense_65 (Dense)	(None, 64)	8256
dropout_66 (Dropout)	(None, 64)	0
dense_66 (Dense)	(None, 7)	455
softmax1 (Activation)	(None, 7)	0

save/20190407/model_0.h5

Total params: 2,445,767
Trainable params: 2,444,807
Non-trainable params: 960

(b)DNN模型架構

Layer (type)	Output Shape	Param #
Flatten (Flatten)	(None, 2304)	0
dense_176 (Dense)	(None, 805)	1855525
dropout_152 (Dropout)	(None, 805)	0
dense_177 (Dense)	(None, 512)	412672
dropout_153 (Dropout)	(None, 512)	0
dense_178 (Dense)	(None, 256)	131328
dropout_154 (Dropout)	(None, 256)	0
dense_179 (Dense)	(None, 128)	32896
dropout_155 (Dropout)	(None, 128)	0
dense_180 (Dense)	(None, 64)	8256
dropout_156 (Dropout)	(None, 64)	0
dense_181 (Dense)	(None, 32)	2080
dropout_157 (Dropout)	(None, 32)	0
dense_182 (Dense)	(None, 32)	1056
dropout_158 (Dropout)	(None, 32)	0
dense_183 (Dense)	(None, 16)	528
dropout_159 (Dropout)	(None, 16)	0
dense_184 (Dense)	(None, 7)	119
softmax1 (Activation)	(None, 7)	0

Total params: 2,444,460
Trainable params: 2,444,460
Non-trainable params: 0

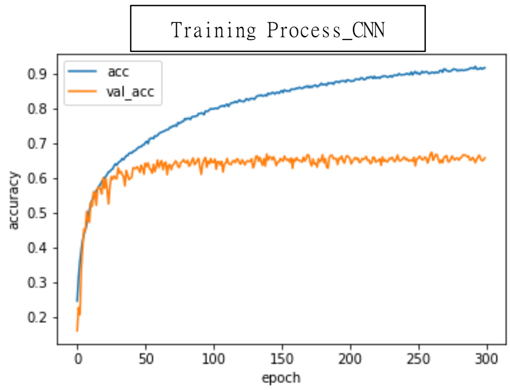
Total params: 2,444,460
Trainable params: 2,444,460
Non-trainable params: 0

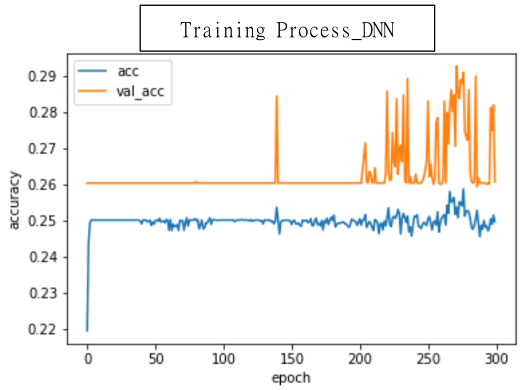
	CNN	DNN
EPOCH	300	300
BATCH_SIZE	256	256
Optimizer	Adam(default)	Adam(default)
Loss Function	categorical_crossentropy	categorical_crossentropy
Kaggle Public Score	0.66118	0.28475
Kaggle Private Score	0.64474	0.28392

如果直接把input接一個fully connected layer會需要大量的參數，在餐數量差不多的條件下，DNN的layer數明顯比CNN少很多。且DNN大部分的參數都集中在某一層，感覺會有問題。

2. (1%) 承上題，請分別畫出這兩個model的訓練過程 (i.e., loss/accuracy v.s. epoch)
(Collaborators:)

答：





上圖是CNN/DNN的training process figure，DNN的架構基本跟CNN後面幾層的FC layer一樣，但卻train不太起來的感覺，可能要再對DNN架構作調整，才有辦法獲得較好的performance。

3. (1%) 請嘗試 data normalization, data augmentation,說明實作方法並且說明實行前後對準確率有什麼樣的影響？
(Collaborators:)

答：

	without data augmentation and data normalization	only with normalization	with data augmentation and data normalization
Kaggle Public Score	0.63443	0.62635	0.66118
Kaggle Private Score	0.61270	0.61744	0.64474

加入Normalization後，private準確率好像有上升一點，而使用data augmentation，則大幅提升了準確率。Data Augmentation我使用了以下幾種方式，並且是在batch裡面再做data augmentation使得每一次epoch看到的data都不太一樣。

rotation_range=10 : 隨機旋轉-10度到10度之間

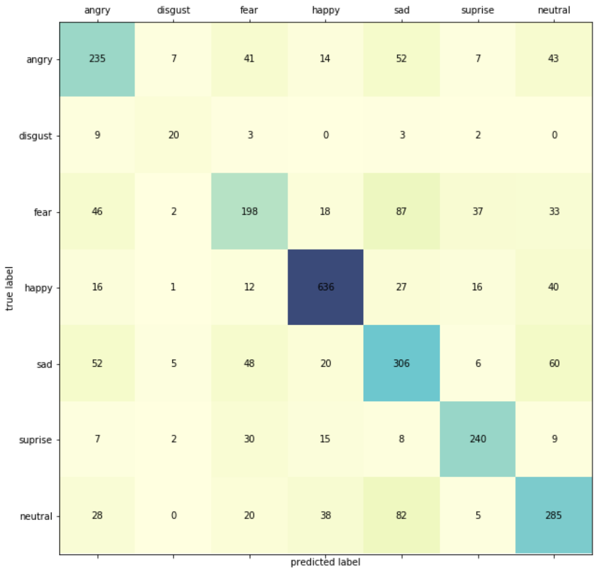
width_shift_range=0.1 : 隨機左右平移0.1(width)

height_shift_range=0.1 : 隨機上下平移0.1(height)

horizontal_flip=True : 隨機左右翻轉

4. (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：



使用validation set做confusion matrix圖，觀察結果如下。

以下括弧中的兩個類別，辨識時特別容易搞混:

{ “neutral”, “sad” }, { “fear”, “sad” }, { “fear”, “angry” }

發現其中“happy”判斷準確度最高。Model學習的最好。