HW3 Report

學號：B04502031 系級：電機二 姓名：施力維

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？

   這次的 model 是使用 Keras 來時做的，可以分成 3 個部分來討論，(1) VGG (2) Data Argumentation (3) Ensemble：

   a. VGG

   這部分是參考 VGG 的 Paper，總共用了 6 層 Conv 層：(64, 64, 128, 128, 128, 128)，kernel 都設為 3，每兩層做一次 Maxpooling，直接 train 大約可以到達 64%的 Accuracy。

   b. Data argumentation

   Data argumentation 總共用了 4 個參數，Rotation=30、Shear=0.2、Zoom=0.2 和 Horizontal flip，不過由於訓練時間過長以及差距不大，並沒有仔細去調整 4 個參數的數值，訓練時生成的資料設成原本的 8 倍長，用了之後 Accuracy 從 64%進步到 69%

   c. Ensemble

   最後是將上面的 model 取 3 個 Public 表現最好的來做 Ensemble，分別是 69.0%、69.3%、70.0%，將預測出來的機率平均再做選擇，效果是從 69%進步到 72%。

   訓練時 Optimizer 使用 Adam，Learning rate 設為 5e-4，總共執行 50 個 epoch。

   最後做出來的成績 (Public, Private) = (0.72053, 0.70381)

2. 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

   Data normalization 一共做了兩種測試：一個是最常見的除以 255，另一個是拿全部的 train data 來做標準化，做出來的結果如下：

不做	除以 255	標準化
68.72%	68.83%	68.86%

可以看出有沒有做標準化對整體的影響其實不大，而兩種標準化的方式也沒有什麼差別。

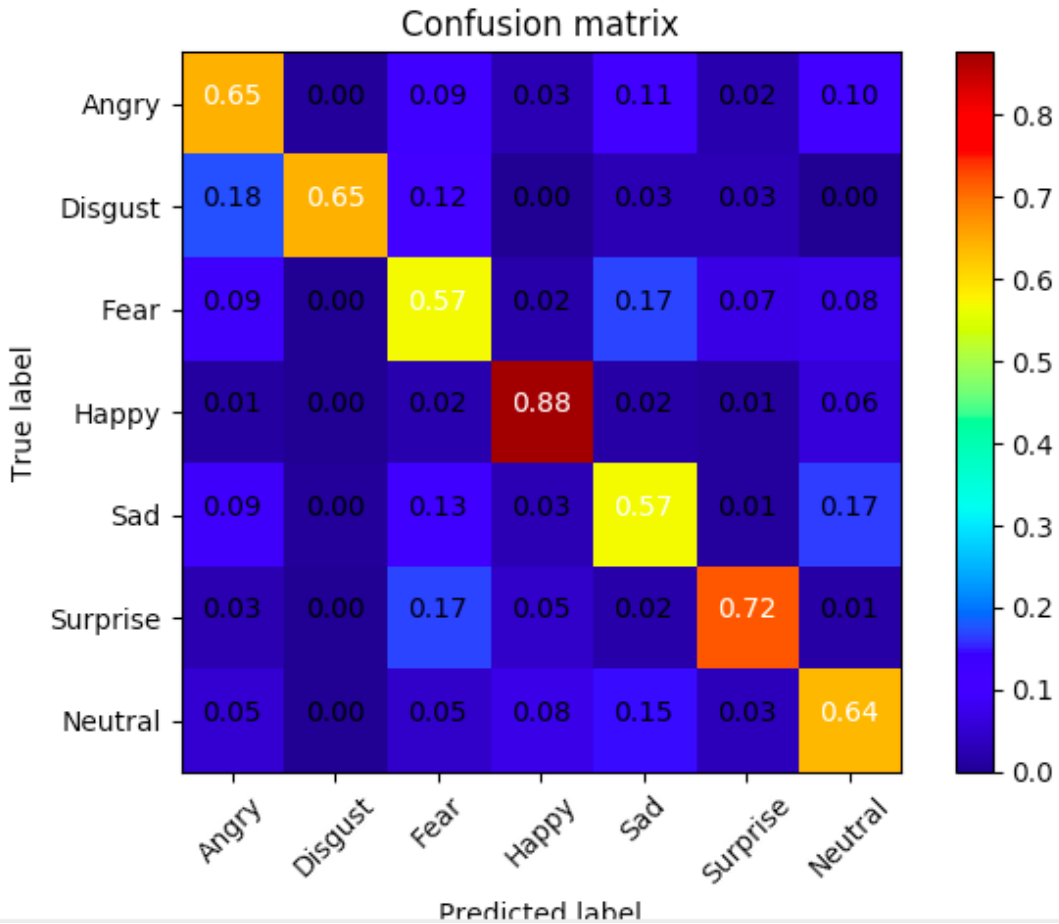
Data argumentation

不做	2 倍 data	4 倍 data	8 倍 data
64.43%	66.32%	68.06%	68.86%

Data argumentation 總共用了 4 個參數，Rotation=30、Shear=0.2、Zoom=0.2 和 Horizontal flip，主要考量，實作的部分是使用 keras 的 fit_genetator，每個 epoch 都會生成測資，生成 data 的數量對於訓練的結果也會有影響，蛋 4 倍與 8 倍時便沒有太大差異了。

3. 觀察答錯的圖片中，哪些 class 彼此間容易用混？[繪出 confusion matrix 分析]

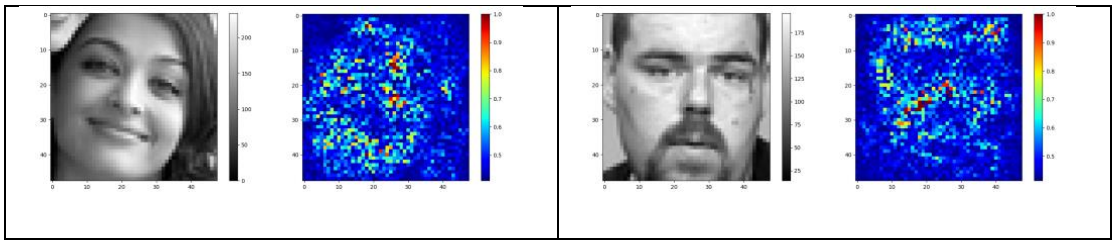
這個 model 是使用第一題的作法中 ensemble 前個別的 model，實作出的 confusoin matrix 如下：

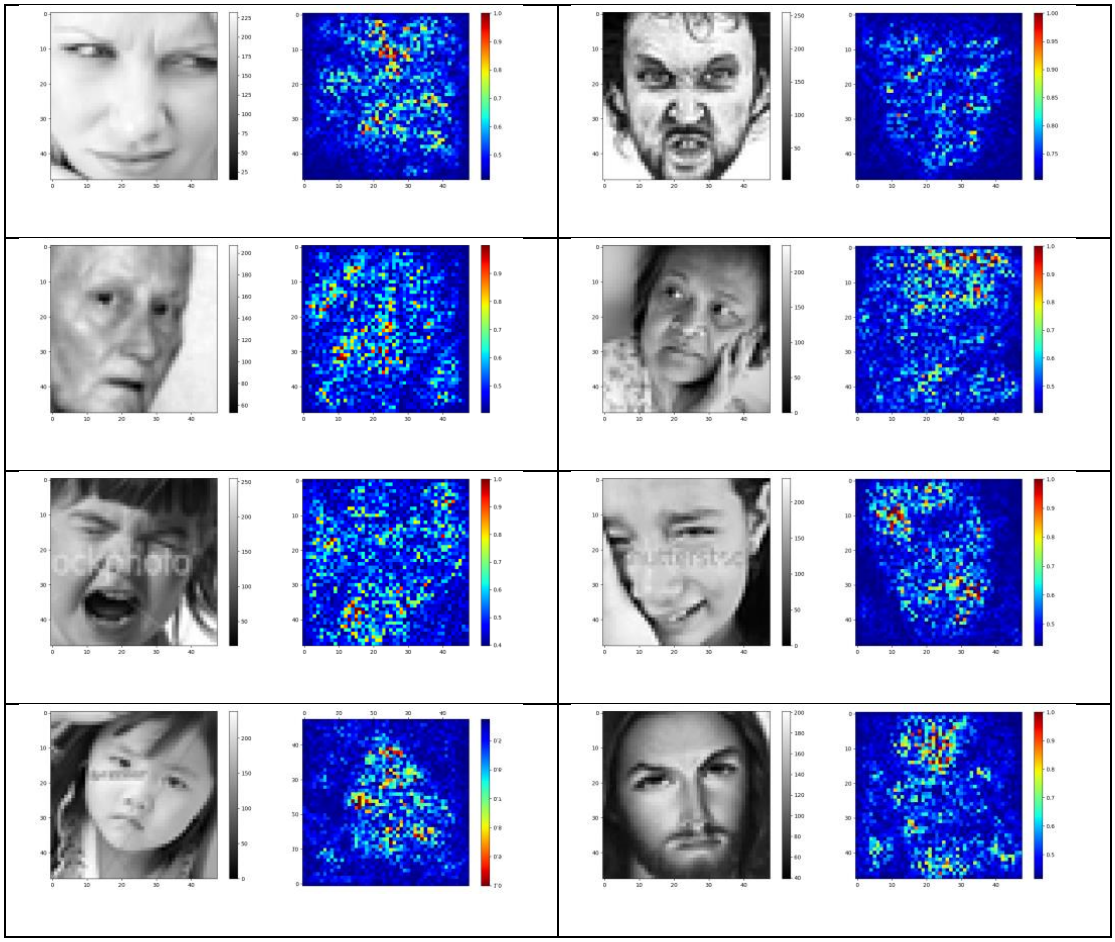


其中相互混淆的部分，互相皆超過 10%的有兩個：(1) Sad 和 Neutral (2) Sad 和 Fear，單方面混淆超過 10%則是(1) Surprise 被當成 Fear (2) Disgust 被當成 Fear。

4. 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？

Model 採用第一題中 Public = 70.0%的模型，隨機挑選 10 張做 saliency maps 如下：

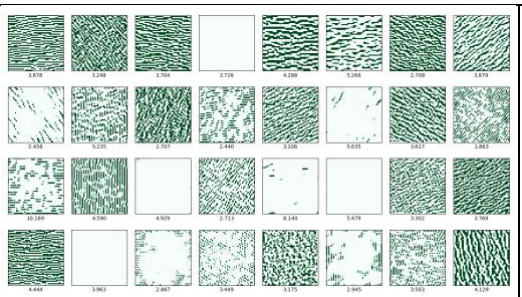
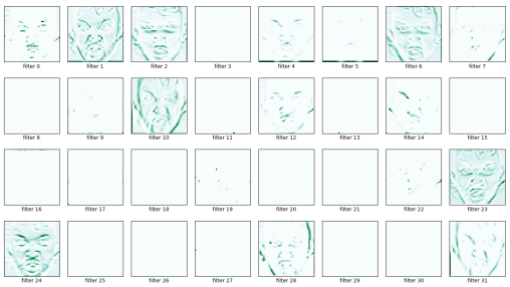
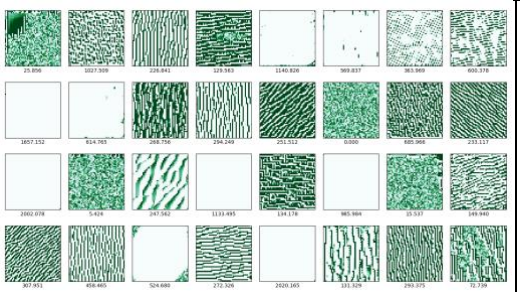
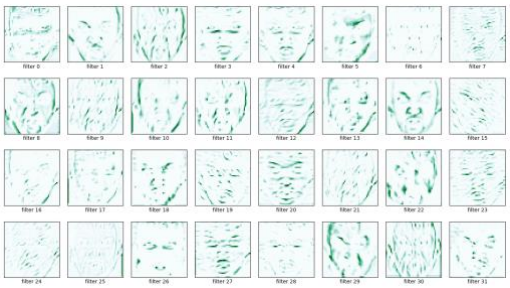
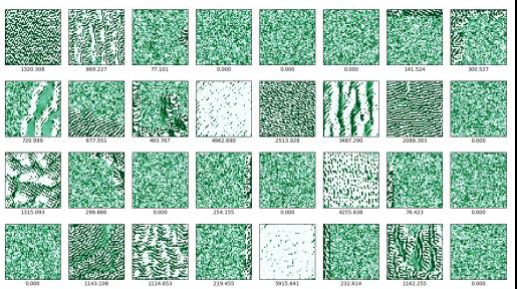
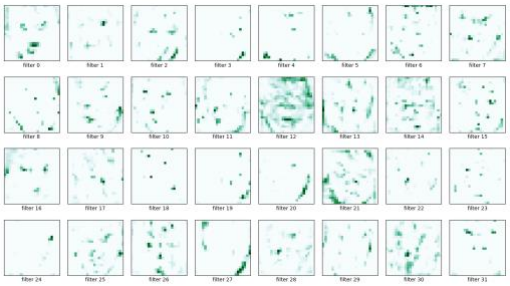




依序比對各張圖，大部分的圖都有把臉型和五官抓出來，最上面兩張圖尤其明顯，臉上除了眼睛、嘴巴以外幾乎都有顏色。其中嘴巴的部分約有 8 張圖有 focus 到，另外則有 6 張圖是 focus 在額頭的位置，也有 3 張圖左右是注意在鼻子的部分，

5. 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。

這邊使用的是(1)Public = 70.0%的 model，取每一層的前 32 個 model，取前四層來看，參考 VGG 的模式每兩層才 pooling 一次。可以看到測試的圖中第一層有蠻多是空白的，而特定幾個則完整抓到臉型，而第二層則繼續加強，幾乎每個 filter 都有抓到五官輪廓。3、4 層側是更重點的把五官的部分抓出來。

1		
2		
3		
4