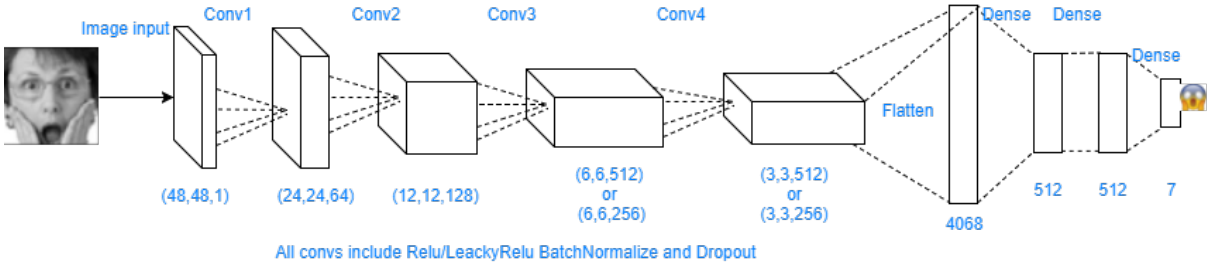
學號：**R06946003** 系級：資料科學碩一 姓名：湯忠憲

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？

答：

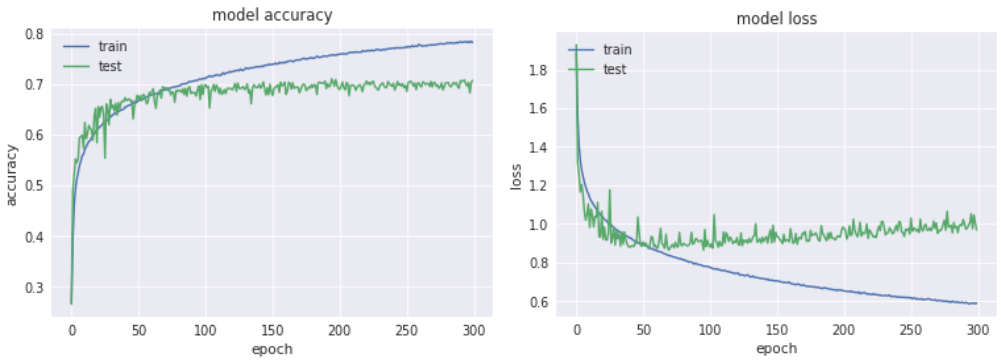
目前有許多做 Image classification 的 related works，例如 AlexNet, VGG, Resnet, GoogleNet 等等。考量到此次圖片分類的資料量以及 class 數量，我使用類似 AlexNet 的架構。整體架構如圖一。



圖一、CNN model 架構

我使用的 CNN 架構中包含四層 convolution layers，每一個的 Conv 模組中包含 Activation, BatchNomalization, Dropout layer。Filter 大致上是越來越多張，而 kernel size 則比照 AlexNet，分別為(5,5)、(3,3)、(3,3)、(3,3)。Dropout 一開始則是設的比較積極一點 (ex: 0.25)，並且隨著層數增加提升百分比。跑完 Conv 層之後則用 Flatten 將圖壓平，再接著兩層 Fully connected 的 Dense layers。最後利用 softmax function 來產生 7 個情緒的分類。Optimizer 用 Adam。CNN 模型參數約為 5,656,199 個 trainable parameters

訓練過程: 本次作業的 Best model 主要有用到 Data augmentation 和 Ensemble 的技巧來訓練的。Data augmentation 是將圖片先經過旋轉、縮放以及平移等操作，使得一張照片可以產生出好幾種樣貌。這部分可以用 Keras 的 ImageDataGenerator 實作。此外，訓練的過程中都會切出 3500 張圖片做 Validation，並挑選 model，但這樣勢必有些資料無法拿來 train。因此，我將 training data 做 bagging 的處理，這樣一來能產生好幾個基於不同 subsample 的 classifier。最後再將這些 model (Best model 大約用了 7 個)對測試資料預測出來的機率"們"相加(有另外試過 major voting 或 accuracy weighting 的方式，但效果沒有比較好) 就可以得到最後的機率分佈。訓練過程的 accuracy 和 loss 如圖二。



圖二、CNN accuracy & loss

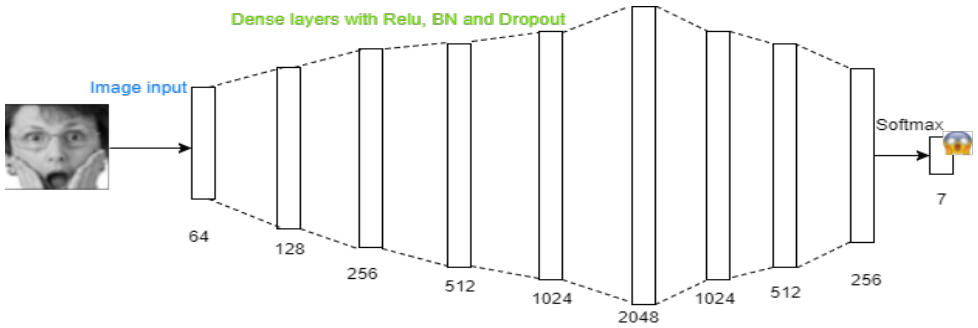
準確率: single model 在 kaggle 上的分數大約在 0.70 附近，我最後將 7 個 model

ensemble 起來，準確度大約 0.72。

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？

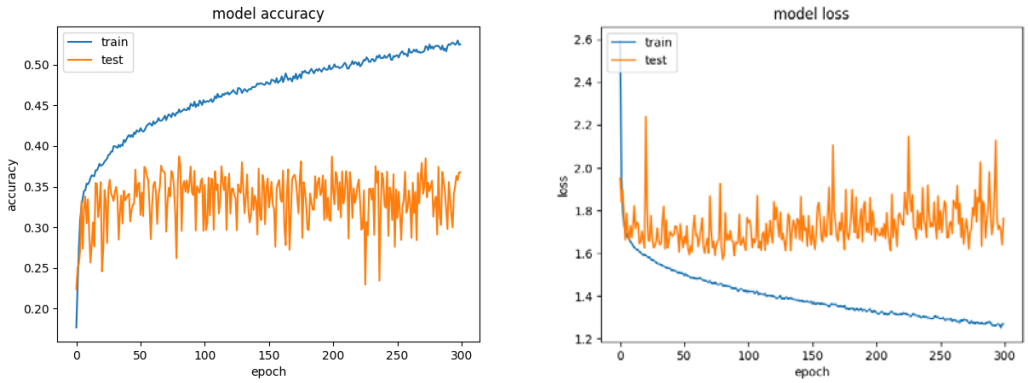
答：

為使得 DNN 模型的參數數量與 CNN 相近，我疊了 9 層的 Fully connected Dense layer，約有 5,712,711 個 trainable parameters。整體架構如圖三。架構中，每一個 Dense 模組均包含 Relu activation function、BatchNomalization 和 Dropout。



圖三、DNN 架構

訓練過程: 在進到 DNN 前先將 input 維度從 (batch size, 48,48,1) 變為 (batch size, 48*48)。如果沒攤平直接進去的話，Dense layer 會在最後一個維度上做計算，有點類似 Keras 中 TimeDistributed 的效果。訓練過程的準確度則如圖四，可以發現 DNN 的準確度大約 0.35，是 CNN 的一半。

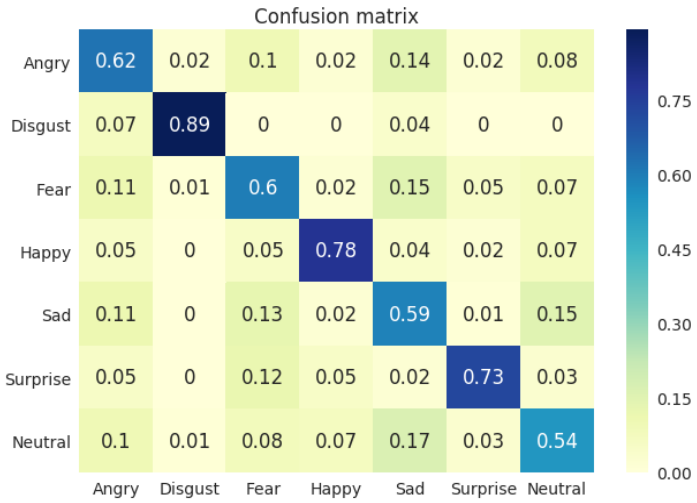


圖四、DNN accuracy & loss

與上題比較: 1) CNN 大約到 100 個 epoches 可以收斂，DNN 在約 50 個 epoches 就收斂了，不過收斂到了不理想的位置。2) 即使 DNN 疊了 9 層，參數還多一點點，D NN 還是跑得比 CNN 快很多，可能是因為卷積的過程中有更多的計算以及操作。3) 比較兩者的準確度可以發現完全不在一個水平上。

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**]**

答：

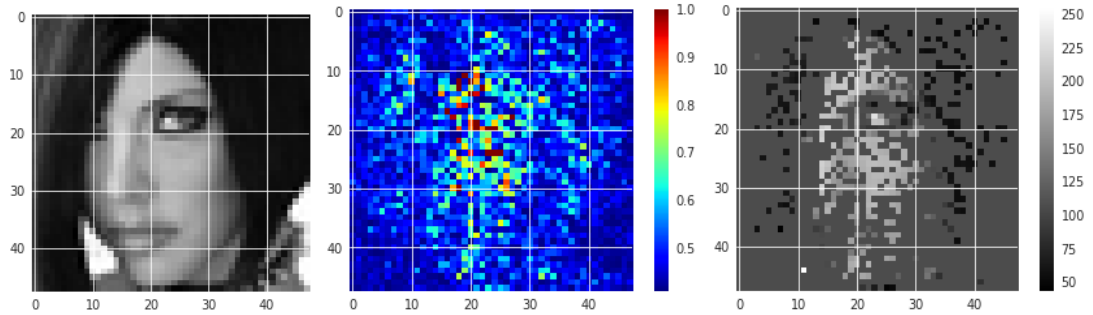


圖五、Confusion matrix for Image Sentiment Recognition

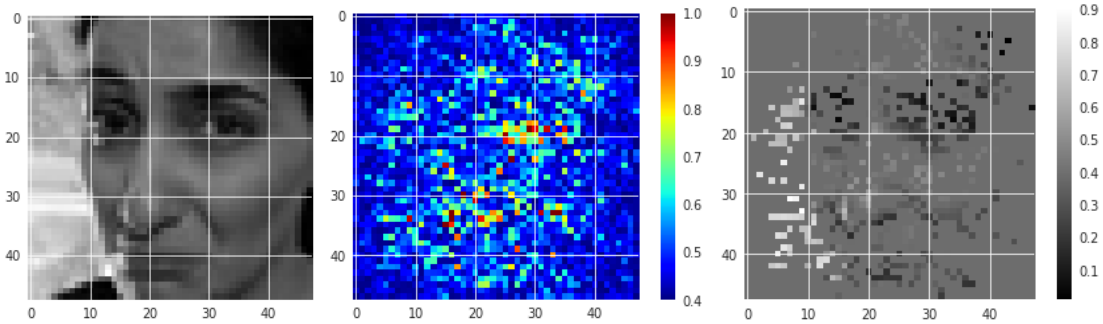
從圖五中可以發現 Fear, Sad 跟 Neutral 都有較低的準確率。觀察其兩兩之間的分錯的比例大約可以到 0.15 或 0.17 的水平，由其是 Sad 容易跟別種情緒混淆 (Sad 的 row 或 column 有比較多顏色較深色的色塊)。兩者之間最容易被分錯的是 Sad 跟 Neutral。

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？

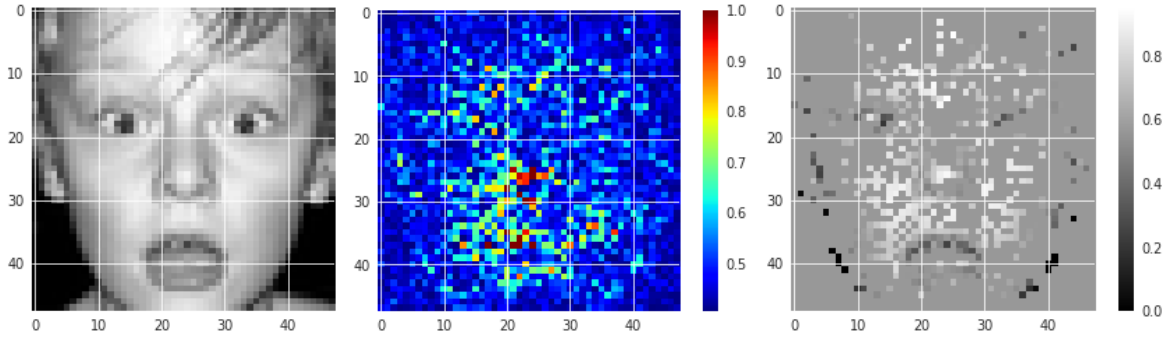
答：



圖六、Saliency maps for Neutral



圖七、Saliency maps for Sad

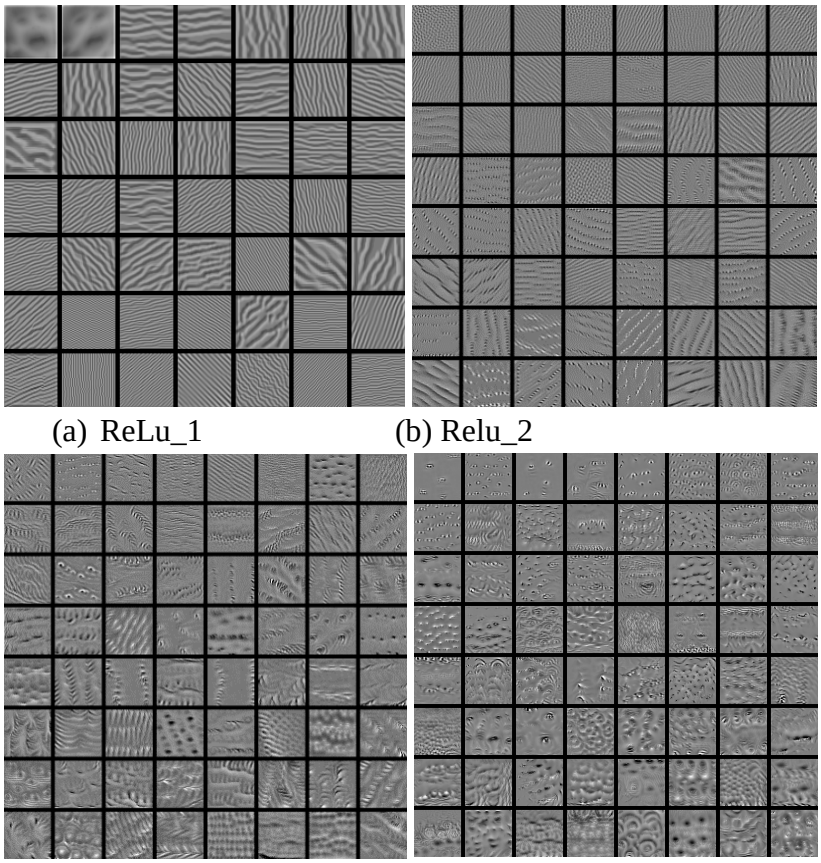


圖八、Saliency maps for Surprise

圖六、圖七和圖八分別繪製出 Neutral, Sad 和 Surprise 的 Saliency。可以發現她們都有 focus 在眼睛和嘴巴的部份。並且顏色較淺的地方在 heatmap 容易有較高的值，例如圖七的背景部份即使不是臉的區域，heatmap 上也有較高的值。

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**。

答：



(a) ReLu_1 (b) Relu_2

(c) ReLu_3 (d) ReLu_4

圖九、Filter Visualization

圖九為四層 activation layer 的 filter visualization，可視為四個 Conv 模組的可視化。每一層 layer 產生出 49 或 64 張 filter visualization。

可以發現(a)和(b) 的輪廓紋理較大面積並且有一致性。有許多紋理相同角度不同的圖片被激活，應該是要抓取各種不同角度的臉部特徵。而(c)(d)則抓到比比較多細部的特徵。例如可以很明顯看到有一堆圈圈像眼睛的圖案，可見得眼睛在這個 task 中是個重要的辨識特徵。而這些圈圈散布在整張圖中可能也是因為眼睛會出現在圖片的不同位置。