

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: )

Input image →

ConvLayer (64 Filters, (3, 3)) → LeakyRelu (alpha=0.05) → MaxPooling → Dropout →  
ConvLayer (128 Filters, (3, 3)) → LeakyRelu (alpha=0.05) → MaxPooling → Dropout →  
ConvLayer (256 Filters, (3, 3)) → LeakyRelu (alpha=0.05) → MaxPooling → Dropout →  
ConvLayer (512 Filters, (3, 3)) → LeakyRelu (alpha=0.05) → MaxPooling → Dropout →  
→ Flatten() →

FullyConLayer (256 units) → Relu → Dropout →

FullyConLayer (512 units) → Relu → Dropout →

Softmax() → output layer

Initializer使用Xavier normal initializer，Optimizer使用Adam，learning rate為預設的0.001。針對每層layer的输出加上Batch Normalization，用來加速training。

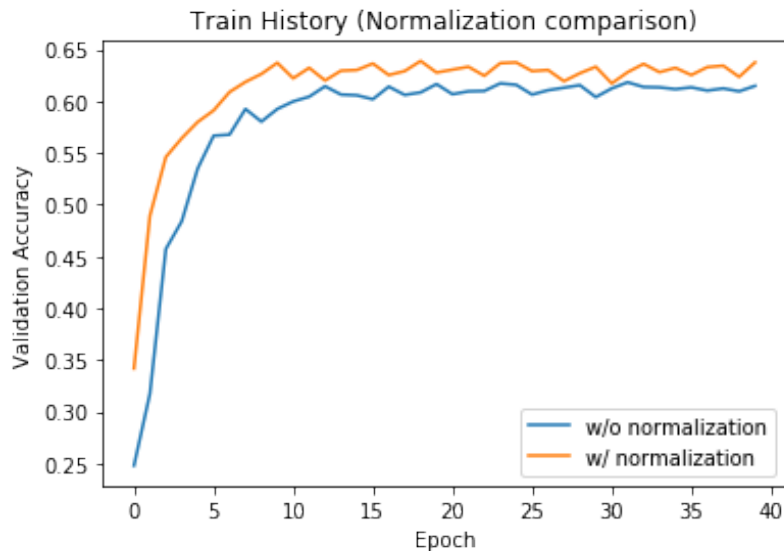
在Validation set上的accuracy為0.69512

在Kaggle上的Testing set的public score為0.69239、private score為0.69517

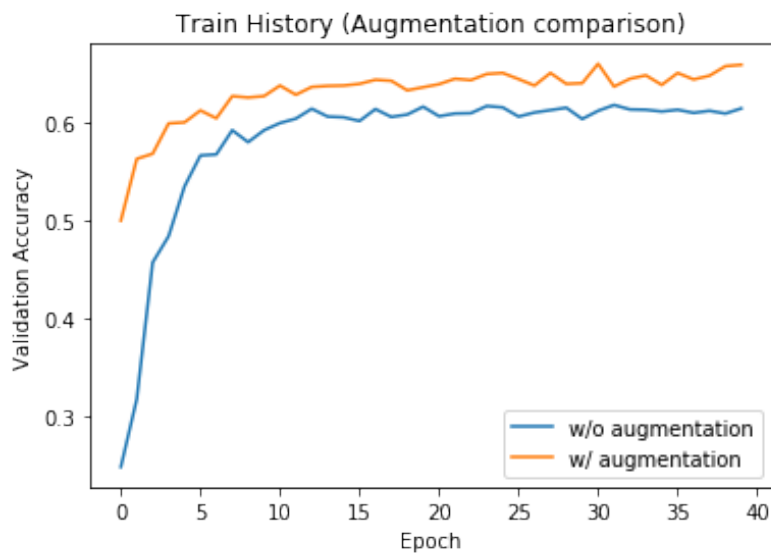
2. (1%) 請嘗試 data normalization, data augmentation,說明實行方法並且說明對準確率有什麼樣的影響？

(Collaborators: )

我使用問題1.中的CNN model，使用相同的training procedure，Train 40個epochs後比較分別實作data normalization和data augmentation的差異。

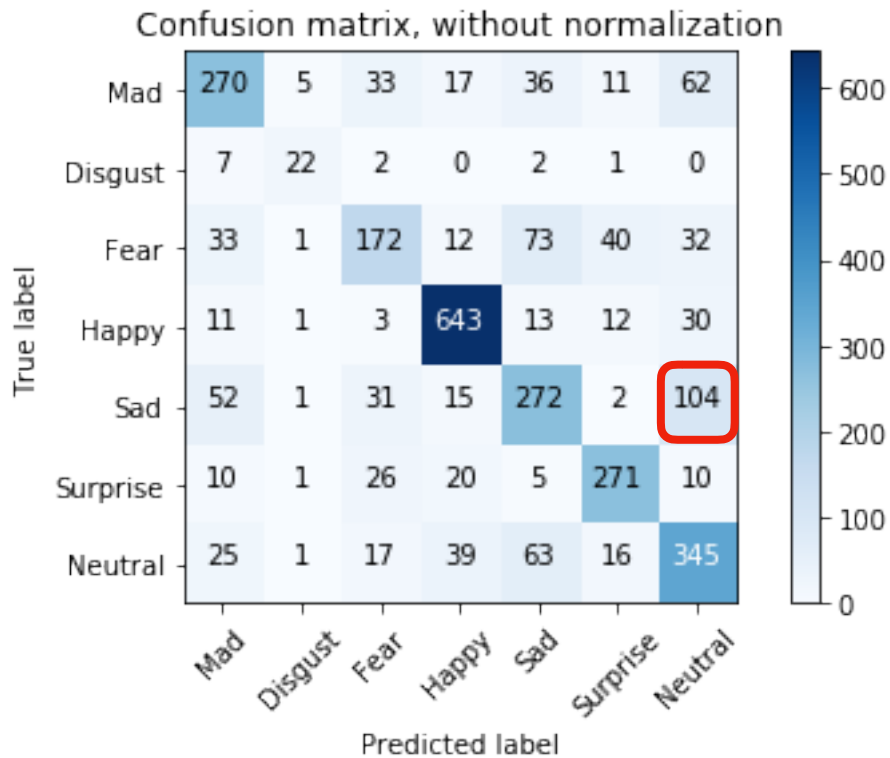


上圖為實作data normalization的比較。我將training data以及testing data全部的features算得一個mean及standard deviation，將每個pixel帶入  $(x - \text{mean}) / \text{std}$  得到mean為0、standard deviation為1的features。由圖中可以看出實作data normalization後的accuracy都高於為實作的accuracy。在validation set上，w/o normalization的accuracy為0.61812，而w/ normalization的accuracy為0.63868，後者accuracy較高。

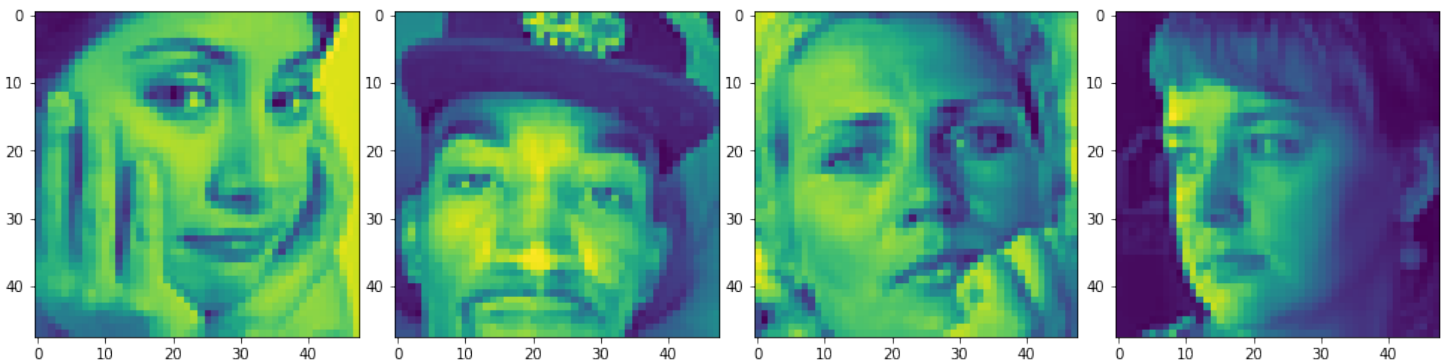


上圖為實作data augmentation的比較。我利用Keras的ImageDataGenerator實作，利用rotation、shear range、zoom range、horizontal flip來產生更多的影像，產生的training data設為原先的3倍。從圖中明顯看出實作data augmentation的accuracy較高，而且提升的幅度比data normalization還要多。在validation set上，w/o augmentation的accuracy為0.61812，而w/ augmentation的accuracy為0.66028，後者accuracy明顯較高。

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: )

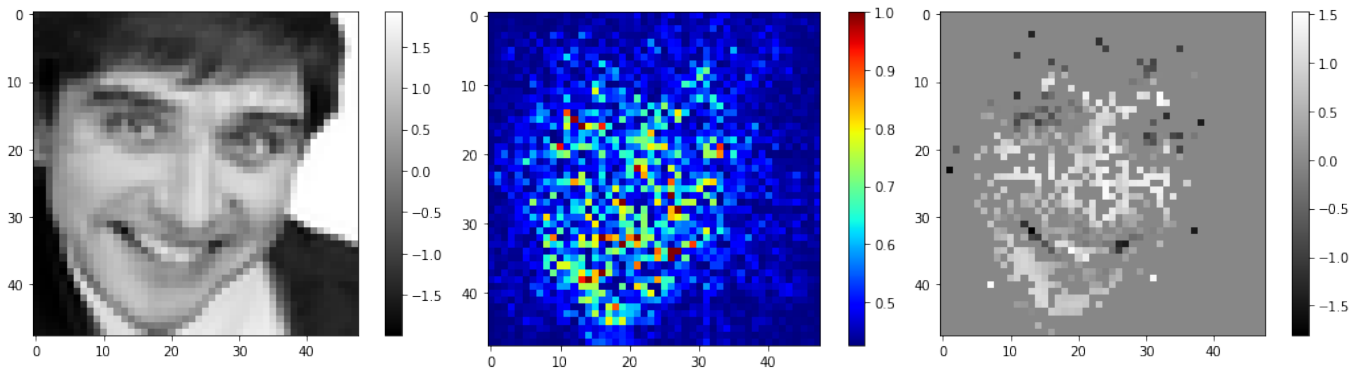


由Confusion matrix可以看出誤判最多的為難過的影像卻分類成中立的影像。選了4張此種誤判的圖形，可以看到雖然影像的原始label為難過，但其實這些表情的難過程並沒有很高，因此肉眼判斷時也可能將它們標示成中立。



Reference: ([http://scikit-learn.org/stable/auto\\_examples/model\\_selection/plot\\_confusion\\_matrix.html#sphx-glr-auto-examples-model-selection-plot-confusion-matrix-py](http://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html#sphx-glr-auto-examples-model-selection-plot-confusion-matrix-py))

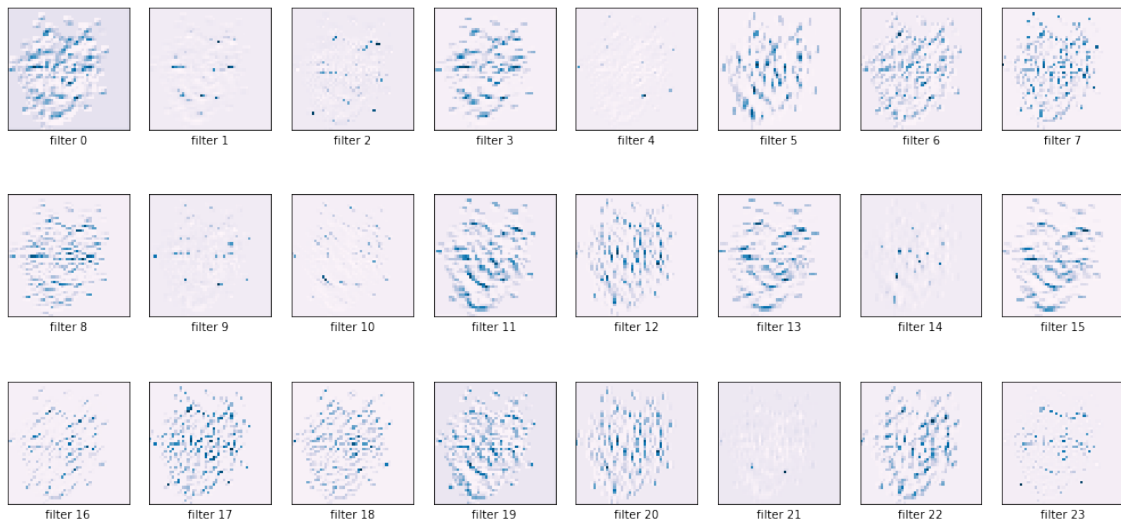
4. (1%) 從(1)(2)可以發現，使用 CNN 的確有些好處，試繪出其 saliency maps，觀察模型在做 classification 時，是 focus 在圖片的哪些部份？  
(Collaborators: )



從saliency maps可以看出模型focus在臉部的眼睛、鼻子、嘴巴這些部位，而這些部位也是表達情緒最重要的部位。

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。(Collaborators: )

利用同上題的影像當作input，選擇第一層convolution layer的前24個filters，可以看到 filters focus在臉部的細紋，包括嘴角的幅度、臉部的線條。



接著我選擇第二層convolution layer的前24個filters，看到focus的部位已經不是細紋了，而是更大的輪廓、臉型。

