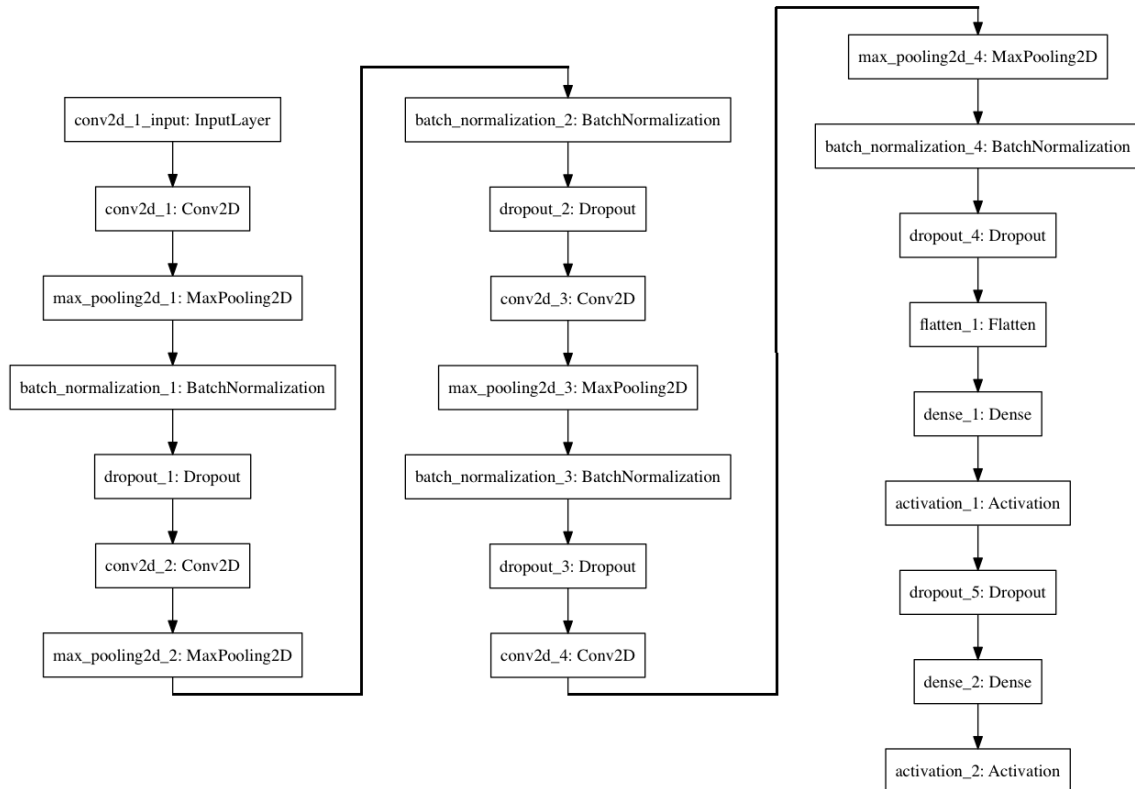


1. (1%) 請說明你實作的 CNN model，其模型架構、訓練過程和準確率為何？

答：



資料處理：

- 刪除第59筆資料 (這筆資料是爛的)
- 對全部資料做 histogram equalization (看網路上的文獻有顯著提升，不過我實際做起來不如預期，但還是有稍微增加正確率)
- 對資料做左右鏡像，使 training data 變兩倍

模型架構：

4層的 Convolution + MaxPooling + BatchNormalization + Dropout

第一層 Convolution 是 (5, 5)，其他都是 (3, 3)

filter 數量依序為 32、128、512、1024

Convolution 的結果都有通過 Activation Function “elu”

MaxPooling 都是 (2, 2)

Dropout rate 依序為 0.5、0.55、0.6、0.65

Flatten 完後接一層 Dense (1024、elu、dropout 0.7)

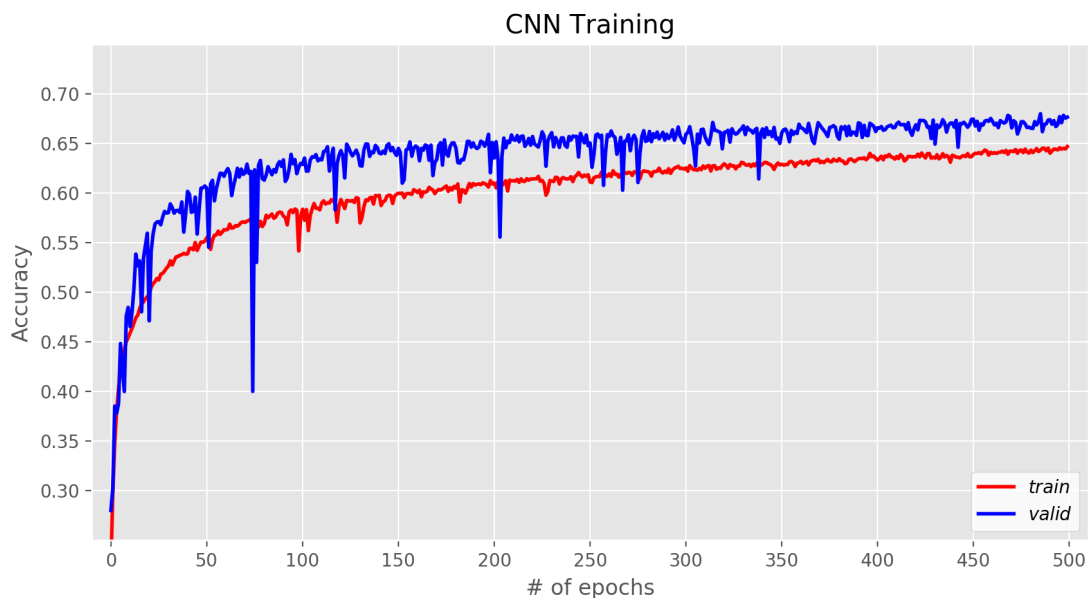
最後接到 output Dense (7、Softmax)

訓練過程：

- 利用 ImageDataGenerator，每個 epoch 在 training 前都會有一些資料被上下左右平移、旋轉，以增加準確率。
- 由於 dropout 設比較大的緣故，使得 training 過程不會太早達到 over-fitting

預測結果：

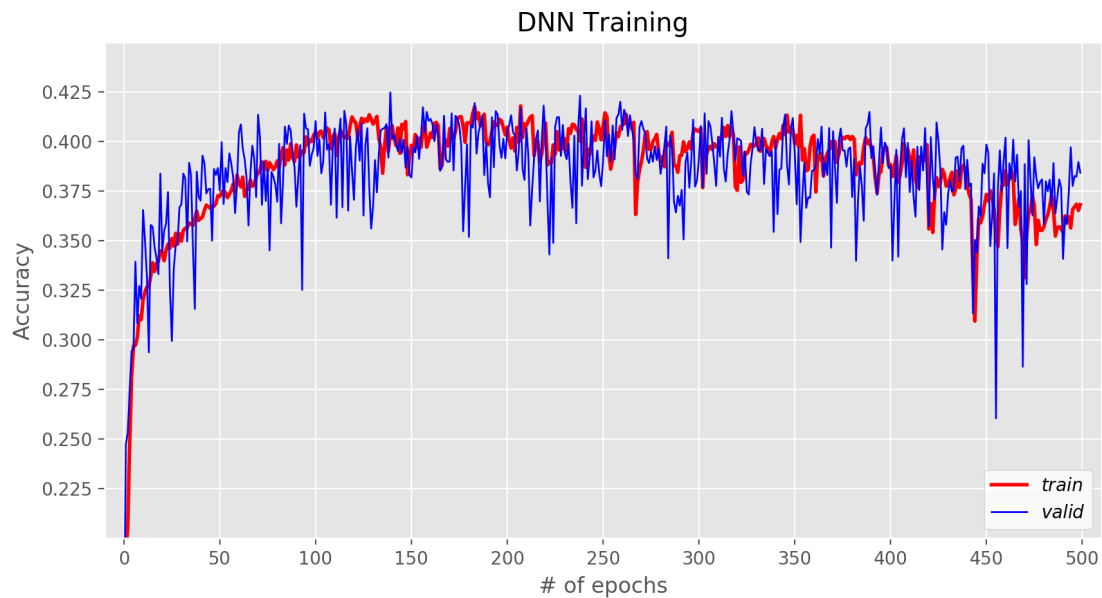
後來發現 train 完數個 model 之後，將各個 model predict 出來的分布做相加取最大值，可以增加 1-2% 正確率，推測有些不容易辨識的圖片經由分佈加成後，不同 model 可以互相彌補的不足，但是也有一個極限。



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：

	參數數量
CNN	6,411,335
DNN	6,565,895



資料處理：

- 刪除第59筆資料 (這筆資料是爛的)
- 對全部資料做 histogram equalization

模型架構：

- 5層的 Dense (1024、elu、dropout 0.5)
- 最後接到 output Dense (7、Softmax)

訓練過程與結果：

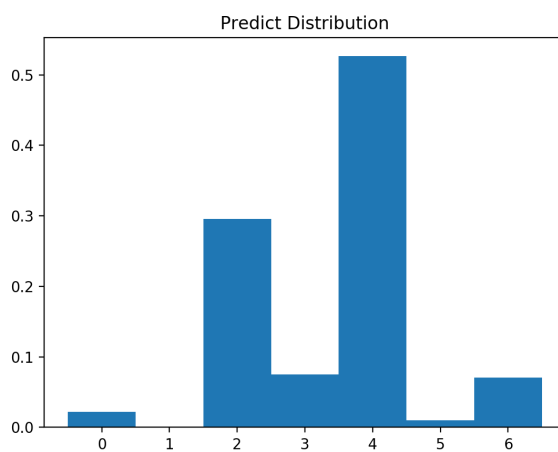
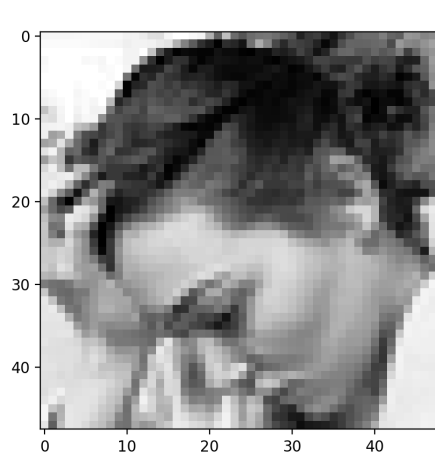
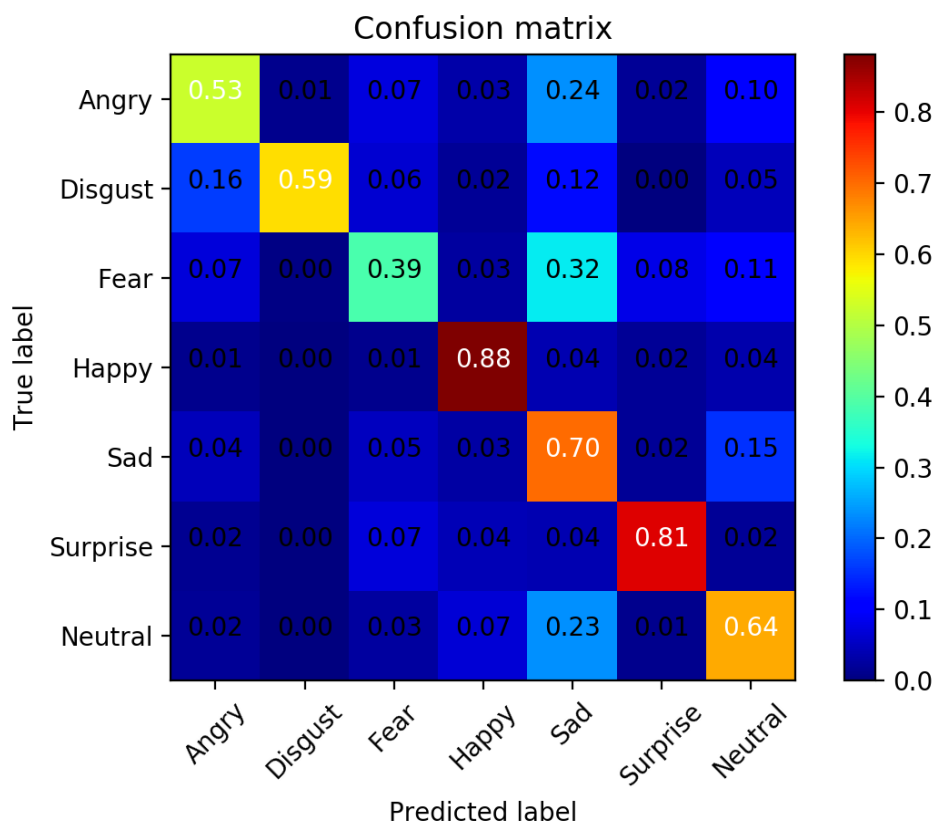
- 不做 reshape 直接把資料拿去的 train
- 發現 DNN train 出來的 accuracy 到 0.4 左右就上不去了，就算 dropout 設大一點也無法提升

比較：

CNN 可以透過 convolution 尋找特定 pattern，再把這些 pattern 拿去 train，這相較於 DNN 直接把每個 pixel 拿去 train，CNN 可以顧慮到附近多個 pixel 的相互關係，能達到較高的正確率。

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]

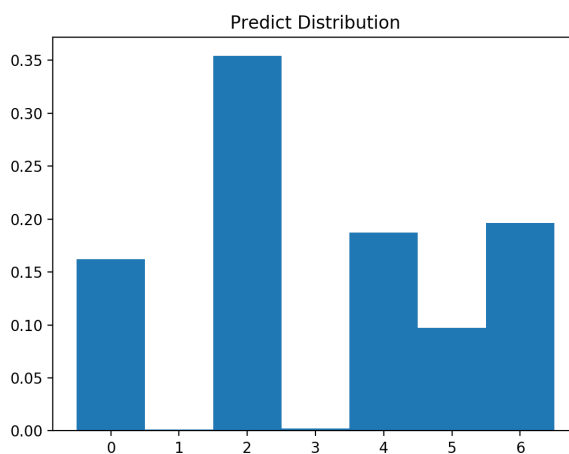
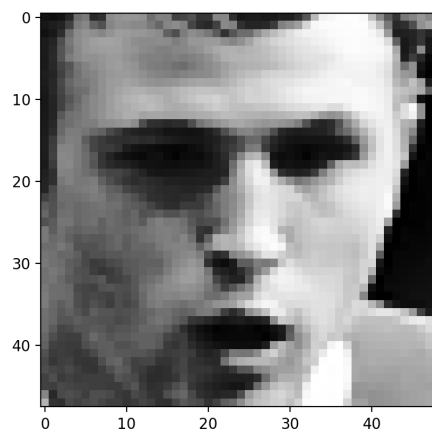
答：Happy、Sad、Surprise 預測的準確率較高，而 Fear 常被誤判為 Sad。  
不過好像有些 training data 的 label 是錯的。



Label:  
Fear(2)

Predict:  
Sad(4)

(這張應該是sad吧...)

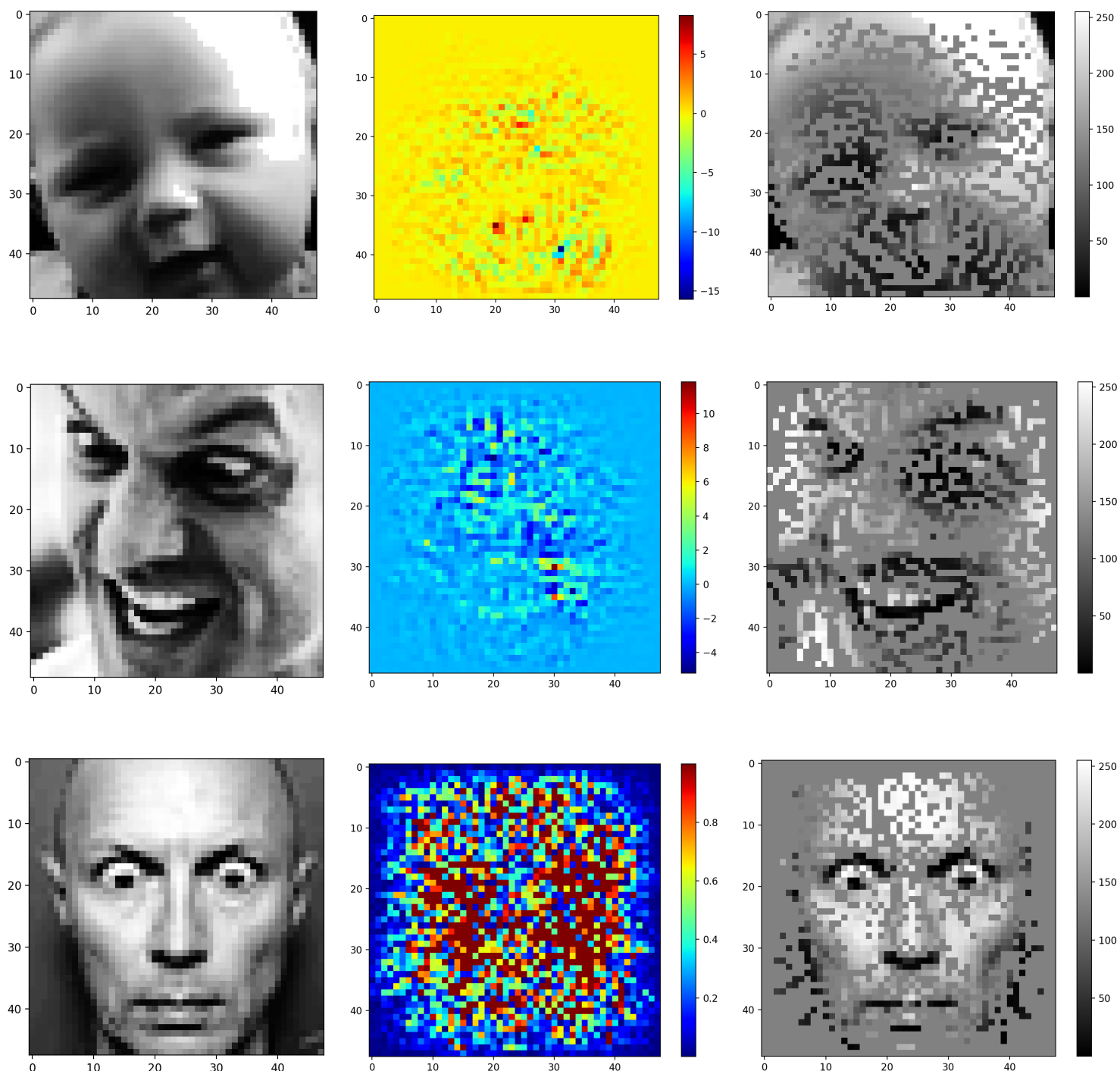


Label:  
Surprise(5)

Predict:  
Fear(2)

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？

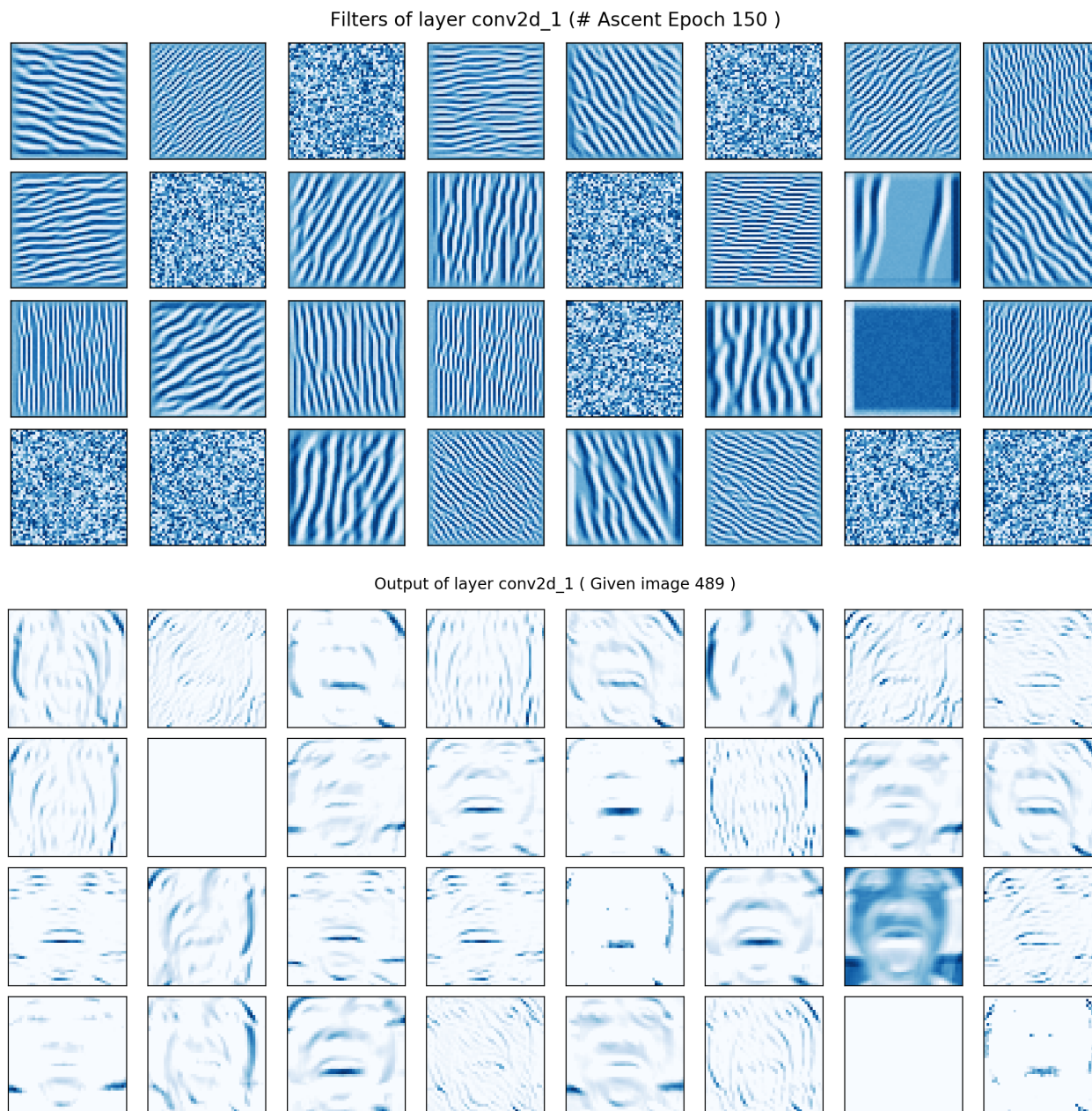
答：



前兩張的 heatmap 沒做任何處理，而第三張把 grad 取絕對值再放到  $[0, 1]$  之間。  
我覺得機器有學習到眼睛與嘴巴的辨識對表情分類有較大的幫助，故 focus 在這些部位較多，而且在臉部的偵測上感覺做的不錯。

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：



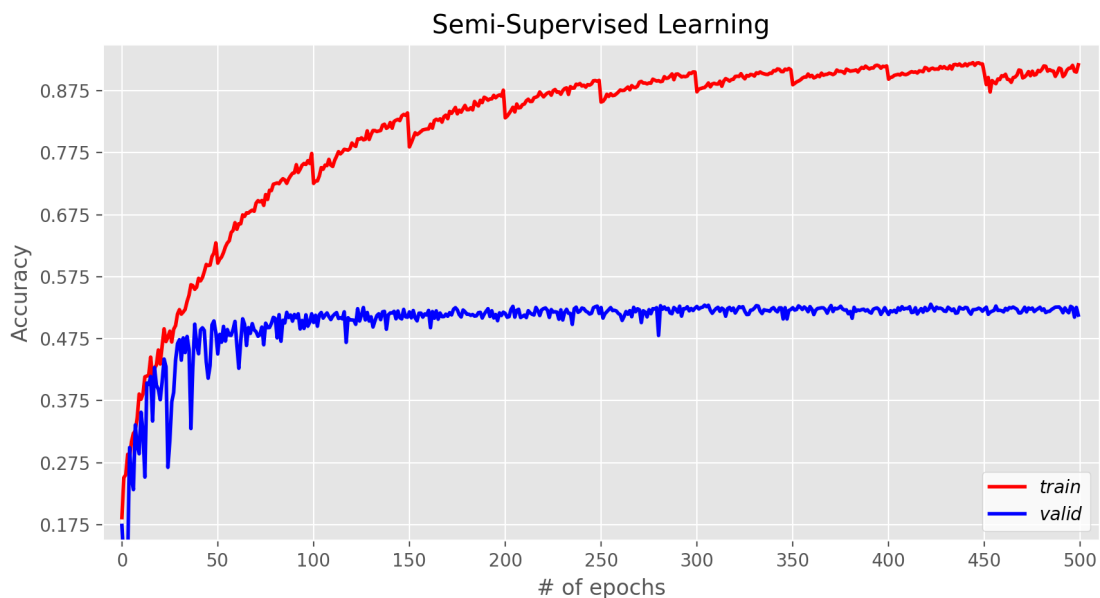
某些 filter 濾出來的圖片感覺像是雜訊，又有些看起來很雷同，我覺得是我 filter 開的太多，導致有許多尋找的 pattern 相似，而且 dropout 設太大，導致有某些 filter 無法被正確完整的 train 到。

[Bonus] (1%) 從 training data 中移除部份 label，實做 semi-supervised learning

答：

我將原本 training data 切出一小塊當 validation 後，將剩餘的分成十等分，只有其中的一等份留有 label，當做第一組的 training set，隨後每 train 完 50 個 epochs 後依序拿一等份去 predict 然後加到 training set 繼續 train，直到所有 training data 跑遍為止。最後 evaluate 原本完整 training data 跟原先的 label 後，發現實際上 accuracy 只有 **0.5584**，與最後 validation accuracy **0.5130** 相差不遠。推測是部分資料在 training 過程中就被分類分錯導致，跟預期結果相近。

註. 因為時間因素我沒有取 predict 完分佈較佳的 data 加到 training set 中，如果有實作這部分 accuracy 應該會再上升。



[Bonus] (1%) 在 Problem 5 中，提供了 3 個 hint，可以嘗試實作及觀察 (但也可以不限於 hint 所提到的方向，也可以自己去研究更多關於 CNN 細節的資料)，並說明你做了些什麼？ [完成 1 個: +0.4%, 完成 2 個: +0.7%, 完成 3 個: +1%]

以下是各個 label 的最能被 activate 的圖片。

Image Activate Angry the Most

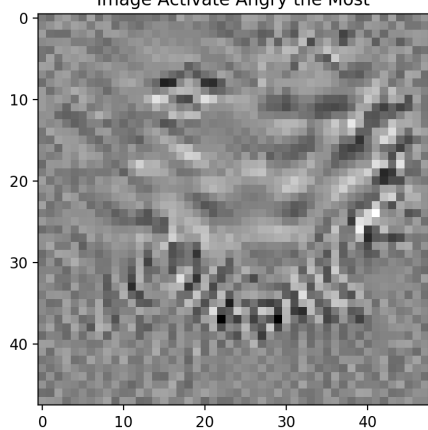


Image Activate Disgust the Most

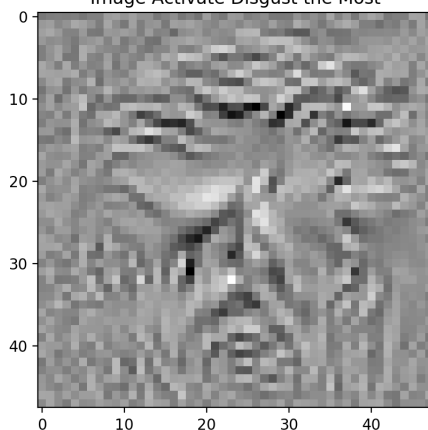
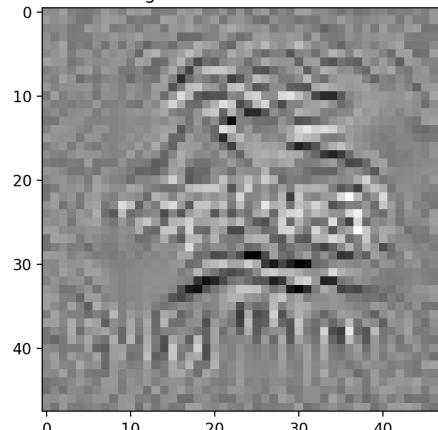
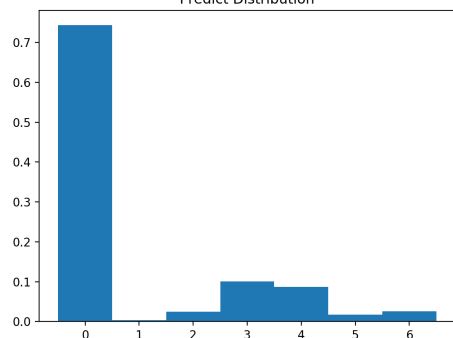


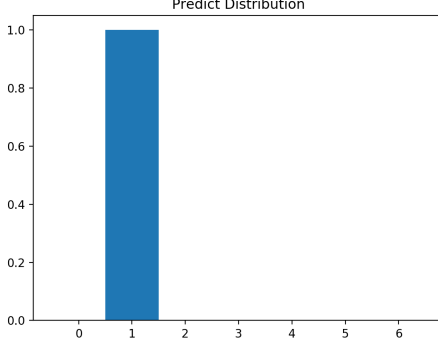
Image Activate Fear the Most



Predict Distribution



Predict Distribution



Predict Distribution

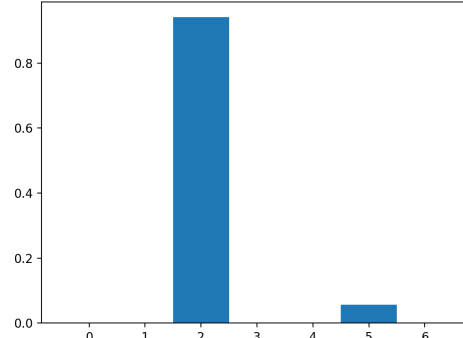


Image Activate Happy the Most

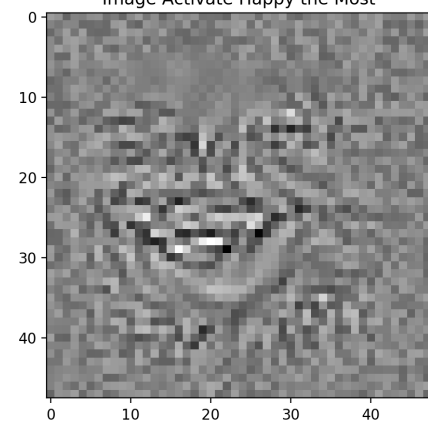


Image Activate Sad the Most

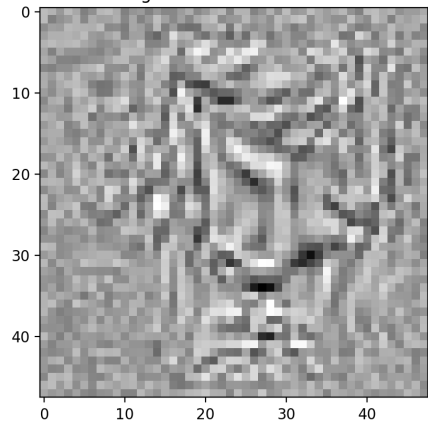
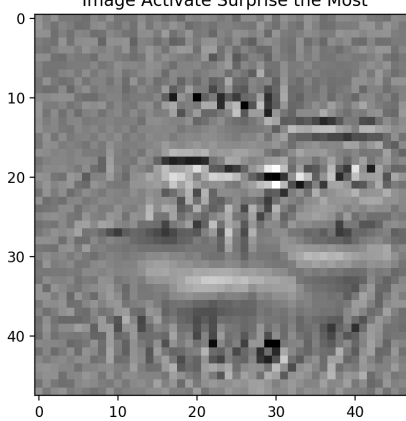
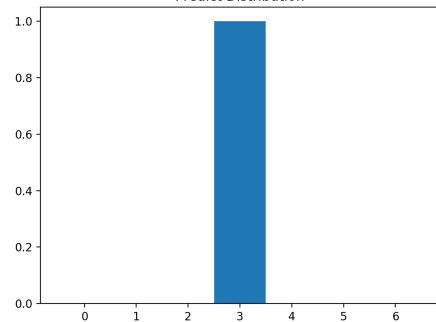


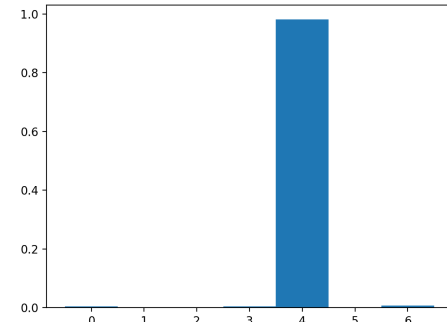
Image Activate Surprise the Most



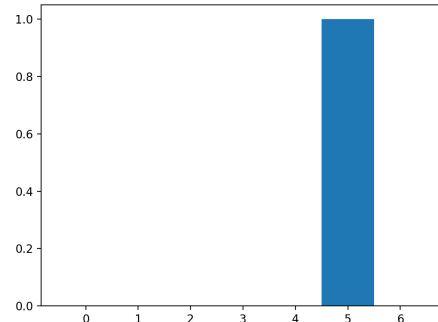
Predict Distribution



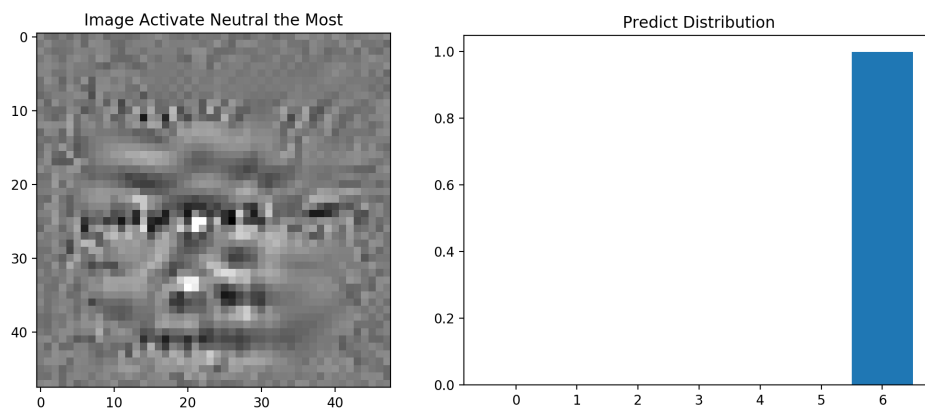
Predict Distribution



Predict Distribution







比較 best model 與其他 worse performance model :

- filter : 當 filter 數量越多時, predict accuracy 也會跟著提升, 不過在找較小 pattern 時 (較上層的 convolution), filter 太多反而會降低 accuracy。而在分析 filter 的影像時, dropout 較大造成很多 filter 看起來像是雜訊 (看不出 pattern)
- dropout : 當 dropout 設越大, accuracy 能穩定上升避免 over-fitting
- layer : 層數越多未必越好, 尤其是後面 dense 的部分, 設的越大反而使 model 的參數直接 fit training data, 使 predict 效果不彰
- 多個 CNN concatenate : 剛開始有變好, 但隨著 layer 和 filter 數量的增加, 整個 model 過於龐大, 且 training 所耗費的時間劇增, 又沒有較好的提升
- 多個 model 分佈相加 : 此作法能有點像「從眾」的感覺, 某個 model 不太確定的圖形依靠別的 model 的結果來做判斷, 以增加整體準確率