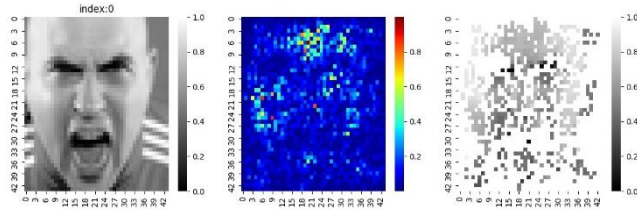
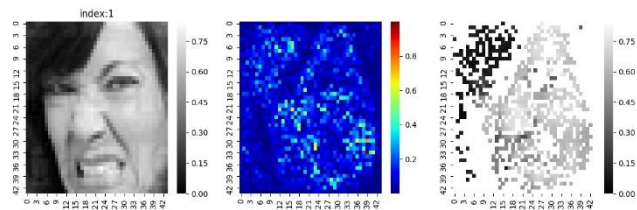


學號：B04104040 系級：工海三 姓名：解正安

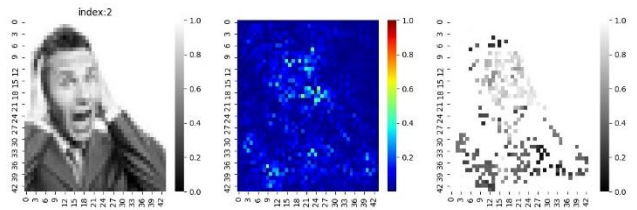
1. (2%) 從作業三可以發現，使用 CNN 的確有些好處，試繪出其 saliency maps，觀察模型在做 classification 時，是 focus 在圖片的哪些部份？(Collaborators:None)



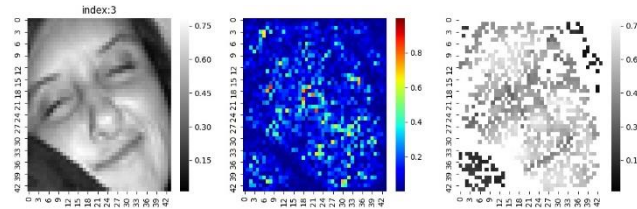
Index:0
著重在眉毛形狀還有額頭



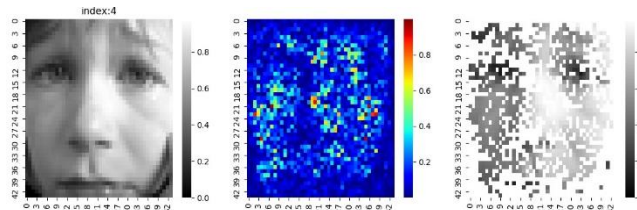
index:1 主要卻是在臉頰凹陷還有少部分的仁中跟鼻子影響，頭髮應該是誤判



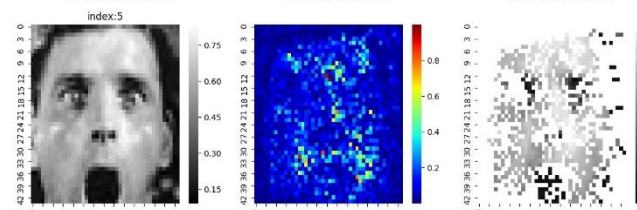
Index:2 恐懼，著重在嘴型、眼神還有手勢



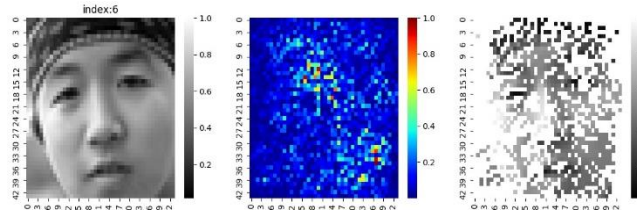
index:3 高興，著重在嘴部的形狀還有酒窩



index:4 悲傷，著重在眼神(眼淚?)和臉頰

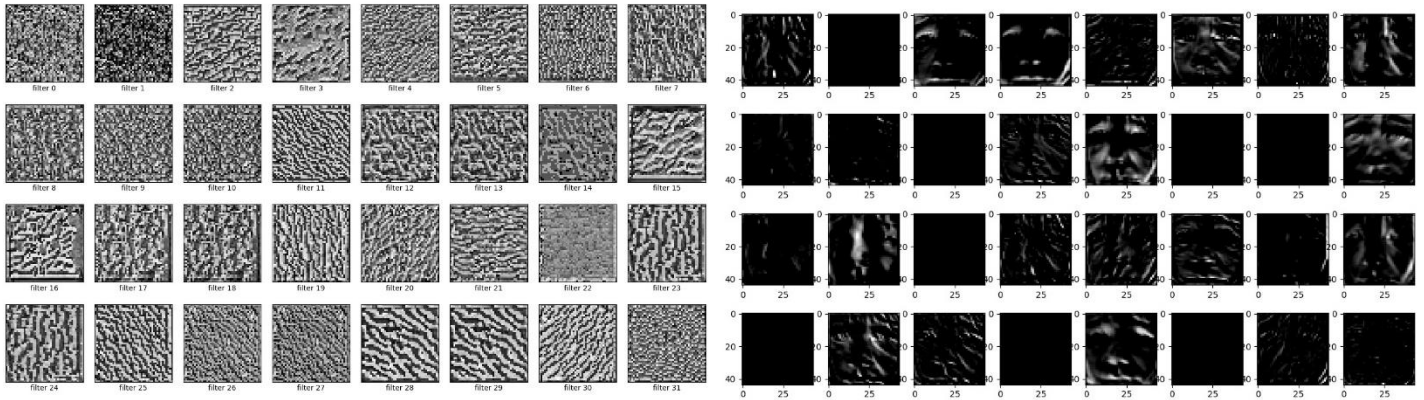


index:5 驚訝，著重在嘴型和眼睛



index:6 中立，著重整個面部表情

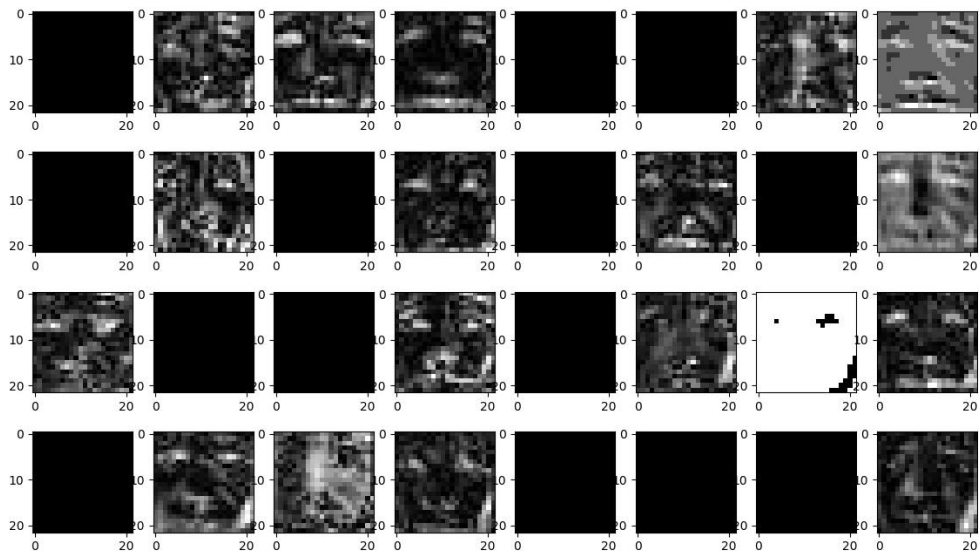
2. (3%) 承(1) 利用上課所提到的 gradient ascent 方法，觀察特定層的 filter 最容易被哪種圖片 activate 與觀察 filter 的 output。(Collaborators:None)



答：filter 有 64 個，在此只取前 32 個做為觀察

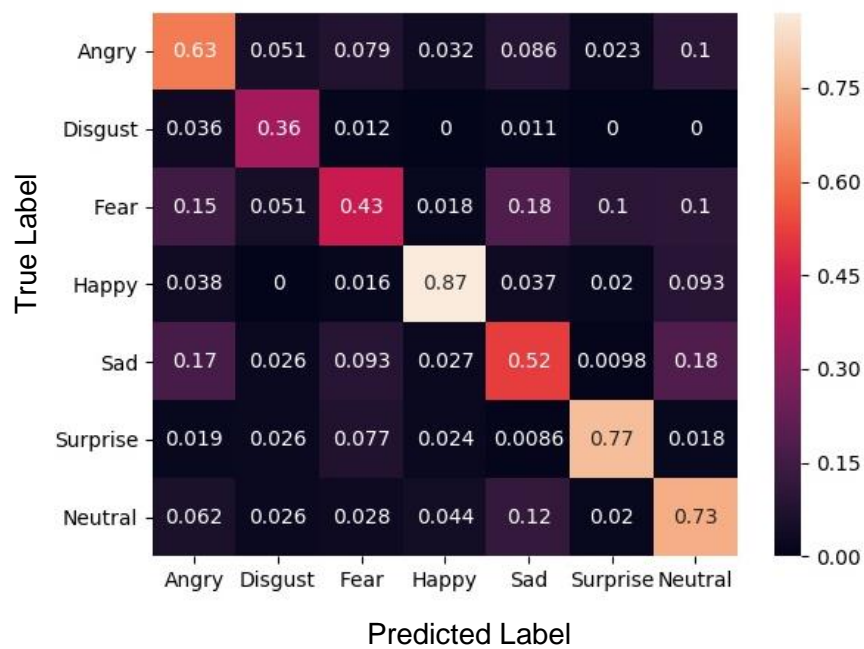
採用第二層 conv2D 的圖形，input 圖形為 index4 悲傷的圖形。由結果可以發現 filter 有點點、垂直線、水平線和斜線等 filter 方式並顯現在 output 的圖形上。此外，因為每張圖片拍攝方式和角度不同，filter 若能涵蓋更多方向角度可以讓結果更好。

而仔細觀察右方的 output 圖片，可以發現第二層 conv2D 中 activate layer 的圖片通常是線條較明顯的圖形，因此線條如果較粗結果較好。若取第三層 conv2D 且經過一次 maxPooling(如下圖)，會發現著重在面積上，此外圖形結果有點難以判斷機器是如何分辨的。



3. (3%) 請使用 Lime 套件分析你的模型對於各種表情的判斷方式，並解釋為何你的模型在某些 label 表現得特別好 (可以搭配作業三的 Confusion Matrix)。

答：



Index:0



Index:1



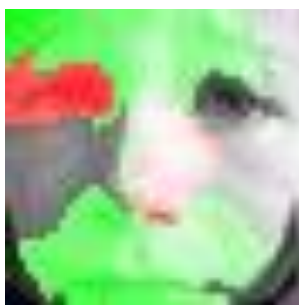
Index:2



Index:3



Index:4



Index:5



Index:6



由 **lime** 可以知道，在生氣時著重在嘴型和眼神；噁心考慮鼻子以下的面部表情；害怕考慮額頭還有誤判的衣服；快樂依據眉頭還有嘴型跟眼睛判斷；傷心則是眼睛和嘴吧；驚訝在眼睛跟 o 形嘴；中立則是整個面部表情。

若配合 **confusion matrix** 觀察，噁心和害怕兩個準確率偏低，推測是因為兩個面部表情和其他有類似的地方，像是皺眉、眼睛變小等等。此外，害怕的這張圖，衣服剛好有皺褶，機器可能因此視為人在害怕時的皺眉，導致最後結果不佳。

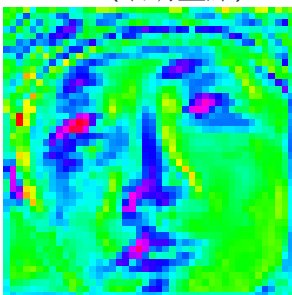
4. (2%) [自由發揮] 請同學自行搜尋或參考上課曾提及的內容，實作任一種方式來觀察 CNN 模型的訓練，並說明你的實作方法及呈現 **visualization** 的結果。

答：

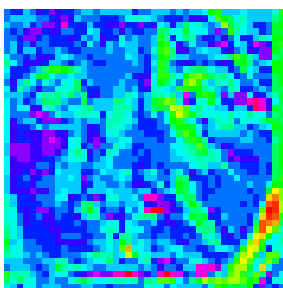
使用 **Grad-Cam** 的方式，已知每個 **feature** 有自己的權重，取出每個 **feature** 和權重就能疊加到原圖上，作出類似 **saliency map** 的效果。而取得權重的方式，**grad-cam** 是利用 **grad** 在整個訓練的平均去計算，加權後即是熱力圖。

利用 **pytorch** 的 **hook** 去取得 **feature**，以下的層數為第二層 **conv2D** 取 **RELU** 厚的結果，可以發現比起上述 **gradient ascent** 方法，更能看出在該層中，較為重要的 **feature** 是哪一部分。照片同第三題。

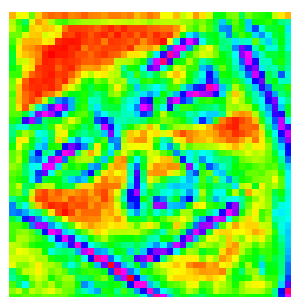
Index=6(眼睛重點)



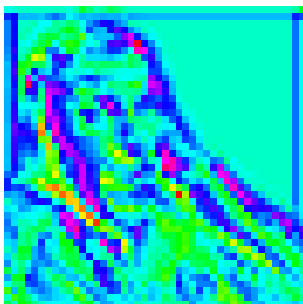
index=4(嘴巴)



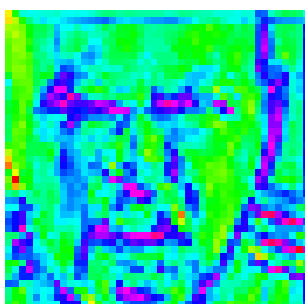
index=3(竟然是額頭跟雙頰)



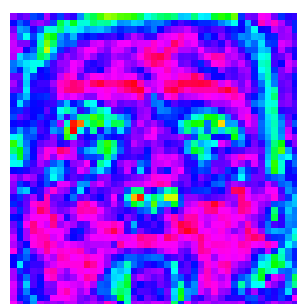
Index=2(我覺得錯了)



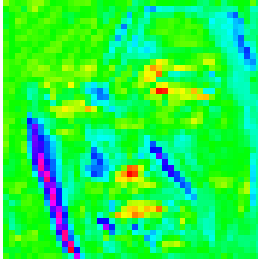
index=0 (五官都重要)



index=5(只看臉部皮膚...)



Index=1(眼鼻口)



公式:

$$L_{\text{Grad-CAM}}^c = \text{ReLU} \left(\underbrace{\sum_k \alpha_k^c A^k}_{\text{linear combination}} \right)$$

参考:

Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization
(Ramprasaath R. Selvaraju¹, Michael Cogswel, Devi Parikh)