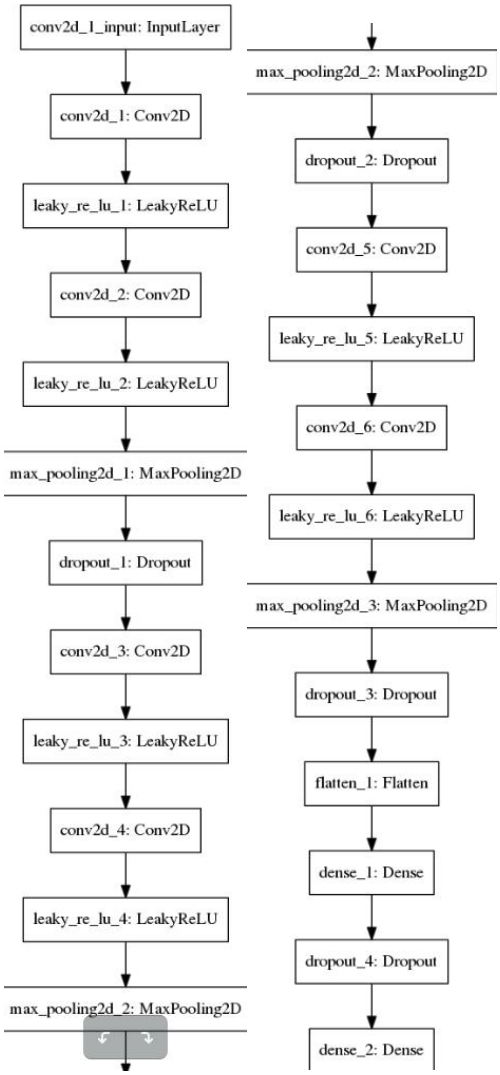
學號：r05922145 系級： 資工碩二 姓名：郁錦濤

1. (1%) 請說明你實作的 CNN model，其模型架構、訓練過程和準確率爲何？ (Collaborators: None)

答：我實作的 model 參數共 8,673,351 個，用了 6 层 conv2d，每一層後 activation function 采用了 LeakyReLu(這樣能夠避免 neural node 出現 dead 的現象)，然後每兩個 conv2d 添加一個 max_pooling2d 和 dropout。此外，我將原來的 training data 的後 4000 筆用作 valid data，其他 data 採用了 image_generator 通過圖片翻轉、旋轉來產生新的 data，增大 training data 的數量，總共訓練 1000 個 epoch。因每次訓練 1000 個 epoch 需要 10 個小時，耗費時間巨大，所以後面的 training procedure 我只訓練了 400 個 epoch 進行分析。

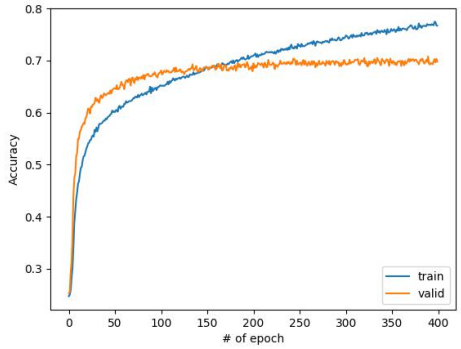
1)模型架構如下：

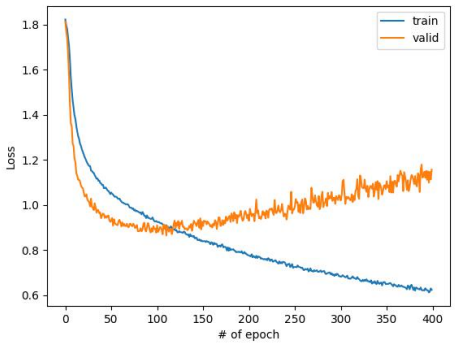


Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 48, 48, 64)	640
leaky_re_lu_1 (LeakyReLU)	(None, 48, 48, 64)	0
conv2d_2 (Conv2D)	(None, 46, 46, 64)	36928
leaky_re_lu_2 (LeakyReLU)	(None, 46, 46, 64)	0
max_pooling2d_1 (MaxPooling2	(None, 23, 23, 64)	0
dropout_1 (Dropout)	(None, 23, 23, 64)	0
conv2d_3 (Conv2D)	(None, 23, 23, 128)	73856
leaky_re_lu_3 (LeakyReLU)	(None, 23, 23, 128)	0
conv2d_4 (Conv2D)	(None, 21, 21, 256)	295168
leaky_re_lu_4 (LeakyReLU)	(None, 21, 21, 256)	0
max_pooling2d_2 (MaxPooling2	(None, 10, 10, 256)	0
dropout_2 (Dropout)	(None, 10, 10, 256)	0
conv2d_5 (Conv2D)	(None, 8, 8, 512)	1180160
leaky_re_lu_5 (LeakyReLU)	(None, 8, 8, 512)	0
conv2d_6 (Conv2D)	(None, 6, 6, 512)	2359808
leaky_re_lu_6 (LeakyReLU)	(None, 6, 6, 512)	0
max_pooling2d_3 (MaxPooling2	(None, 3, 3, 512)	0
dropout_3 (Dropout)	(None, 3, 3, 512)	0
flatten_1 (Flatten)	(None, 4608)	0
dense_1 (Dense)	(None, 1024)	4719616
dropout_4 (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 7)	7175

Total params: 8,673,351
Trainable params: 8,673,351
Non-trainable params: 0

2)訓練過程如下（400 個 epoch）：





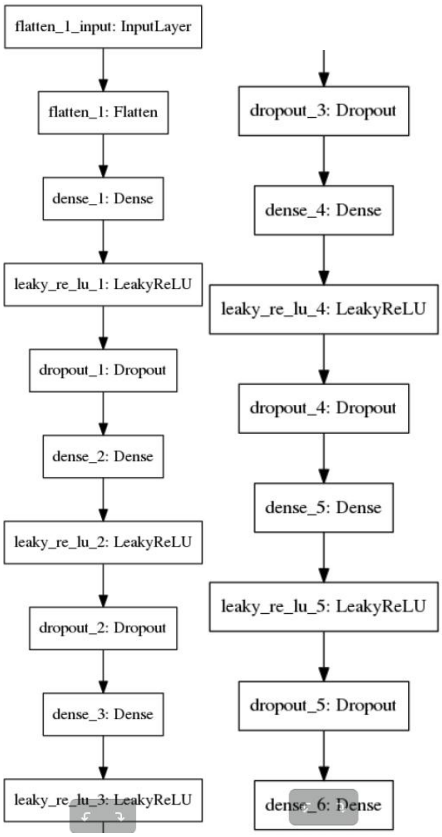
3)準確率：在 valid data 上有 0.70120 的準確率，在 kaggle 的 public set 上的準確率爲 0.70214。

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、訓練過程和準確率爲何？試與上題結果做比較，並說明你觀察到了什麼？

(Collaborators: None)

答：DNN 實作 model 中我採用了 6 層 Fully Connected，分別用 1024 和 2048 的節點進行 connect，最後使得 DNN model 的參數達到 8,664,071 多個。在訓練過程中，我發現 CNN 和 DNN 的 model 有以下兩個差別。首先，直觀感受是，DNN 的模型在大致相同參數的情況下訓練速度比較快，1000 個 epoch 只需要 2 個小時即可訓練完成，CNN 的模型則訓練時間較長，在 1000 個 epoch 中可以達到 10 個小時的 training 時間。但是，另外一個不同的是，CNN 訓練出來的模型準確率遠高於 DNN，DNN 訓練出的模型不容易收斂，在 valid set 上表現不佳。

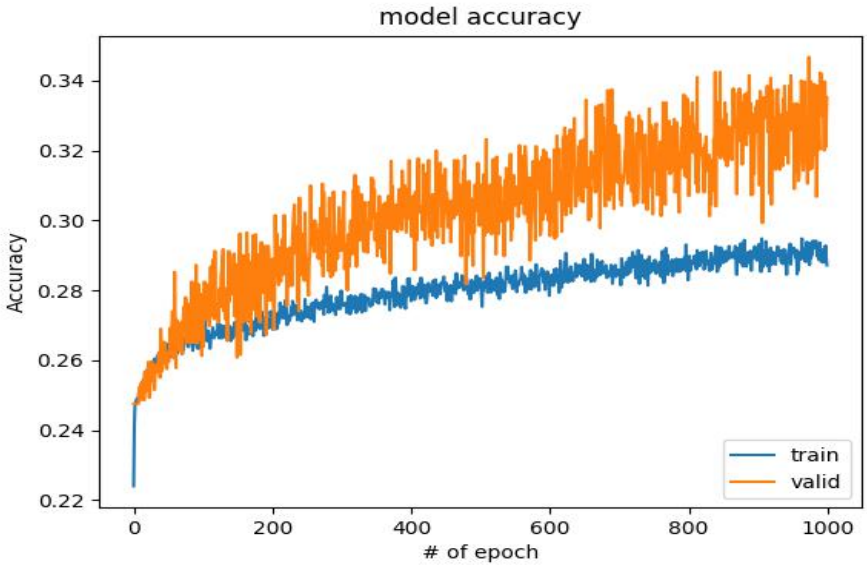
1)模型架構如下：



Layer (type)	Output Shape	Param #
flatten_1 (Flatten)	(None, 2304)	0
dense_1 (Dense)	(None, 1024)	2360320
leaky_re_lu_1 (LeakyReLU)	(None, 1024)	0
dropout_1 (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 1024)	1049600
leaky_re_lu_2 (LeakyReLU)	(None, 1024)	0
dropout_2 (Dropout)	(None, 1024)	0
dense_3 (Dense)	(None, 1024)	1049600
leaky_re_lu_3 (LeakyReLU)	(None, 1024)	0
dropout_3 (Dropout)	(None, 1024)	0
dense_4 (Dense)	(None, 2048)	2099200
leaky_re_lu_4 (LeakyReLU)	(None, 2048)	0
dropout_4 (Dropout)	(None, 2048)	0
dense_5 (Dense)	(None, 1024)	2098176
leaky_re_lu_5 (LeakyReLU)	(None, 1024)	0
dropout_5 (Dropout)	(None, 1024)	0
dense_6 (Dense)	(None, 7)	7175

Total params: 8,664,071
Trainable params: 8,664,071
Non-trainable params: 0

2)訓練過程如下：



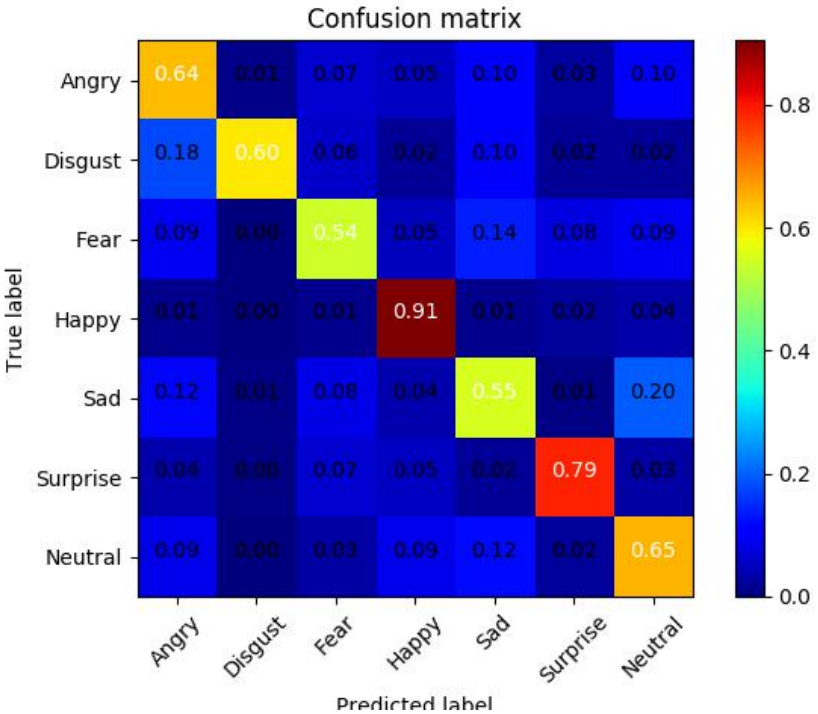
3)準確率：在 valid set 上只有 0.3353 的準確率，且準確率不易收斂，還有較大幅度的搖擺。

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: None)

答：觀察 confusion matrix 進行分析，發現 disgust 和 angry， fear 和 sad，sad 和 angry，

neutral 和 sad 之間 rate 都大於 0.1，尤其是 disgust 和 angry，fear 和 sad。以 disgust 和 angry 爲例子，這兩種情感分別爲厭惡和生氣，這兩種情感在人的主觀情感上來看是比較接近的，所以人工分類時也會有一定的誤差，在進過 model 訓練，也會有誤差產生。



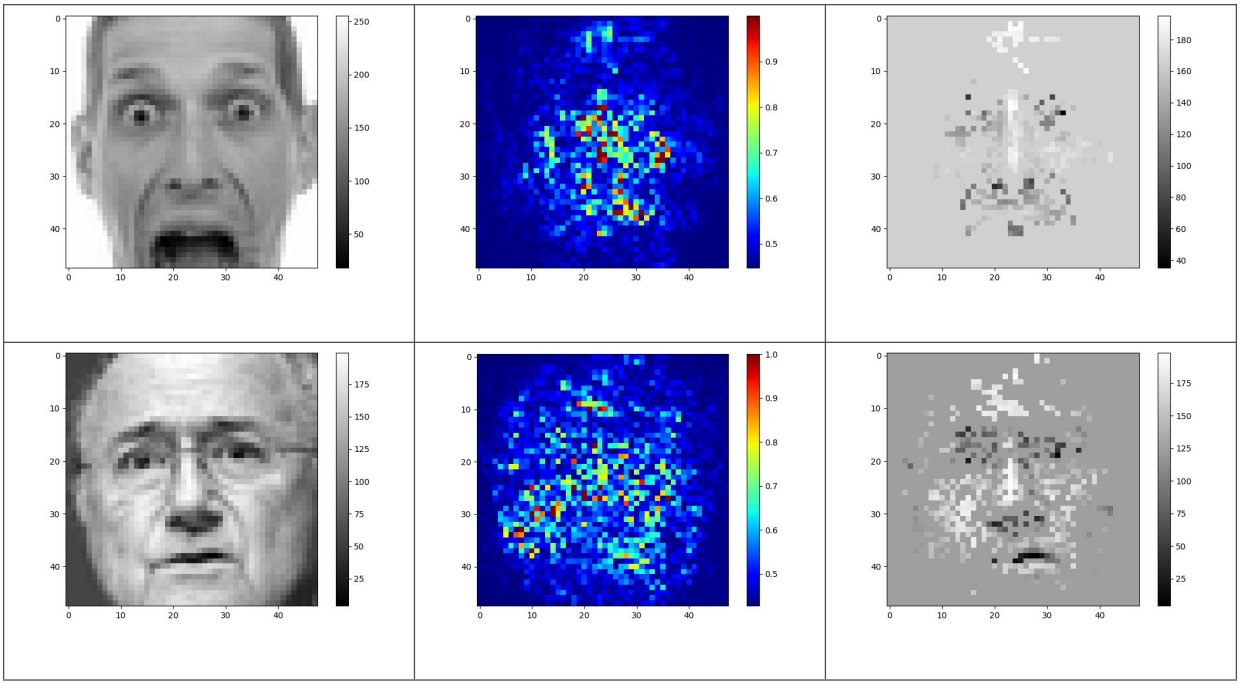
4. (1%) 從(1)(2)可以發現，使用 CNN 的確有些好處，試繪出其 saliency maps，觀察模型在做 classification 時，是 focus 在圖片的哪些部份？
(Collaborators: None)

答：我從 training data 中取出第 5 和 9 張圖片進行比較，二者都是恐懼的表情。從下圖中我們可以看出，在經過 saliency maps 以後，表情中的眼睛、鼻子、嘴巴和鼻孔周圍具有很高的 heat，通過去掉 heat 以後，我們可以發現，在臉部的紋路處（皺紋）一般特徵非常明顯，比如眼角周圍，鼻樑周圍的皺紋，嘴巴附近的皺紋處都可以清楚地從最後一列圖片中看見顯著的特徵。

除此以外，在 classfication 時，臉部的主要器官，如鼻子、眼睛、嘴巴都能夠快速保留特徵。

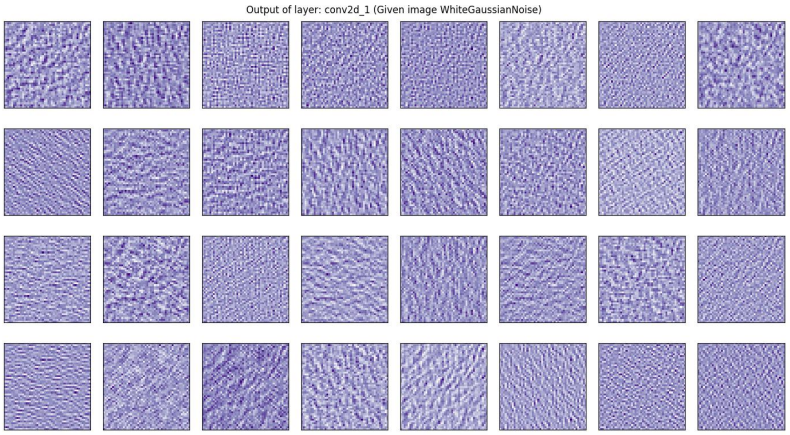
因此，我覺得 focus 在圖片的一些臉部器官和紋路。

原圖	Saliency Map	Mask 掉 heat 小的部分

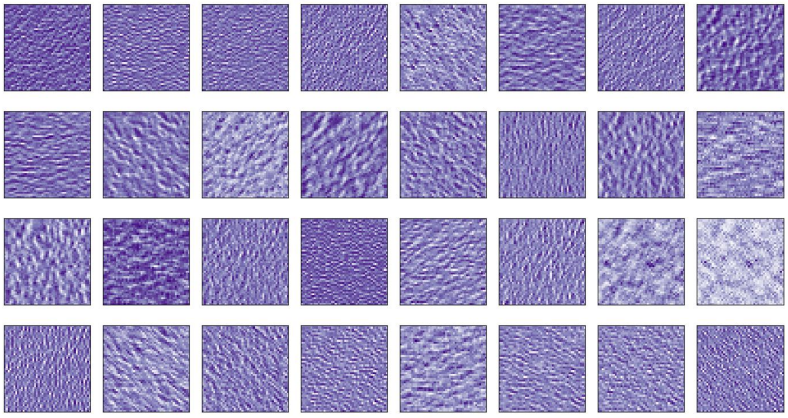


5. (1%) 承(1)(2)，利用上課所提到的 gradient ascent 方法，觀察特定層的 filter 最容易被哪種圖片 activate。
(Collaborators: None)
答：

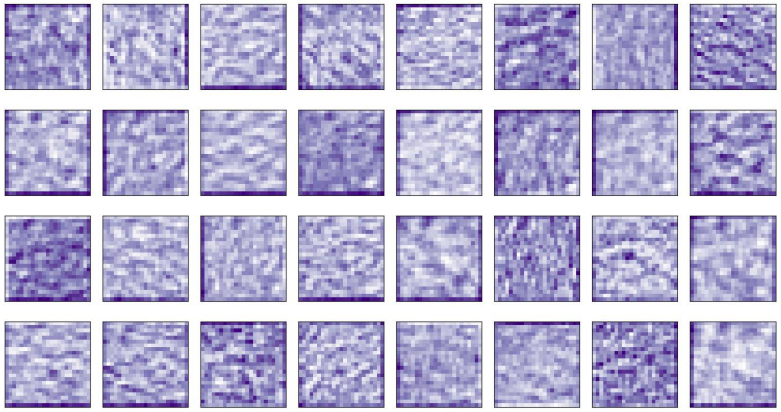
1）使用高斯白噪聲進行輸入，觀察每個 conv2D 層共計 6 層的輸出，觀察其特點，發現每個層的輸出包含了各個方向的紋路，應該是用來鑑別圖像的主要輪廓特徵。從第一層到第六層，圖片紋理慢慢變得模糊，但是後面幾層慢慢顯示出一些輪廓。從第一層可以看出，該層 filter 主要是用來抓取圖像的細致特徵，主要是圖像周圍的紋理方向和變化。



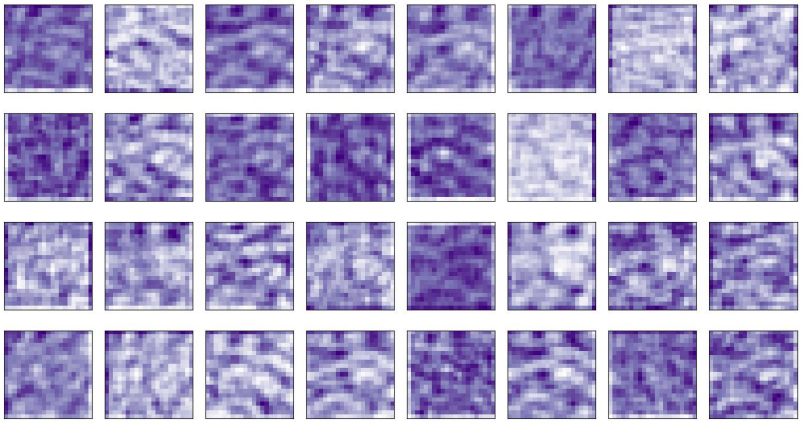
Output of layer: conv2d_2 (Given image WhiteGaussianNoise)

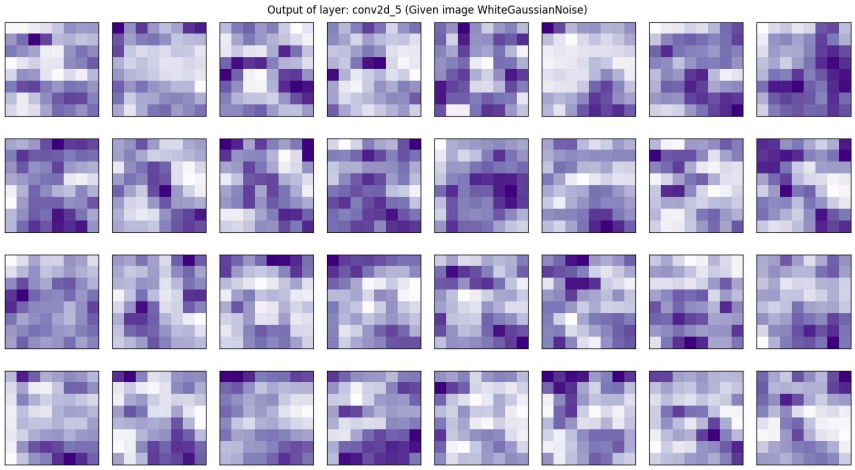


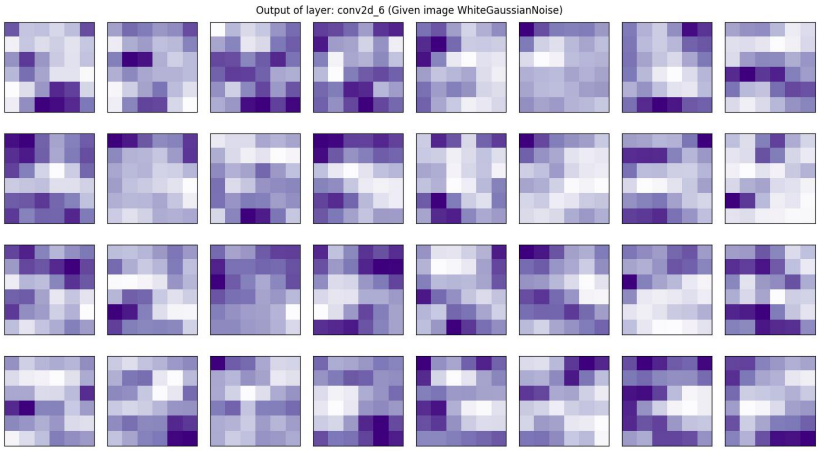
Output of layer: conv2d_3 (Given image WhiteGaussianNoise)



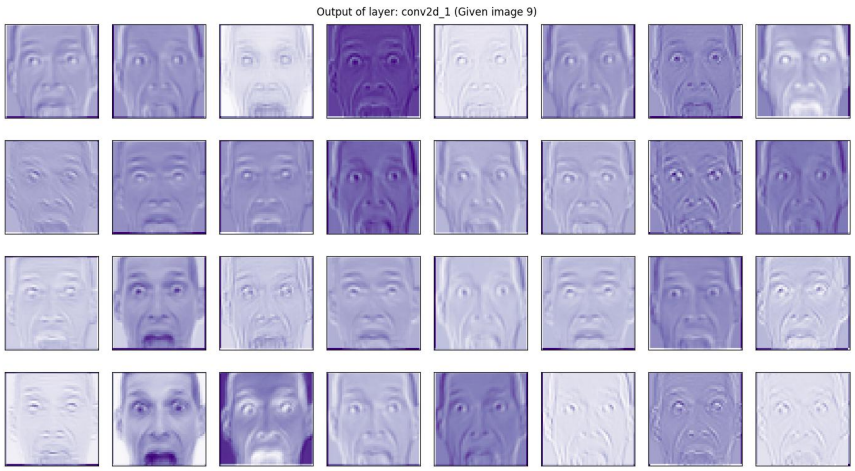
Output of layer: conv2d_4 (Given image WhiteGaussianNoise)



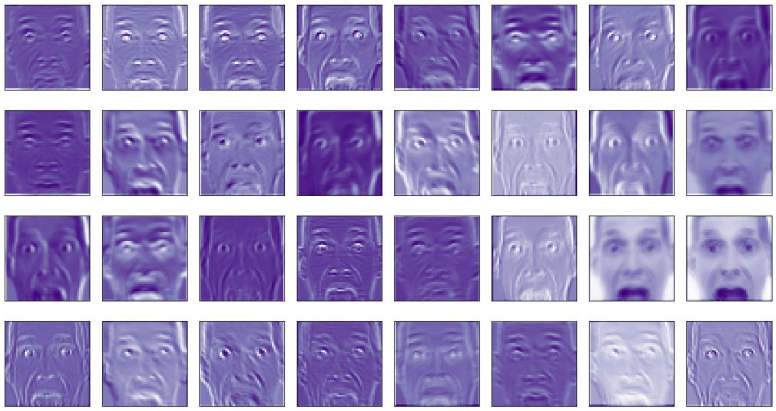




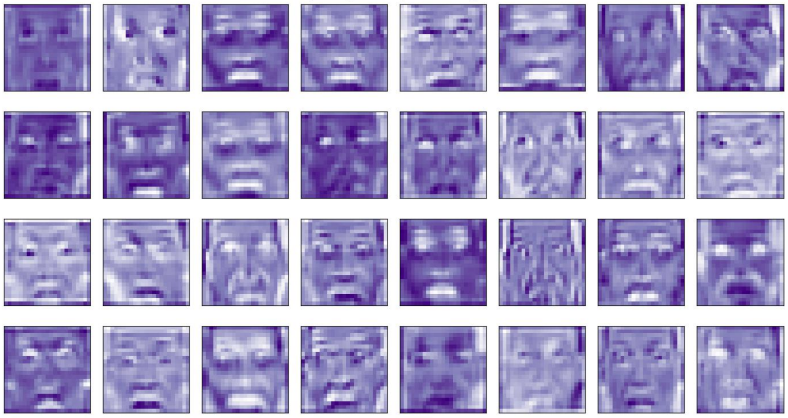
2）使用 training data 中的第九張圖片進行輸入，同樣觀察每個 conv2D 層共計 6 層的輸出，觀察其特點。可以從下圖看出，在第一層中，大多數的圖片主要刻畫了人物表情的一些輪廓，而在接下的幾層當中，人物臉部的器官特徵開始漸漸清晰，眼睛、鼻子、嘴巴等特徵清楚，由於層數越多進行了 max_pooling2d 操作，像素減半，雖然圖片模糊，但是臉部整體的特徵清晰，臉部器官的朝向也能清楚看到。



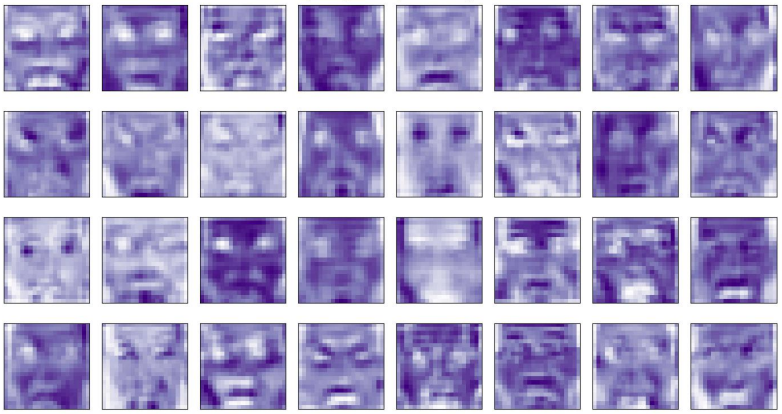
Output of layer: conv2d_2 (Given image 9)



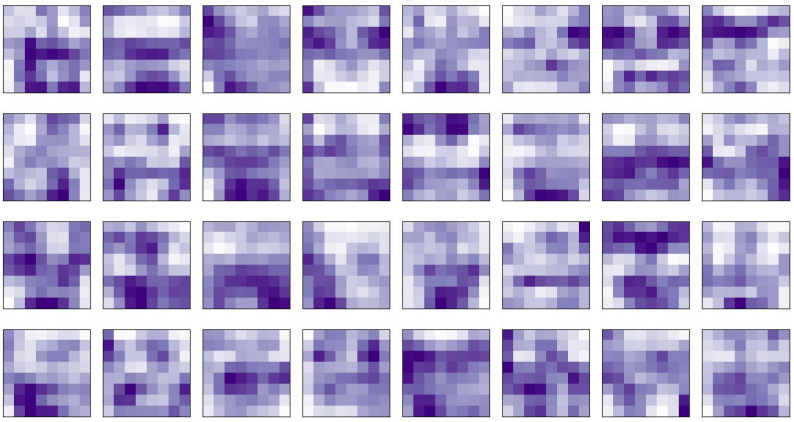
Output of layer: conv2d_3 (Given image 9)

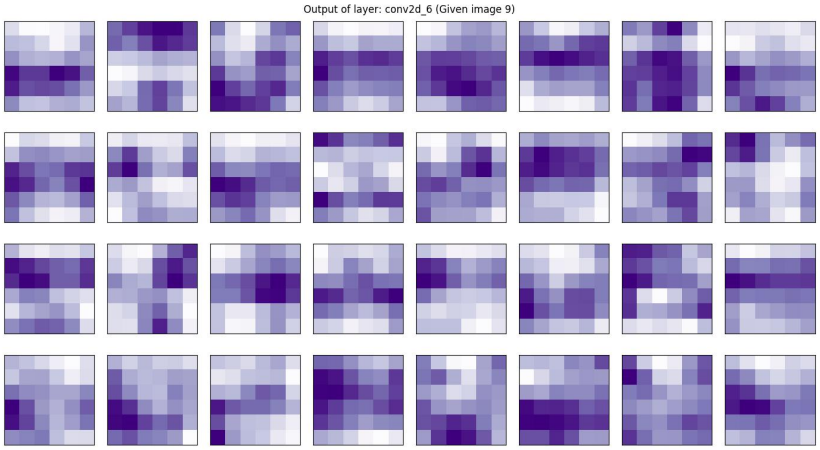


Output of layer: conv2d_4 (Given image 9)



Output of layer: conv2d_5 (Given image 9)




Output of layer: conv2d_6 (Given image 9)