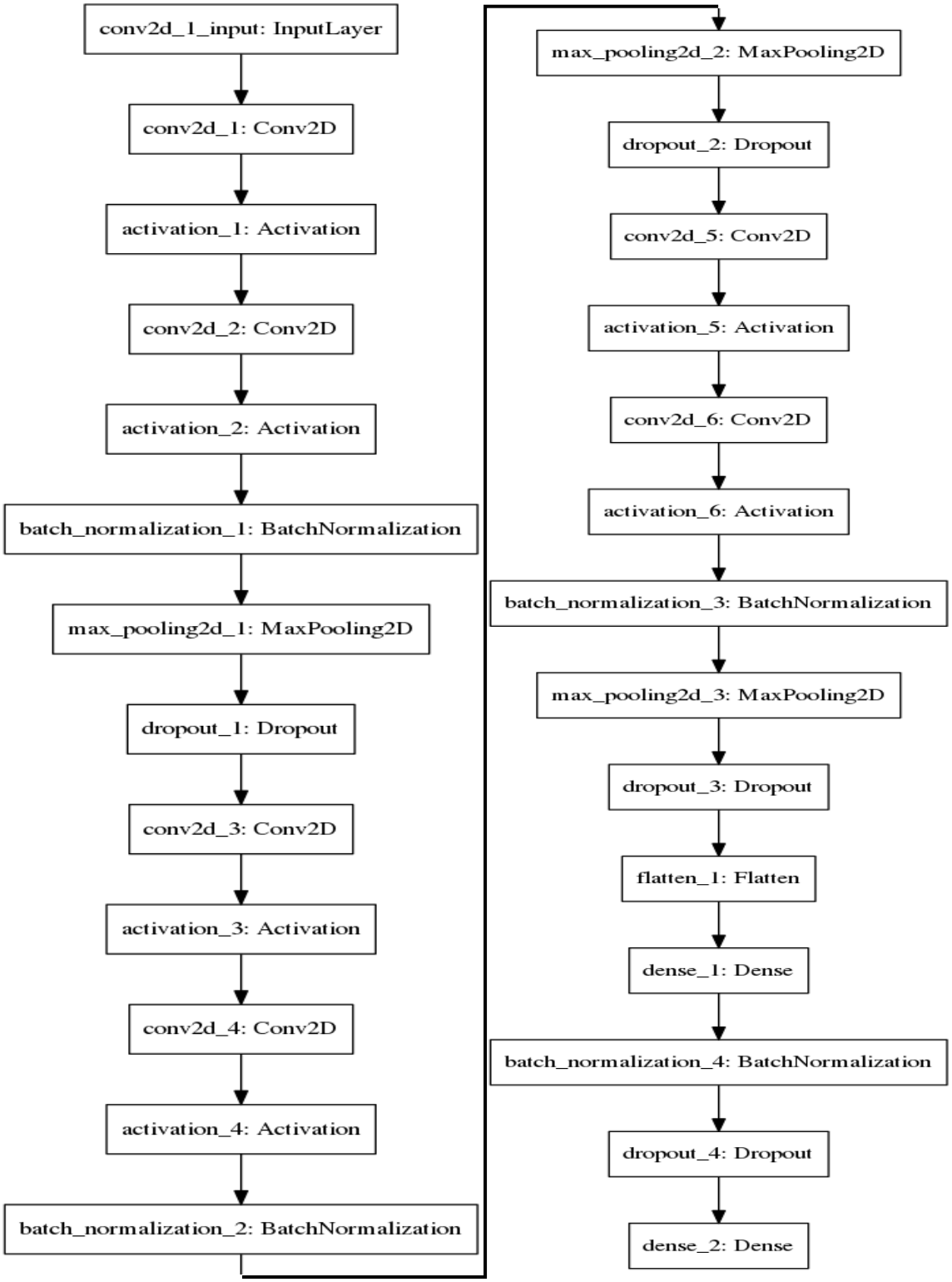
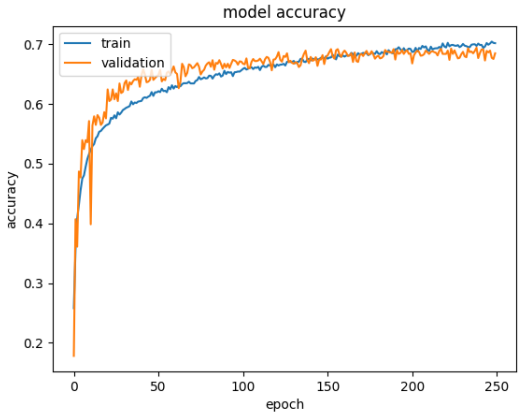
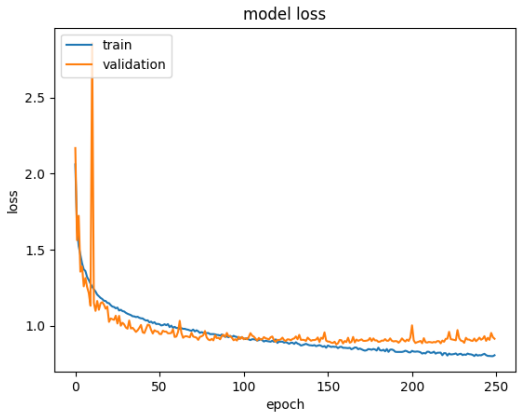
ML2017 HW3 Report

學號：B02901124 系級： 電機四 姓名：黃柏翔

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：







我所 train 的 CNN 模型及訓練過程如上面幾張圖所示

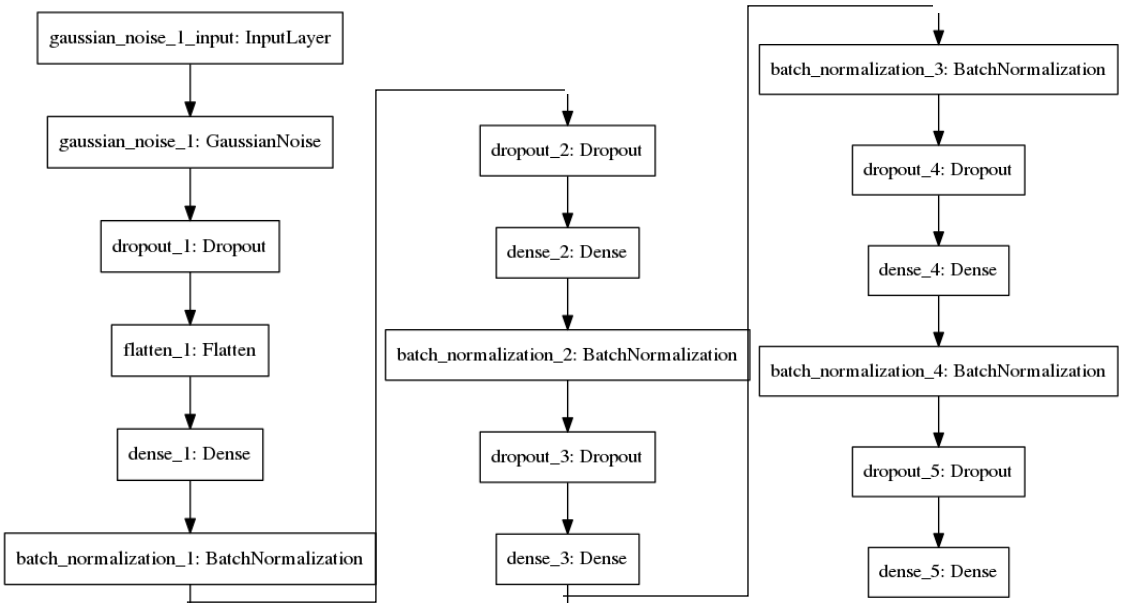
CNN 架構有六層的 Conv2D layer，filter 數量分別為 32, 32, 64, 64, 128, 128，每隔兩層做一次 batch_normalization, max_pooling, dropout，最後在接到 2 層的 Dense layer，node 數量分別為 512, 7。

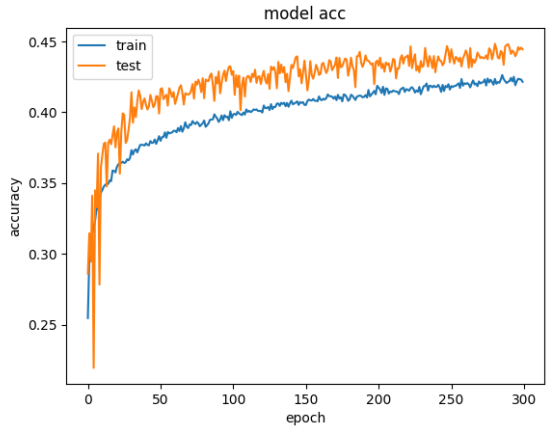
在訓練的過程中，將 training data 切出 2000 筆的 validation data，並用 ImageDataGenerator 增加 training data 的數量。使用了水平翻轉、上下左右平移 0.1 倍、旋轉角度 10 度。

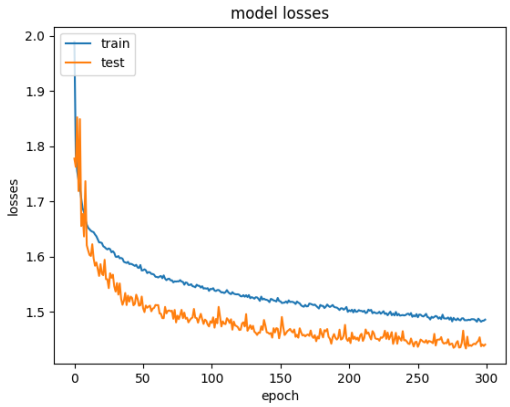
而這樣的架構在 train 到 250 個 epochs 時的 training accuracy 為 70%左右。可以發現在前 25 個 epoch 準確率上升相對十分快速(到達 63%左右)，而後來的 225 個 epoch 上升了 10%左右。而在 validation accuracy 約為 68%。最後 Kaggle 上的 public accuracy 為 0.677。

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：







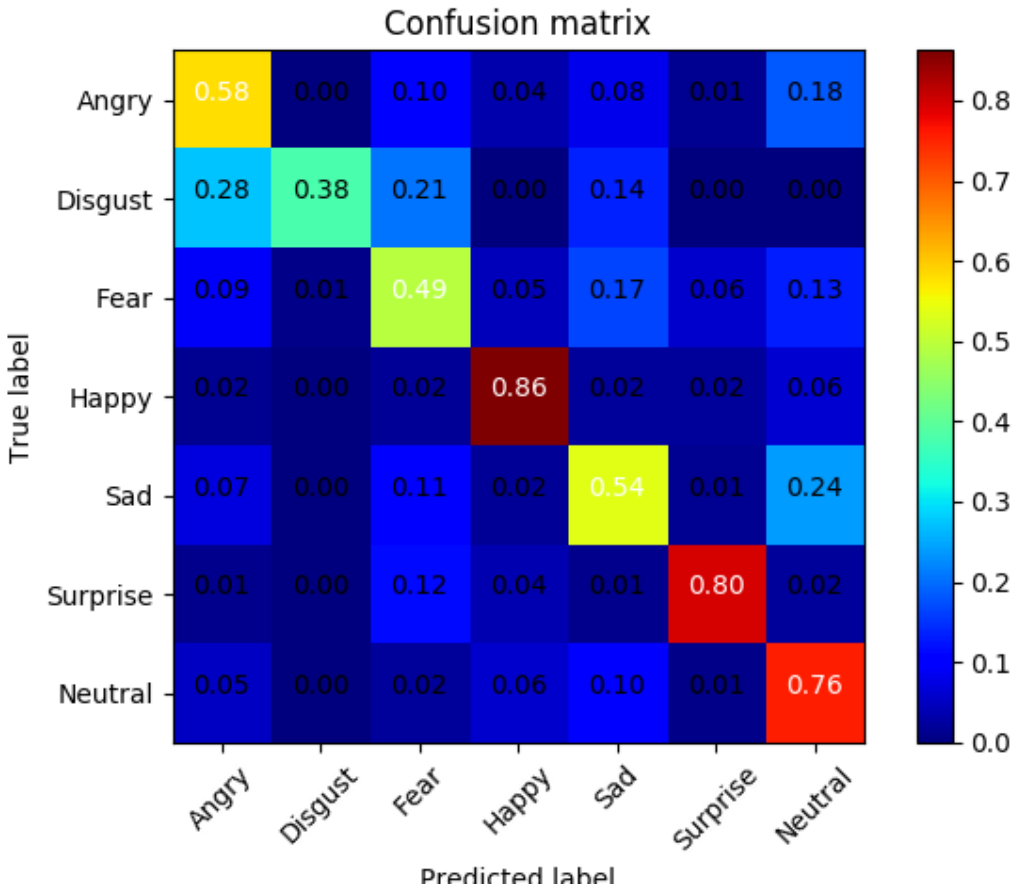
利用 summary()得知上題的 CNN 參數量為 134 萬左右，為了疊出參數量差不多的 DNN，我的架構為五層的 Dense layer，node 數量分別為 256,256,512,1024 及最後的分類層 7 個 node，而每層後面皆有做 batch_normalization, dropout。這樣架構下的參數量為 132.8 萬左右。

訓練過程 train 到約 300 個 epochs，最後 validation accuracy 到約 43.6%，可以發現在前 50 個 epoch 準確率上升相對十分快速，但後來的 250 個 epoch 只上升了 5%。

相對於上題的 CNN 在 25 個 epochs 就達到轉折點，DNN 在 converge 的時間(epoch 數)比較慢，可以發現 CNN 的學習效率大於 DNN（但實質上 DNN update 參數的時間較快，可能是在計算 gradient 的時候相對容易）。而最後準確率的差距達到 20%左右，也是 CNN 在圖片分類中優於 DNN 的證明。

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：畫出 confusion matrix 如下



藉由 validation set 中的圖片畫出的 confusion matrix，X 軸方向的合為 1，表示給定 true label 後 predict label 所佔的百分比。觀察可以看出斜對角線上的分數都比較高，表示 model 預測的結果有一定的可信度。但還是可以看出在 Angry, Surprise 上的表現相對較低，而在 Disgust, Fear 上的的預測結果更不理想。

從淺藍色的部分(預測結果錯誤率較高)，可以發現 model 容易混淆的配對：
Disgust 分類為 Angry(0.28),Fear(0.21), Sad(0.14)；Fear 分類為 Sad(0.17),Neutral(0.13)；Sad 分類為 Fear(0.11), Neutral(0.24)。這些表情大部分具有負面的情緒，推測若 model 學到的 feature 是可信的話，人類在這些負面情緒下所產生的表情是十分雷同的。

而在 Happy 及 Surprise 的準確率十分高，可能代表這兩個表情的 feature 具有獨特性，使 model 在學習及判斷的過程中不會與其他情緒混淆。

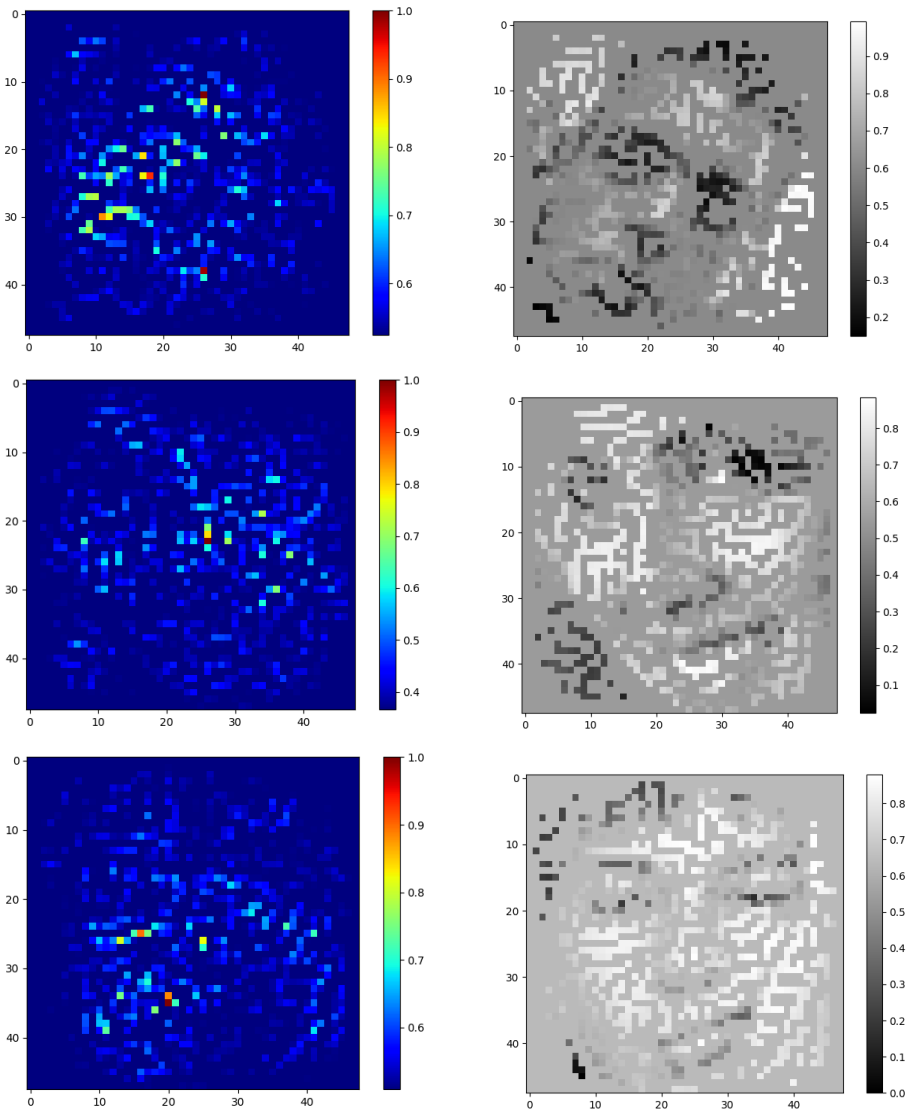
4. (1%) 從(1)(2)可以發現，使用 CNN 的確有些好處，試繪出其 saliency maps，觀察模型在做 classification 時，是 focus 在圖片的哪些部份？

答：

下面三張圖分別是用 testing data 的第 5,16,35 張來做 saliency maps。雖然 heatmap 的結果看起來沒有助教的範例這麼好，可能是 normalization 的方式不同，但可以看到 heatmap 分數較高的地方大致上在五官的位置。

我所 normalize 的方式是先採用 MinMax normalization 將算出的 gradients 弄在[0, 1]區間，而 threshold 的設置是取“平均值加上 0.2 個標準差”，最後一個比較 tricky 的作法是將所有小於 threshold 的點都設為 0.999 倍的 threshold，在畫 heatmap 時才能將小於 threshold 的點都畫成藍色(比較像範例圖)，否則大部分的 gradients 都會集中在 0.5(綠色)的部分。

觀察 mask 後出來的灰階影像，根據計算每張圖大約畫出 30%的像素，便已經可以約略看出一個人表情的輪廓，表示 model 大致根據這些位置去做分類。根據我目測 mask 的結果，猜測三張圖的 label 分別為 neutrual, happy, disgust。

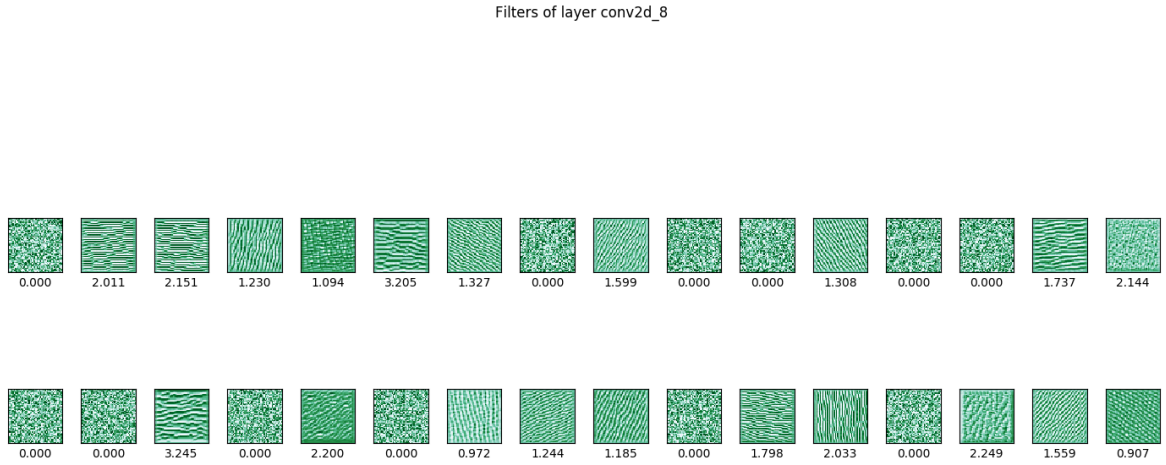


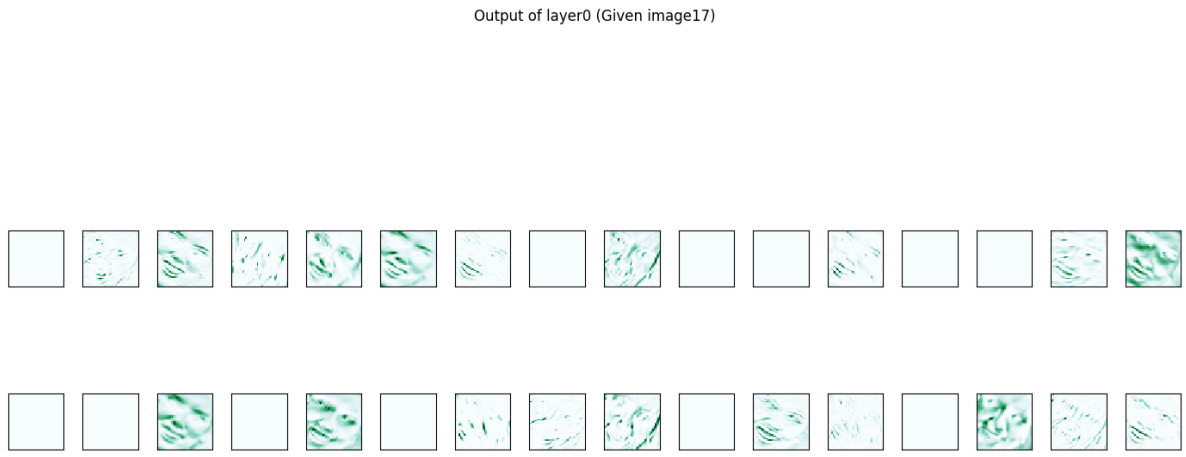
5. (1%) 承(1)(2)，利用上課所提到的 gradient ascent 方法，觀察特定層的 filter 最容易被哪種圖片 activate。

答：

下面第一張圖是第一層 Conv2D 在經過 20 次 gradient ascent 由 white noise update 而成的圖片，看起來帶有不同的 texture 的特性，顯示出這層的 filters 最容易被哪些圖片激活。

下面後三張圖分別是給定 validation data 第 17 張照片在第 2,4,6 層 Conv2D 的 filter output,可以看出一開始還略帶有圖片本身的輪廓，到後面 resolution 變比較低時便比較只有一格一格的資訊。





Output of layer2 (Given image17)

Output of layer4 (Given image17)