

Machine Learning HW 3

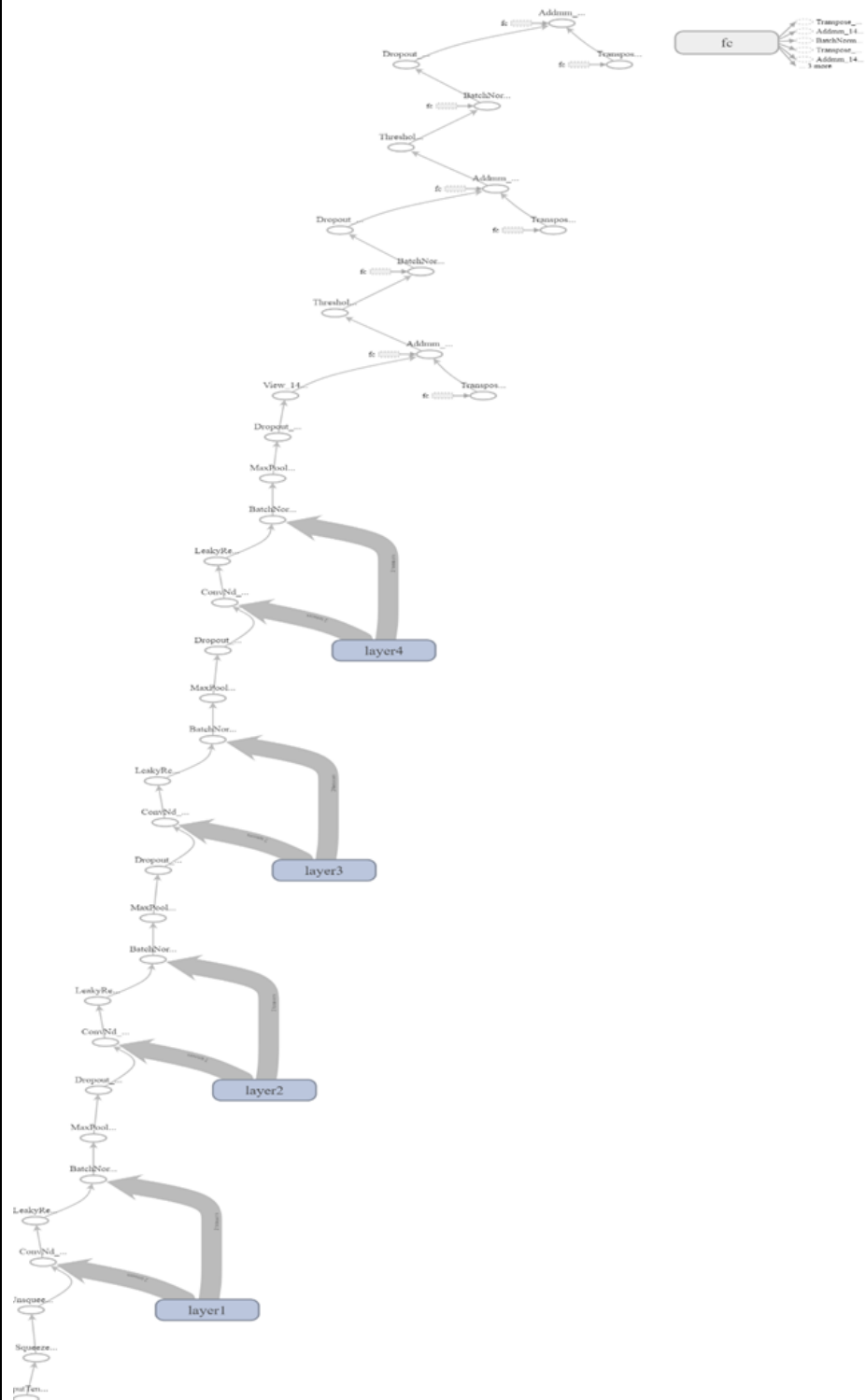
B04705003 資工三 林子雋

Model 部分參考：資工四宋子維同學的模型

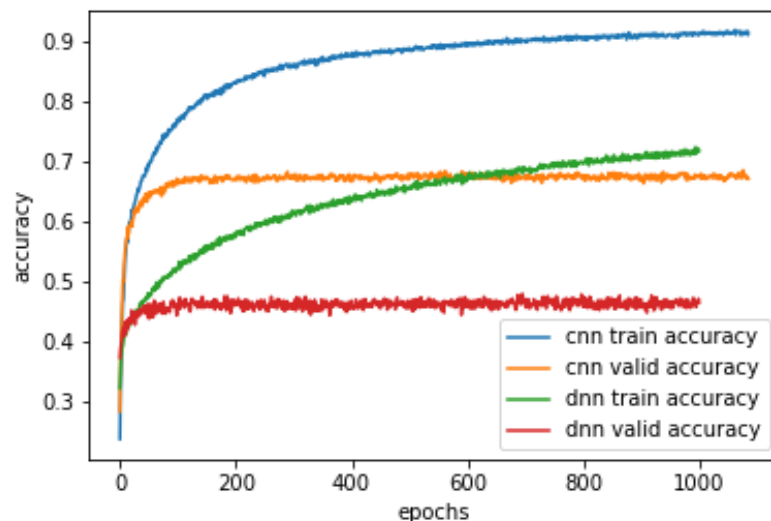
1. 請說明你實作的 CNN model，其模型架構、訓練過程和準確率為何？

A：

模型架構（使用 pytorch
建立模型，tensorboardX
輸出模型圖到
tensorboard 上）：



使用 CNN 訓練的過程
(同下題討論 CNN 和
DNN 的圖片)：



2. 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？

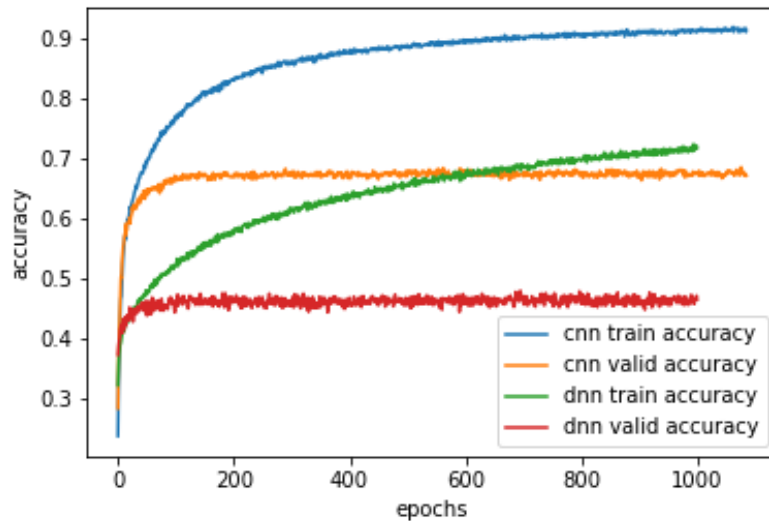
A：

我所建立的 CNN 架構需要的參數如下：

參數類別	參數細節	參數總數
CNN + DNN	$CNN = (5*5*1)*64(Conv2d) + 64(bias) + 64*2(BatchNorm2d) +$ $(3*3*64)*128(Conv2d) + 128(bias) + 128*2(BatchNorm2d) +$ $(3*3*128)*256(Conv2d) + 256(bias) + 256*2(BatchNorm2d) +$ $(3*3*256)*512(Conv2d) + 512(bias) + 512*2(BatchNorm2d)$ $fully\ connected : (512*4*4)*512(Linear) + 512(bias) + 512*2$ $(BatchNorm1d) + 512*512(Linear) + 512(bias) + 512*2$ $(BatchNorm1d) + 512*7(Linear) + 7(bias)$	6015879 個參數
DNN	$48*48*1800 + 1800 + 1800*1050 + 1050 + 1050*7 + 7$	6047407 個參數

模型	訓練集和驗證集準確率
----	------------

CNN 和
DNN 準確
率比較



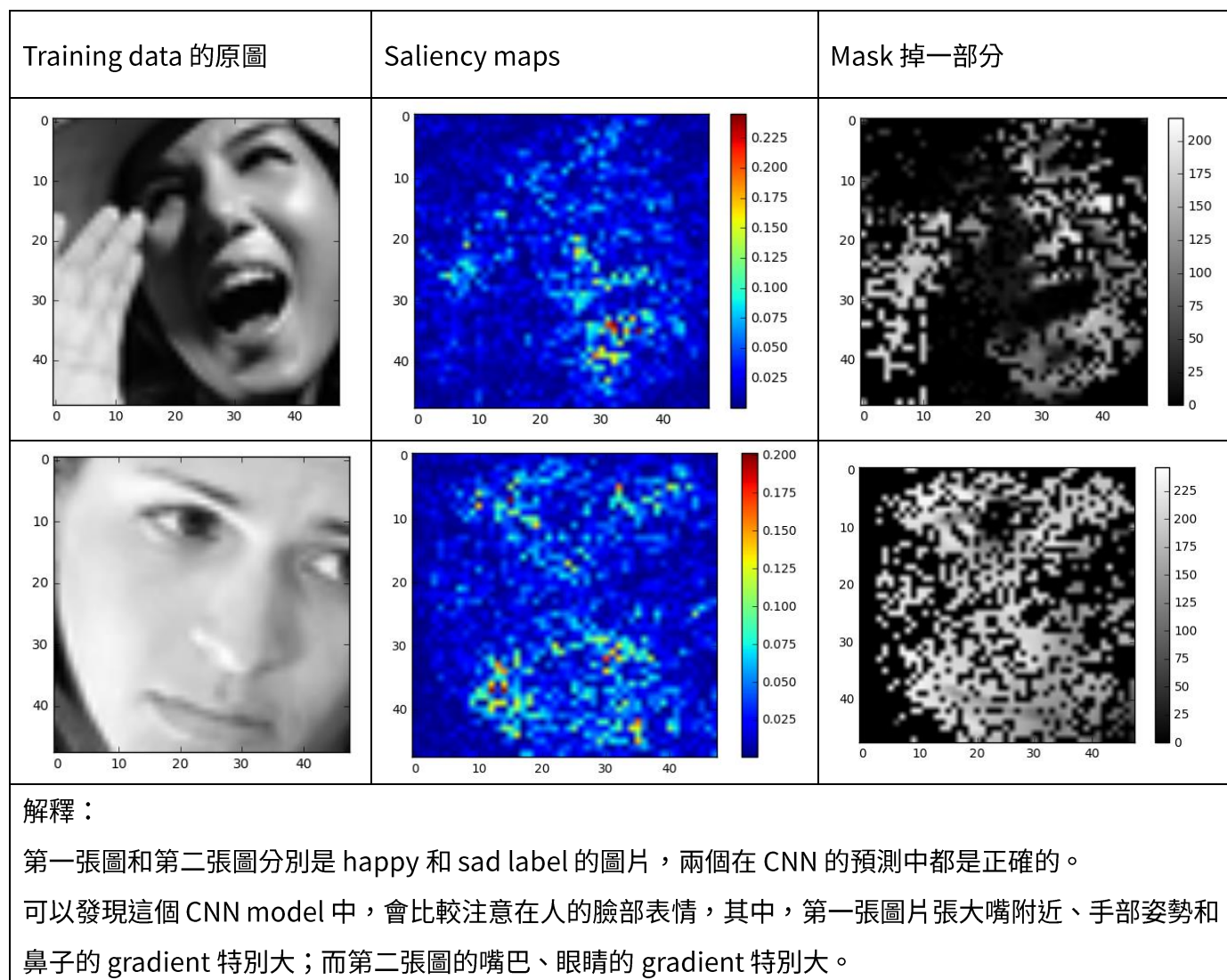
解釋：可以發現到，DNN 在後期的 epoch 中，沒辦法再提升 validation 準確率，但是 training set 的準確率仍然穩定上升。相比之下，CNN 的 validation 從一開始就遠遠勝過 DNN，而且 training set 的準確率也比 DNN 上升的快速。

3. 觀察答錯的圖片中，哪些 class 彼此間容易用混? [繪出 confusion matrix 分析]

資料	Confusion matrix																																																																
20%的 training data	<div><p>Confusion matrix</p><table><tr><th>True label \ Predicted label</th><th>Angry</th><th>Disgust</th><th>Fear</th><th>Happy</th><th>Sad</th><th>Surprise</th><th>Neutral</th></tr><tr><th>Angry</th><td>0.56</td><td>0.01</td><td>0.07</td><td>0.06</td><td>0.14</td><td>0.03</td><td>0.13</td></tr><tr><th>Disgust</th><td>0.13</td><td>0.60</td><td>0.04</td><td>0.03</td><td>0.11</td><td>0.03</td><td>0.07</td></tr><tr><th>Fear</th><td>0.09</td><td>0.00</td><td>0.48</td><td>0.05</td><td>0.14</td><td>0.11</td><td>0.13</td></tr><tr><th>Happy</th><td>0.02</td><td>0.00</td><td>0.01</td><td>0.90</td><td>0.02</td><td>0.02</td><td>0.04</td></tr><tr><th>Sad</th><td>0.09</td><td>0.00</td><td>0.08</td><td>0.05</td><td>0.56</td><td>0.01</td><td>0.20</td></tr><tr><th>Surprise</th><td>0.03</td><td>0.00</td><td>0.05</td><td>0.05</td><td>0.02</td><td>0.81</td><td>0.04</td></tr><tr><th>Neutral</th><td>0.05</td><td>0.00</td><td>0.04</td><td>0.11</td><td>0.10</td><td>0.01</td><td>0.70</td></tr></table></div>	True label \ Predicted label	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	Angry	0.56	0.01	0.07	0.06	0.14	0.03	0.13	Disgust	0.13	0.60	0.04	0.03	0.11	0.03	0.07	Fear	0.09	0.00	0.48	0.05	0.14	0.11	0.13	Happy	0.02	0.00	0.01	0.90	0.02	0.02	0.04	Sad	0.09	0.00	0.08	0.05	0.56	0.01	0.20	Surprise	0.03	0.00	0.05	0.05	0.02	0.81	0.04	Neutral	0.05	0.00	0.04	0.11	0.10	0.01	0.70
True label \ Predicted label	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral																																																										
Angry	0.56	0.01	0.07	0.06	0.14	0.03	0.13																																																										
Disgust	0.13	0.60	0.04	0.03	0.11	0.03	0.07																																																										
Fear	0.09	0.00	0.48	0.05	0.14	0.11	0.13																																																										
Happy	0.02	0.00	0.01	0.90	0.02	0.02	0.04																																																										
Sad	0.09	0.00	0.08	0.05	0.56	0.01	0.20																																																										
Surprise	0.03	0.00	0.05	0.05	0.02	0.81	0.04																																																										
Neutral	0.05	0.00	0.04	0.11	0.10	0.01	0.70																																																										
	<p>解釋：</p> <p>可以發現到，前三最容易弄錯的類別是 fear, angry, sad 的 label。也可以顯著發現 happy 的命中率特別高。</p>																																																																

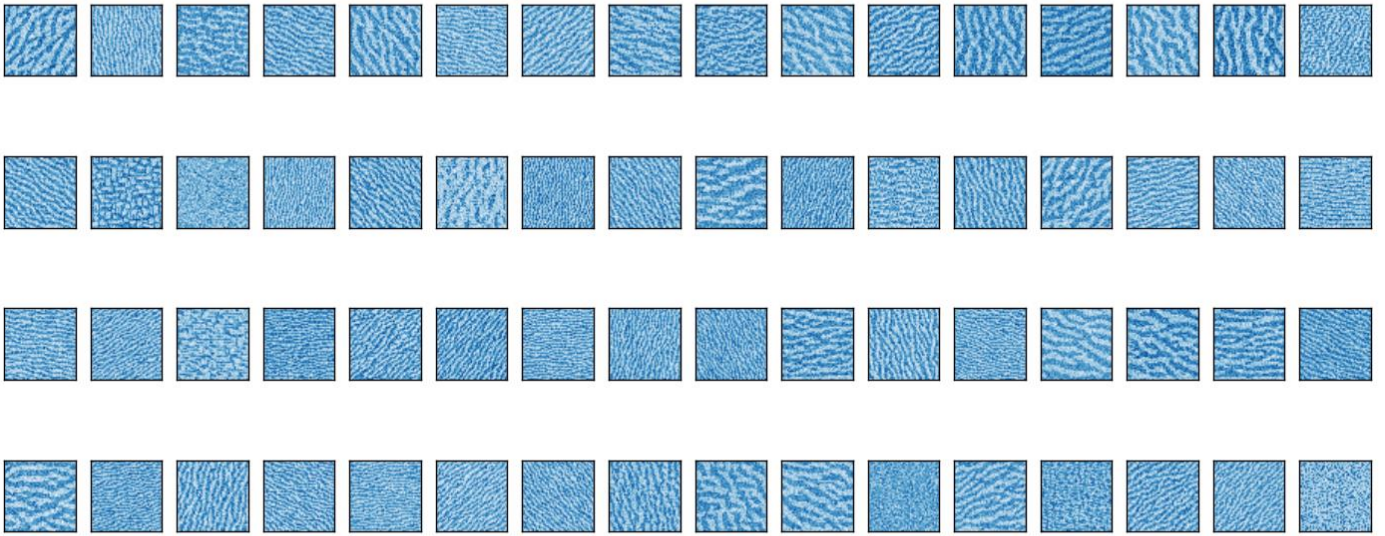
4. 從(1)(2)可以發現，使用 CNN 的確有些好處，試繪出其 saliency maps，觀察模型在做 classification

時，是 focus 在圖片的哪些部份？



5. 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。

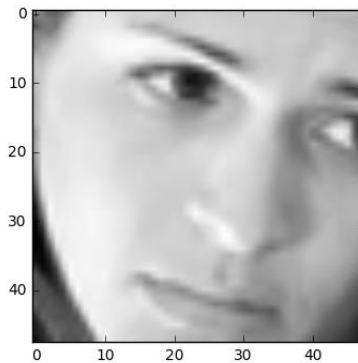
Image that can activate filters 64 (Given image 300)



解釋：

這是第一層 Convolution 64 種 filter，分別輸入這些圖片會使得那一種 filter 卷積出來的結果最大。可以發現到第一層 filter 所抓取到圖片的資訊主要在於圖片紋路的樣式。

原始圖片：



被第一層的 64 種 filter 卷積過後的輸出圖片：

Output of layer64 (Given image300)



解釋：

這是經過 64 種 filters 卷積出來的結果，可以發現到抓取到的資訊都以人臉的輪廓為主，譬如像是眼睛、鼻子、嘴巴的顏色都比較深。