HW3 Report

學號：r07521603　系級：　土木所碩二　姓名：蔡松霖

1. (1%) 請說明這次使用的 model 架構，包含各層維度及連接方式。

這次作業我嘗試 train 了幾個不同的 model，最後用到的有 torchvision 中的 ResNeXt50 跟 Inception v3，將 3 個 ResNeXt50-32x4d model 取平均跟 1 個 inception v3 的 model 做 ensemble 得到最後的結果。

ResNeXt50-32x4d (https://arxiv.org/pdf/1611.05431.pdf)

各層維度及連接方式如下表的右側欄所示。

stage	output	ResNet-50	**ResNeXt-50 (32×4d)**
conv1	112×112	7×7, 64, stride 2	7×7, 64, stride 2
conv2	56×56	3×3 max pool, stride 2	3×3 max pool, stride 2
		[1×1, 64 3×3, 64 1×1, 256] ×3	[1×1, 128 3×3, 128, C=32 1×1, 256] ×3
conv3	28×28	[1×1, 128 3×3, 128 1×1, 512] ×4	[1×1, 256 3×3, 256, C=32 1×1, 512] ×4
conv4	14×14	[1×1, 256 3×3, 256 1×1, 1024] ×6	[1×1, 512 3×3, 512, C=32 1×1, 1024] ×6
conv5	7×7	[1×1, 512 3×3, 512 1×1, 2048] ×3	[1×1, 1024 3×3, 1024, C=32 1×1, 2048] ×3
	1×1	global average pool 1000-d fc, softmax	global average pool 1000-d fc, softmax
# params.		$\mathbf{25.5}\times10^6$	$\mathbf{25.0}\times10^6$
FLOPs		$\mathbf{4.1}\times10^9$	$\mathbf{4.2}\times10^9$

Table 1. (**Left**) ResNet-50. (**Right**) ResNeXt-50 with a 32×4d template (using the reformulation in Fig. 3(c)). Inside the brackets are the shape of a residual block, and outside the brackets is the number of stacked blocks on a stage. "C=32" suggests grouped convolutions [24] with 32 groups. *The numbers of parameters and FLOPs are similar between these two models.*

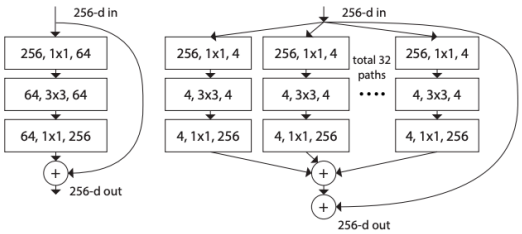


Figure 1. **Left**: A block of ResNet [14]. **Right**: A block of ResNeXt with cardinality = 32, with roughly the same complexity. A layer is shown as (# in channels, filter size, # out channels).

最後的 fc 有換掉，換成 out 為 7-d 的 linear layer。

Inception v3: (https://arxiv.org/pdf/1512.00567.pdf)

各層維度及連接方式如下表所示，最後面的 linear 層改為 output 7-d。

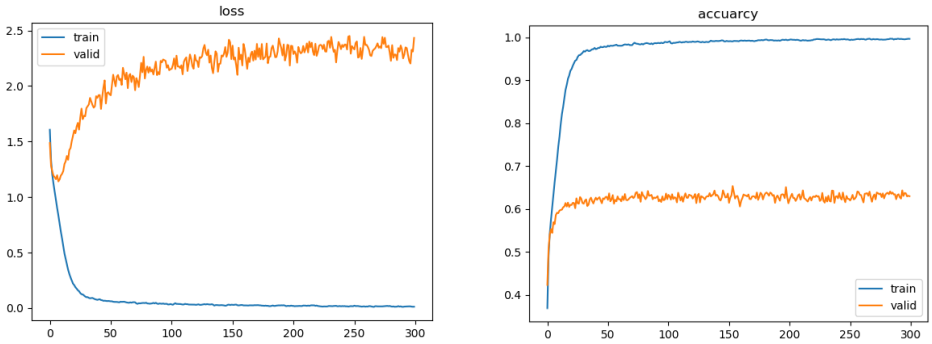
type	patch size/stride or remarks	input size
conv	3×3/2	299×299×3
conv	3×3/1	149×149×32
conv padded	3×3/1	147×147×32
pool	3×3/2	147×147×64
conv	3×3/1	73×73×64
conv	3×3/2	71×71×80
conv	3×3/1	35×35×192
3×Inception	As in figure 5	35×35×288
5×Inception	As in figure 6	17×17×768
2×Inception	As in figure 7	8×8×1280
pool	8 × 8	8 × 8 × 2048
linear	logits	1 × 1 × 2048
softmax	classifier	1 × 1 × 1000

Table 1. The outline of the proposed network architecture. The output size of each module is the input size of the next one. We are using variations of reduction technique depicted Figure 10 to reduce the grid sizes between the Inception blocks whenever applicable. We have marked the convolution with 0-padding, which is used to maintain the grid size. 0-padding is also used inside those Inception modules that do not reduce the grid size. All other layers do not use padding. The various filter bank sizes are chosen to observe principle 4 from Section 2.

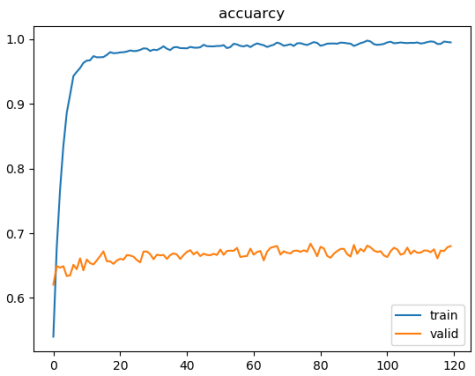
Ensemble 是由 resnext50 兩次的 training 取其中 3 個 epoch 的 model 與 inceptionV3 的 model 做加權平均：prediction = ((ResNeXt50-1 + ResNeXt50-2 + ResNeXt50-3) / 3 + inceptionV3) / 2，在 Kaggle 上的 public score 是 0.964。

2. (1%) 請附上 model 的 training/validation history (loss and accuracy)。

ResNeXt50-32x4d:

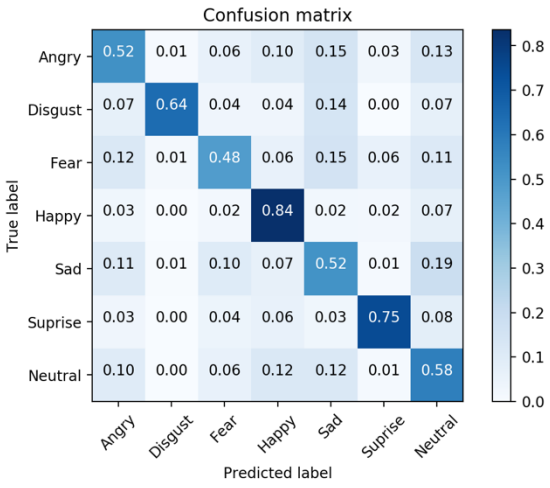


Inception v3:

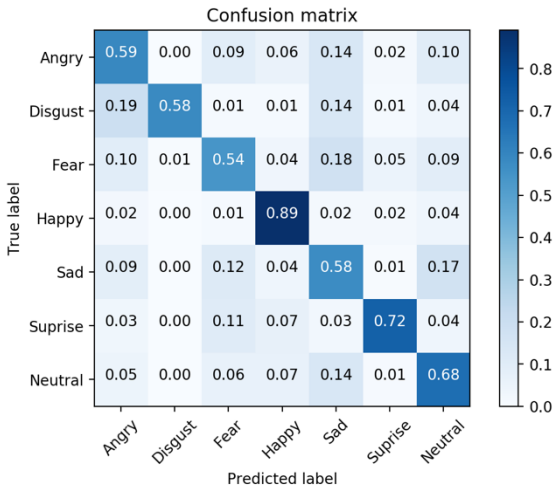


3. (1%) 畫出 confusion matrix 分析哪些類別的圖片容易使 model 搞混，並簡單說明。(ref: https://en.wikipedia.org/wiki/Confusion_matrix)

ResNeXt50-32x4d



Inception V3



不管是 ResNeXt50 還是 Inception v3，Happy 預測的準確度都有不錯的表現，其他的表現則各有優劣。像是 ResNeXt50 在 Disgust, Surprise 比起 Inception V3 有較好的準確度，而其他項目 Angry, Fear, Sad, Neutral 則是 Inception V3 表現較好。總地來看，可以發現 Angry, Disgust, Fear 跟 Sad 這幾個負面情緒辨識上比較容易混淆，也導致這幾個的準確度沒那麼高。

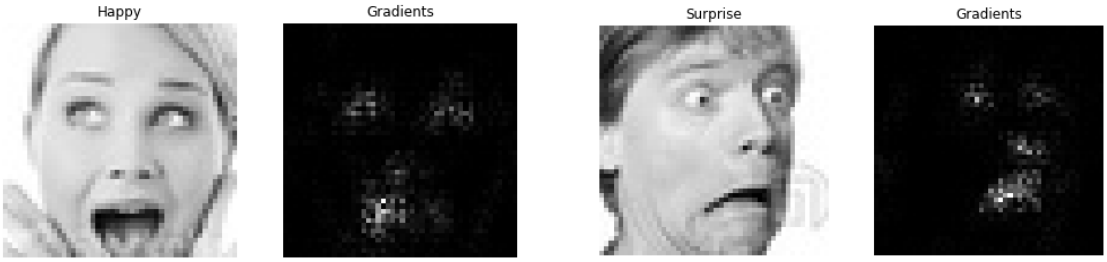
[關於第四及第五題]

可以使用簡單的 3-layer CNN model [64, 128, 512] 進行實作。

4. (1%) 畫出 CNN model 的 saliency map，並簡單討論其現象。

(ref: https://reurl.cc/Qpjg8b)

Ref:
https://github.com/wmn7/ML_Practice/blob/master/2019_07_08/Saliency%20Maps/Saliency%20Maps%20Picture.ipynb



Happy 跟 Surprise 這兩個是分類比較成功的兩類，可以看到五官位置都抓的蠻精準的。可以想像在判斷是哪種情緒時，機器也是從圖片中的五官作為判斷的重要依據。

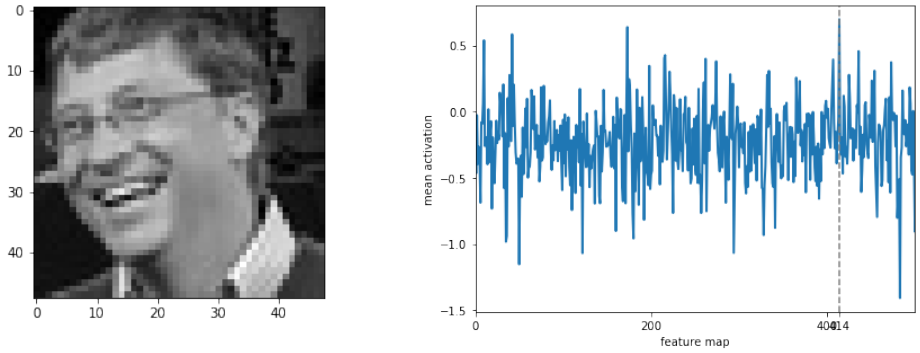
5. (1%) 畫出最後一層的 filters 最容易被哪些 feature activate。

(ref: https://reurl.cc/ZnrgYg)

Ref: https://github.com/utkuozbulak/pytorch-cnn-visualizations

參考 reference 的分析方法，Plots the mean activation per feature map for a specific layer given an input image (https://reurl.cc/Gknv4A)

下圖 mean activation 最大的幾個 filter 分別是 [10, 42, 173, 414, 507]



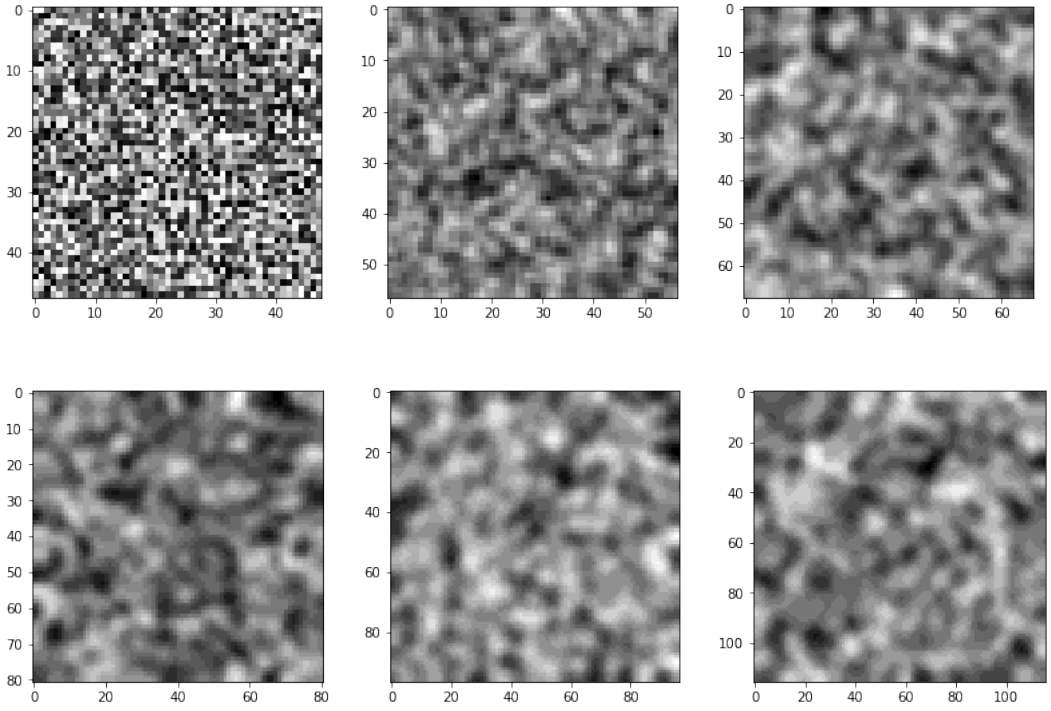
分別將這幾個 filters 吃的 features 畫出來，感覺有點像雜訊不太能看出什麼。

filter 10　filter 42　filter 173　filter 414　filter 507

推測原因可能是因為這個三層 CNN 過於簡單，預測能力本身就不大強。

嘗試 reference 的作法，進行 upscaling 的步驟，並加上 blur，來看看：

Filter 414: (upscaling_factor=1.2, blur=3) 由左至右，由上至下 upscaling 的步數分別是由 1 往上遞增至 6。

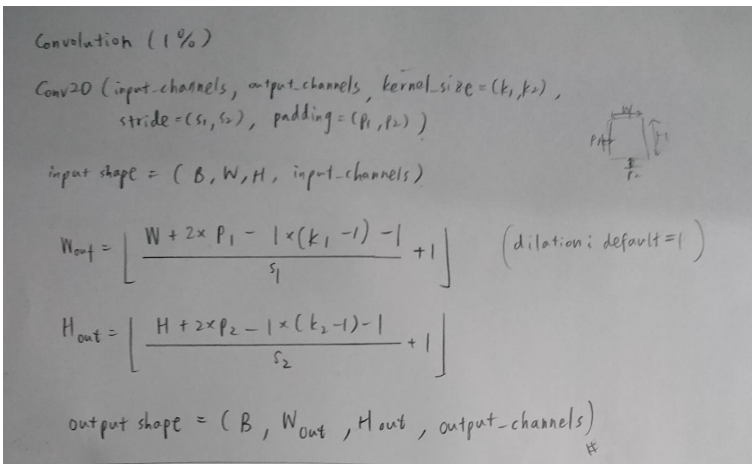


可惜仍然看不出什麼所以然 QQ

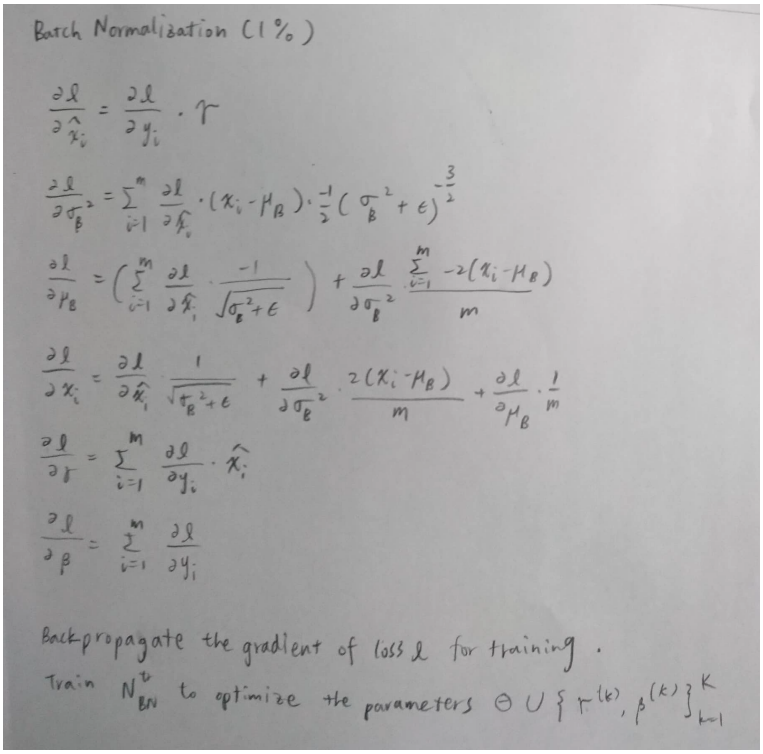
6. (3%)Refer to math problem

https://hackmd.io/JIZ_0Q3dStSw0t0O0w6Ndw

Convolution (1%)



Batch Normalization (1%)



Softmax and Cross Entropy (1%)

Ref: https://peterroelants.github.io/posts/cross-entropy-softmax/