學號：B05902042 系級： 資工二 姓名：林瑋毅

1. (1%) 請說明你實作的 CNN model，其模型架構、訓練參數和準確率為何？
(Collaborators:)
答：架構如下：

```
AlexNetBN2(
  (layer1): Sequential(
        (0): Conv2d(1, 32, kernel_size=(5, 5), stride=(1, 1))
        (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True)
        (2): ReLU(inplace)
        (3): MaxPool2d(kernel_size=(2, 2), stride=(2, 2), dilation=(1, 1),
ceil_mode=False)
  )
  (layer2): Sequential(
        (0): Conv2d(32, 64, kernel_size=(5, 5), stride=(1, 1))
        (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True)
        (2): ReLU(inplace)
        (3): MaxPool2d(kernel_size=(2, 2), stride=(2, 2), dilation=(1, 1),
ceil_mode=False)
  )
  (layer3): Sequential(
        (0): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
        (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True)
        (2): ReLU(inplace)
  )
  (layer4): Sequential(
        (0): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
        (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True)
        (2): ReLU(inplace)
  )
  (layer5): Sequential(
        (0): MaxPool2d(kernel_size=(3, 3), stride=(1, 1), dilation=(1, 1),
ceil_mode=False)
        (1): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1))
        (2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True)
        (3): ReLU(inplace)
  )
  (layer6): Sequential(
        (0): Dropout(p=0.5)
        (1): Linear(in_features=6272, out_features=1024, bias=True)
        (2): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True)
        (3): ReLU(inplace)
  )
  (layer7): Sequential(
        (0): Dropout(p=0.5)
        (1): Linear(in_features=1024, out_features=1024, bias=True)
        (2): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True)
        (3): ReLU(inplace)
  )
  (layer8): Sequential(
        (0): Linear(in_features=1024, out_features=7, bias=True)
        (1): LogSoftmax()
  )
)
```

訓練參數： epoch =1000, optimizer = adam(lr=0.001, betas=(0.9, 0.999))

準確率：private / public = 0.67483 / 0.68765

2. (1%) 請嘗試 data normalization, data augmentation,說明實行方法並且說明對準確率有什麼樣的影響？
(Collaborators:)
答： 寫出實作data normalization過程、與實作前、實作後準確率。
寫出實作data augmentation過程、與實作前、實作後準確率。
    - Data normalization
        - 過程：已知各 pixel 的範圍為 0-255 ，所以將各個 pixel 的值除以 255，使其範圍在 0-1 的區間。

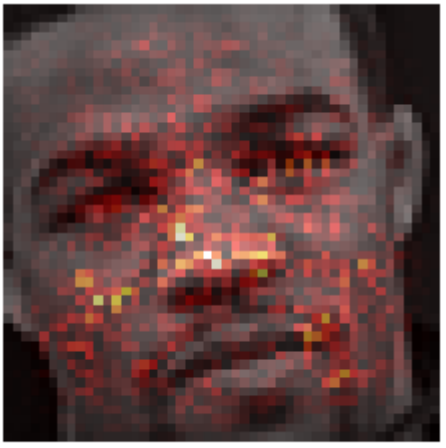
        - 實作前：private / public = 0.60434 / 0.61131
        - 實作後：private / public = 0.60769 / 0.60044
        - 兩者並未相差太多，推測是因為原本的 model 裡已經有 batch normalization，所以訓練的時候不會因為 input 範圍太大而不穩定。
    - Data augmentation
        - 過程：首先先隨機調整亮度、對比、飽和度 (+-10%)，接著隨機水平翻轉，接著隨機旋轉 (+-30度)，接著上下左右 pad 四個像素，再隨機 crop 出一個 48x48 的影像做為輸入。
        - 實作前：private / public = 0.60434 / 0.61131
        - 實作後：private / public = 0.66035 / 0.65840
        - 經過 augmentation 後有顯著的進步，因為更多的資料可以增強模型的泛化能力。

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：Confusion matrix 如下

```
[[ 440,    7,  105,   26,  109,   22,   96],
 [  21,   31,    7,    1,    9,    2,    3],
 [  83,    3,  350,   31,  174,   76,   62],
 [  32,    1,   31, 1239,   28,   41,   73],
 [  91,    4,  113,   42,  542,    9,  170],
 [  19,    1,   89,   33,   21,  502,   19],
 [  62,    0,   71,   61,  166,   28,  596]]
```

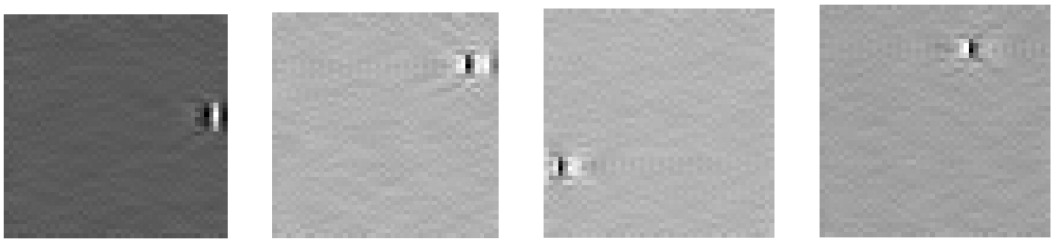
   其中容易誤認的組合有 (2,0), (2,4), (4,0), (4,2), (4,6), (6,6)。

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：合理說明test的圖片和觀察到的東西 -> 0.5分
   貼出saliency圖片 -> 0.5分

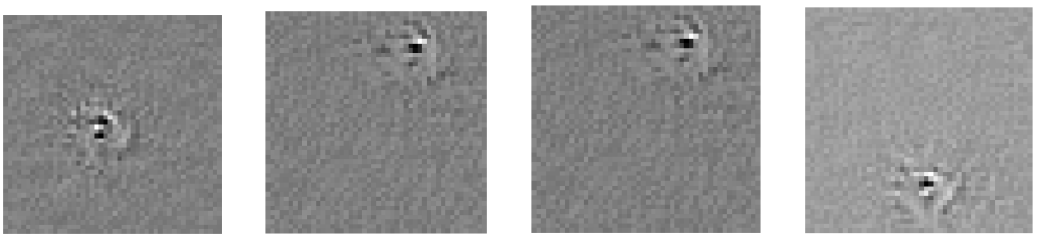
   觀察到的東西為：鼻子、臉頰和一點眼睛，原因應該是情緒可以從眼神、臉頰看出。

5. (1%) 承(4) 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:)
   答：合理說明test的層數和觀察到的東西 -> 0.5分
   貼出filter input and output的圖片 -> 0.5分
   首先是 layer 1 的 filter



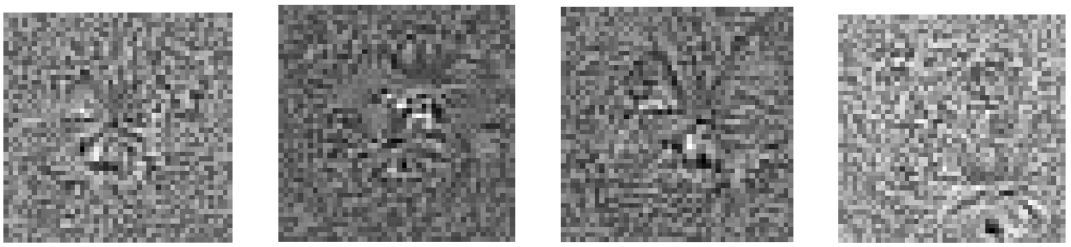
由圖可見，主要是觀察一些局部、簡單的 pattern。

接著是 layer2 的 filter



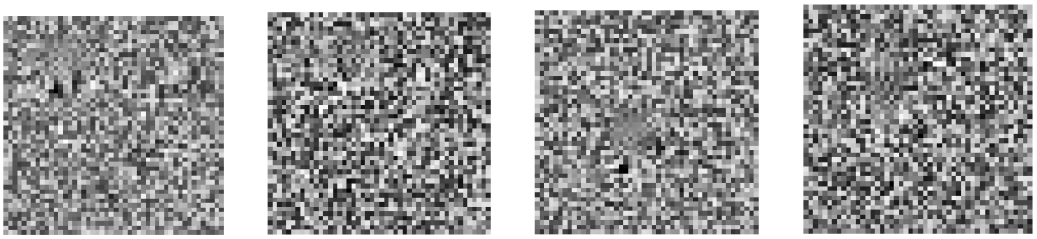
可以發現，觀察的 pattern 變得較為複雜（看起來可能像是眼睛之類的）。

以下是 layer3 的 filter



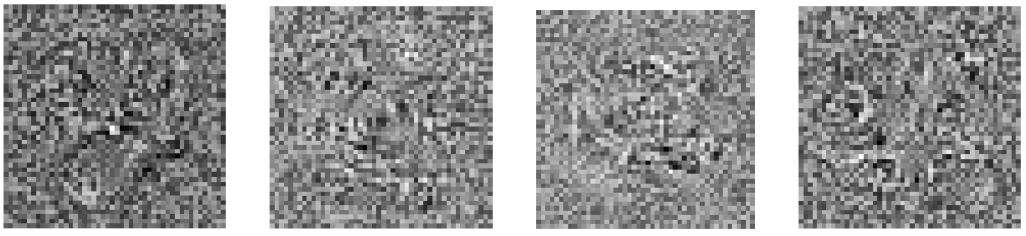
這層的 filter 顯示的特徵更為高階，已經可以看出一點臉的雛形（有眼窩、眉毛、鼻子等）。

以下是 layer4 的 filter



看起來已經接近是雜訊了。

以下是 layer 5 的 filter



好像帶有一種 pattern 又有很多雜訊。