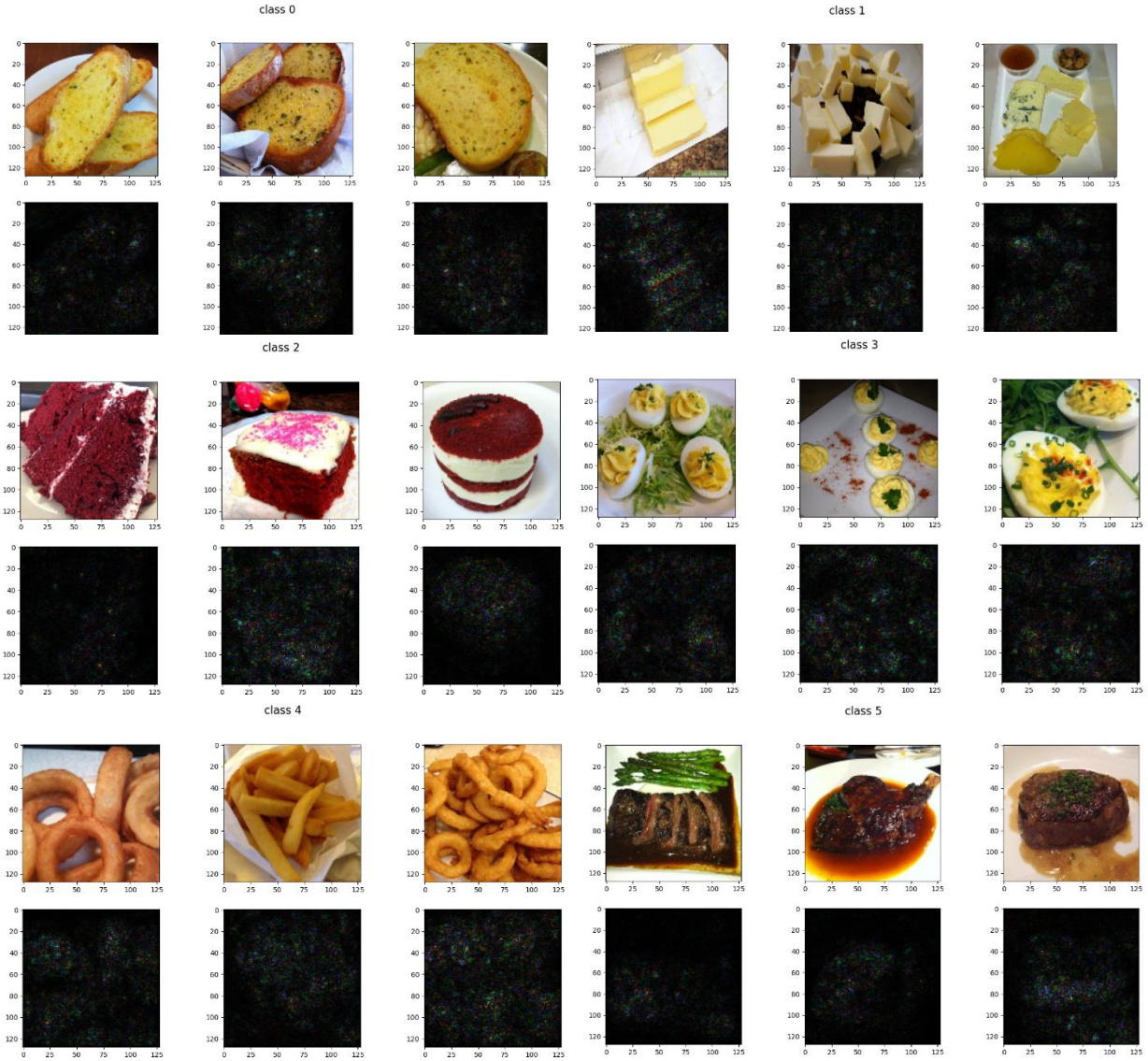
HW5 Report

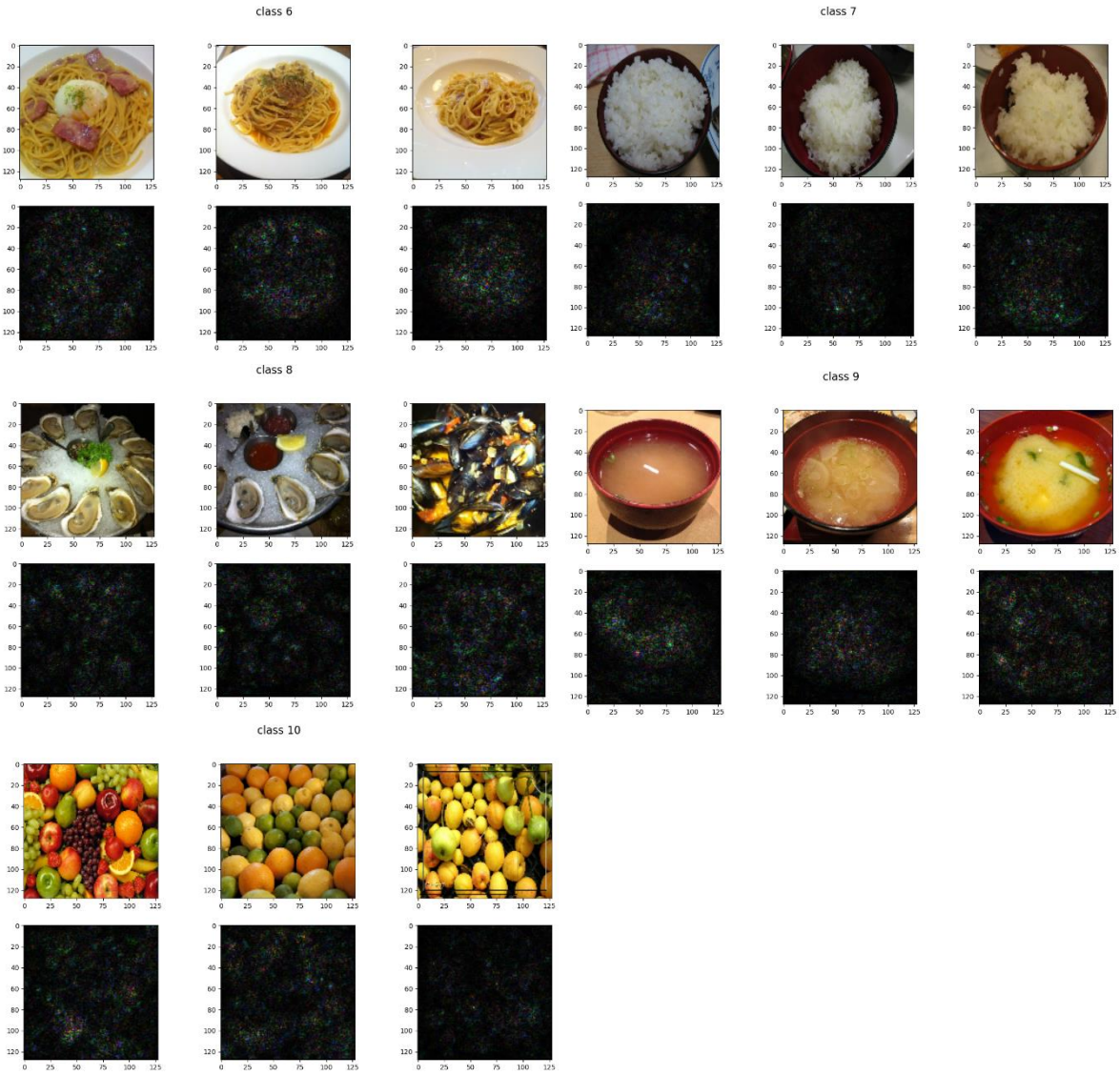
學號：b0701039　系級：電機二　姓名：劉知穎

1. (2%) 從作業三可以發現，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: 無)

   用之前 pretrained 好的 model 在 training set 上做預測，標準化預測出的原始 y 值，選擇在該類別正確預測且標準化後該類別的 y 值最大的前三者，做 saliency maps。





雖然看不出模型是用食物的那些細節再判斷，但可以看出模型有認出食物的大致輪廓。例如，在義大利麵、飯和湯的圖片裡，模型有辨識出碗或盤子的圓形狀，也就是有裝食物的範圍。在蛋、海鮮和水果的圖片中，模型有辨識出一顆顆的形狀。

2. (3%) 承(1) 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: 無)

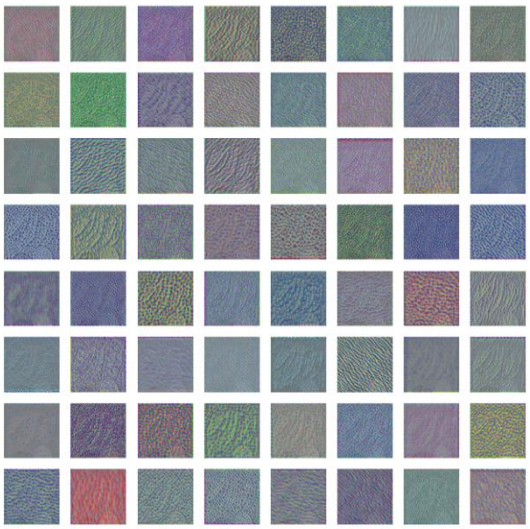
(1) 觀察不同層數 Conv2d 的 filter visualization (“num Con2d”： num 指的是層數，例如: 8 Conv2d 指的是第八層，第三個 Conv2d。)

模型架構：{nn.Conv2d()、nn.BatchNorm2d()、nn.ReLU()、nn.MaxPool2d(2, 2, 0)} 重複疊 5 次

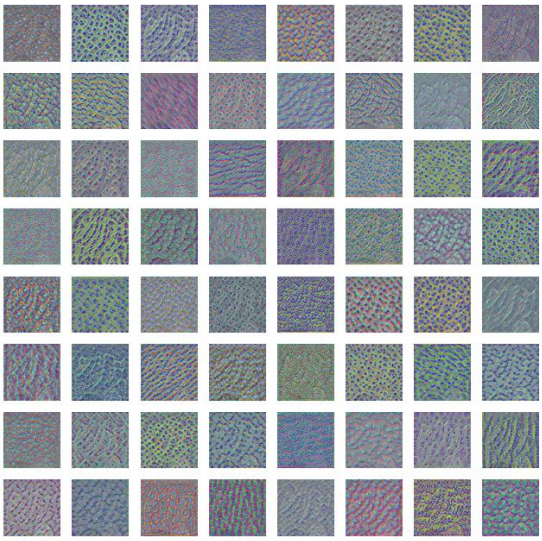
```
Classifier(
  (cnn): Sequential(
    (0): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (4): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (5): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (6): ReLU()
    (7): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (8): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (9): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (10): ReLU()
    (11): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (12): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (13): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (14): ReLU()
    (15): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (16): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (17): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (18): ReLU()
    (19): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (fc): Sequential(
    (0): Linear(in_features=8192, out_features=1024, bias=True)
    (1): ReLU()
    (2): Linear(in_features=1024, out_features=512, bias=True)
    (3): ReLU()
    (4): Linear(in_features=512, out_features=11, bias=True)
  )
)
```

first 64 filters of 4 Conv2d

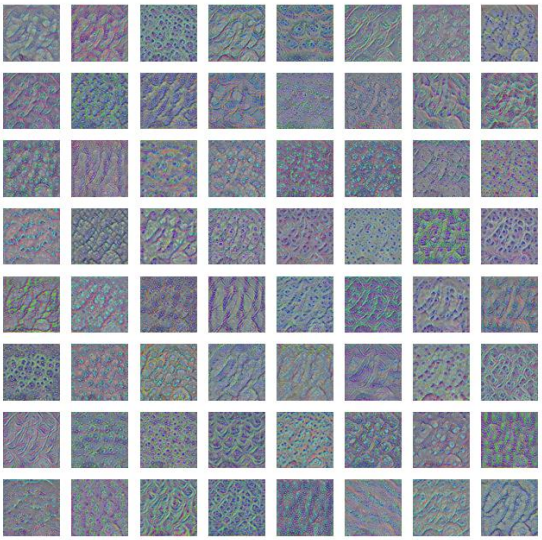
first 64 filters of 8 Conv2d



first 64 filters of 12 Conv2d



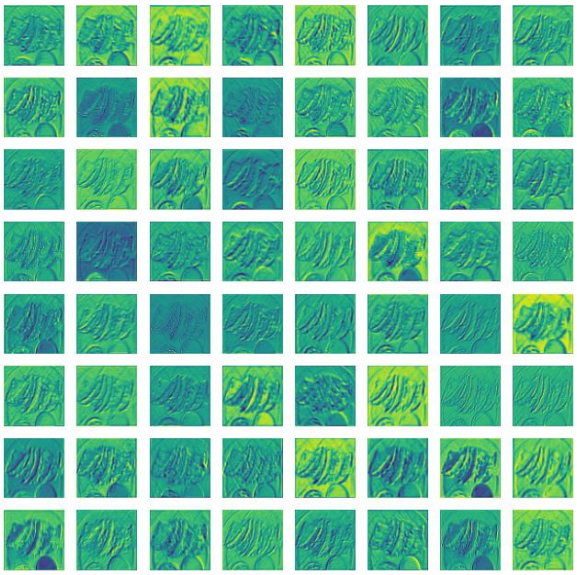
first 64 filters of 16 Conv2d



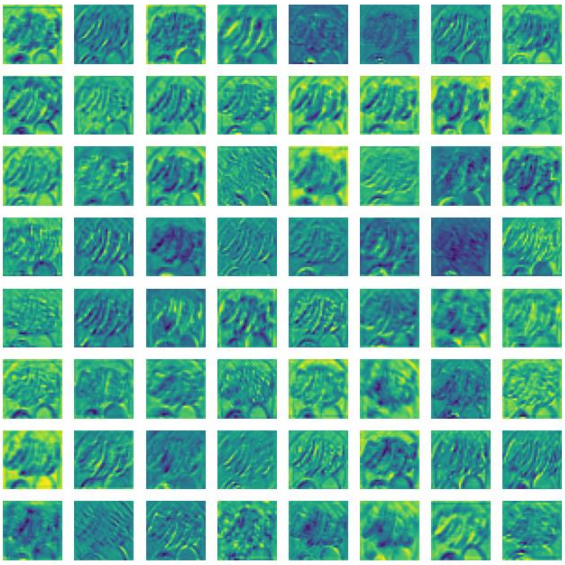
層數越深的 filter 學習到越細緻、複雜的花紋。在第四層時，學習到的花紋還不明顯；第八層時，學習到一些斜線、斑點的花紋；第九和十二層，學習到像螺旋、菱形、波浪等更複雜的花紋。

(2) 觀察不同層數 Conv2d 的 Filter activation (output)

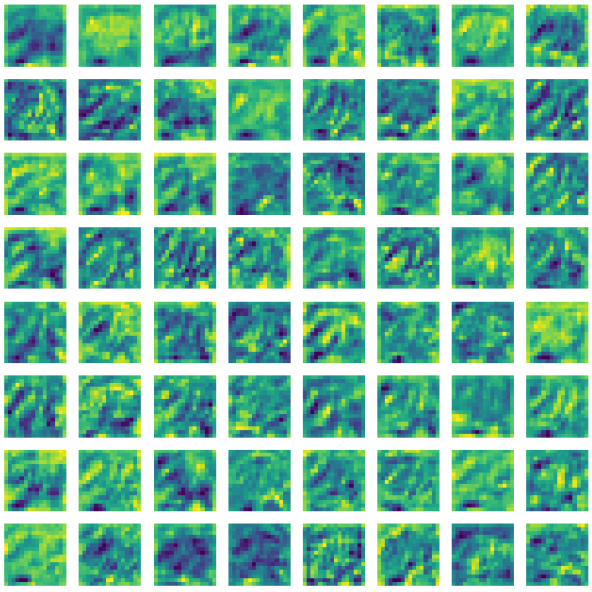
output of first 64 filters of 4 Conv2d



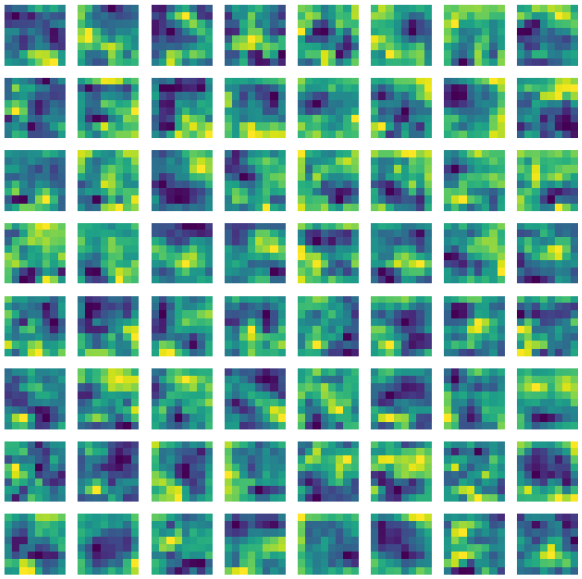
output of first 64 filters of 8 Conv2d



output of first 64 filters of 12 Conv2d



output of first 64 filters of 16 Conv2d



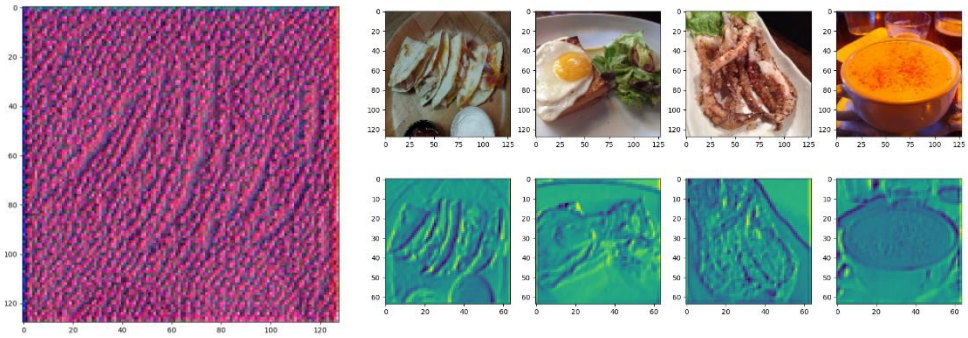
層數淺時，filter output 能清楚看到食物的輪廓，層數越深，filter output 越模糊。因為在層數深時，filter 學習到的 pattern 變複雜，因此 filter activation 比較「看不到」指定的 pattern，所以不會有肉眼能辨識的輪廓。另外因為 max

pooling 的關係，每次 conv2d 輸出的大小都會減小，在第 16 層（第五次 conv2d）輸出的大小為 8*8，當放大到同樣的 size 時，就會比較模糊。

(3) 觀察特定 filter 的 visualization 和 activation

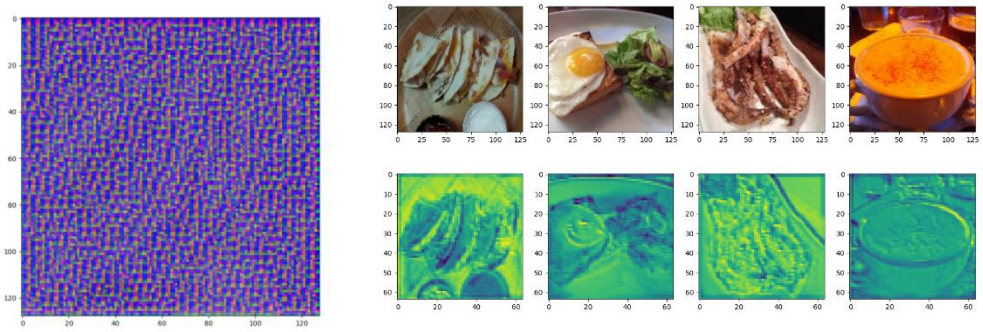
因為深層的 filter activation 看不出明顯的花紋，因此選擇用第四層的 filter 來觀察。分別取線狀和點狀各一個 filter。

a. cnn_id = 4, filter_id = 48



這個 filter 學習到較深的斜線線條。圖片中的邊界被 activate，尤其是第一張圖可以觀察到麵包邊緣的清楚的斜線。

b. cnn_id = 4, filter_id = 29



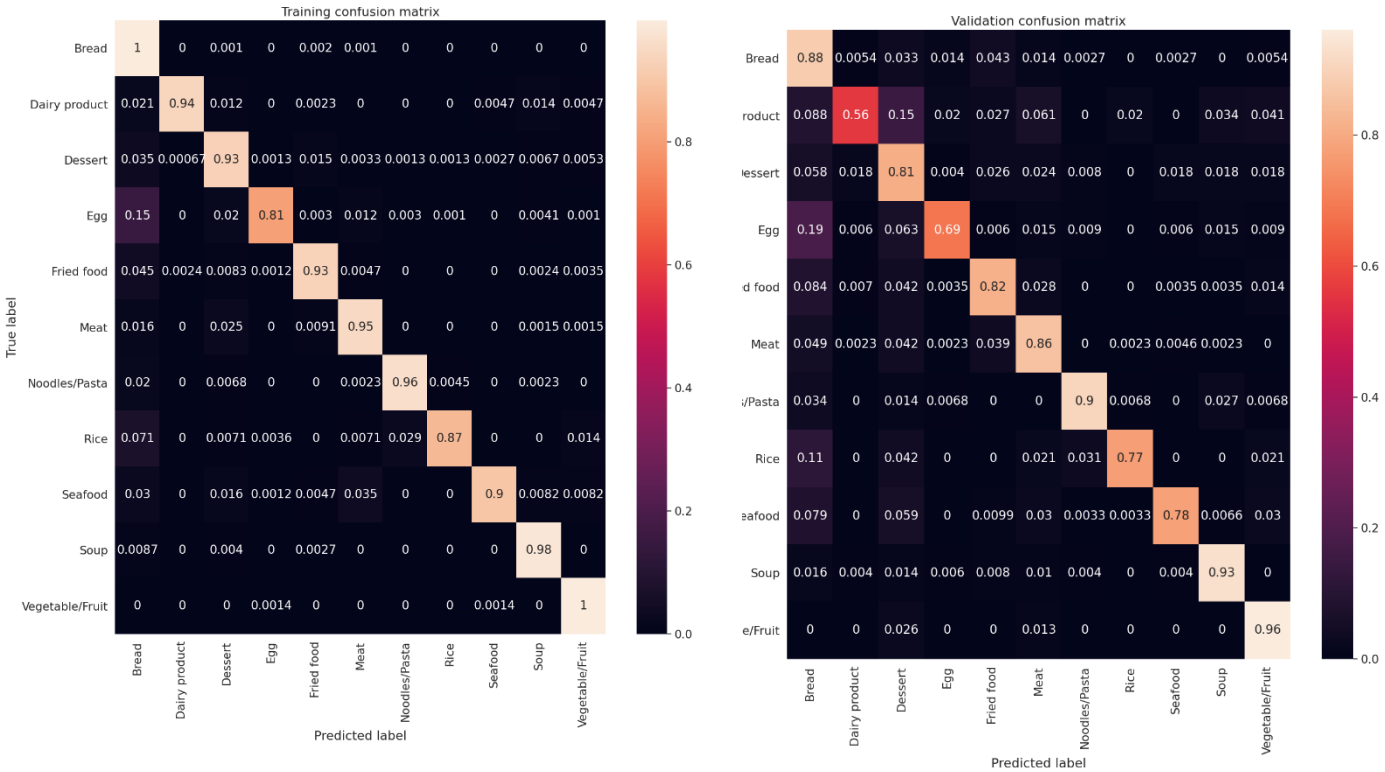
這個 filter 學習到邊點狀的圖形。和(a)相比，filter activation 中物品的邊緣較沒有那麼清楚。第四張圖中，可以看到湯中央點狀的結構有被 activate。

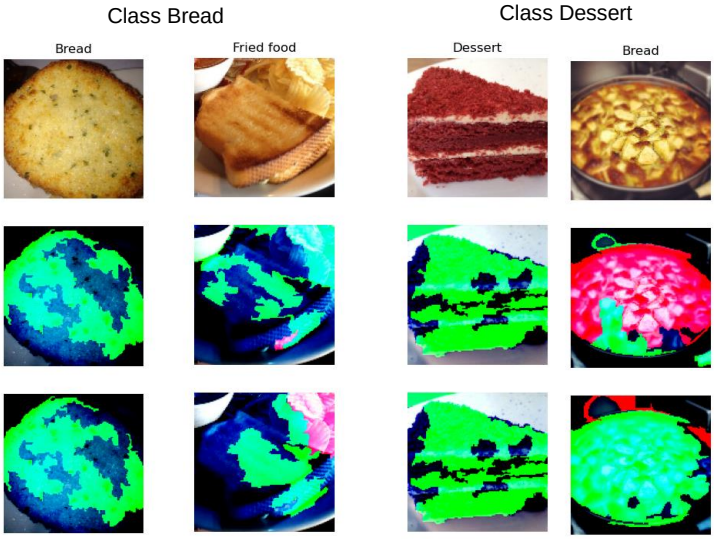
3. (2%) 請使用 Lime 套件分析你的模型對於各種食物的判斷方式，並解釋為何你的模型在某些 label 表現得特別好 (可以搭配作業三的 Confusion Matrix)。

(Collaborators: 無)

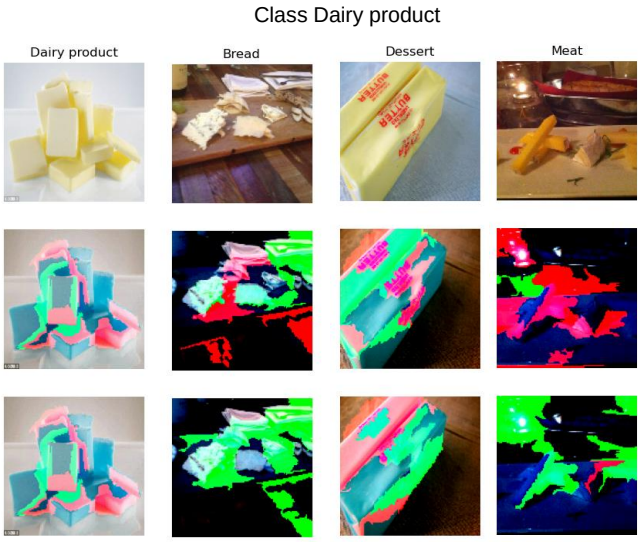
因為在 training set 上判斷的準確率太高，不容易觀察模型為何在某些 label 容易判斷成另一個 label，因此本題以 validation set 的資料實驗。針對每個 label，分別選擇一個正確預測的圖片、數張誤判成其他 label 的圖片（選擇在 validation set 誤判機率大於 0.05 者或若沒有一項大於 0.05 則選擇最高的一項），用 lime 分析。（將原始預測出的 y 值做標準化，選擇在該類別的分量有最大值者，例如挑選正確判斷成類別 1 的圖片時，我比較所有是類別 1 且正確判斷成類別 1 的圖片的標準化後 y[1]，選擇最大者）

因為我有做 normalize 的 transform，因此 lime 套件畫出的背景圖片顏色會失真，所以在第一橫排附上各圖的原始圖片。每一個類別的第一直排，上至下分別是，該類別正確判斷的原始圖片、lime 根據正確 label 解釋的圖片、lime 根據預測 label 解釋的圖片（第二、三張相同）。第二直排之後，直排上方的標題是誤判成的類別，三張圖上至下分別為，原始圖片、lime 根據正確 label 解釋的圖片、lime 根據預測出的 label 解釋的圖片。

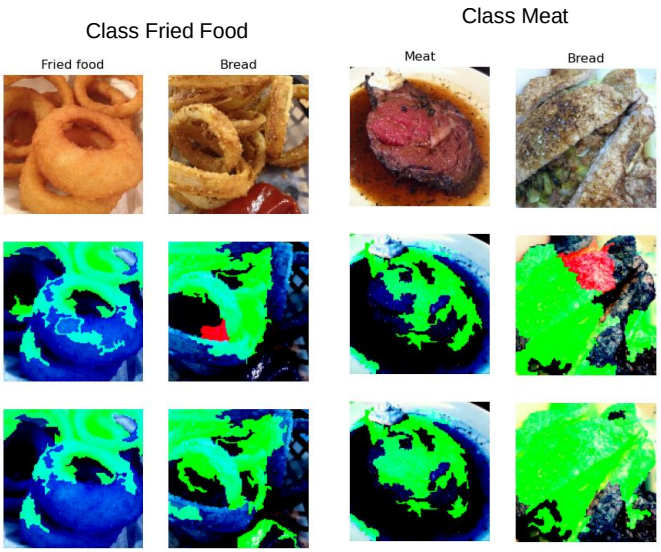




觀察 Dessert 判斷成 bread 的圖，發現食物整體在判斷成”dessert”上呈現負向影響，反倒在判斷成 ”bread”上是正向影響。推測可能原因是模型會將偏黃棕色和圓形的圖像判斷成”bread”。



觀察 Dairy Product 判斷成 bread 和 meat 的兩張圖，發現這兩張圖的背景物品，像是木板、桌子、桌燈在針對”Dairy Product”做解釋時產生了負向的影響。甚至第四直列的圖可能是食物與背景太相似，模型判斷的重點不是在食物身上。

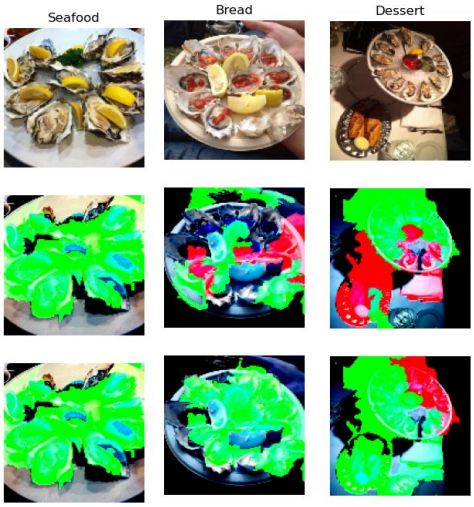


Fried Food 和 Meat 最容易誤判成的類別都是 Bread。食物本體在判斷成正確或錯誤類別大多是貢獻正向的影響，所以模型有正確抓到食物，沒有受背景影響。推測容易判斷錯的理由之一可能是受光線影響，因為兩張圖在判斷成正確類別的解釋圖中，分別都有一小塊在光線有明顯變化的區域呈現負向的影響。另一個原因，推測是 bread 這類別本身的形狀變化性太大，因此模型對於棕色的塊狀物都容易誤判成 bread。

Class Egg

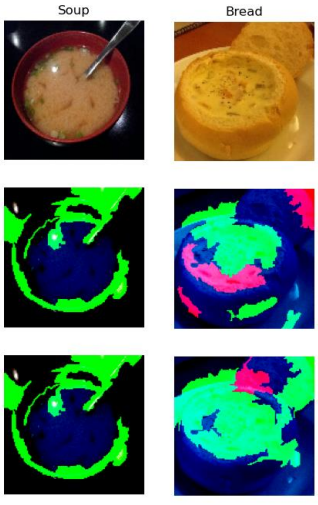
從第一直列的圖，發現模型判斷 egg 的重點在蛋黃以及圓形的形狀。第二直列的圖，由於是棕黃色塊狀的食物，因此被誤判成 bread。第三直列的圖，荷包蛋下方的兩片吐司在判斷成 egg 上產生負向影響。而在判斷成 dessert 上產生正向影響，可能是因為兩片土司的結構長得很像夾心蛋糕。另外，蛋黃的部分則在判斷成 egg 上產生正向影響；判斷成 dessert 上產生負向影響。當食物同時出現兩種食物時，若模型判斷錯哪個食物才是主角，就會分類錯誤。

Class Seafood

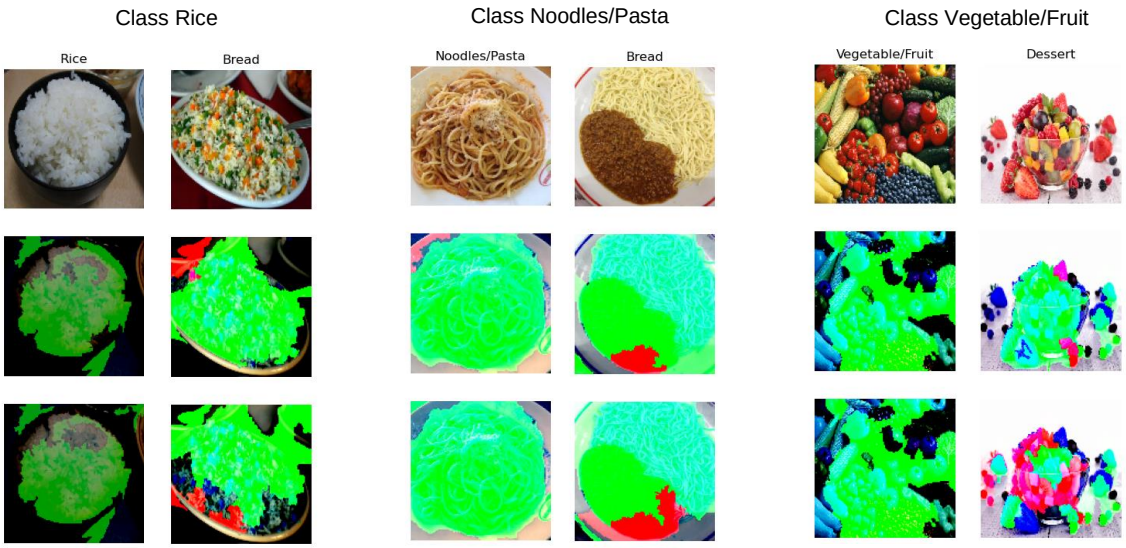


第二直列的圖在判斷成 seafood 時，模型的重點大多在中間的檸檬上，沒有抓到旁邊的海鮮。第三直列的圖，在判斷成 seafood 時，左下角小圓盤上的食物產生了負向的影響。

Class Soup



從第一直列的圖發現模型學習 soup 的重點在於邊緣圓形的形狀，而中間湯的顏色則不是重點。第二直列的圖，雖然乍看是麵包，但實際上是以麵包裝的湯。模型學習的重點是整片黃棕色的圓形圖案，所以誤判成了 bread。



結論：

(1) 有許多類別都容易判斷成 bread。推測是因為 bread 的形狀太多樣化，因此只要看到塊狀黃棕色的物品，模型都容易判斷錯誤。

(2) 模型在判斷時會受光影的影響。當同一塊食物有光影的變化時，可能就不會在判斷上產生正向影響，甚至可能產生負向的影響。

(3) 模型會被背景影響。如在 dairy product 類別所述。

(4) 當畫面同時有多種食物，模型著重的重點可能會擺錯。如再 egg 和 seafood 所述。

4. (3%) [自由發揮] 請同學自行搜尋或參考上課曾提及的內容，實作任一種方式來觀察 CNN 模型的訓練，並說明你的實作方法及呈現 visualization 的結果。

   (Collaborators: 無)

   實做 Grad-CAM。

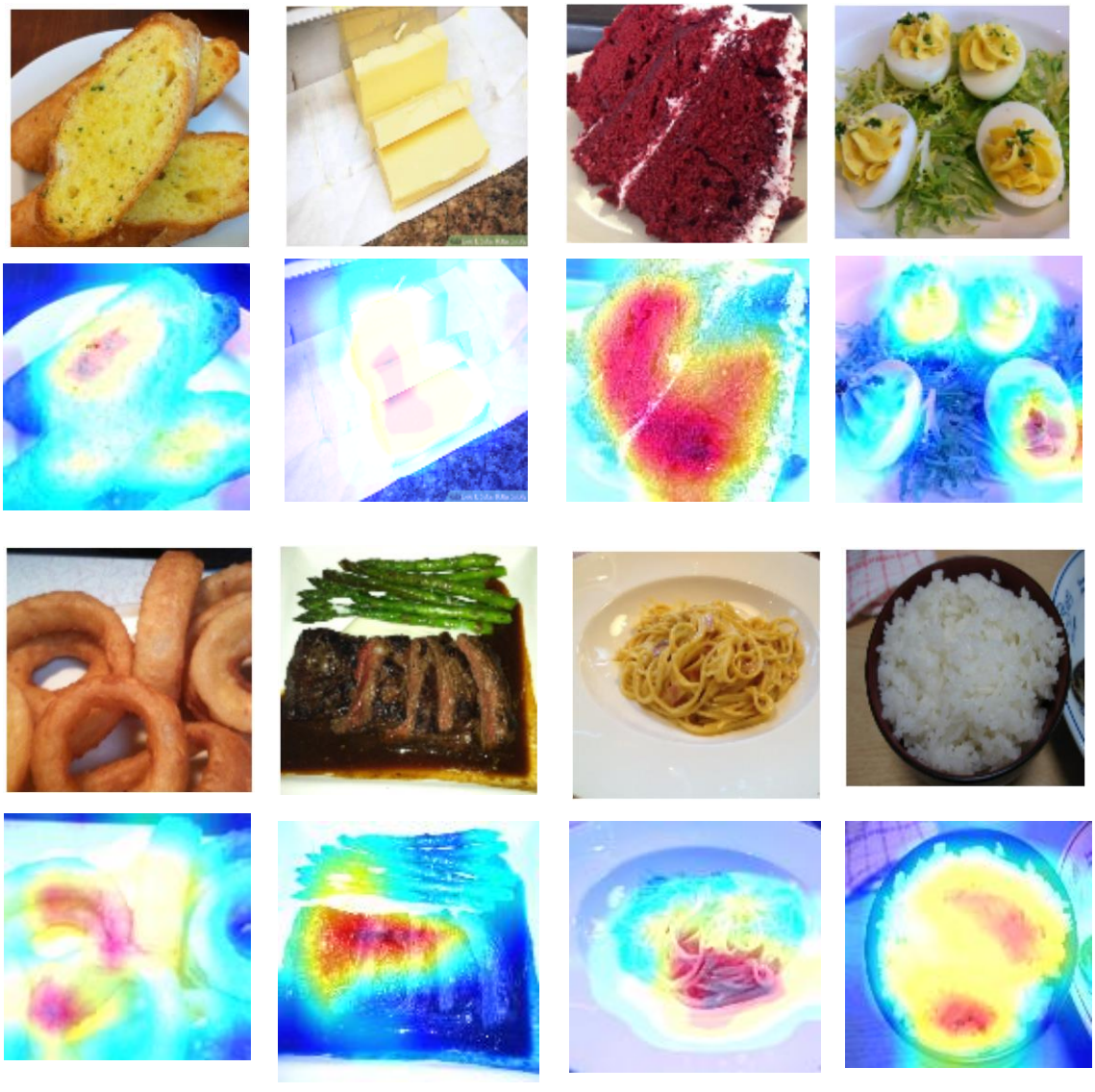
   實作方法：

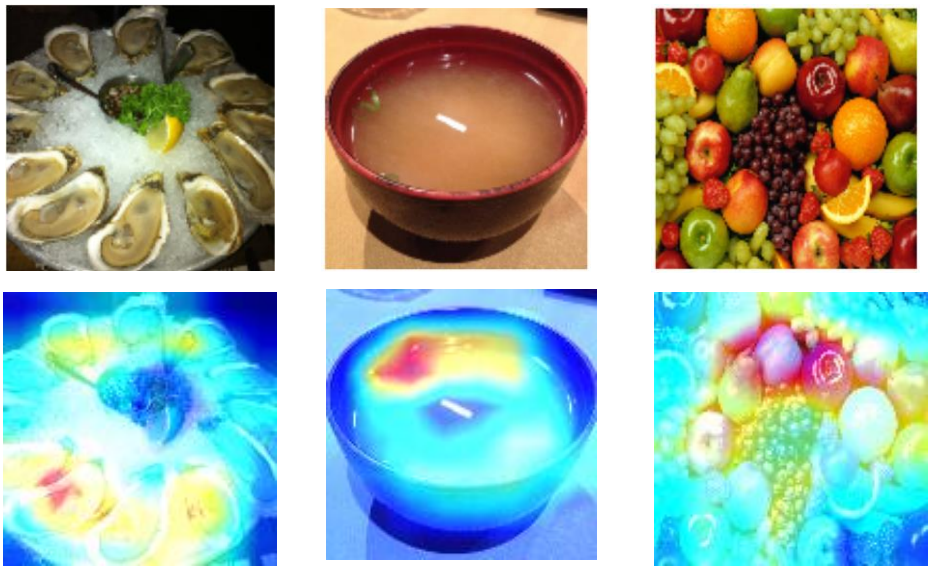
   創建一個新的 model — CAM_class ifier 繼承原先訓練好的 cnn classifier 的架構和參數。在經過最後一次 ReLU 後，產生一個 hook，紀錄當層的 activation map 和

gradient。之後再補上一次 max pool 和剩下的 feed forward network，使新的模型和舊的模型架構相同。從欲解釋的類別的 output 做 back propagation，求之前 hook 設好的 activation map 的 gradient。將 acitivation map 取出，與平均後的 gradient 加權，再經平均、relu、normalize，形成 heatmap。因為經過四次 Max pooling，此時的 heatmap 只有 8*8 的大小。將 heatmap resize 到 128*128，再將所有數值乘上 225（使原本的值域從[0,1]轉成[0,255]），然後用 cv2.COLORMAP_JET 轉換成 RGB 的圖像（紅色代表重要、藍色代表不重要）。和原本的圖片重疊，形成最終輸出的圖片。

參考資料：https://medium.com/@stepanulyanin/implementing-grad-cam-in-pytorch-ea0937c31e82

模型架構：與第二題附的圖片相同。





從 training set 每個類別各挑一個正確判斷的圖片做實驗。附上原圖比較。

觀察：

(1) 模型有將判斷重點放在食物本體上。例如：在 egg 的判斷中，蛋黃影響最大。

(2) 當有兩種食物同時出現在畫面中，例如：蛋和生菜、牛排和蘆筍、海鮮和檸檬片，模型也有學習到哪一項食物才是主角。

(3) 觀察 rice 和 soup 兩個類別。兩個類別同樣是用碗乘裝食物。模型在判斷 rice 類別時將重點放在碗內部的白飯上，而在半段湯時，重點放在周遭的碗型。和第三題用 Lime 得到的結果相同。可以推測是因為白飯有固定的顏色、形狀，而各種湯的顏色都不一樣，所以模型在判斷湯時的重點不是在湯本身長的樣子。