

Machine Learning HW5 Report

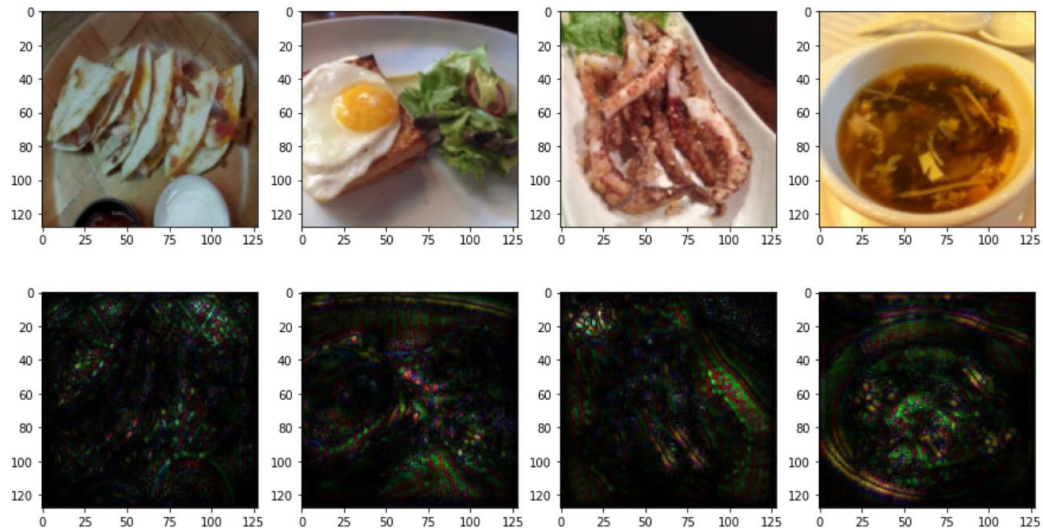
學號：B06901045 系級：電機三 姓名：曹林熹

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: b06303097 陳子晴)

ANS：

['Bread', 'Dairy product', 'Dessert', 'Egg', 'Fried food', 'Meat',
'Noodles/Pasta', 'Rice', 'Seafood', 'Soup', 'Vegetable/Fruit']

1. 第一張圖 (bread) 我們 focus 在餅皮的部分，旁邊的深色盤子也有一點被認出來是麵包，可能是顏色相近關係。
2. 第二張圖 (egg) 我們並沒有明顯的看出蛋黃的特徵，反而是白色的盤子以及蛋白讓我們辨識出蛋。蛋的蛋白也可以看到機器有明顯的辨認出來，蔬菜也很完美的避免到被辨認。
3. 第三張圖 (fried food) 看起來是炸魷魚，model 有順利辨識出一點酥皮的地方，觸角也蠻明顯被辨認，因為觸角外裹著酥皮。盤子白色部分並沒有被認為是炸物，因為白色圓形高機率被機器認為是蛋。
4. 第四張圖 (soup) 幾乎把整個碗圈出來，代表模型判定有碗就高機率是湯，而湯的表面也是有 focus 到，算是有把湯的輪廓描摹出來。



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: b06303097 陳子晴)

ANS :

可以先來看看我使用的 model，總共有 27 層。而總共做了 7 次的 convolution

，與同學討論後，認為在經過 ReLU 後得到的 filter 會較為準確，因此選定了 CNN (2)、(5)、(9)、(13)、(17)、(21)、(25) 的其中某幾層。在這裡，我選了 (2)、(9)、(17)、(25) 觀察。

```

Classifier(
  (cnn): Sequential(
    (0): Conv2d(3, 64, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
    (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (4): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): ReLU()
    (6): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (7): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (8): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (9): ReLU()
    (10): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (11): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (12): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (13): ReLU()
    (14): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (15): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (16): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (17): ReLU()
    (18): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (19): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (20): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (21): ReLU()
    (22): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (23): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (24): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (25): ReLU()
    (26): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (fc): Sequential(
    (0): Linear(in_features=2048, out_features=1024, bias=True)
    (1): Dropout(p=0.5, inplace=False)
    (2): ReLU()
    (3): Linear(in_features=1024, out_features=512, bias=True)
    (4): Dropout(p=0.5, inplace=False)
    (5): ReLU()
    (6): Linear(in_features=512, out_features=11, bias=True)
  )
)

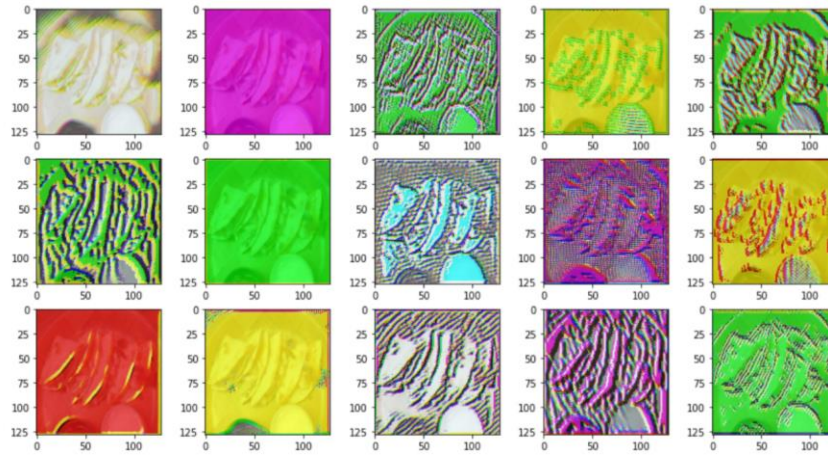
```

我們可以稍微看一下 bread 的 filter output，我只取其中 60 個做為觀察。

我們先用以下表格，對每一個 filter 編號。

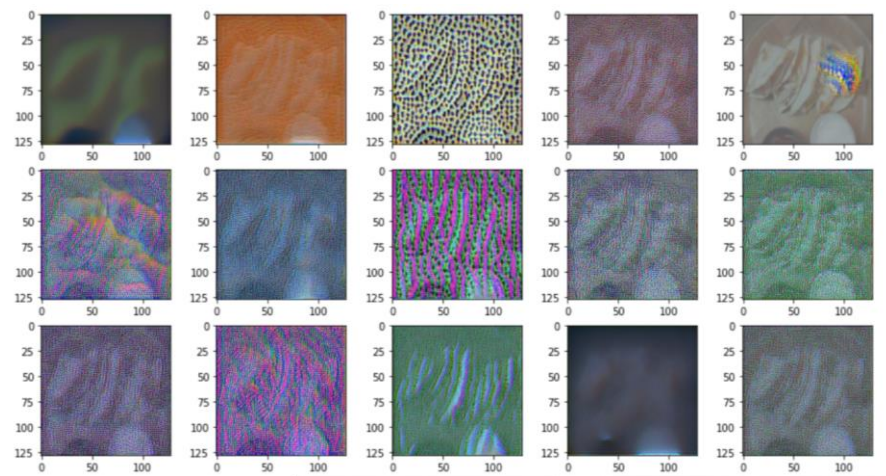
1	2	3	4	5
6	7	8	9	10
11	12	13	14	15

(2):



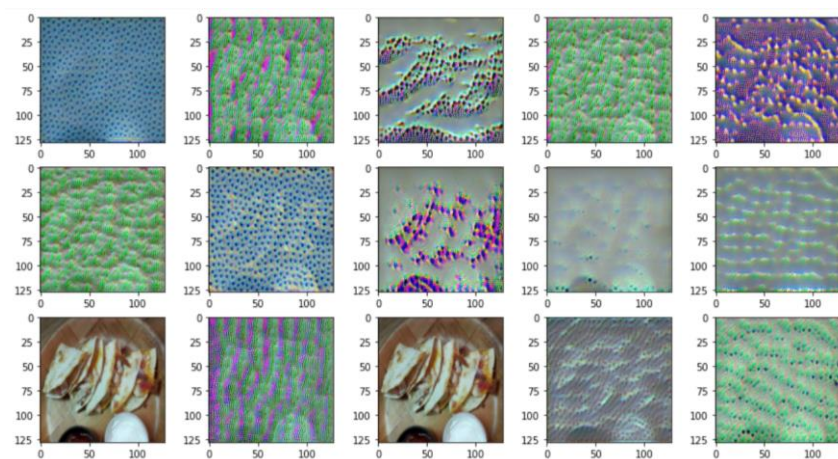
因為層數算淺，所以大致能看出 filter 有勾勒出麵包的圖樣。

(9):



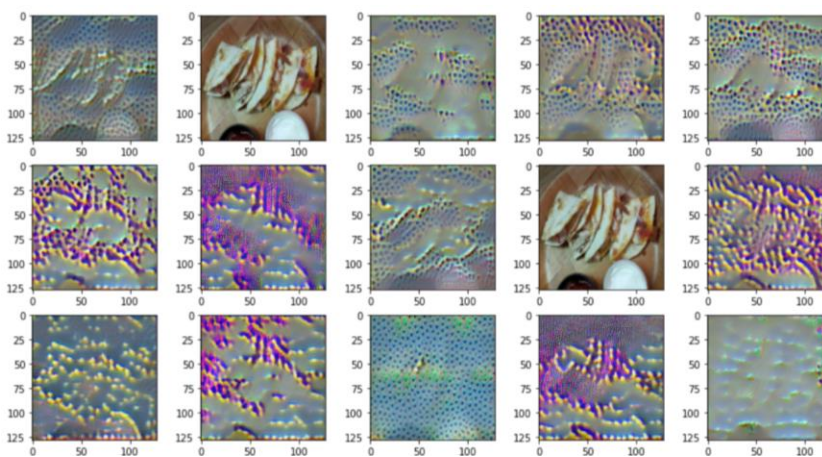
因為層數算中間，第一張圖有大致勾勒出輪廓，第五張圖跑出很奇怪的圖樣。

(17):

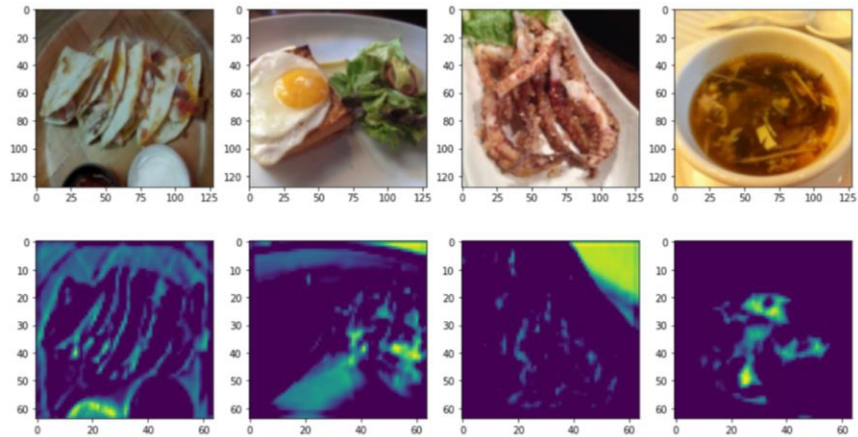


層數比較深了，開始有機器產生出的細節出現，甚至在 11、13 我們好像看不出機器產生什麼。

(25):



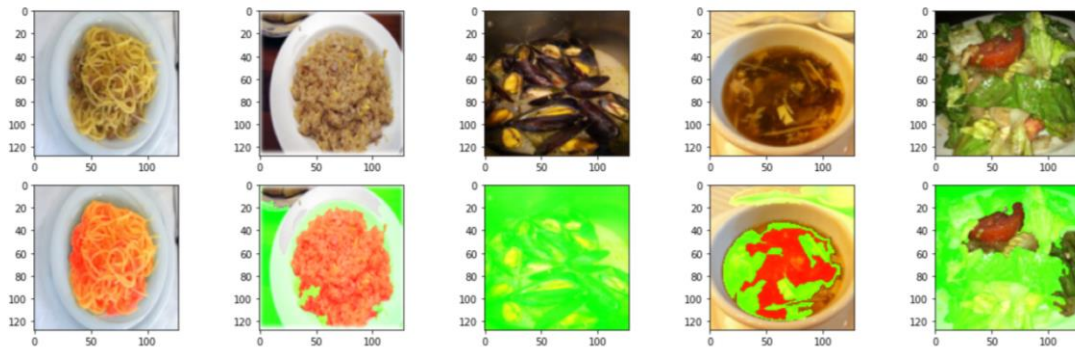
可以清楚看到機器認為麵包的重點，外型變得非常怪異，人眼無法辨識。



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: b06303097 陳子晴)

ANS :

這裡我列舉了幾項食物分析，列舉兩種差的跟三種好的判斷，以下將逐步介紹，並且將他的 Lime 圖片分析列舉幾張。



1. Noodles / pasta :

很差的判斷，我認為機器把它判定為飯，由肉眼看到下一張飯的圖也是很像。

2. Rice :

也是很差的判斷，甚至飯都沒有判斷正確，我認為與麵搞混。

3. Seafood :

可以看到機器很準確的判斷出整張圖，我認為是顏色與貝殼特徵的關係。

4. Soup :

有圓形的碗，還有湯的表面，不過可以看到湯的正表面反而是紅色，代表「單看碗中」的東西反而有礙辨認。

5. Vegetable / Fruit :

綠色的食物很好辨認為蔬果。

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: NONE)

ANS :