

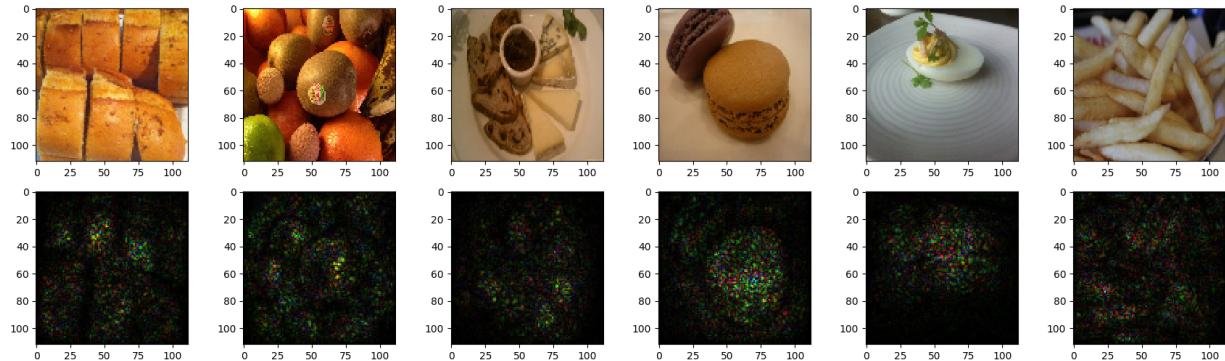
學號：r07921001

系級：電機所二

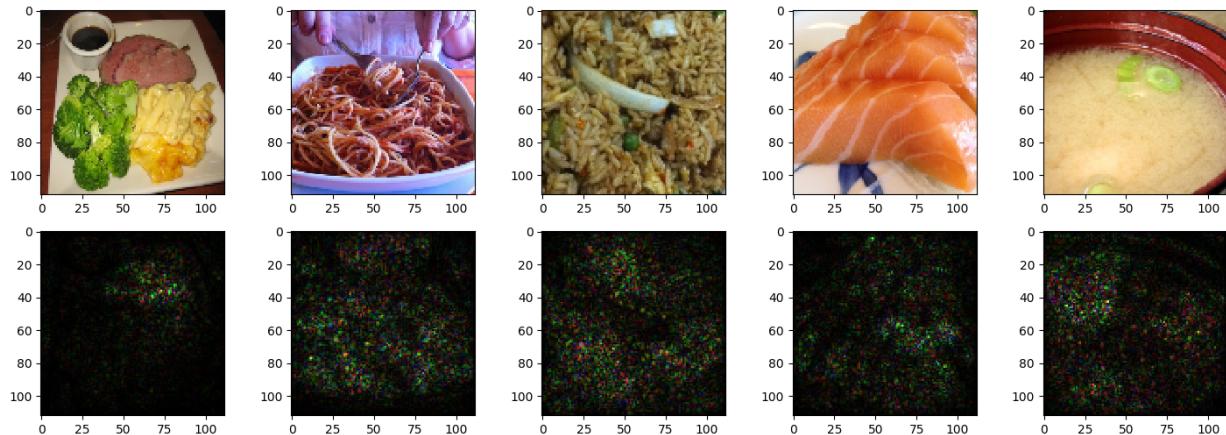
姓名：李尚倫

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: no)

img/saliency_maps_800_1602_2001_3201_4001_4800



)_5600_5600_7000_7400_8003



saliency map 的原理是對圖片中的 pixel 做出一個微小的變化，看看 predict 出來的分數會不會有什麼很大的變化，來判斷該 network 是否知道哪些 pixels 對於正確分類是重要的，寫作：

$$\{x_1, \dots, x_n, \dots, x_N\} \longrightarrow \{x_1, \dots, x_n + \Delta x, \dots, x_N\}$$

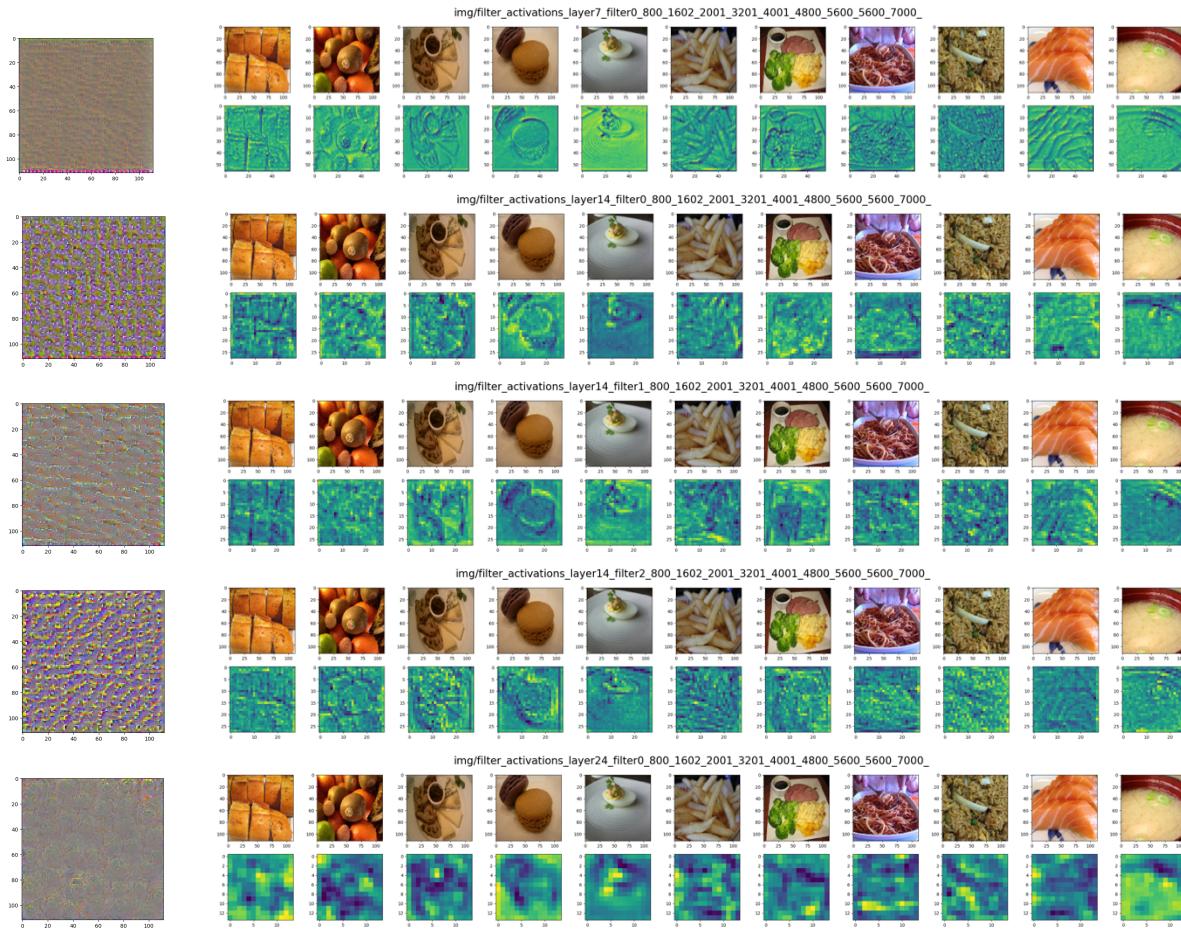
$$y_k \longrightarrow y_k + \Delta y$$

$$\left| \frac{\Delta y}{\Delta x} \right| \rightarrow \left| \frac{\partial y_k}{\partial x_n} \right|$$

可以透過對 x (input) 的偏微來取得。

從上面十一種分類的圖片的 saliency map 的結果來看，我們可以明顯清楚的看到，[乳製品, 甜點, 蛋, 肉] 的 saliency map 把目標物在有其他不相干的東西在旁邊的情況下標示得非常清楚，在正確的物品上 gradient 比較大，而其他區域則保持黑色，而[麵, 飯, 海鮮, 湯]的 saliency map 則是在影像中目標物幾乎滿版的情況下，正確地把是目標類別的地方標示出來，極少數區域不是目標類別的保持黑色，其他則因 gradient 大而顯示亮色。而最後剩下的類別[麵包, 蔬果, 炸物] 雖沒有完整的標示出全部的目標類別，但有標示到的也都有對。

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: no)

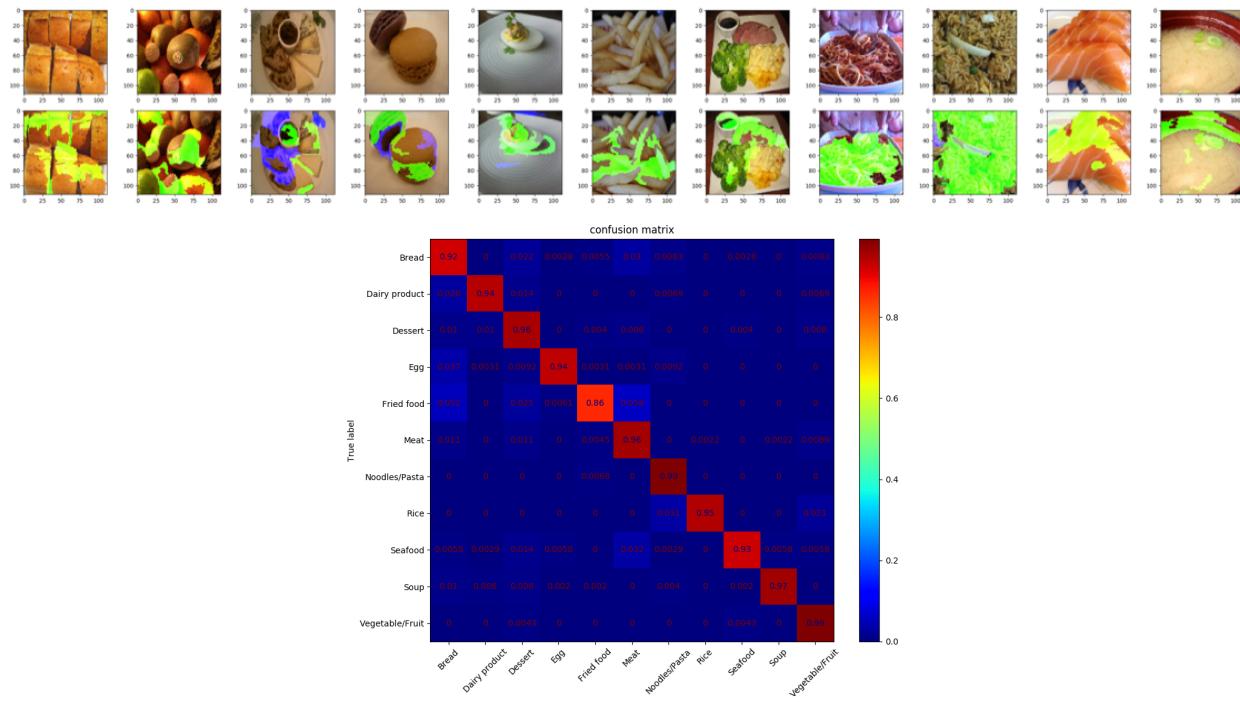


左圖是觀察 filter (filter visualization)是透過直切抽出 certain layer 然後 visualize。右圖則是透過 gradient ascent 來找到一個 x (image)能使 y (分類的 score)最大，藉而觀察 filter 在影像變成怎麼樣的時候會有最大的反應 (filter activation)，寫作：

$$x^* = \arg \max_x y_i$$

由上面的結果可以看到，layer7->layer14->layer24 因為 CNN network 有加入 maxpooling 的關係所以 filter 在辨識的層次上會越來越高(廣)，從認非常細緻的物體的線條到顆粒較大的且對比度明顯的邊界。而在第 14 層的 layer 上，選第 0, 1, 2 個 filter 也各自有有不同的效果，從 visualization 上可以看出他們分別代表的是直條紋、左上往右下紋、右上往左下紋，而能 activate 該 filter 的圖片也能看出對應的現象，雖然看起來都是顯示物體邊界的陰影，但較重的方向卻各自不同，由湯的 filter0 和 filter1(左上邊緣陰影 v.s. 右上邊緣陰影)、炸物的 filter1 和 filter2(右上往右下方向陰影 v.s. 左上往右下方向陰影)、麵包的 filter1 和 filter2(右上往右下方向陰影 v.s. 左上往右下方向陰影)可以看得比較明顯，證明這個 CNN network 內部眾多的 filter 裡真的是有學會邊緣的偵測和一些方向性 pattern 的判斷，並由這些數以萬計的 filter 得到的 feature 做非線性的疊加得到到分類結果。

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)。



本題使用的是 Local Interpretable Model- Agnostic Explanations (LIME) Library，原理是先透過 skimage library 的 segmentation function 把圖片切成一組 super pixels，然後再 fitting with linear (interpretable) model，可以寫作：

$$y = w_1x_1 + \dots + w_mx_m + \dots + w_Mx_M$$

$$x_m = \begin{cases} 0 & \text{Segment } m \text{ is deleted.} \\ 1 & \text{Segment } m \text{ exists.} \end{cases}$$

M is the number of segments.

因為式子很簡單，所以可解釋，如果那塊 super pixel 的 patch 的 weight 是正的的話就代表這個區塊對於辨識出正確的類別很有幫助，mask 標示為綠色(上圖因為經過轉色所以偏黃)，負的表示為這個區域不是該目標類別，mask 標示為紅色(上圖因為經過轉色為藍色)。由結果可以看出[蛋, 肉, 麵, 飯]都很明顯地標示出了目標類別所在的位置，[乳製品]則不但標出了乳酪蛋糕還正確地把巧克力蛋糕標示為非乳製品，[甜點]則標示出了馬卡龍側邊多層次的地方，推測可能是因為甜點類的東西，像蛋糕也都是側邊有很多層，所以是判斷甜點的關鍵，[湯]則看起來偏向於在辨識碗的形狀出現，就會被認為是湯，不過選出來的這張照片還有一個重點，就是湯上加的蔥也被 label 為重點，看來喝湯就是要加蔥啊，湯類別的添加物也會是 classifier 判斷的一個重點，畢竟大多照片的湯都只有一個平面太 texture-less 了，[蔬果]也有點異曲同工之妙，奇異果上的標籤也有被 label 到，或許就是判斷眾多圓形物體是否是蔬果的一個關鍵，而根據上次的 confusion matrix，倒數三名 0.86~0.93 score 的類別分別為[炸物, 麵包, 海鮮]，確實也都標示的沒有很好，雖然沒有標錯，但標得不是很明確。

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。

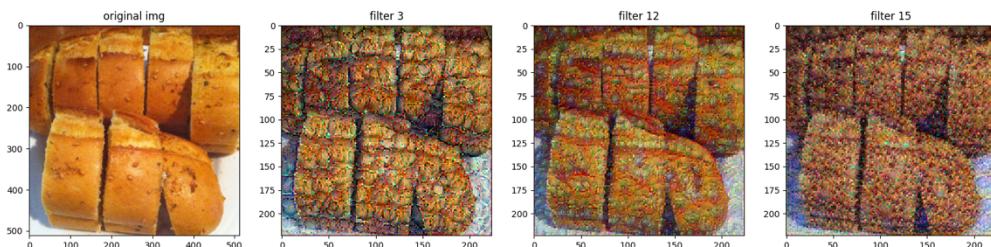
deep dream in different class, filter3, iter201, layer30



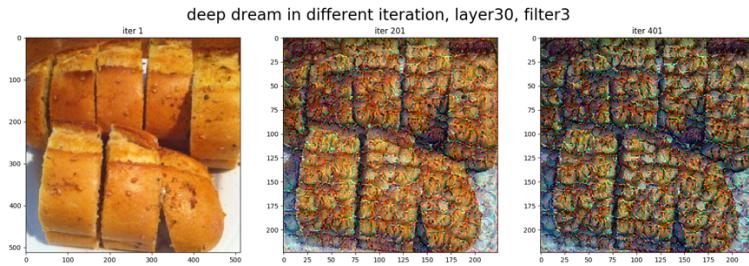
這題我實作的是 google 提出的 deep dream，其實做方法類基本上和第二題一樣，是使用 gradient ascent 去找一張 input 圖片 x 使 y 分類的 score 分數越大越好，可以寫作：

$$x^* = \arg \max_x y_i$$
 原文是：We ask the network: “Whatever you see there, I want more of it!”而其中與第二題的差別就在於，deep dream 的 x 是用圖片(非雜訊)開始，並且在輸出時保持 x 維持在圖片的狀態， x 餵進去時位於 0~1 的區間，經過 activate 後可能會超過，而超過的 trim 掉，再 *255 還原回 pixel 數值的圖片(非第二題的單色梯度 map)，因此可以看到 filter activate 後的圖片長什麼樣子，如果是 train 在動物的 dataset 上可能能看到一些動物頭的 pattern 出現。而根據我做出來的結果挑的是 CNN 的第 30 層，第 3 個 filter，201 個 iteration 的結果，雖然不到 pattern 看起來直接就像個什麼，但根據[飯]的結果，我覺得這個 filter 偏向反映出米飯的 pattern。而若選用同層不同的 filter，如下圖：

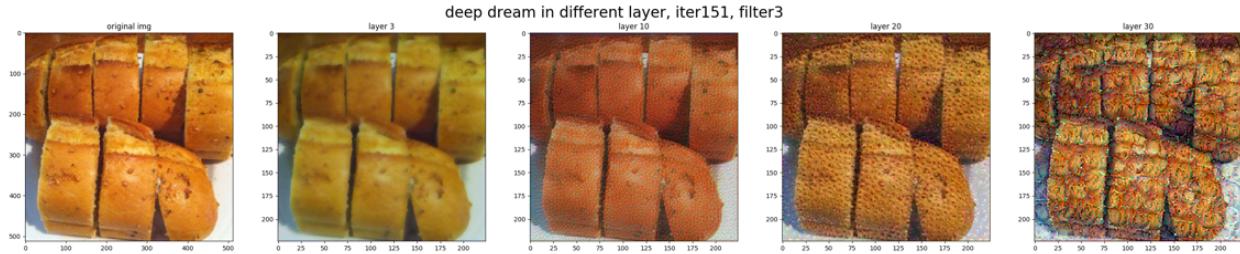
deep dream in different filter, iter201, layer30



也會有不一樣的效果除了 filter3 像是飯，filter12 很像是青豆，而且很密集，看了蠻噁心的 QQ，filter15 也蠻奇妙的，加讓去之後看起來很像是毛的感覺，像是奇異果表面的毛。而若是不同的 iteration，往更多走，則會有更深的 pattern 出現，如下圖：



而若是不同的 layer，從淺到深，如下圖有左到右：



在還很淺的時候還沒有什麼 high level 的 pattern，越來越深之後則會反映出越來越明顯的 pattern，由上圖看起來是從很細微的 blur，到很細的點點，到粗點點，到米飯形狀的 pattern or 蝦仁 or 咖啡豆 or 杏仁狀。不過綜合的來說這個方法比較適合拿來觀賞，相比前面，較無法客觀的分析 model 的能力，但確實非常的 attractive，尤其是原文 google blog 發表的那些圖片，非常的驚艷，應該還是有經過一些 regularization terms and hyperparameter tuning。

Reference:

[1] Problem1: Saliency Map

Karen Simonyan, Andrea Vedaldi, Andrew Zisserman, “Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps”, ICLR, 2014

[2] Problem2: gradient ascent

Yosinski, Jason, et al. "Understanding neural networks through deep visualization." *arXiv preprint arXiv:1506.06579* (2015).

[3] Problem3: LIME

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. "" Why should i trust you?" Explaining the predictions of any classifier." *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016.

[4] Problem4: deep dream

Mordvintsev, Alexander, Christopher Olah, and Mike Tyka. "Inceptionism: Going deeper into neural networks." (2015).

<https://github.com/utkuozbulak/pytorch-cnn-visualizations>