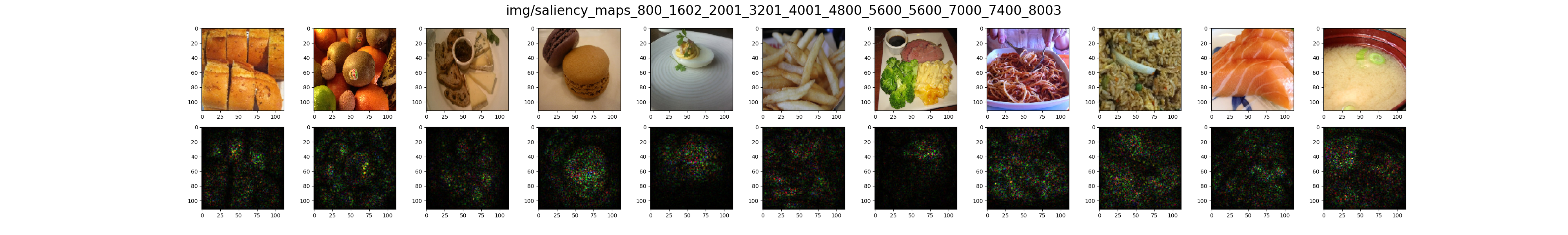
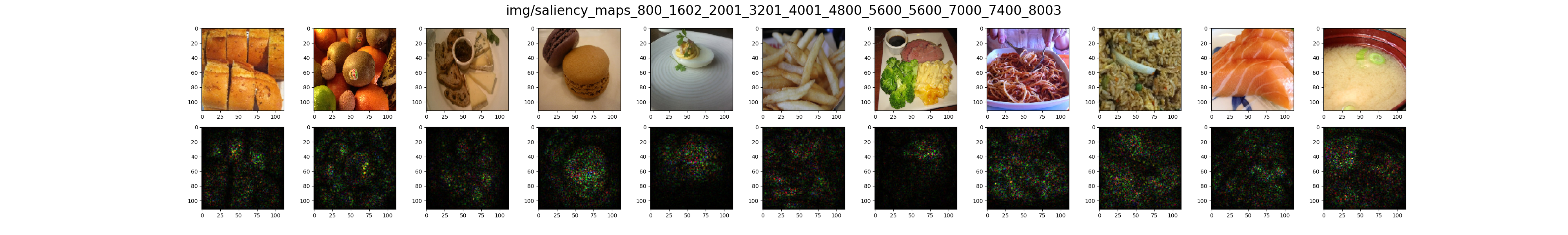
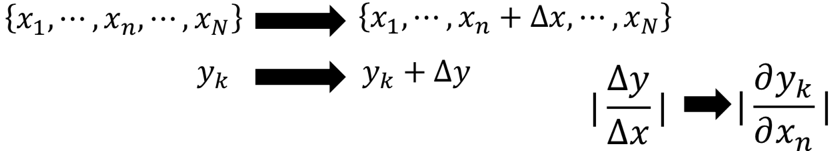
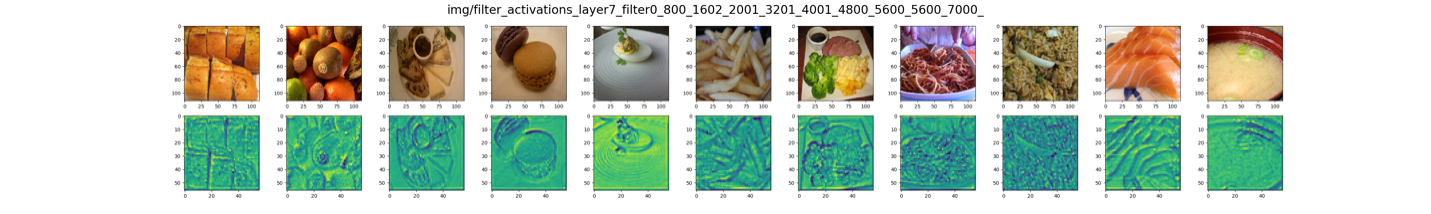
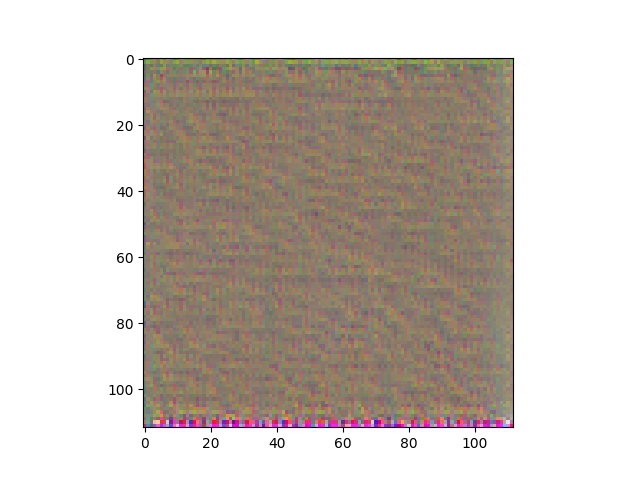
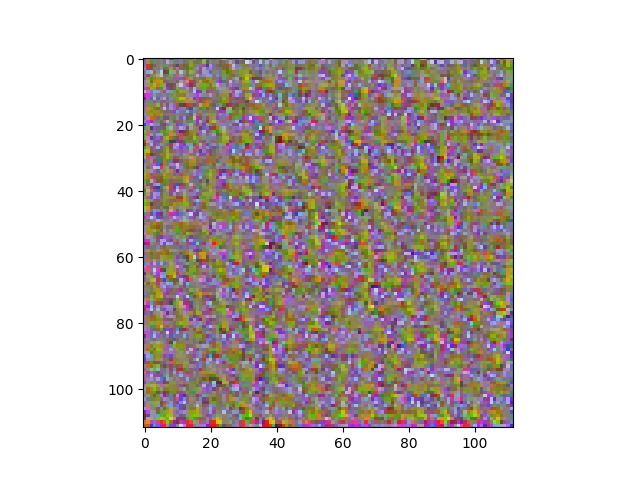
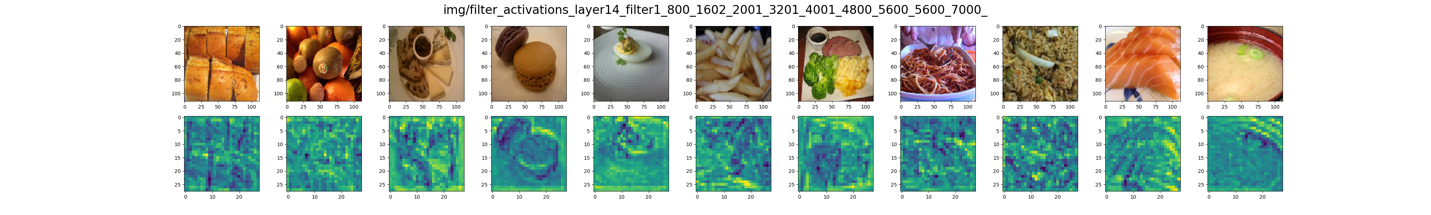
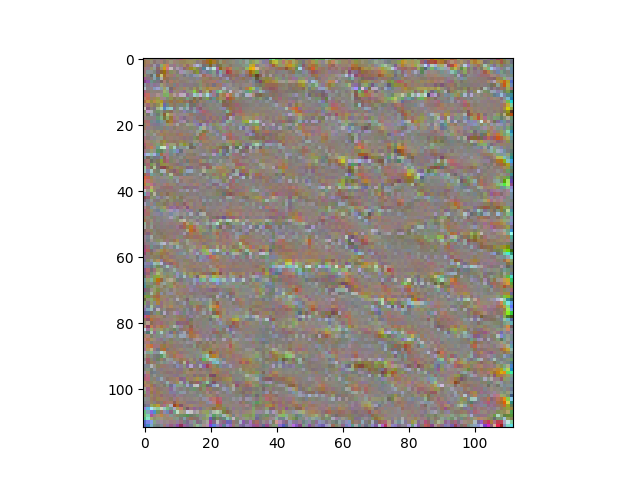
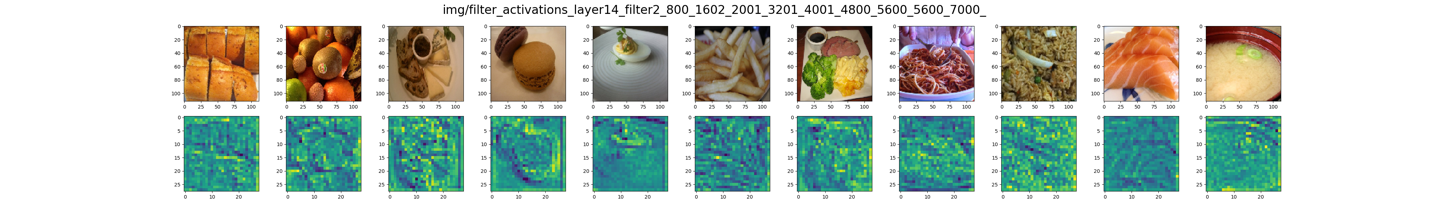
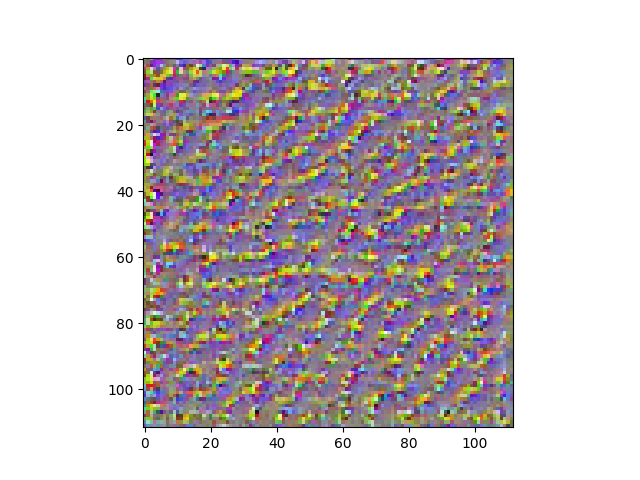
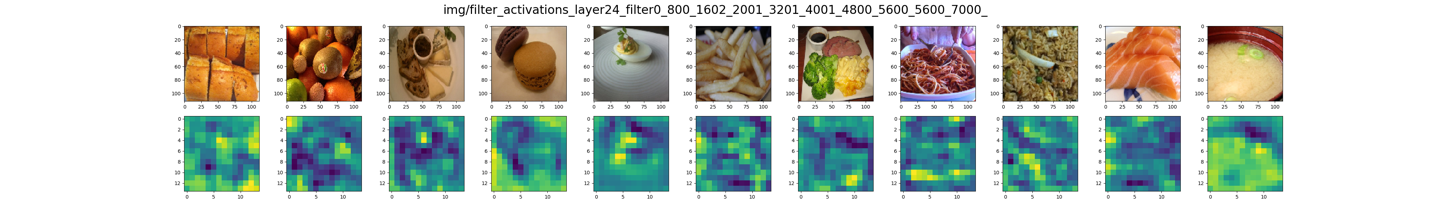
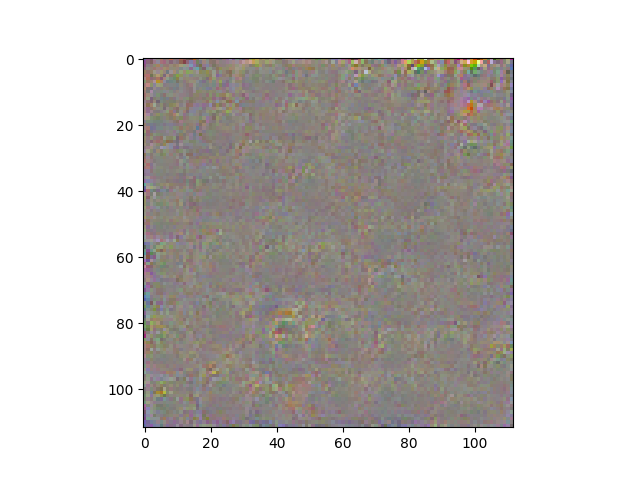
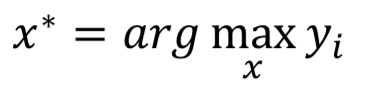
學號：r07921001 系級：電機所二 姓名：李尚倫

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)

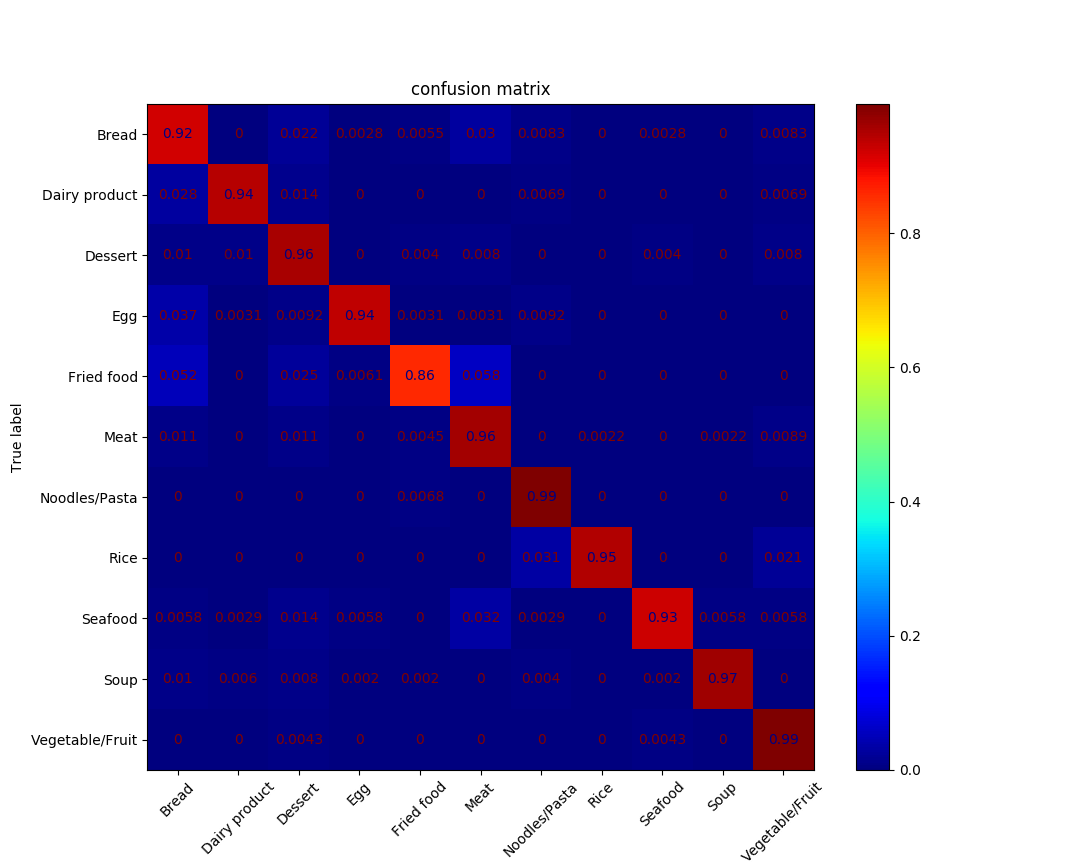
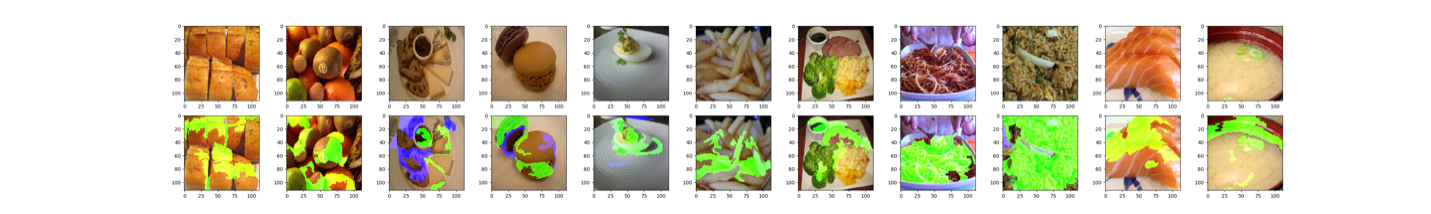
  
  
saliency map 的原理是對圖片中的pixel做出一個微小的變化，看看predict出來的分數會不會有什麼很大的變化，來判斷該network是否知道哪些pixels對於正確分類是重要的，寫作：  
 可以透過對x (input) 的偏微來取得。  
從上面十一種分類的圖片的saliency map的結果來看，我們可以明顯清楚的看到，[乳製品, 甜點, 蛋, 肉] 的saliency map把目標物在有其他不相干的東西在旁邊的情況下標示得非常清楚，在正確的物品上gradient比較大，而其他區域則保持黑色，而[麵, 飯, 海鮮, 湯]的saliency map則是在影像中目標物幾乎滿版的情況下，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)

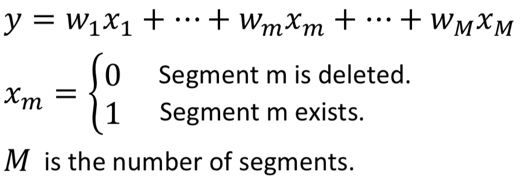
    

左圖是觀察filter (filter visualization)是透過直切抽出 certain layer然後visualize。右圖則是透過gradient ascent來找到一個x (image)能使y(分類的score)最大，藉而觀察filter在影像變成怎麼樣的時候會有最大的反應 (filter activation)，寫作：  
 。  
由上面的結果可以看到，layer7->layer14->layer24因為CNN network有加入maxpooling的關係所以filter在辨識的層次上會越來越高(廣)，從認非常細緻的物體的線條到顆粒較大的且對比度明顯的邊界。而在第14層的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)。

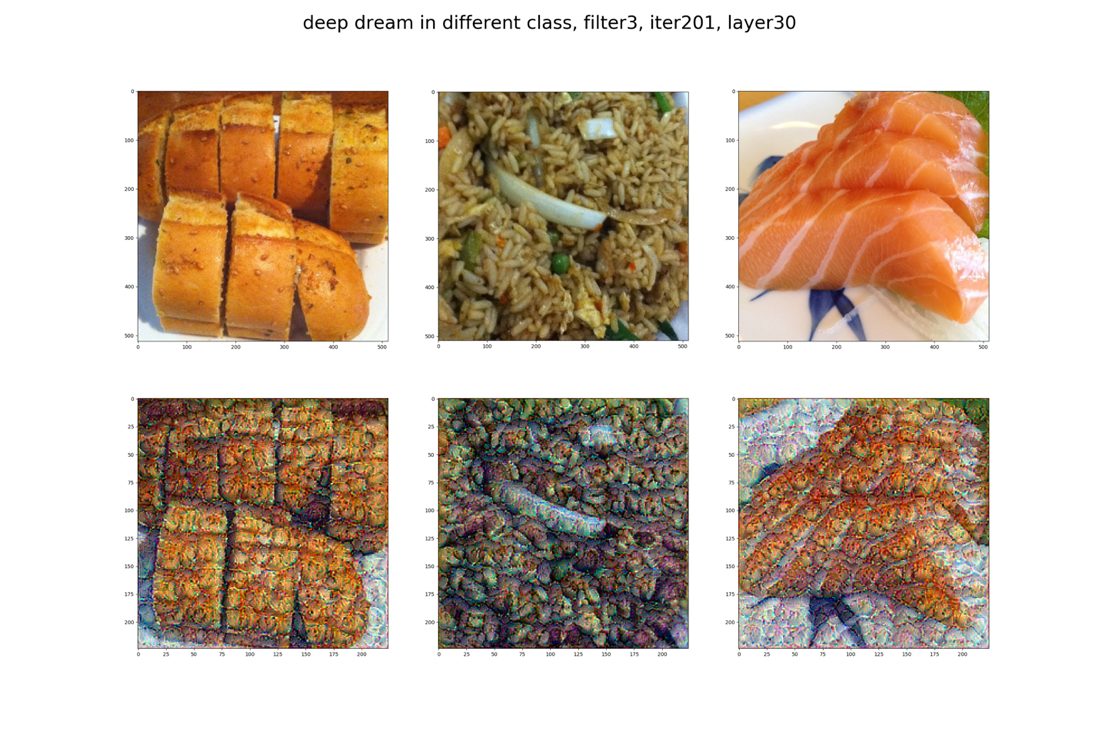


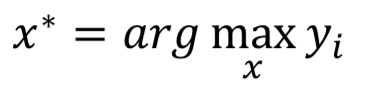
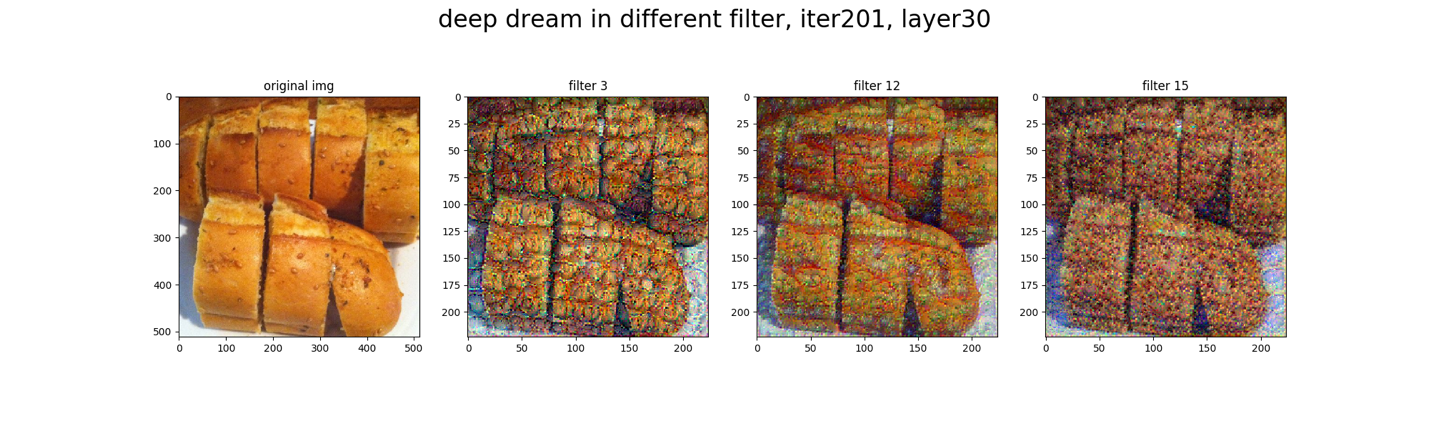
本題使用的是Local Interpretable Model- Agnostic Explanations (LIME) Library，原理是先透過skimage library 的segmentation function把圖片切成一組super pixels，然後再fitting with linear (interpretable) model，可以寫作：

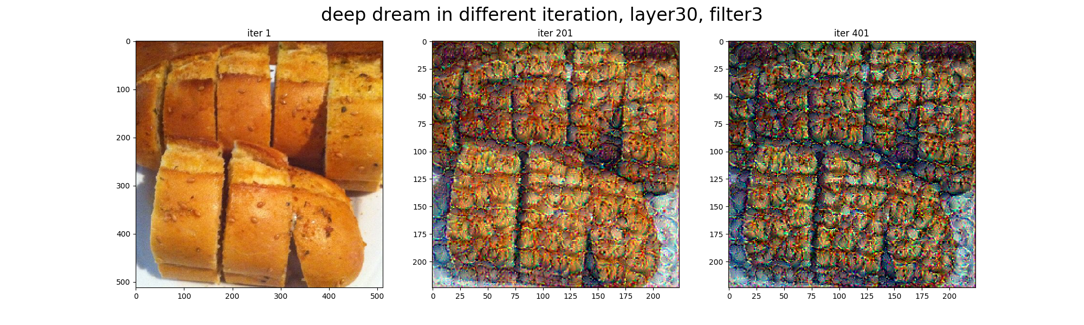


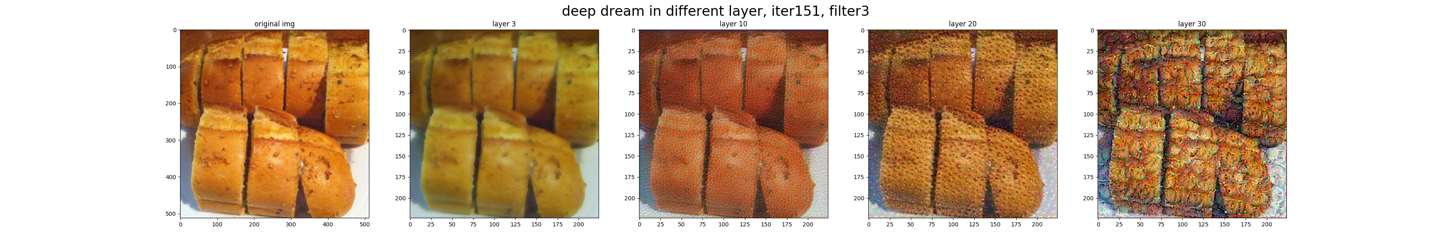
因為式子很簡單，所以可解釋，如果那塊super pixel的patch 的weight是正的的話就代表這個區塊對於辨識出正確的類別很有幫助，mask標示為綠色(上圖因為經過轉色所以偏黃)，負的表示為這個區域不是該目標類別，mask標示為紅色(上圖因為經過轉色為藍色)。由結果可以看出[蛋, 肉, 麵, 飯] 都很明顯地標示出了目標類別所在的位置，[乳製品]則不但標出了乳酪蛋糕還正確地把巧克力蛋糕標示為非乳製品，[甜點]則標示出了馬卡龍側邊多層次的地方，推測可能是因為甜點類的東西，像蛋糕也都是側邊有很多層，所以是判斷甜點的關鍵，[湯]則看起來偏向於在辨識碗的形狀出現，就會被認為是湯，不過選出來的這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。



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ㄧ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： 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：  
  
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>