

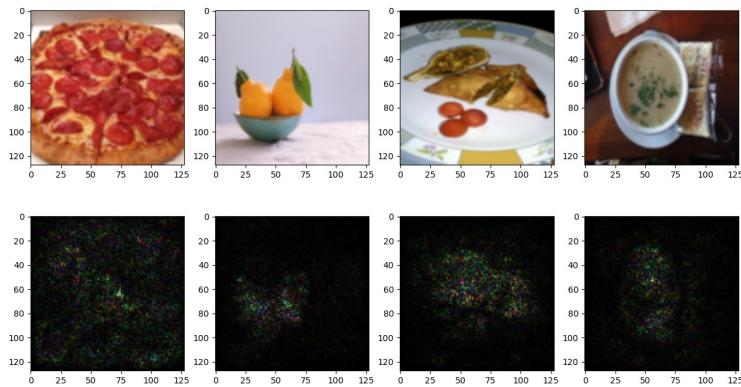
學號：B06902125 系級：資工三 姓名：黃柏瑋

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？

答：

dataset: training



就這四張圖而言，第一張圖(左一)對模型來說最為容易，因為目標占滿整張圖片，模型在做classification時不需要再花力氣偵測目標。至於第二張圖到第四張圖，其中有包含一些目標物以外的物品(像碗、裝飾物等等)，原以為模型在做判斷時會受到這些雜質影響，但根據saliency map，模型在為這些圖片進行classification時，也能很精準的判斷出目標物的本體，不會太過分心，藉此得出較為精準的分類結果。

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**。

答：

dataset: training

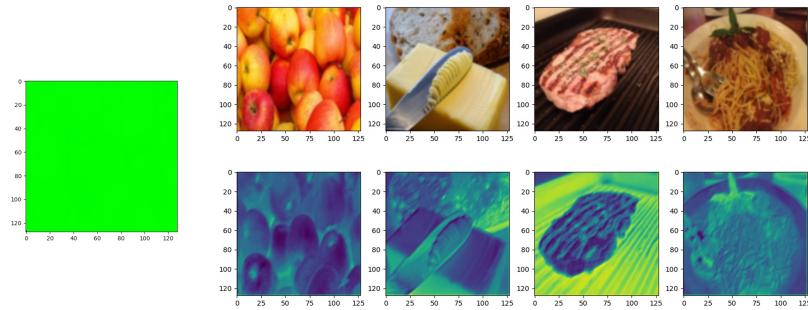
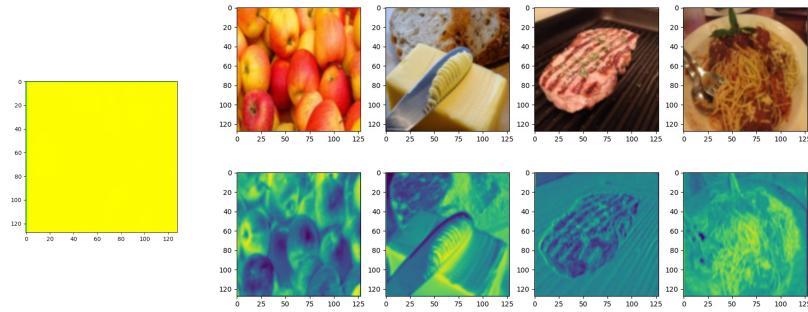
首先先來看模型架構。以下，我會挑出CNN的第零層、第四層與第八層進行討論。

```

self.cnn = nn.Sequential(
    nn.Conv2d(in_channels=3, out_channels=64, kernel_size=3, stride=1, padding=1), # [64, 128, 128]
    nn.BatchNorm2d(64),
    nn.PReLU(),
    nn.MaxPool2d(kernel_size=2, stride=2, padding=0), # [64, 64, 64]
    nn.Conv2d(in_channels=64, out_channels=128, kernel_size=3, stride=1, padding=1), # [128, 64, 64]
    nn.BatchNorm2d(128),
    nn.PReLU(),
    nn.MaxPool2d(kernel_size=2, stride=2, padding=0), # [128, 32, 32]
    nn.Conv2d(in_channels=128, out_channels=256, kernel_size=3, stride=1, padding=1), # [256, 32, 32]
    nn.BatchNorm2d(256),
    nn.PReLU(),
    nn.MaxPool2d(kernel_size=2, stride=2, padding=0), # [256, 16, 16]
    nn.Conv2d(in_channels=256, out_channels=512, kernel_size=3, stride=1, padding=1), # [512, 16, 16]
    nn.BatchNorm2d(512),
    nn.PReLU(),
    nn.MaxPool2d(kernel_size=2, stride=2, padding=0), # [512, 8, 8]
    nn.Conv2d(in_channels=512, out_channels=512, kernel_size=3, stride=1, padding=1), # [512, 8, 8]
    nn.PReLU(),
    nn.MaxPool2d(kernel_size=2, stride=2, padding=0)) # [512, 4, 4]

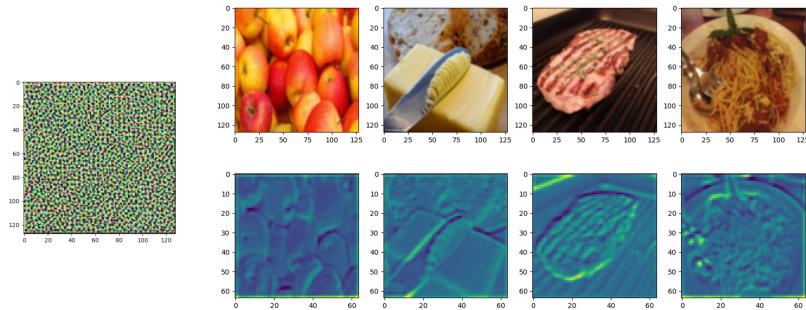
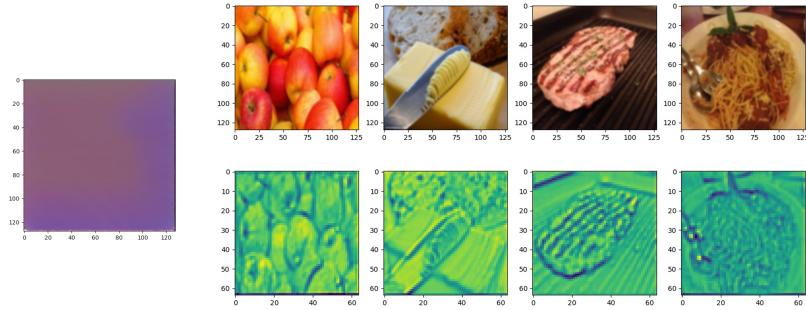
```

在第零層中，第25個和第55個filter的visualization和activation如下：



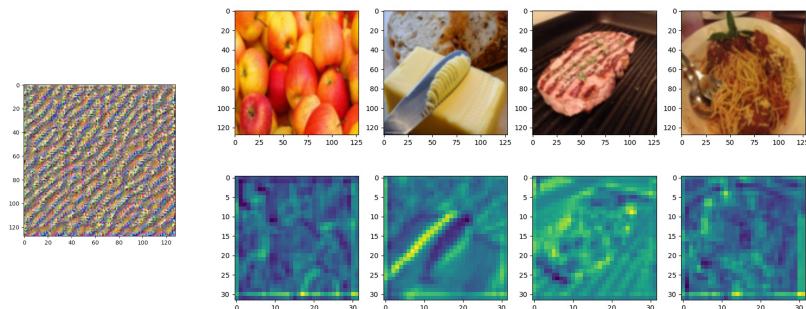
很顯然地，第零層主要會判斷物體的顏色。當圖片通過第25個filter時，黃色區域越多越深就越會activate該filter；而第55個filter則比較注重綠色或深色，只要有帶綠色的地方(如右一麵上的葉子)或像是黑色的地方(右二的煎鍋)，就會受到該filter較多的關注。

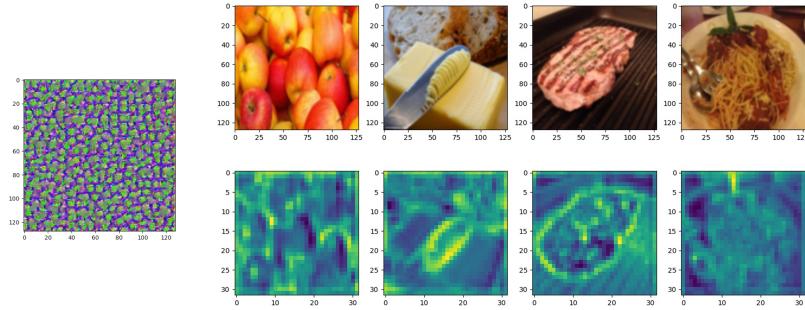
在第四層中，第64個和第120個filter的visualization和activation如下：



在第64個filter中，只要是漸層色就會被標註起來，像是左一蘋果的表面、左二刀子的反光等等；而在第120個filter中，我們能看到visualization的對比度和顆粒感相當重，再對照看右方的activations，就能發現物體的外框，或是分界比較明顯的部分會被特別注意。

在第八層中，第45個和第85個filter的visualization和activation如下：





第45個filter相當明顯，是用來找出右上至左下的斜直線，但主要還是比較適用於清楚的輪廓上，像是左二的刀背和右二的肉排邊緣；而第85個filter相當有趣，visualization上滿是小小的孔洞，一圈一圈地併在一起，對照到右方的activations來看，該filter應該是用來圈出物體完整的外框，左二和右二的圖最為明顯，奶油和肉排的外框都被完整的圈了出來。

最後，總結一下不同深度之間的差別：

- 第零層的filters主要用來判斷大方向的特徵，例如顏色
- 第四層的filters在顏色上有著更高的要求，不單純觀察純色而已，還觀察了顏色的變化。此外，也涉略一些形狀上的偵測，漸漸將物體的輪廓描繪出來。然而，有可能因為第四層處於過渡階段，activations都稍顯模糊，沒有畫出格外突出或深刻的重點。
- 第八層的filters對於物體形狀的描繪更加細膩，相較於第四層，filter的特色也相當鮮明，無論從visualization或是activations都能快速找到重點，可以說是模型中用來定位目標的好幫手，提升模型進行classification時的準確性。

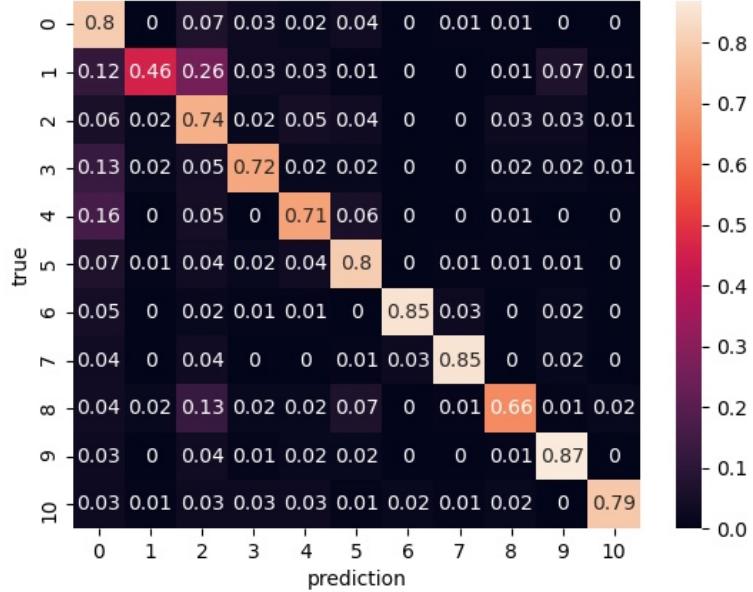
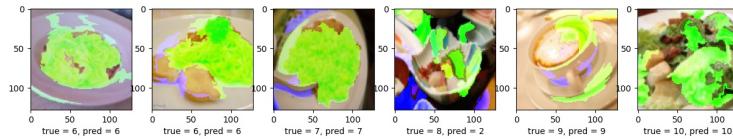
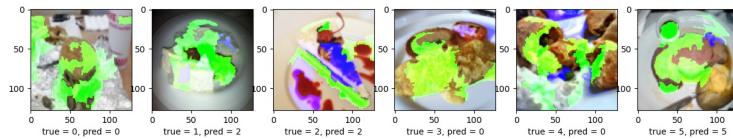
3. (2%)

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**)。

答：

dataset: validation

以下為利用lime套件畫出來的結果，以及模型在validation set上的confusion matrix(以true label為axis進行normalization)：



在lime的結果中，綠色為相關藍色為不相關，搭配confusion matrix，可以討論為何有些label很容易被混淆，而有些不會：

- true label為1但被predict成2的圖片不在少數，例如上排左二的圖。依照lime的分析，在分類該圖片時著重在奶油(class 1)本體，而他和蛋糕(class 2)有幾分相似，才會高機率被誤導成甜點。
- 接著解釋模型為何會常把true label為3的圖片 predict成0？因為class 3為蛋類，而蛋類經常和麵包放在一起，尤其當蛋已經打散時，長相更像麵包。由上排右三來看，模型定位的地方除了有真正的麵包之外，蛋的焦邊也很像某些麵包的顏色和花紋。
- 至於true label為4的圖片，經常會被判斷成0的原因可能是因為炸物有時會有一些包裝紙(如上排右二)，而許多麵包(class 0)的圖片中也有包裝紙。此外，炸物顏色也和麵包相近，所以模型才會時常搞混。
- 接著來看true label為8的照片，模型在判斷該照片時大多沒有將焦點放在對的目標上，真正的海鮮都沒被注意到，反倒是旁邊的碗或是裝飾的葉子等等，這和class 2的特徵相近，食物與碗之間有些偏方正的稜角，還有些點綴。
- 總歸而言，當模型將目標放錯位置時就會判斷錯誤，反之，表現可能就會相當不錯。

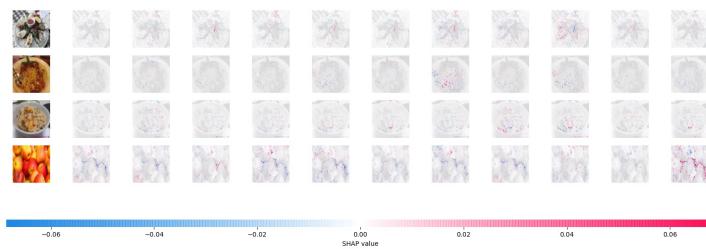
像模型在面對class 5, 6, 7, 9, 10的時候，都能準確抓住食物本體，或食物與餐具的關係(像下排右二中靠近碗邊緣的液體比較容易受到模型關注)。其中最成功的應該是下排左二，除了成功抓出麵的位置之外，還將旁邊的裝飾歸為負相關，考慮相當周詳。

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。

答：

dataset: training



本題我選用shap套件的deep explainer來解釋模型行為，每張圖片上會有一些紅點或藍點，對應不同的shap value。而對每個class label都會有一張shap value的分布，圖中某處的shap value越高，代表模型認為該處屬於該label的可能性越大。

由上圖我們能發現一些模型在分類時的判斷依據：

- 第一張圖片在class 8中，貝殼上的shap value比較高，這也代表貝殼的形狀讓模型判斷出他是屬於class 8。
- 第二張圖片在class 6中，麵體上的shap value上比較高，而在其他class中(如class 2、4或10)，麵體反而出現負值，這也表示麵體成功讓模型預測正確。
- 第三張圖片的結果有點讓人出乎意外，我原本預期和第二張照片一樣，飯的部分會是模型判斷該照片為class 7的主要依據，但根據shap做出來的結果，shap value在靠近碗的部分反而最高，而在會出現許多碗的class 9中，那些碗的部分卻沒有得到特別高的shap value。

可見模型在判斷這張照片時並不只是注意容器而已，更著眼於內容物在容器邊緣的狀況，像飯會與碗產生凹凸不平的交界，而湯與碗則會產生平滑的交界等等。

- 第四章圖片相當好解釋，蘋果的輪廓在class 10中擁有很多的shap value，而在其他的class中卻幾乎都是負值，所以模型可能會根據水果的外框將這張圖分類為class 10。