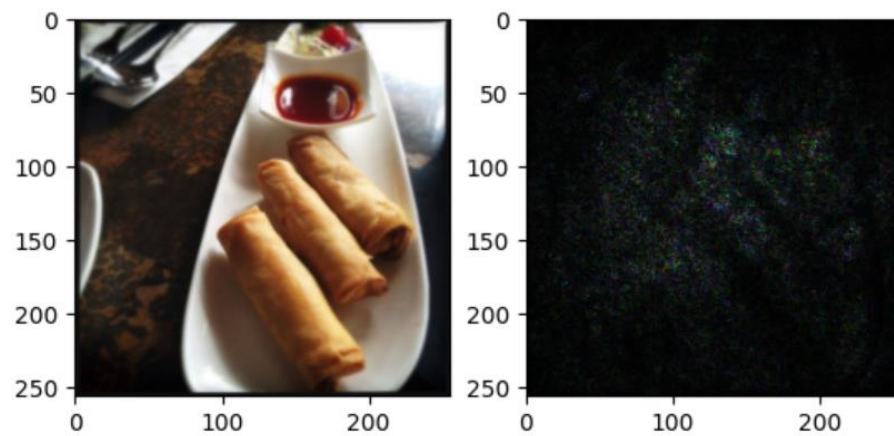
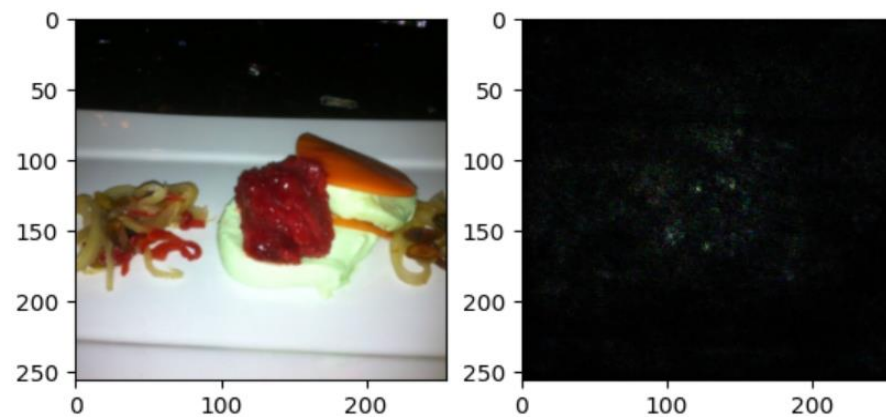
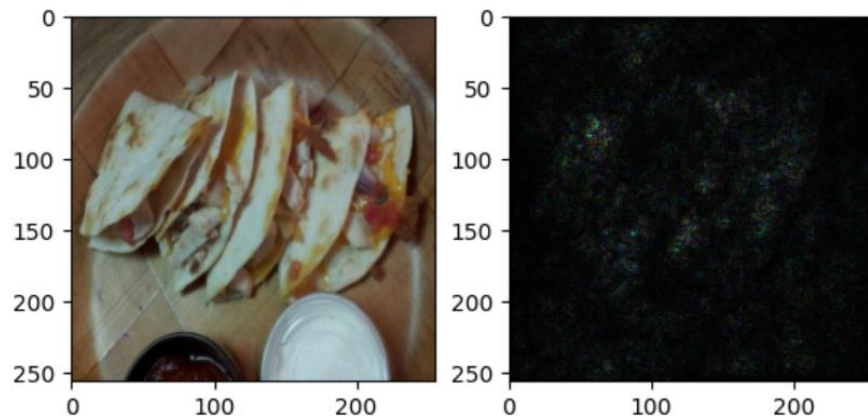


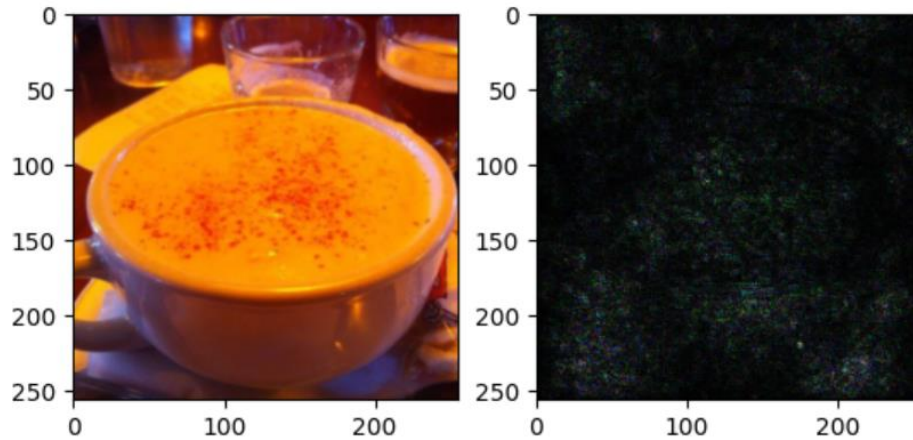
學號：b05901070 系級：電機四

姓名：蔡昌廷

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:)



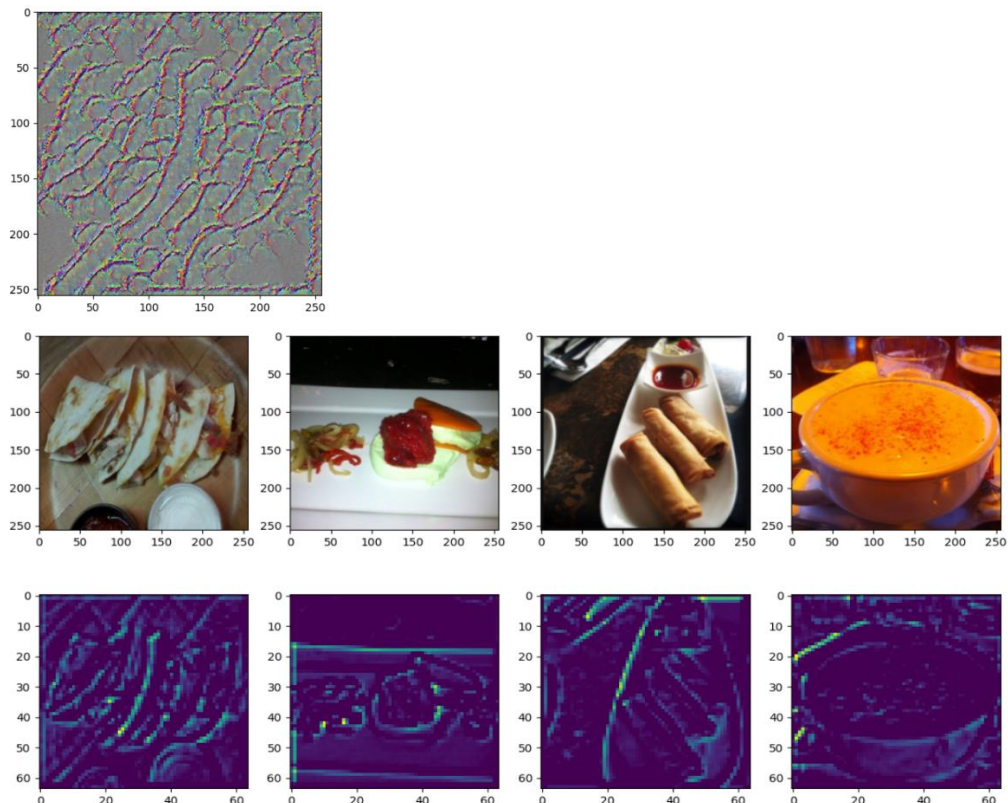


其實從上面四張圖中可以發現 saliency map 多多少少有 focus 在食物的某些部分，雖然顏色的色差不太明顯，第二張圖主要 focus 在醬料的部分，第三張圖有部分是 focus 在中間那條上面，最後一張則有 focus 在整個湯碗上面。

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:)

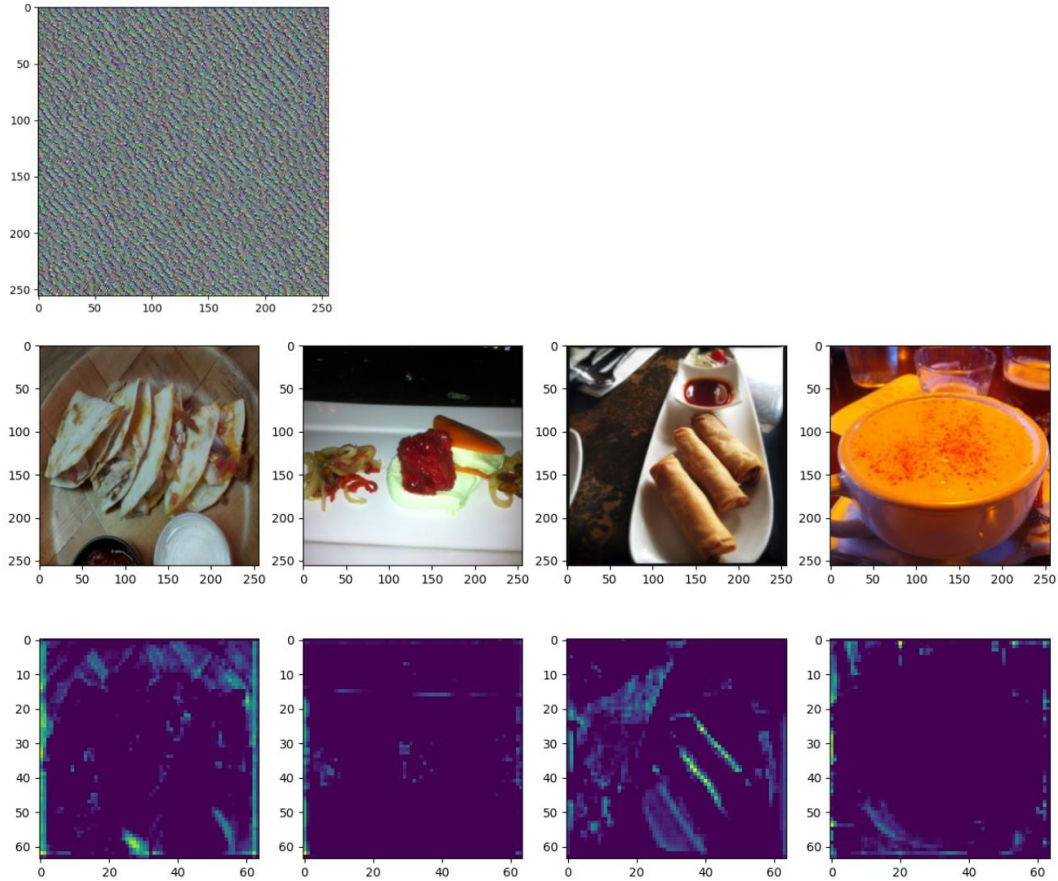
以下我以第 10 層之 CNN 的第 10、15 個的 filter 以及第 5 層 CNN 之第 20 個 filter 做呈現。

第 10 層第 10 個 filter :



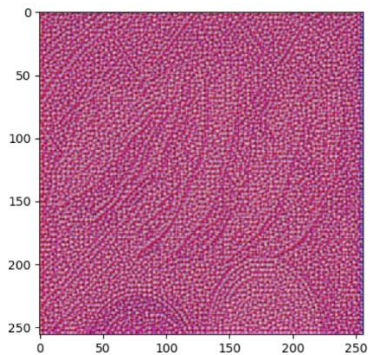
從以上可以發現此 **filter** 最主要是判斷是否存在右上到左下的斜線，圖片裡也可以看出比較亮的部分大多都是上述的斜線。

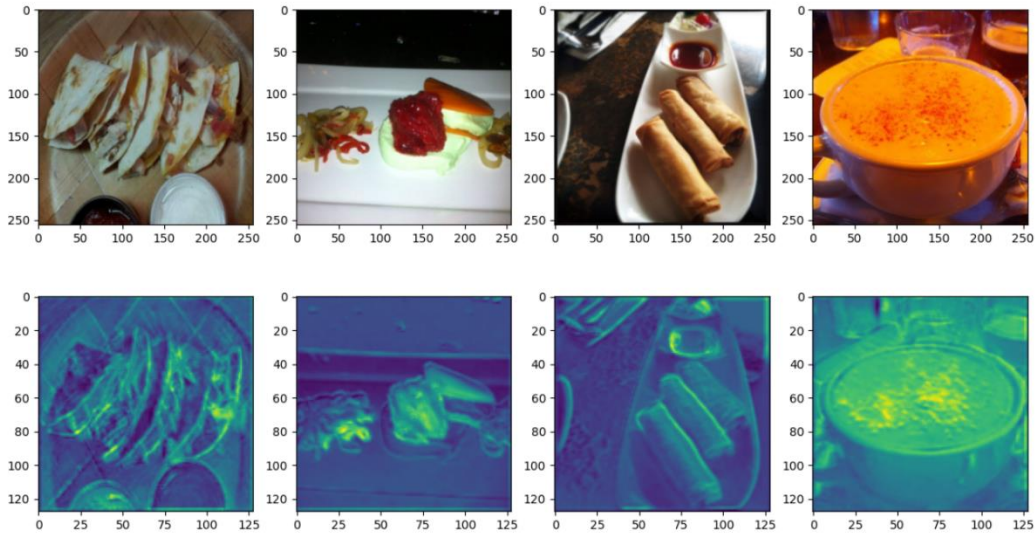
第 10 層第 15 個 filter：



從 **filter** 的 **visualization** 可以看出是在判斷是否存在左上到右下的斜線，相較於第二張跟第四張圖片幾乎沒有亮處，第三張圖片很明顯可以看出此現象。

第 5 層第 20 個 filter：



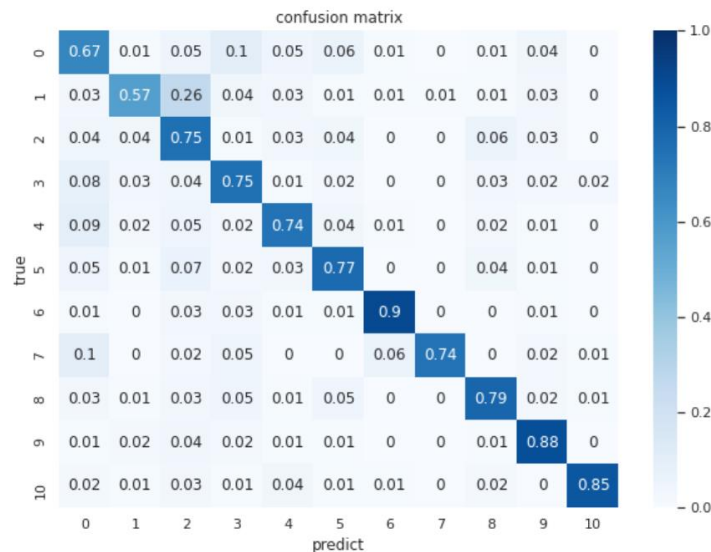


從 **filter** 來看，似乎是在判斷是否有紅色的成分，而下面的圖片 **activate** 的地方大多都是紅色成分偏高的現象，尤其第二張圖片的醬汁很明顯可以 **activate filter**，而第四張主要是因為整張照片偏向橘紅色，因此 **activate** 的區域也遠比其他圖片多。

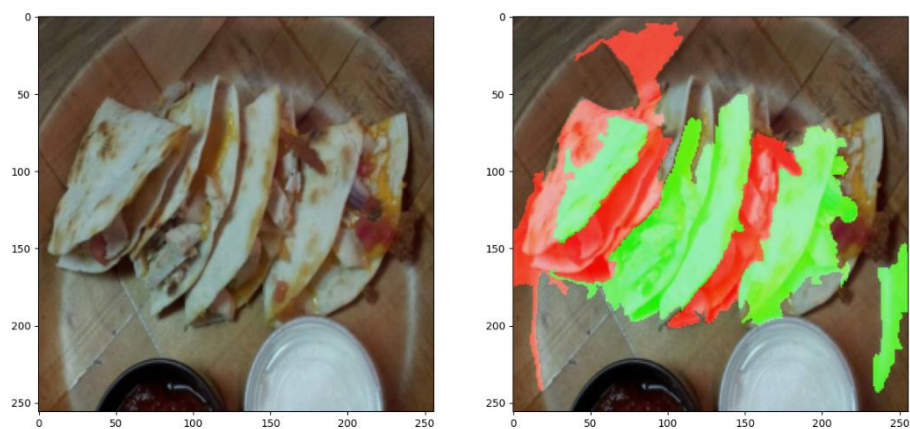
由上述三個不同 **filter** 可以看出每個 **filter** 都有屬於自己不同的任務，在第 10 層主要是判斷一些斜線，而第 5 層是判斷顏色的成分，可見在同一層之間 **filter** 的功能會比較有關聯。

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**)。

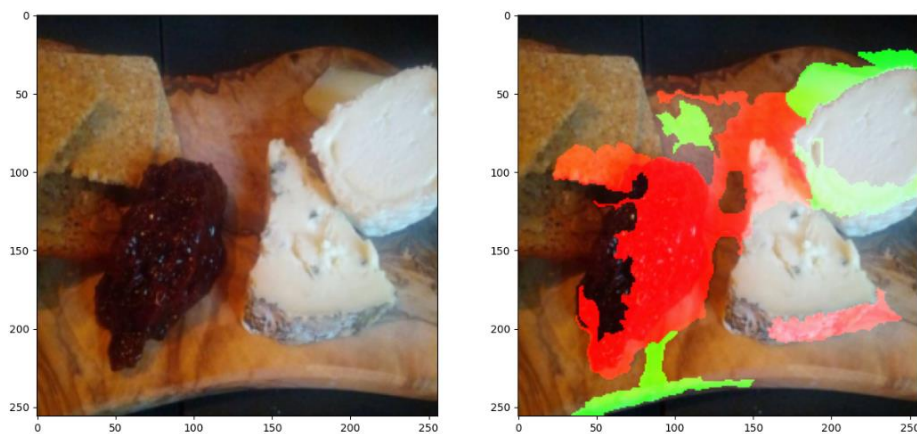
根據 **confusion matrix**，發現自己的 **model** 在 **label 6** 表現的最好，而在 **label 1** 上則表現最差。



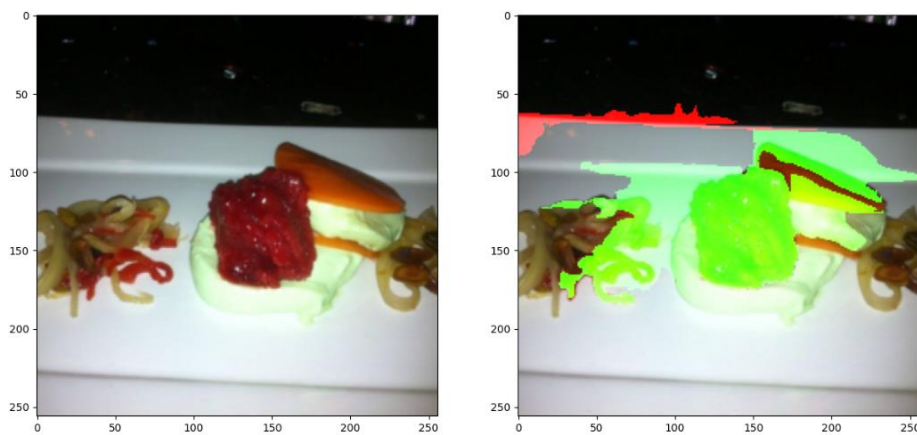
以下為各 label 從 0 至 10 的原始圖片以及使用 lime 分析的圖片。



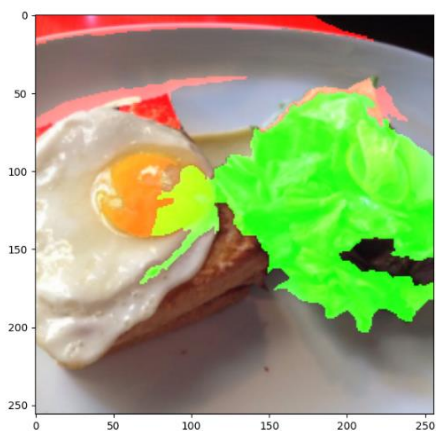
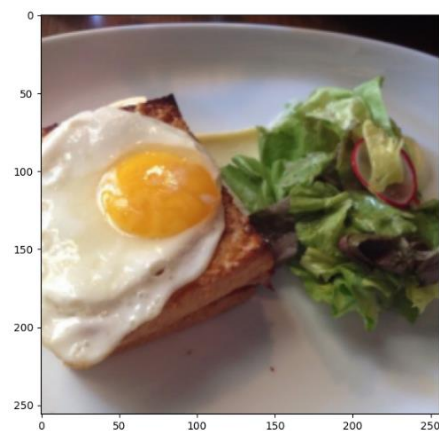
Label 0



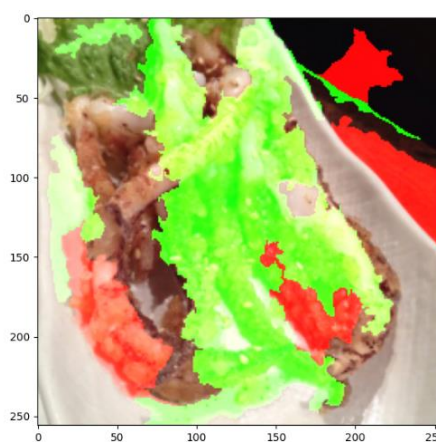
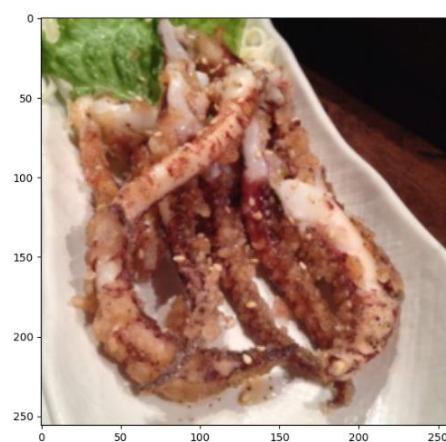
Label 1



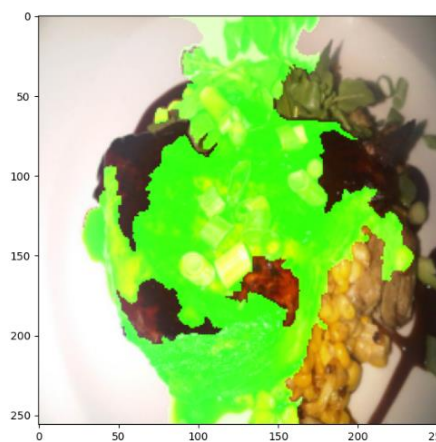
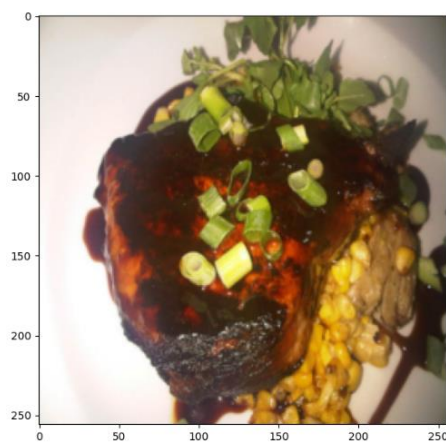
Label 2



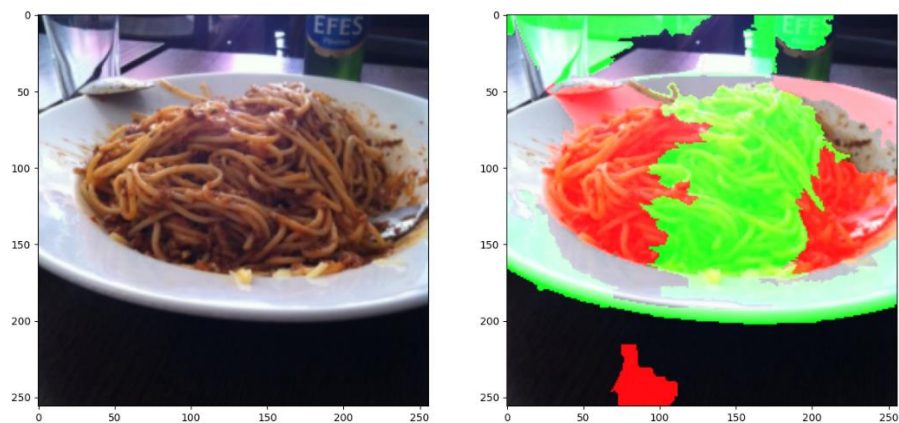
Label 3



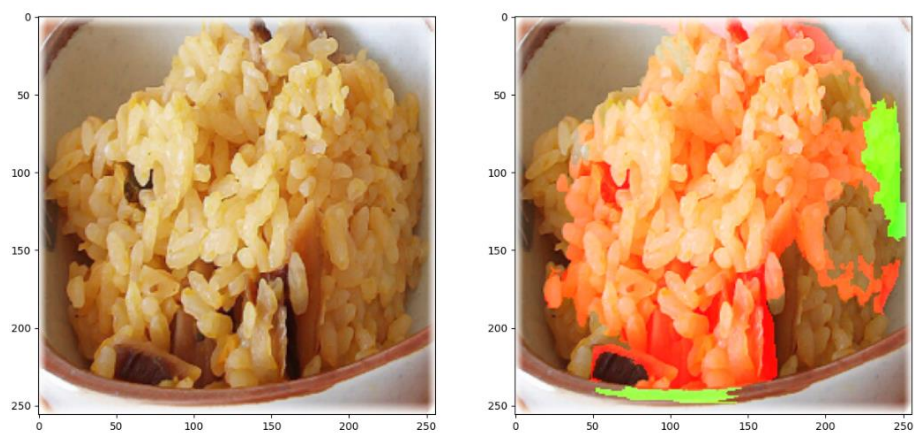
Label 4



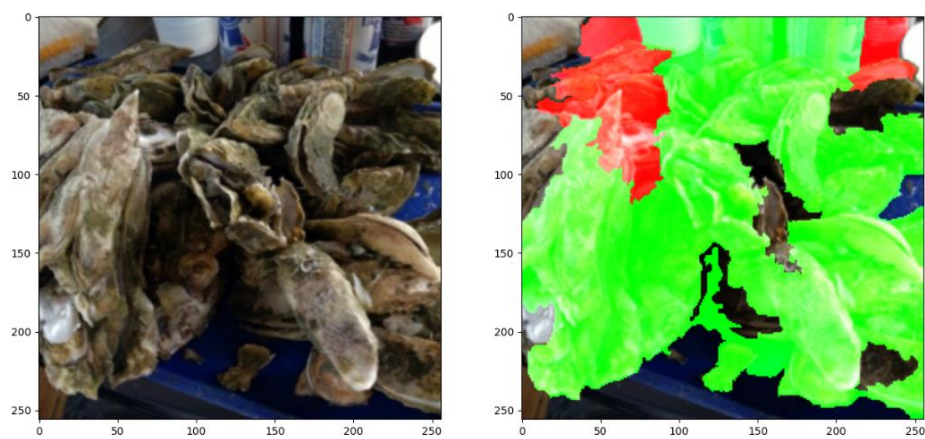
Label 5



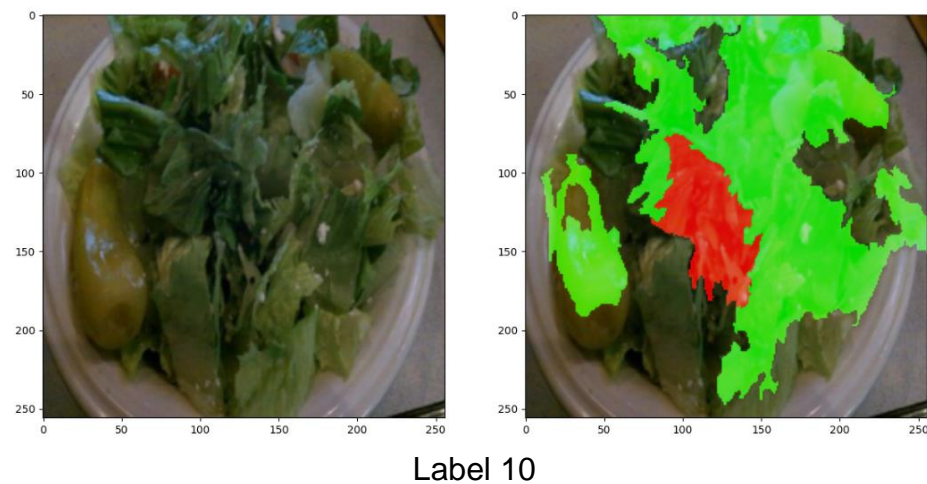
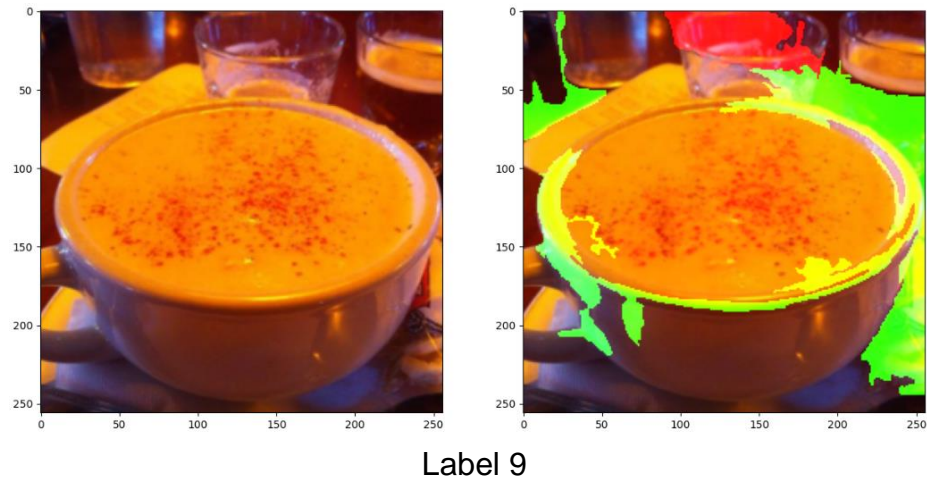
Label 6



Label 7



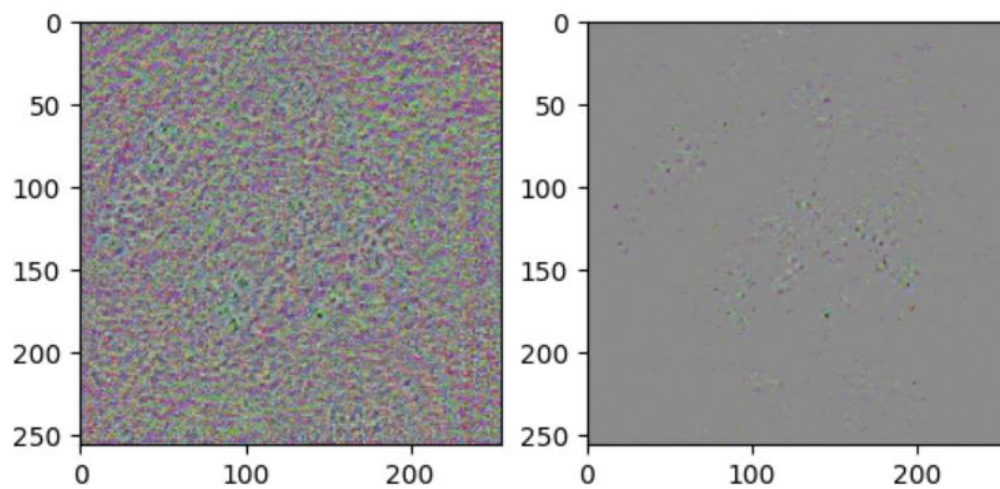
Label 8



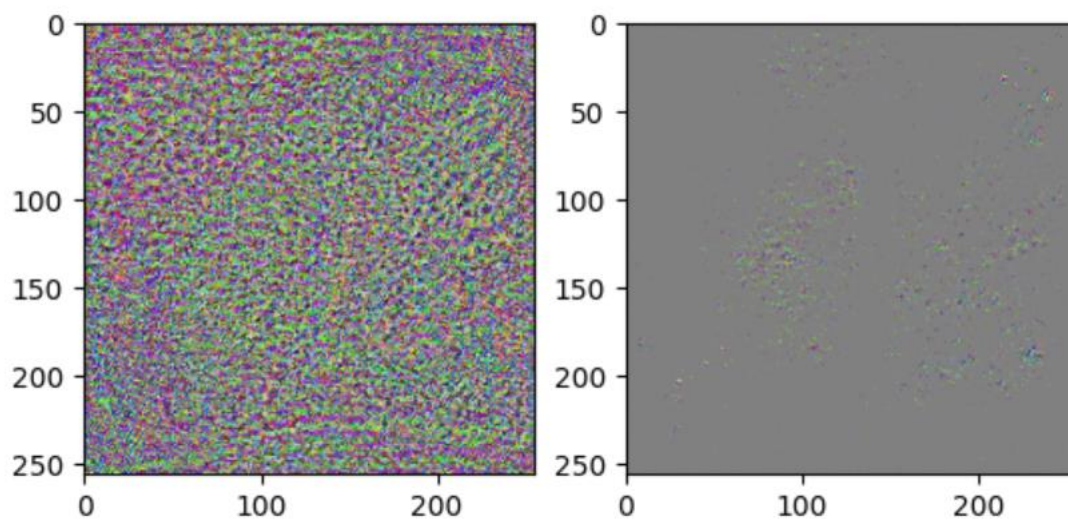
由於自己 model 在 label 6 表現最好，因此特別觀察 label 6 的圖片，發現 model 會除了部分透過麵本身去判斷之外，其還特別針對義大利麵盤子的形狀去判斷此類別，其中也發現 label 9 同樣也會根據碗的形狀去辨識，從 confusion matrix 同樣可以看出 label 9 的辨識成功率跟其他類別比起來相對較高，仔細想想，麵跟湯通常會用碗或是盤子來盛，因此 model 可能就是透過辨識出這些形狀來分類這兩個 label，跟其他食物比起來可能是個相對好的 feature。

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。

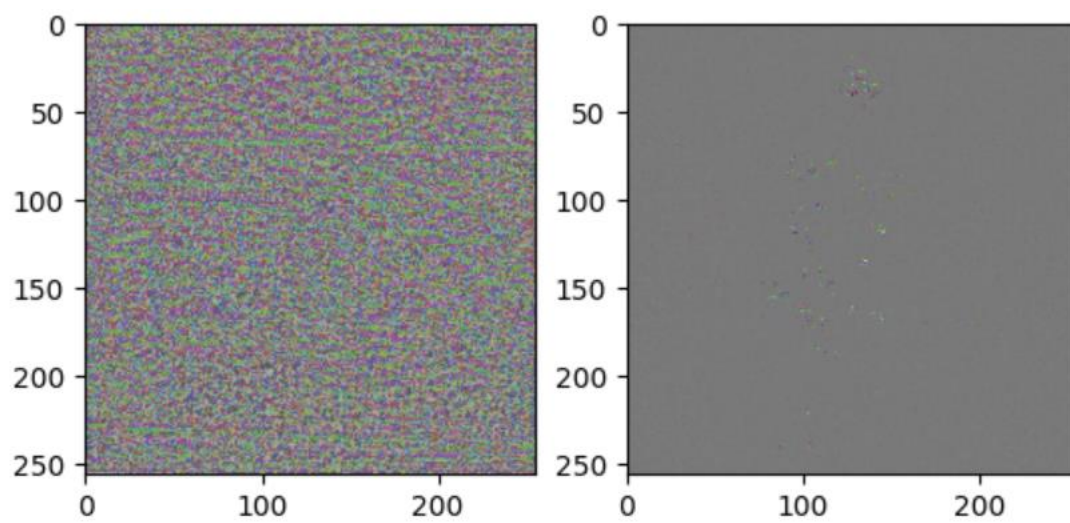
以下使用 gradient descent 的方式，透過調整 input image，看看甚麼樣的圖片可以使該 class 的機率最大，左圖是調整的 image，而右圖則是左圖經過 regularization 後的圖片。



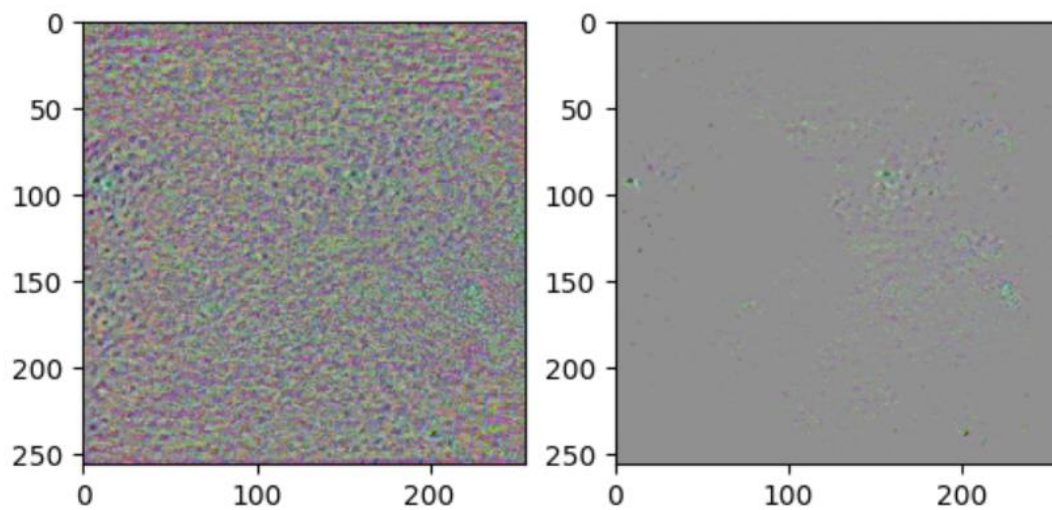
Label 0



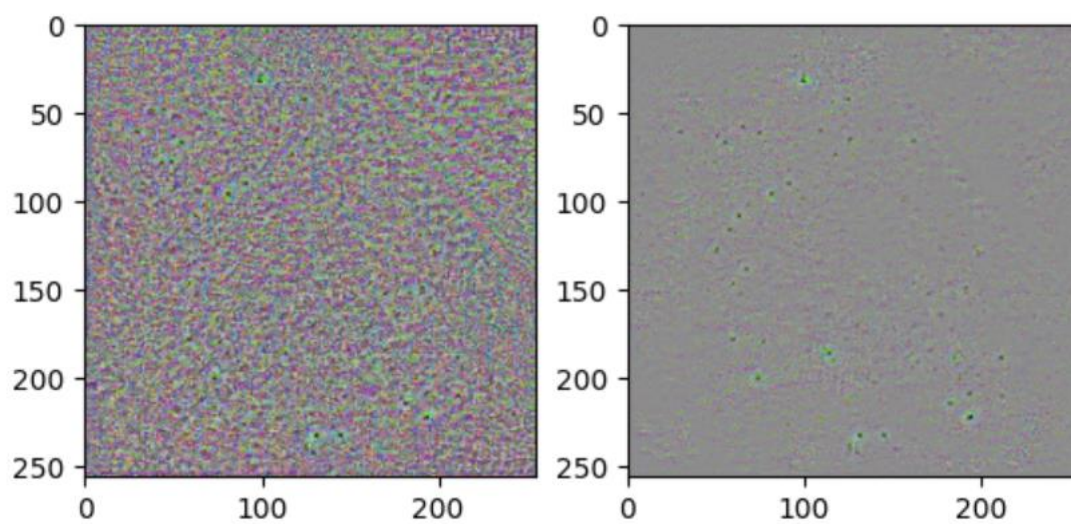
Label 1



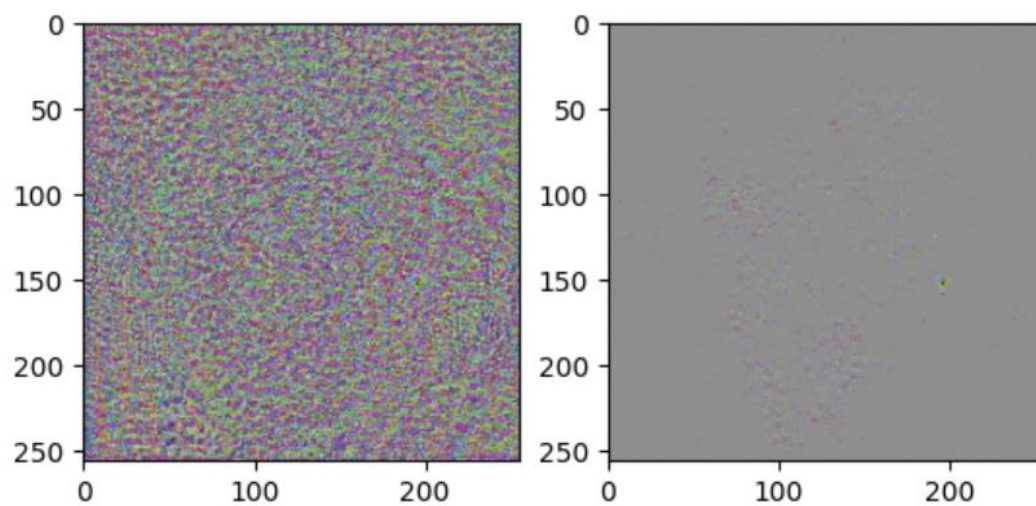
Label 2



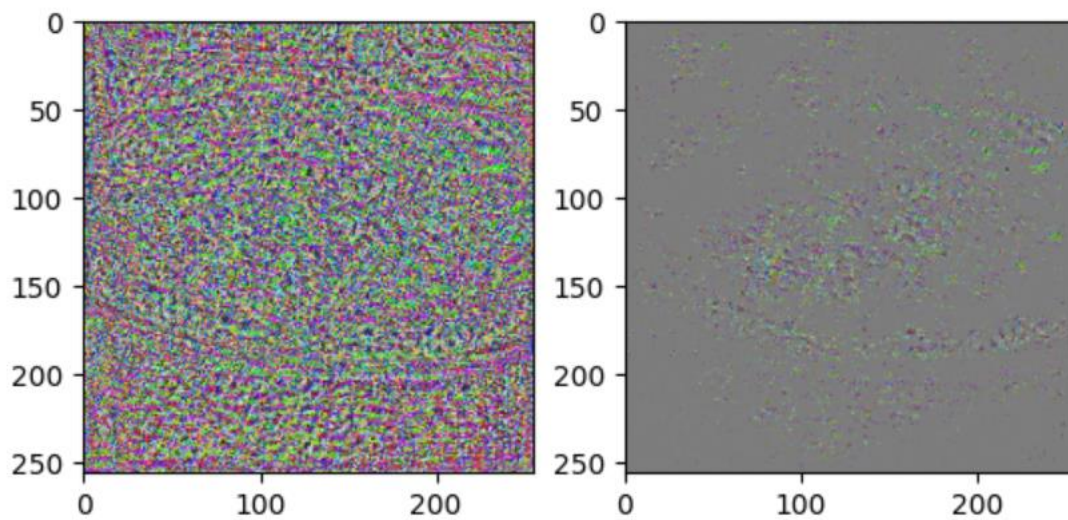
Label 3



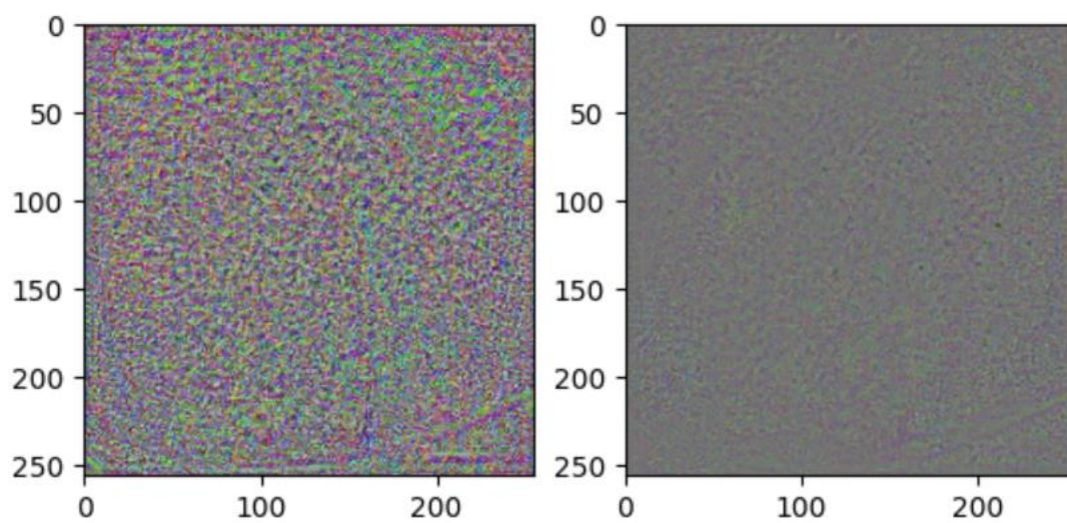
Label 4



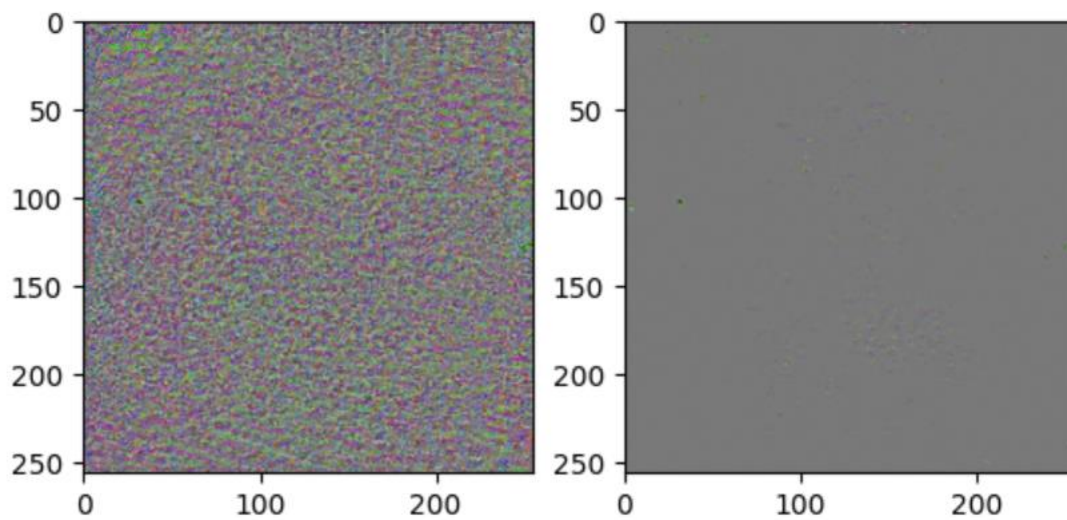
Label 5



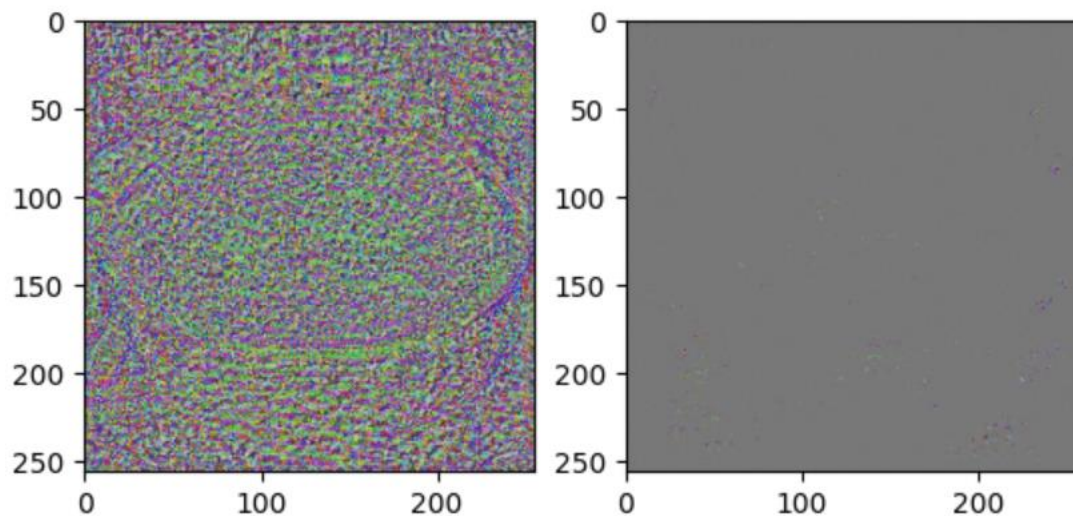
Label 6



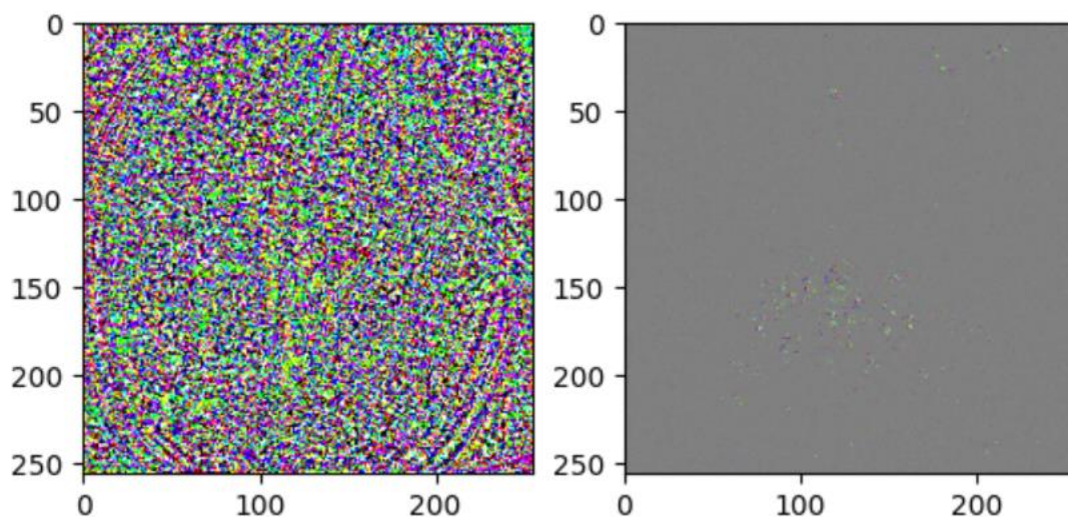
Label 7



Label 8



Label 9



Label 10

由上圖的結果來看，會發現如果不斷調整 Input image 去 fit 該 training label 的話，其實 model 最後學出來的大多都是一些看不懂的雜訊，如果說此 model 真的有學會分類的食物似乎有點牽強，但是這個現象也符合老師上課提過 mnist 數字辨識的例子，雖然 model 經過 regularization 後，大部分也很難看出端倪，但是藉由上一題提到過 label 6 會去認盤子的形狀，會發現 label 6 的圖似乎有類似橢圓的形狀，更能驗證 label 6 似乎正是以盤子的形狀做辨認。