Machine Learning HW5 Report

學號：R07943107 系級：電子所碩一 姓名：徐晨皓

1. **(1%) 試說明hw5\_best.sh攻擊的方法，包括使用的proxy model、方法、參數等。此方法和FGSM的差異為何？如何影響你的結果？請完整討論。(依內容完整度給分)**
2. **攻擊方法**
3. Proxy model：ResNet-50
4. 方法：

首先，hw5\_best.sh使用PyTorch pretrained model ResNet-50。在preprocessing部分，使用線上torchvision.models 說明文件中的方法。即先將data normalize至[0,1]，再用與

normalize一次。

hw5\_best.sh的攻擊方法是基於FGSM的方法再做改良。我們可以發現在FSGM中epsilon的大小與success rate有很大的關係，在助教提供的sample code裡只有對image加一次noise，這樣並不能保證產生出來的圖片能讓model錯誤辨識。因此我採用iterative的方法來加noise，若加一次noise無法成功就會進入下一個iteration，直到model錯誤辨識或達到最大iteration數才停止。為避免的大小讓圖片振盪無法收斂，的值會隨著iteration數量增加而減小，本次作業設定為，其中為第i次iteration的，為初始值。

1. 參數：
2. : 0.06
3. Iterations: 50
4. **與FGSM的差異**

主要的差異為，當當前的image無法讓model錯誤辨識時，hw5\_best.sh會對image加數次noise直到model錯誤辨識。這樣比起simple FGSM能有更多機會產生出成功錯誤辨識的圖片。

1. **如何影響結果**

從實驗結果中，使用simple FGSM僅有92.5%的success rate，L-inf. norm為6.0。但使用iterative FGSM能有100%的success rate，L-inf. norm 為4.17。因此我認為iterative的方法能更有效的產生出有效攻擊model的圖片。

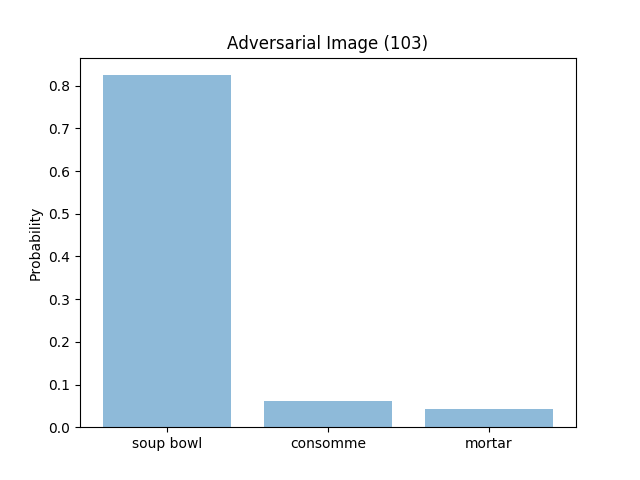
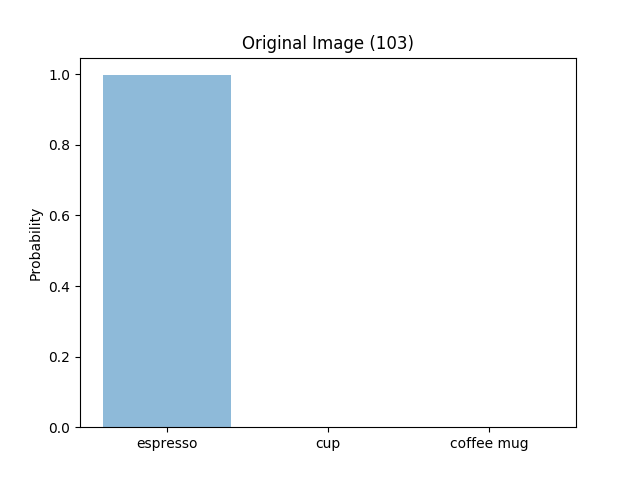
1. **(1%) 請列出hw5\_fgsm.sh和hw5\_best.sh 的結果 (使用的proxy model、success rate、L-inf. norm)。**
2. **hw5\_fgsm.sh**
3. Proxy model: ResNet-50
4. Success rate: 0.925
5. L-inf. norm: 6.0000
6. **hw5\_best.sh**
7. Proxy model: ResNet-50
8. Success rate: 1.000
9. L-inf. norm: 4.1700
10. **(1%) 請嘗試不同的proxy model，依照你的實作的結果來看，背後的black box最有可能為哪一個模型？請說明你的觀察和理由。**
11. 最有可能的black box：**ResNet-50**
12. 觀察與理由：
13. 從助教所提示的六種pretrained models辨識200張images的成功率，我們就可以猜到black box為何。下表為六種pretrained models 對於200 images的辨識成功率 (使用torchvision.models說明網頁中的preprocess方法)。ResNet-50有100%的正確率，因此在實作FGSM前基本上就已確認black box為ResNet-50。

|  |  |
| --- | --- |
| **Model** | **Classification Success Rate** |
| VGG-16 | 173/200 (86.5%) |
| VGG-19 | 174/200 (87.0%) |
| ResNet-50 | 200/200 (100.0%) |
| ResNet-101 | 186/200 (93.0%) |
| Densenet-121 | 185/200 (92.5%) |
| Densenet-169 | 183/200 (91.5%) |

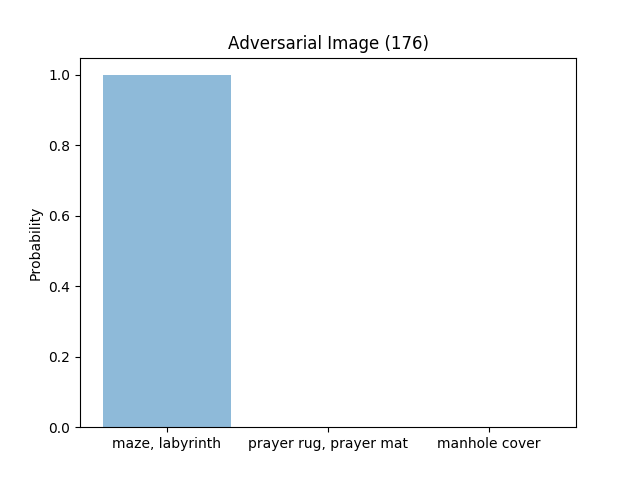
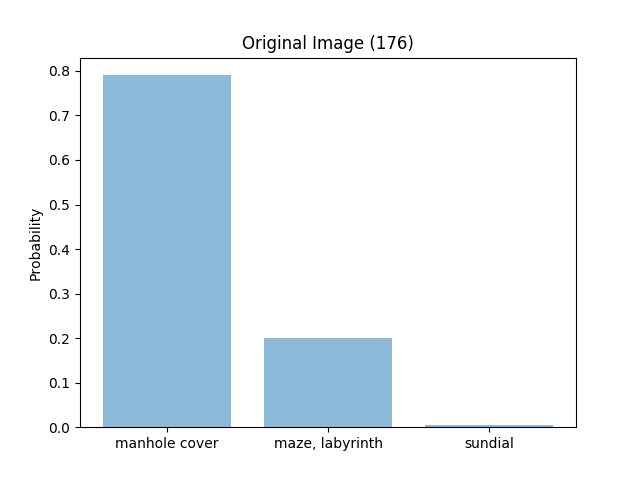
1. 從實作的FGSM中，當我們使用不同的proxy model時，攻擊成功率有顯著的差別。下表為六種pretrained model對於FGSM (使用hw5\_fgsm.sh)產生出的攻擊圖片的辨識失敗率。從表中，ResNet-50有最高的辨識失敗率92.5%，其餘五種model的辨識失敗率皆低於50%，因此ResNet-50最有可能是black box。

|  |  |
| --- | --- |
| **Model** | **Classification Failure Rate** |
| VGG-16 | 59/200 (29.5%) |
| VGG-19 | 55/200 (27.5%) |
| ResNet-50 | 185/200 (92.5%) |
| ResNet-101 | 97/200 (48.5%) |
| Densenet-121 | 78/200 (39.0%) |
| Densenet-169 | 77/200 (38.5%) |

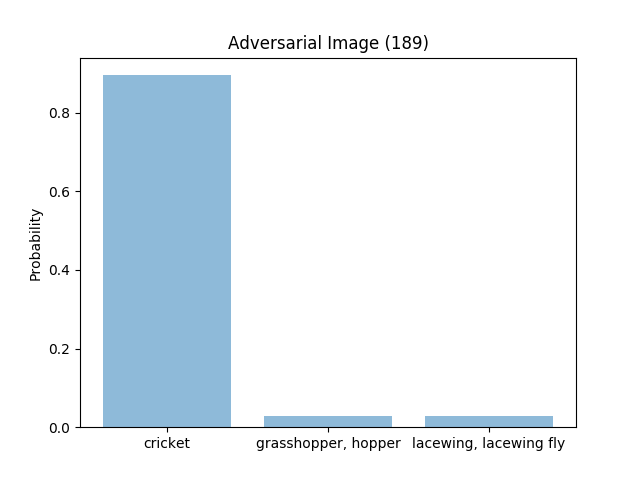
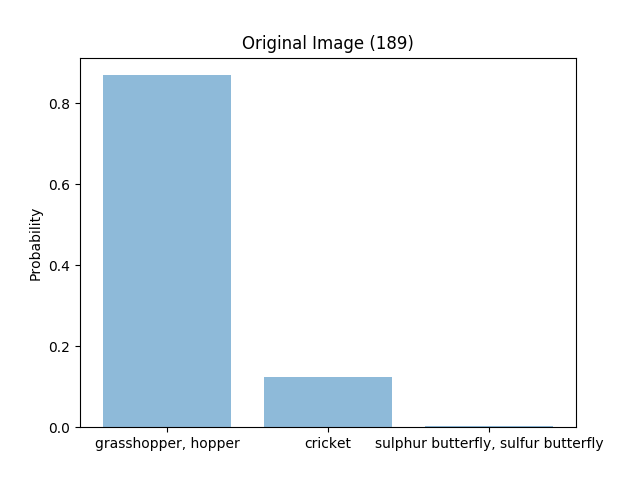
1. **(1%) 請以hw5\_best.sh的方法，visualize任意三張圖片攻擊前後的機率圖 (分別取前三高的機率)。**



103.png



176.png



189.png

1. **(1%) 請將你產生出來的 adversarial image，以任一種smoothing的方式實作被動防禦 (passive defense)，觀察是否有效降低模型的誤判的比例。請說明你的方法，附上你防禦前後的success rate，並簡要說明你的觀察。另外也請討論此防禦對原始圖片會有什麼影響。**

本題選用Gaussian filter作為smoothing的方式。使用scipy.ndimage中的gaussin\_filter實作，sigma設為1。

1. 在對由hw5\_best.sh產生出的adversarial images使用Gaussian smoothing後，實驗結果如下:

|  |  |  |
| --- | --- | --- |
|  | Before smoothing | After smoothing |
| Success rate | 100% | 64% |
| L-inf. norm | 4.17 | 116.05 |

原先成功率為100%的adversarial images，在使用Gaussian smoothing後，成功率降為64%。因此我認為以smoothing的方法並無法有效的降低model辨識成功率。

1. 選用Gaussian filter作為smoothing的方式。在對原始圖片使用Gaussian smoothing後，實驗結果如下:

|  |  |
| --- | --- |
| Success rate | 27.5% |
| L-inf. norm | 116.13 |

原始圖片在使用Gaussian smoothing後，success rate僅為27.5%。因此我認為將圖片模糊化或平滑化並無法很有效的成功攻擊一個model。要攻擊一個model較好的方法還是了解其結構，對其結構的弱點攻擊(如FGSM)，這樣才能有效的攻擊model。