Machine Learning HW5 Report

學號：B07901069 系級：電機一 姓名：劉奇聖

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 的差異爲何？如何影響你的結果？請完整討論。(依內容完整度給分)

答:

我使用的 proxy model 爲 pytorch(torchvision) pretrained 過的 resnet50，對於每張圖片我都先用 FGSM 攻擊一次，FGSM 的 epsilon 設爲 1，確保其 L-infinity norm 爲 1。若 FGSM 無法讓 model 辨識錯誤，則用 iterative 的 gradient attack 去攻擊。(FGSM 無法攻擊成功的圖片有 27 張)

Iterative gradient attack 方法如下: 首先將原始圖片複製一份當作要攻擊的圖片。最大的 iteration 次數設爲 40000，對於每次 iteration，將欲攻擊圖片通過 model，計算 loss(crossentropy loss)對於欲攻擊圖片的 gradient，將此 gradient 乘上 learning rate(設爲 1)後加到欲攻擊圖片上，之後將欲攻擊圖片與原始圖片間所有差距大於 1 的 pixel 強制拉回 1，確保其 L-infinity norm 爲 1(因爲圖片有經過預處理，即除以 255 使值變爲 0 到 1 之間後再對整個 imagenet 的資料做標準化，因此這裡的差距大於 1 指的是反處理回去之後差距大於 1)。接著將此修改過的圖片再通過 model 一次，並通過 softmax 函數，若能讓 model 辨識錯誤且使其辨識原標籤的機率降到 0.1 以下則跳出迴圈(因此 iteration 不會到 40000，大部份圖片在 1000 個 iteration 內即跳出迴圈)。加上使其辨識機率降到 0.1 以下這個條件是因爲反處理回去時要把小數點變成整數，勢必有些值的精度會下降，導致攻擊的效果消失，可能使 model 再次辨識正確，例如可能辨識原標籤的機率降到了 0.45，辨識爲另一錯誤標籤的機率升爲 0.48，若此時就跳出迴圈，將圖片反處理回去，可能使辨識爲原標籤的機率上升成 0.51，model 將會再次辨識正確。

經過了上面的攻擊後仍有 7 張圖片無法攻擊成功，對此我仍然使用上面提到的 iterative gradient attack 做攻擊，但對每張圖片都給一個客製化的 learning rate，而非上面的統一設爲 1，如此一來只剩 1 張圖片無法攻擊成功。

最後無法攻擊成功的圖片爲 121.png，我只好把對此圖片可容忍的 L-infinity norm 提升爲 2，此外對這張圖片我採取的是 iterative gradient sign attack，此方法和上面的 iterative gradient attack 的唯一差別即不是將 gradient 直接乘上 learning rate 後加到欲攻擊圖片上，而是將 gradient 通過 sign 函數，僅取正負值，乘上 learning rate(設爲 0.004)再加到圖片上。

我使用的 iterative gradient attack 和 FGSM 的差別是 FGSM 只計算一次 gradient 即產生出攻擊的圖片，雖然快速但有時無法找出最佳解。使用 iterative 的方法可以漸漸逼近最佳解，比較不會發生跳過最佳解的情況，因此攻擊的成功率便會較 FGSM 高。

2. (1%) 請列出 hw5_fgsm.sh 和 hw5_best.sh 的結果 (使用的 proxy model、success rate、L-inf. norm)。

答: 使用的 proxy model 皆爲 pytorch pretrained 過的 resnet50。

	hw5_fgsm.sh	hw5_best.sh
success rate	0.925	1.000
L-inf. norm	5.0000	1.0050

3. (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 最有可能爲哪一個模型？請說明你的觀察和理由。

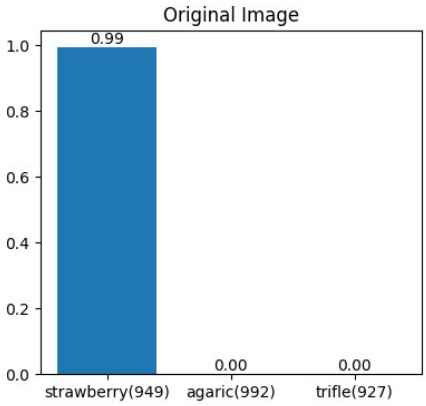
答:

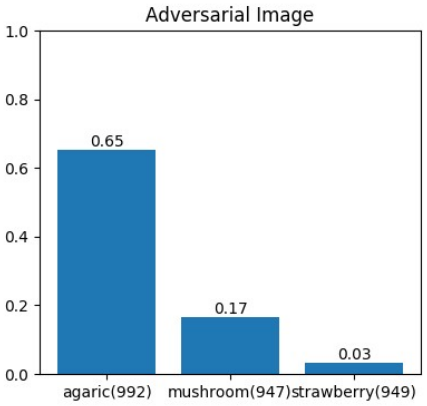
	vgg16	vgg19	resnet50	resnet101	densenet121	densenet169
同 model	0.985	0.975	0.925	0.935	0.980	0.905
上傳	0.330	0.335	0.925	0.475	0.385	0.395

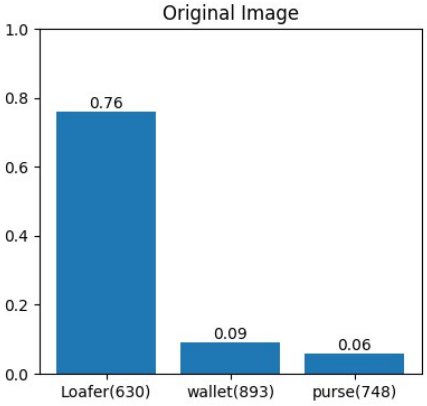
上表爲各個 model 分別使用 FGSM，epsilon 設爲 5 攻擊的結果。「同 model」列代表的是攻擊後的圖片對於 attack 時使用的同一個 proxy model 的 success rate，「上傳」列代表的是上傳 black box 的 success rate。由此表可看出各個 proxy model 對於同 model 的 success rate 都相當高，代表個別的攻擊都有成功，但上傳到 black box 後 resnet50 的 success rate 最高，其餘 model 的 success rate 皆大幅下降，故我推測背後的 black box 爲 resnet50。

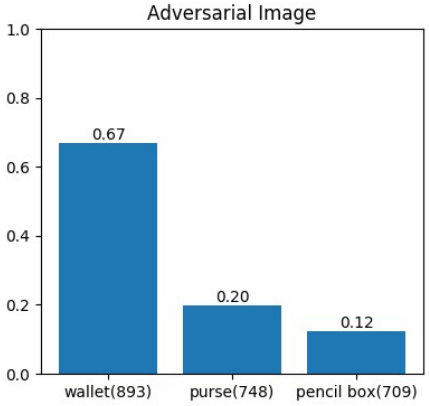
4. (1%) 請以 hw5_best.sh 的方法，visualize 任意三張圖片攻擊前後的機率圖 (分別取前三高的機率)。

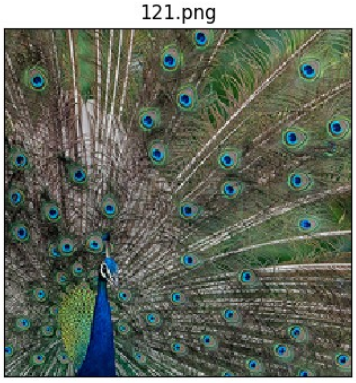
答:

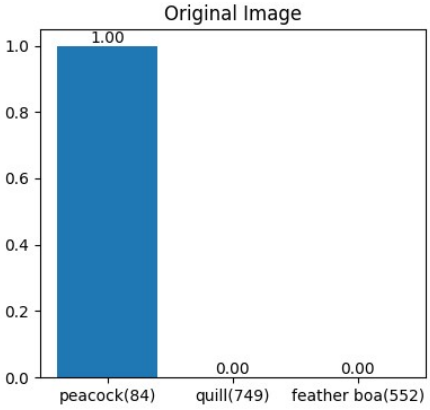


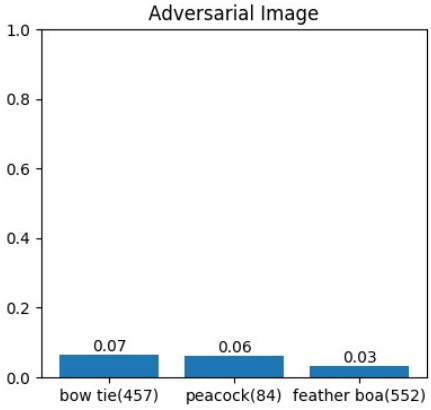












5. (1%) 請將你產生出來的 adversarial img，以任一種 smoothing 的方式實作被動防禦 (passive defense)，觀察是否有效降低模型的誤判的比例。請說明你的方法，附上你防禦前後的 success rate，並簡要說明你的觀察。另外也請討論此防禦對原始圖片會有什麼影響。

答:

	防禦前	防禦後
Success rate	1.000	0.425

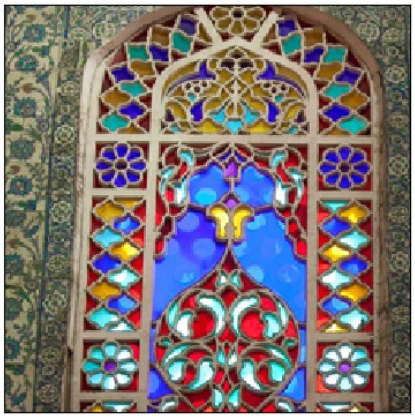
上表是我使用 Gaussian filter，sigma 設爲 1 作爲防禦方法，防禦前後的 success rate。可以發現在使用了此防禦方法後攻擊的成功率下降至低於一半，有效地減低了模型誤判的比例。

使用了 Gaussian filter 之後圖片會變模糊、鮮豔程度下降，看起來就像是有一層半透明的灰色霧狀遮罩蓋在上面。以下是幾張用了 Gaussian filter 前後的圖片。

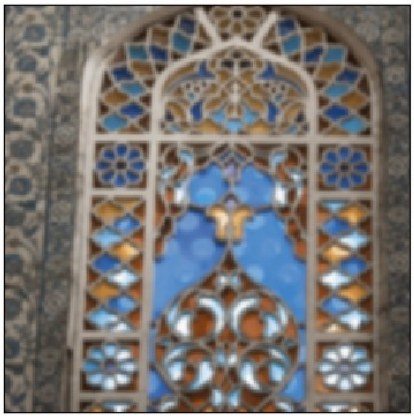
before defense

after defense

before defense



after defense



before defense

after defense