

1. (2%) 試說明 hw6_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)

使用 DenseNet-121作為Proxy model，採用Iterative FGSM方法，其中 $\epsilon = 0.035$ ，具體方法是先做判斷，如果本身錯誤就跳過攻擊，正確則進行無窮迴圈：每做一次 $\epsilon=0.035$ 的FGSM即重新做classification直到與label不相同即跳出迴圈並做下一張圖片。而FGSM方法 $\epsilon 0.2$ ，只做一次。

由於只能做一次，FGSM方法的 ϵ 需要較大來使Attack Success Rate提高，因此會犧牲L-inf，兩者需要取得平衡無法同時達成；Iterative FGSM可以重複多次，因此設小一點的 ϵ 可以使L-inf壓低，同時因為每張圖片的Iteration次數不同，可以對每張圖片個別處理，因此不會有無法取得平衡的問題，達成高成功率低L-inf的結果。

<input checked="" type="checkbox"/>	9	2020-04-27 15:49:06	1.000	3.0900
<input type="checkbox"/>	10	2020-04-29 20:03:59	0.910	11.1750

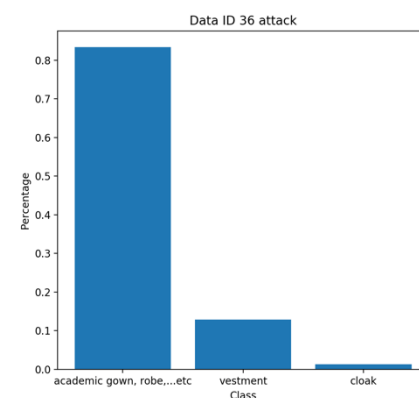
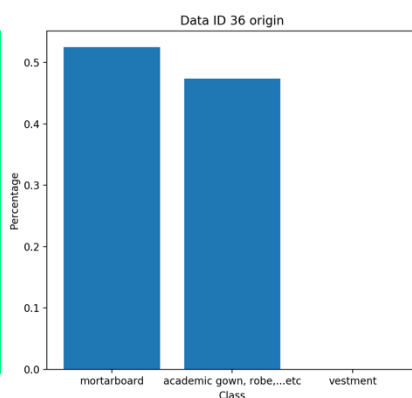
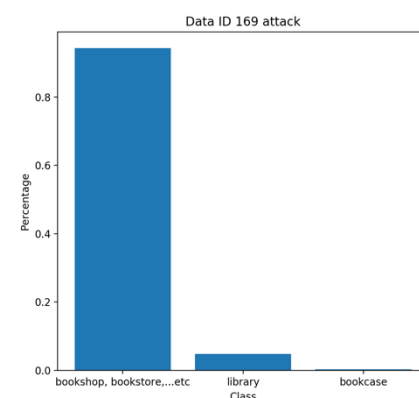
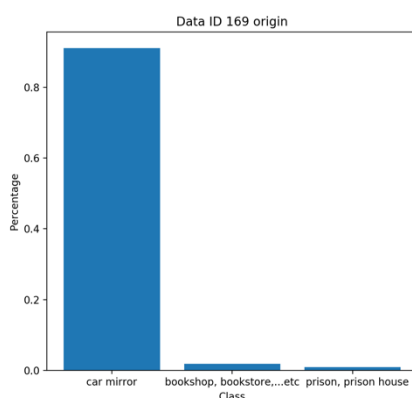
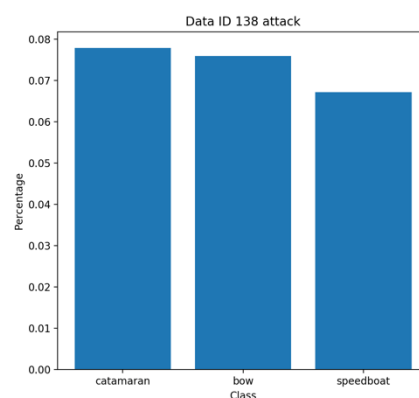
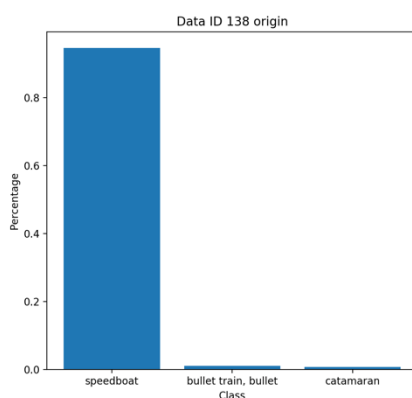
編號9為Iterative FGSM，10為FGSM，可見L-inf與成功率如上所述

2. (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。

Proxy model	Attack Success Rate	Submit Success Rate
VGG-16	99.5%	11.0%
VGG-19	100%	13.0%
ResNet-50	100%	25.5%
ResNet-101	100%	21.5%
DenseNet-121	100%	100%
DenseNet-169	100%	25.5%


DenseNet-121，由上圖顯示，六種models attack完後重新判斷正確率都接近或等於100，然而最後結果卻只有DenseNet-121維持100，其餘皆低於33.3%，故猜測black box 為 DenseNet-121。

3. (1%) 請以 hw6_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)。



通常將最正確的Attack完後，如果第二機率的相較第三機率的比列大很多，很有可能在Attack完後出現第二正確的機率很高的情況。

4. (2%) 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。

使用的方式做smoothing，下圖為比較，左方為原圖，右方為smoothing結果。



```
chinyi0523@SpeechLab531:~/hw6-chinyi0523$ python3 hw6_report.py ./data ./test -test
Start Testing...
-----
Attack Success: 100.0% [200/200]
chinyi0523@SpeechLab531:~/hw6-chinyi0523$ python3 hw6_report.py ./data ./test -test
Start Testing...
-----
Attack Success: 54.5% [109/200]
```

防禦後Attack的成功率剩下54.5%，防禦確實有效果，成功率砍半。

由smoothing前後可以發現，smooth後圖片模糊很多，顏色飽和度也降低，可能因此將attack的雜訊功能降低。