Machine Learning HW5 Report

學號：R07922134　　系級：資工碩一　　姓名：陳紘豪

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)

Proxy model：Resnet 50

方法：Basic iterative method（https://arxiv.org/abs/1607.02533）

$$\boldsymbol{X}_0^{adv} = \boldsymbol{X}, \quad \boldsymbol{X}_{N+1}^{adv} = Clip_{X,\epsilon}\left\{\boldsymbol{X}_N^{adv} + \alpha\,\mathrm{sign}\left(\nabla_X J(\boldsymbol{X}_N^{adv}, y_{true})\right)\right\}$$

- Number of iteration：4
- alpha：1
- epsilon：0.01

說明：

Basic iterative method 是 FGSM 的延伸作法，不像 FGSM 是一次就能得到可以拿來攻擊的 image，Basic iterative methond 是經過多次迭代得到結果，而這個迭代次數可以自己選擇，paper 作者是建議次數=min(eps + 4, 1.25*eps)。

基本作法就是先像 FGSM 那樣得到一張 adversarial image，這邊的參數 alpha 就是 FGSM 的參數 epsilon，再來拿這張 adversarial image 去和 original image（最一剛開始的，完全沒有雜訊的 image）去做 pixel-wise clipping，使得 adversarial image 的 pixel 值只能落在 original image pixel 值的 epsilon-neiborhood 上。（舉個例子：若 epsilon 取 0.01，則 adversarial image 和 original image 相減只能落在 [-0.01, 0.01] 上，小於 -0.01 就設成 -0.01，大於 0.01 就設成 0.01。）

做完 clipping 後等於做完一次的迭代，把做完 clipping 的 image 拿來再重新跑一次剛剛的流程，最後所有迭代都結束後所得到的圖片即為可以拿來攻擊的 image。

結果：Success rate 上升，而且 L-inf norm下降。

探討：

alpha 的效果就是和 FGSM 的 epsilon 一樣，提供一個 step size 告訴程式該讓 image pixel 值變動多少，alpha 設 1 就是讓新的 image 和舊的 image 之間的每個 pixel 值只能相差 1 ，epsilon 就是為了讓你隨著迭代產生新的 image 時離最原本的 image 不能相差太大，這個參數是 FGSM 所沒有的，有了 epsilon ，即使是 iterative method，最終的結果也不會離最原本的 image 太遠，所以自然而然可以在讓 Success rate 上升（iteration、alpha）的同時，也讓 L-inf norm（epsilon） 下降。

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)。

	Proxy model	Success rate	L-inf. norm
hw5_fgsm.sh	Resnet 50	0.925	6
hw5_best.sh	Resnet 50	0.995	2

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 最有可能為哪一個模型？請說明你的觀察和理由。

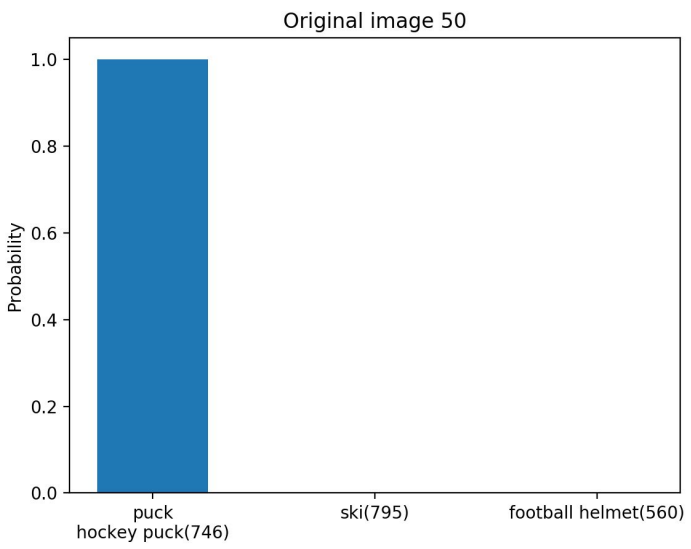
使用不同的 Proxy model 去實作 FGSM，且彼此之間都採用相同的參數，並去觀察其 Success rate 和 L-inf. norm，如下：

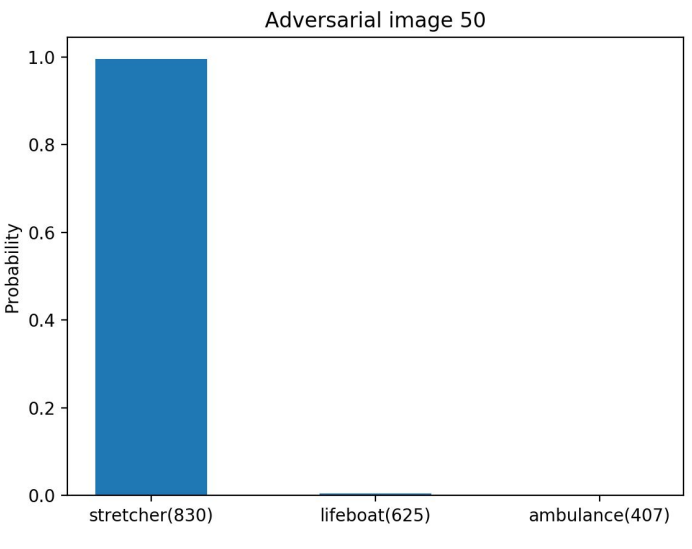
Proxy model	Success rate	L-inf. norm
Vgg 16	0.360	6
Vgg 19	0.355	6
Resnet 50	0.925	6
Resnet 101	0.515	6
Densenet 121	0.415	6
Densenet 169	0.425	6

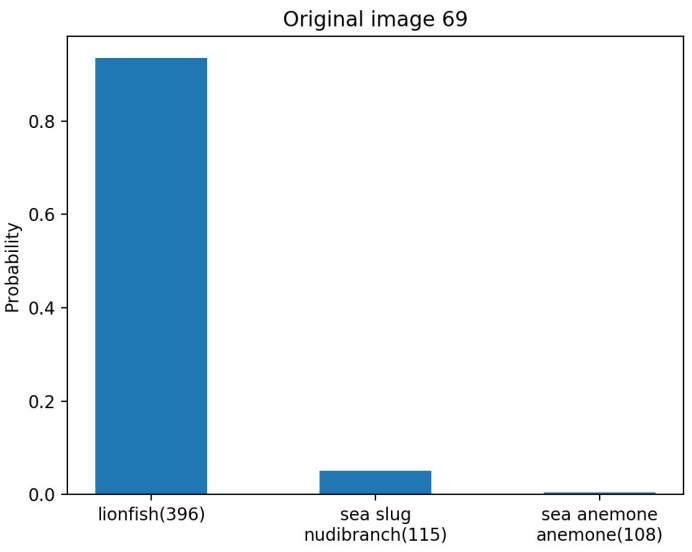
由觀察看到，其中 Resnet 50 所產生出來的 images 的 Success rate 是最高的，而且遠遠大於其他所有的 Proxy model，因此可以推定助教所用的 black box 是 Resnet 50。

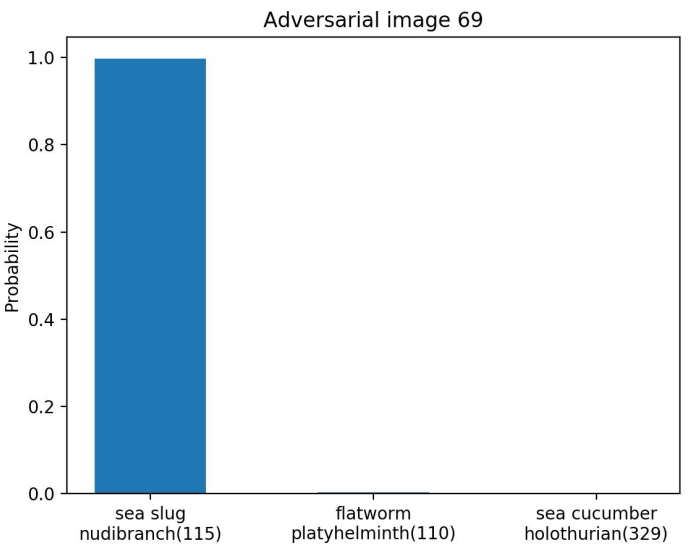
4. (1%) 請以 hw5_best.sh 的方法，visualize 任意三張圖片攻擊前後的機率圖 (分別取前三高的機率)。

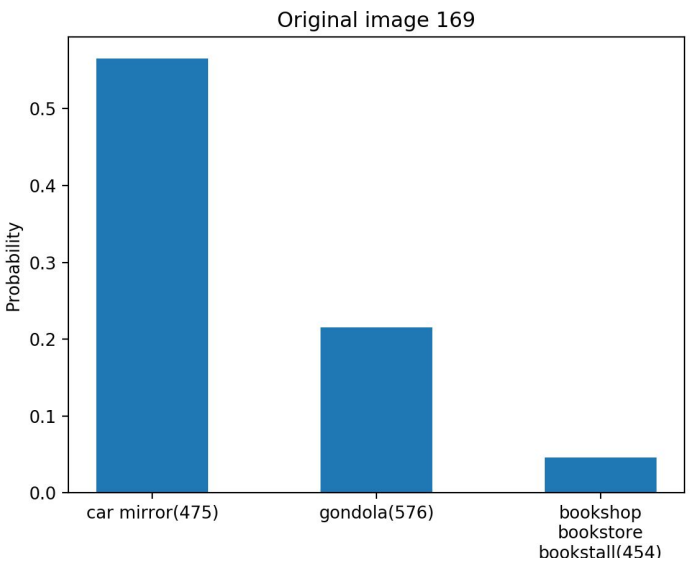
圖片	攻擊前	攻擊後

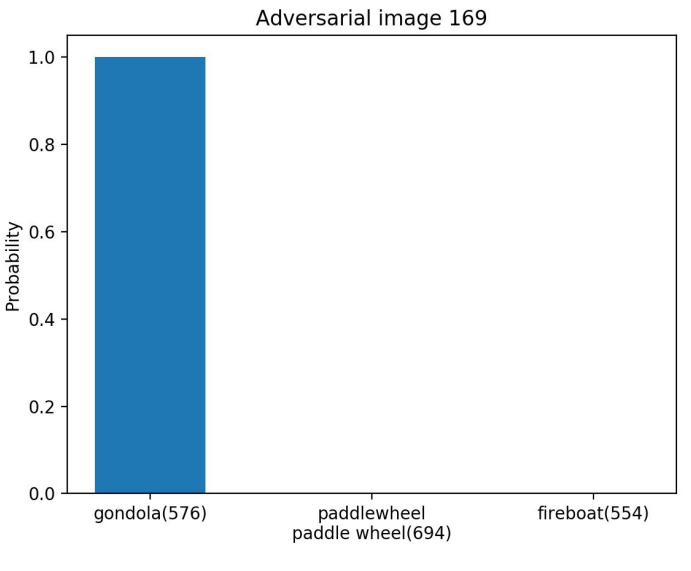












5. (1%) 請將你產生出來的 adversarial img，以任一種 smoothing 的方式實作被動防禦 (passive defense)，觀察是否有效降低模型的誤判的比例。請說明你的方法，附上你攻擊有無的 success rate，並簡要說明你的觀察。

方法：Gaussian filter

對象：hw5_best.sh 中產生出的 adversarial image

Sigma	Kernel size	Success rate	L-inf. norm
0	0	0.995	2
1	3	0.740	94.3900
1	5	0.530	107.4750
2	3	0.625	105.8300
2	5	0.370	136.1950

本次 Smoothing 方法採用 Gaussian filter 來實作，觀察在不同的 Sigma 值和 Kernel 大小的情況下對 Success rate/L-inf. norm 的影響，在完全沒有實作被動防禦的情況下攻擊成功的機率為 0.995，L-inf. norm 也只有 2，但隨著越來越大的 Sigma 值和 Kenel 大小的情況下，攻擊的成功機率越來越低，而且 L-inf. norm 也越來越大，代表要成功防禦攻擊所換取來的代價就是圖片會變得跟原圖比較不同（但肉眼還是察覺不出，只能觀察到有點模糊），若要採取被動防禦而且想要防禦成功機率高一點的話，使用 Gaussia filter 並採用較大的 Sigma 值和 Kernel size 應該能有不錯的效果。

對象：Original image

Sigma	Kernel size	Success rate	L-inf. norm
0	0	0	0
1	3	0.120	94.1800
1	5	0.150	107.2800
2	3	0.135	105.7000
2	5	0.190	136.1000

另外一個實驗的對象是原始圖片，觀察 Gaussian filter 對它的影響，由結果發現，雖然 filter 會讓新的圖片和舊的圖片之間具有很大的 L-inf. norm，但是對於 model 對圖片的辨識率卻沒有太大的影響，產生僅僅不到 1% 的錯誤率而已，隨著 Sigma 和 Kernel size 的增加，錯誤率也會些微的上升。

結論：對於不明來源的 data，不知道裡面是否含有具有攻擊性的圖片時，可以試著對所有 data 去做 Gaussian filter smoothing，此作法對於正常圖片只有些微的提高錯誤率（不到1%），但是對於 adversarial image 可以大大的降低它的錯誤率，整體來說是利大於弊。