Machine Learning HW5 Report

學號：R07922004　系級：資工碩一　　姓名：吳星耀

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)

   我使用pretrained過的ResNet-50作為proxy model，攻擊的方法為Basic Iterative Method，參數為N=3，alpha=1/255，也就是更新三次，每次改變每個pixel一個單位的大小(0~255中的1個單位)。

   我認為這個方法其實就相當於多次對一個圖片做FGSM，因此與FGSM最大的不同就是他會同一張圖片更新很多次，另外此方法參數alpha與FGSM的epsilon的物理意義相當，唯一差別就是alpha在作者論文中有提到說他們在一次更新的量值(alpha)預設為一個單位(0~255中的1個單位)，epsilon則是沒有特別對應物理單位的直接調。因此我在FGSM中是直接對normalize後的圖片直接以epsilon乘上梯度正負值更新後，再轉回沒有normalize的圖片，而在Basic Iterative Method則是每次更新都先把normalized的圖片denormalize成原本圖片再更新。

   如此影響的結果為：由於Basic Iterative Method一次都更新一個單位大小，因此每張攻擊後的圖片L-infinity norm一定會是N，也就是更新的次數，而FGSM攻擊後的圖片L-infinity norm則不一定一致。

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)。

   下表為兩支程式的執行結果：

	proxy model	success rate	L-inf norm
hw5_fgsm.sh	ResNet-50	0.905	2.000
hw5_best.sh	ResNet-50	0.995	3.000

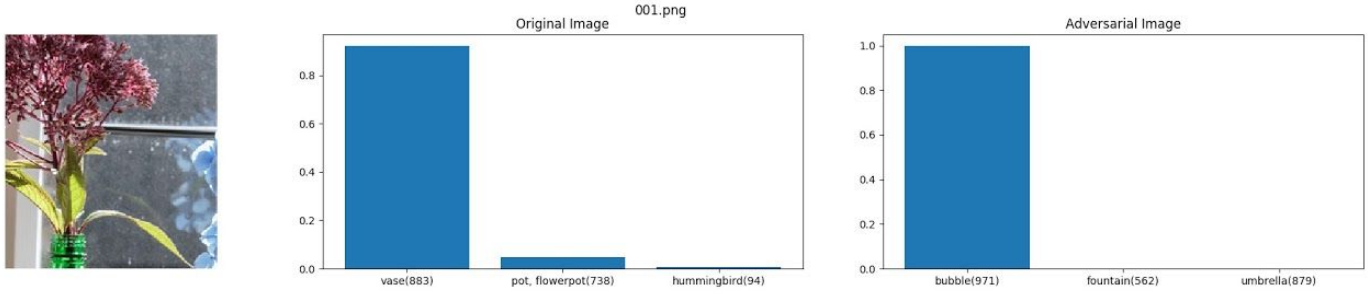
3. (1%) 請嘗試不同的 proxy model，依照你的實作的結果來看，背後的 black box 最有可能為哪一個模型？請說明你的觀察和理由。

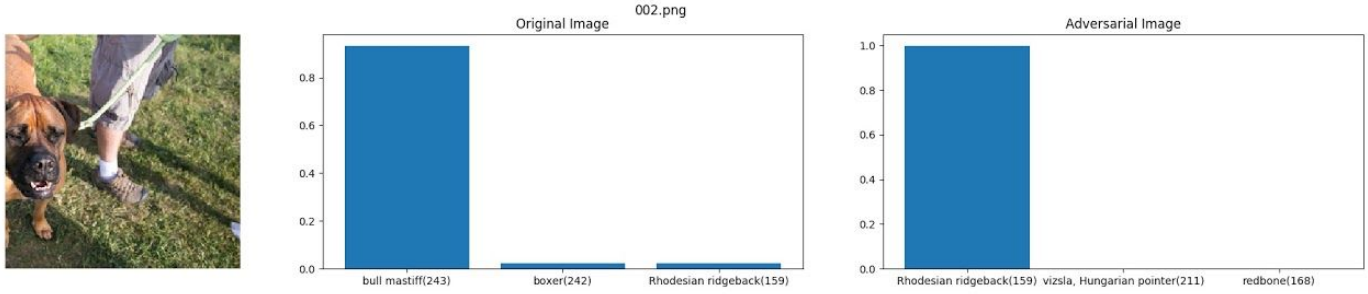
   我用epsilon = 0.015 的 FGSM (hw5_fgsm.sh是用epsilon = 0.03)測試不同proxy model的影響。以下為結果：

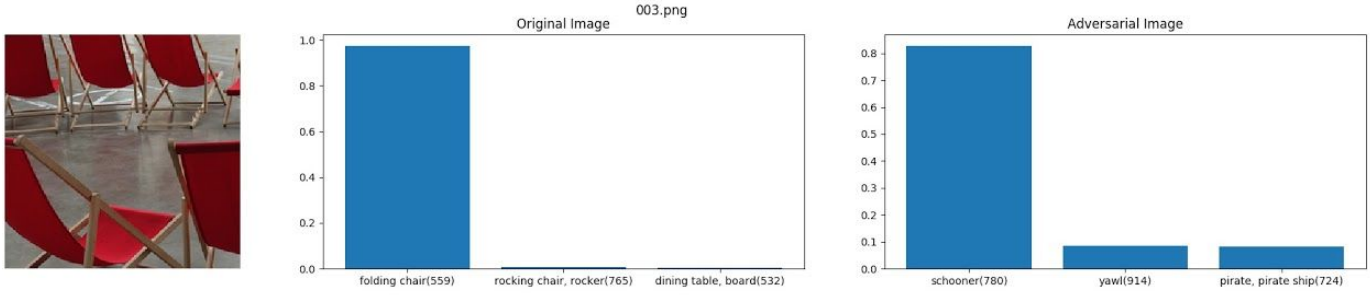
Proxy model	VGG-16	VGG-19	ResNet-50	ResNet-101	DenseNet-101	DenseNet-169
success rate	0.1	0.11	0.865	0.175	0.135	0.13
L-inf norm	1.000	1.000	1.000	1.000	1.000	1.000

可以很明顯的看出除了ResNet-50之外，其餘的模型成功率皆不佳，故可以判斷black box模型應該是ResNet-50。由於我在使用pretrained的ResNet-50時success rate為0.855，但是black box模型卻為0.865，與pretrained結果不一致，因此可以猜測助教的模型可能為ResNet-50架構，但是參數有再額外train過使結果不完全一致。

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)。







上三張圖是三張圖片在attack前後的模型預測機率。在第一張照片中，未被攻擊前模型本來預測前三大機率的label為vase(883), pot,flowerpot(738), hummingbird(94)，攻擊後的圖則為bubble(971), fountain(562), umbrella(879)；在第二張照片中，未被攻擊前模型本來預測前三大機率的label為bull mastiff(243), boxer(242), Rhodesian ridgeback(159)，攻擊後的圖則為 Rhodesian ridgeback(159), vizsia, Hungarian pointer(211), redbone(168)；在第三張照片中，未被攻擊前模型本來預測前三大機率的label為folding chair(559), rocker chair, rocker(765), dining table, board(532)，攻擊後的圖則為 schooner(780), yawl(211), pirate, pirate ship(724)。

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。

   我用Gaussian Filter來smoothing hw5_best.sh所產生的圖片，以下為不同sigma的成功率與L-inf norm：

sigma	無filter	0.4	0.8	1	1.2	1.6	2	2.4
success rate	0.995	0.995	0.85	0.650	0.485	0.42	0.46	0.535
L-inf norm	3	18.445	90.93	108.43	121.02	137.57	147.83	154.52

   由表格數據可以看出：在sigma=1.6的時候success rate降低得最多，證明smoothing可以一定程度的防禦攻擊。在sigma>1.6後，sigma的提昇反而會造成success rate提高。這可能是因為當圖片被smooth到一定的程度之後，因為模糊的程度已經大到讓原本的model都無法判斷圖片所造成的。

   另外，smoothing也會造成L-infinity norm上升，這是因為圖片模糊化造成與原圖差異變大造成。因此sigma越大，圖片模糊化越嚴重，L-infinity norm也越大。