Machine Learning HW5 Report

學號：B04505028 系級：工科四 姓名：林秀銓

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

   方法與差異 : 我所使用的方法同樣是FGSM，但是在hw5_fgsm.sh中我是用keras來實作，而在hw5_best.sh中我是用PyTorch來實作，兩者所用的proxy model也不相同，hw5_fgsm.sh是用VGG19，hw5_best.sh是用ResNet-50，且所有的圖片都有先經過normalization再進行運算，結果success rate 和 L-inf皆有明顯的改進，另外，為了提高攻擊的成功率，每張圖片在經過一次攻擊後都會再用proxy mdel預測一次，若結果與原圖相同則表示攻擊失敗，則對圖片再進行一次攻擊，結果success rate由0.925小幅上升至0.94，但同時L-inf也由5上升.至5375。

   參數 : epsilon = 0.08

   結果 : hw5_fgsm.sh的成績為 success rate : 0.495, L-inf : 17.325

   hw5_best.sh的成績為 success rate : 0.94, L-inf : 5.375

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)。

   hw5_fgsm.sh :

   proxy model : VGG19, success rate : 0.495, L-inf : 17.325

   hw5_best.sh :

   proxymodel : ResNet-50, success rate : 0.94, L-inf : 5.375

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。

   proxy model : ResNet-50, success rate : 0.94, L-inf : 5.375

   proxy model : VGG16, success rate : 0.31, L-inf : 5.075

proxy model : VGG19, success rate : 0.315, L-inf : 5.125

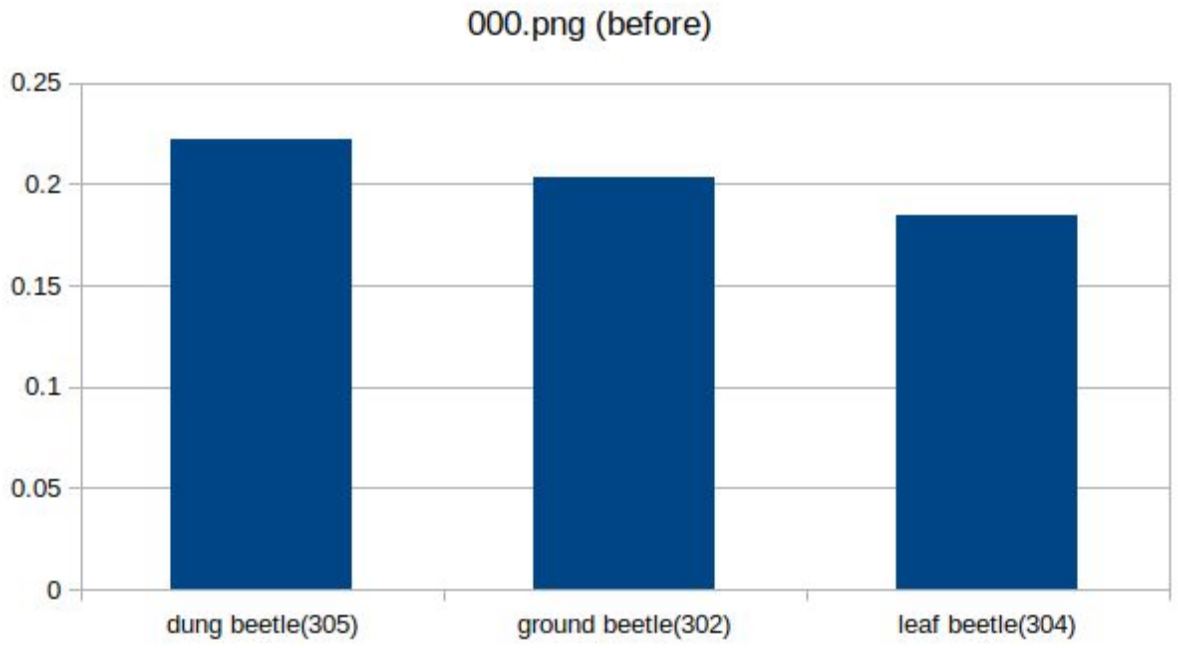
proxy model : ResNet-101, success rate : 0.455, L-inf : 5.325

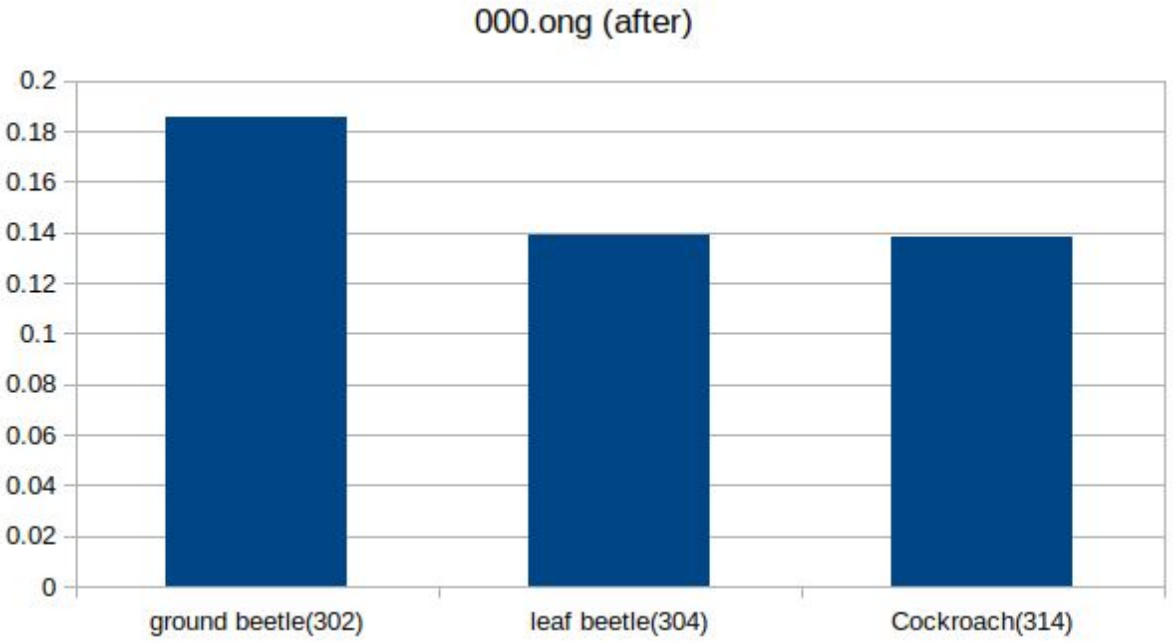
proxy model : DenseNet-121, success rate : 0.365, L-inf : 5.1

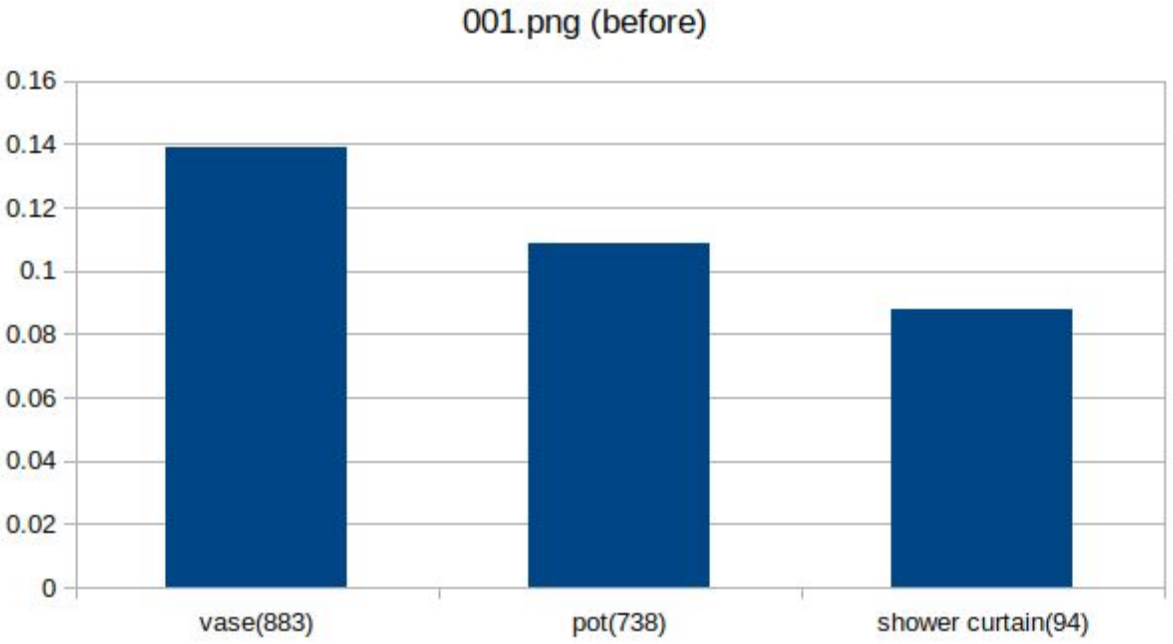
proxy model : DenseNet-169, success rate : 0.4, L-inf : 5.45

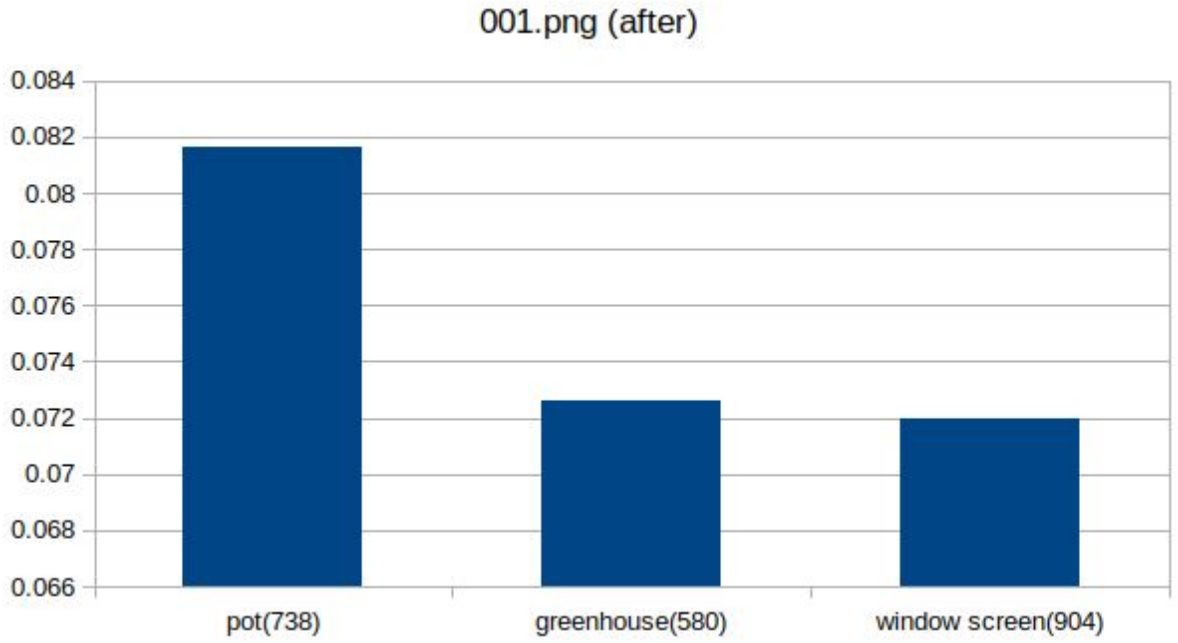
觀察結果可以發現ResNe-50實作結果比其他model都好非常多，所以我推測black box的model最有可能為ResNet-50。

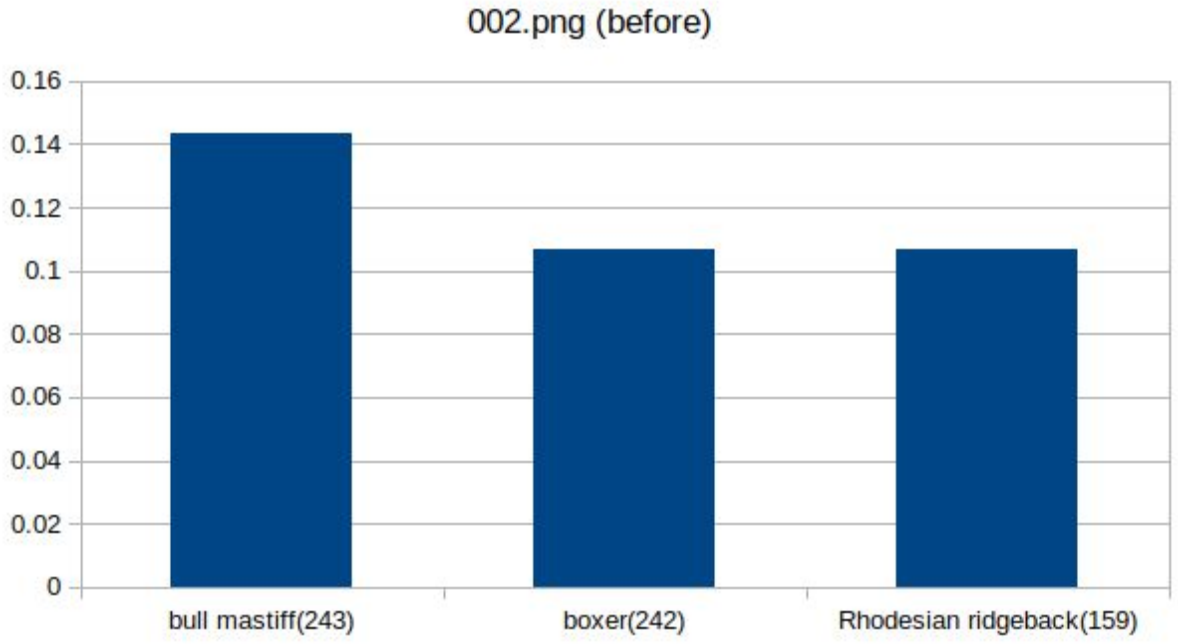
4. (1%) 請以 hw5_best.sh 的方法，visualize 任意三張圖片攻擊前後的機率圖 (分別取前三高的機率)。

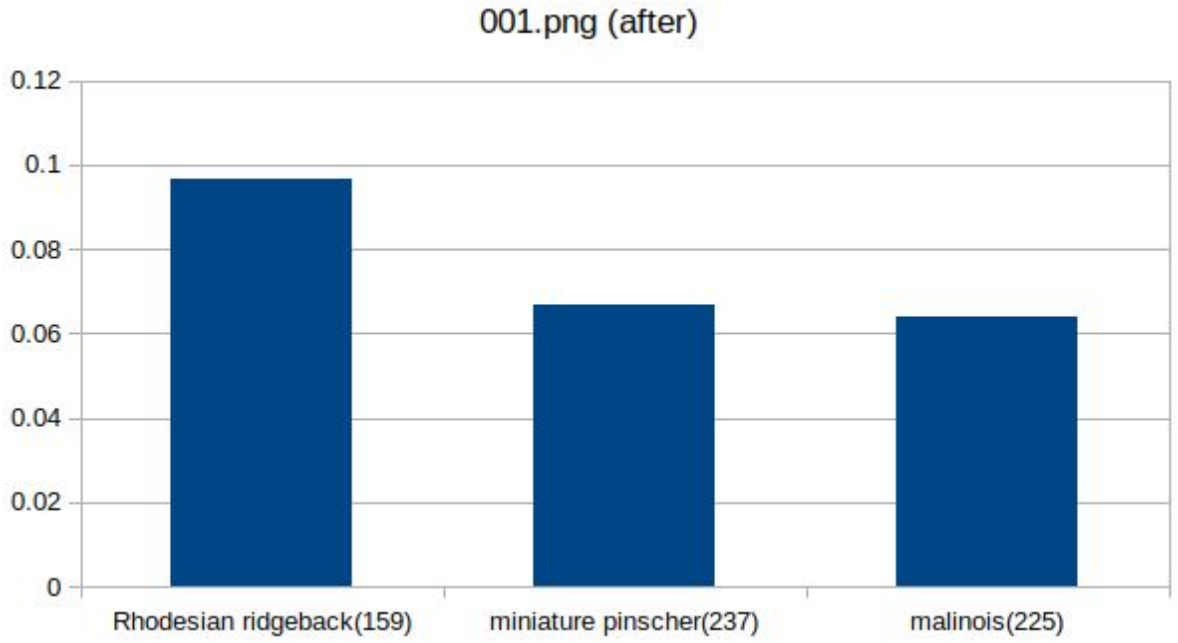



000.ong (after)
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
ground beetle(302)
leaf beetle(304)
Cockroach(314)


001.png (before)
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
vase(883)
pot(738)
shower curtain(94)


001.png (after)
0.084
0.082
0.08
0.078
0.076
0.074
0.072
0.07
0.068
0.066
pot(738)
greenhouse(580)
window screen(904)


002.png (before)
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
bull mastiff(243)
boxer(242)
Rhodesian ridgeback(159)




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。

我使用Gausian filter來進行防禦，先利用Gausian function計算出一個5X5的filter，再跟攻擊後的圖片做convolution。防禦前的success rate為94%，防禦後的success rate為68%，確實可以有效降低誤判的比例，因為Gausian filter會將每個pixel和周圍的pixel做權重相加，已達到模糊化合降低雜訊的作用，所以我們可以觀察到進行防禦後的圖片與原圖相比確實變得比較模糊。