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)

   嘗試過使用 deep fool 對 keras.resnet50 進行攻擊，礙於速度問題只進行迭代五次，但是卻能非常成功的攻擊 keras resnet50，成功率為 1(比 FGSM 的 0.92 高很多)。可惜的是可能 pytorch 與 keras 的 model 不盡相同，因此在 judgeboi 上的結果很差。
   這個方法和 FGSM 的差異是 deep fool 解決了 fgsm 中選擇 ε 的問題，並且能夠利用公式最小化擾動。
   因為在結果還是 fgsm 比較好，因此 hw5_best.sh 中的 code 還是 fgsm。

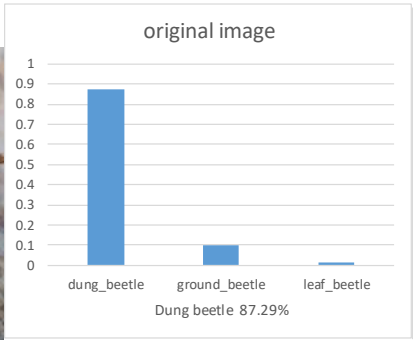
2. (1%) 請列出 hw5_fgsm.sh 和 hw5_best.sh 的結果 (使用的 proxy model、success rate、L-inf. norm)。

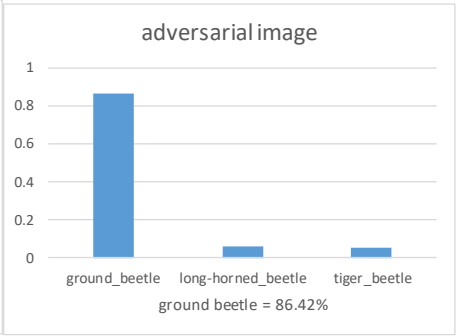
   hw5_fgsm.sh 對 VGG19 進行攻擊，在 judgeboi 上的成績為 success rate = 0.350，L-inf. norm = 4.9750。
   在死線前無法寫出更好的其他方法，因此 hw5_best.sh 也是 fgsm，後來另外實作了 deep fool，對 Resnet50 進行攻擊，success rate = 0.09，L-inf. norm =8.925，

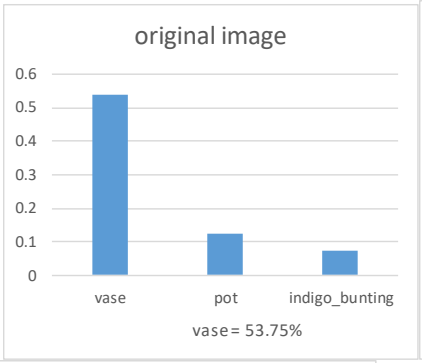
3. (1%) 請嘗試不同的 proxy model，依照你的實作的結果來看，背後的 black box 最有可能為哪一個模型？請說明你的觀察和理由。

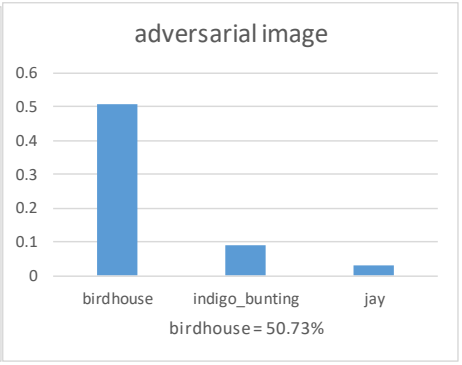
   我認為應該是 resnet50，因為在使用 deep fool 上傳時，雖然可能因為 pytorch 與 keras model 不同的問題造成 success rate 低，但是能明顯看出 resnet 50 的 success rate 遠大於其他 model。

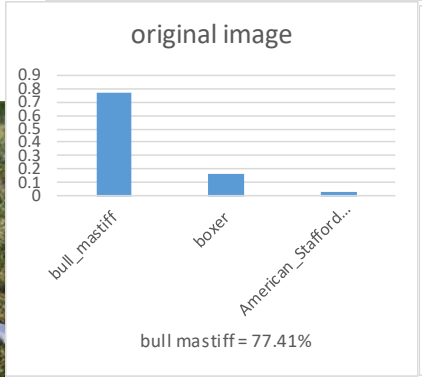
4. (1%) 請以 hw5_best.sh 的方法，visualize 任意三張圖片攻擊前後的機率圖 (分別取前三高的機率)。
   我的 hw5_best.sh 裡面是 fgsm，000.png、001.png、002.png 機率圖為：

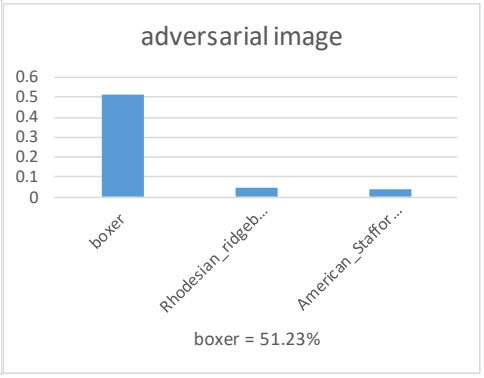













5. (1%) 請將你產生出來的 adversarial img，以任一種 smoothing 的方式實作被動防禦(passive defense)，觀察是否有效降低模型的誤判的比例。請說明你的方法，附上你防禦前後的 success rate，並簡要說明你的觀察。另外也請討論此防禦對原始圖片會有什麼影響。

   我選擇使用 Gaussian filter，並且使用 vgg19 作為模型，本來被攻擊後模型預測的 success rate 為 90.5%，經過 Gaussian filter 之後 success rate 為 61.5%。
   Gaussian filter 會讓原始圖片變得模糊。