Machine Learning 2020 - Homework 6 Report

學號：b08902100, 系級：資工一, 姓名：江昱勳

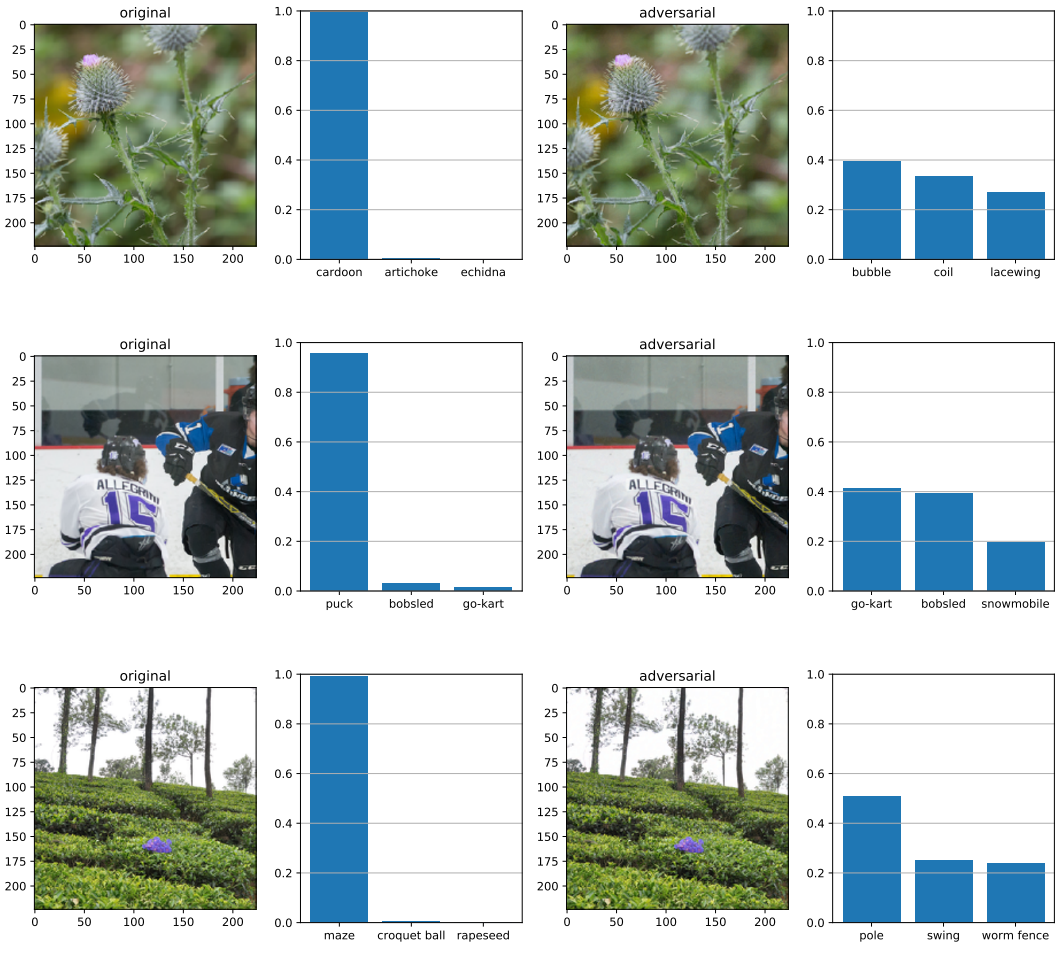
1. 試說明 hw6_best.sh 攻擊的方法，包括使用的 proxy model、方法、參數等。此方法和 FGSM 的差異為何？如何影響你的結果？請完整討論。

   我採用的攻擊方法是將 FGSM 稍作修改，執行多次的 FGSM，並且加入 momentum 的參數，讓我們在維持一定的 EPS 下也能慢慢找出最佳的噪音。而實際上我讓我的程式對於一張圖最多執行 50 次的 FGSM，而若是有先發現已成功攻擊，則會停下來，不繼續執行 FGSM；EPS 的部份我則是選擇 0.02，實際測量過 0.01 的時候會無法將全部的圖片都成功攻擊，因此我 EPS 選擇 0.02。momentum 的參數我則覺得差異不大，有嘗試過設置成 0.6 也能在差不多的 EPS 下成功攻擊圖片，而我最後選擇設製成 0.8。proxy model 的部份稍微實驗過後我採用 PyTorch 內建的 densenet121，理由如第二題。

2. 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。

   我認為背後的 black box model 最有可能是 PyTorch 內建的 densenet121，因為我有嘗試將我的 proxy model 設置成其他的 model，在相同的條件下，成功率都會大幅下降。此外，嘗試將一份稍作攻擊過後的圖片傳至 Judge 上後，可以觀察到 densenet121 的 accuracy 與 Judge 上的 success rate 最為相近，故可以推測背後的 model 為 densenet121。

3. 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)。



4. 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。

   這裡以 PyTorch 內建的 densenet121 做測試，使用 2×2 的高斯模糊。在未加入高斯模糊前，原始圖片正確率為 0.925，被攻擊的圖片正確率則為 0。而加入高斯模糊後，原始圖片的正確率降至 0.595，但被攻擊的圖片正確率則升回 0.52，可以發現兩者差異不大，但是準確率的確被減少了許多。以下附上與第 3 題相同，但是高斯模糊後的結果。