Machine Learning Spring 2020 - HW6 Report

學號: B07902064 系級: 資工二 姓名: 蔡銘軒

1. (2%)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。

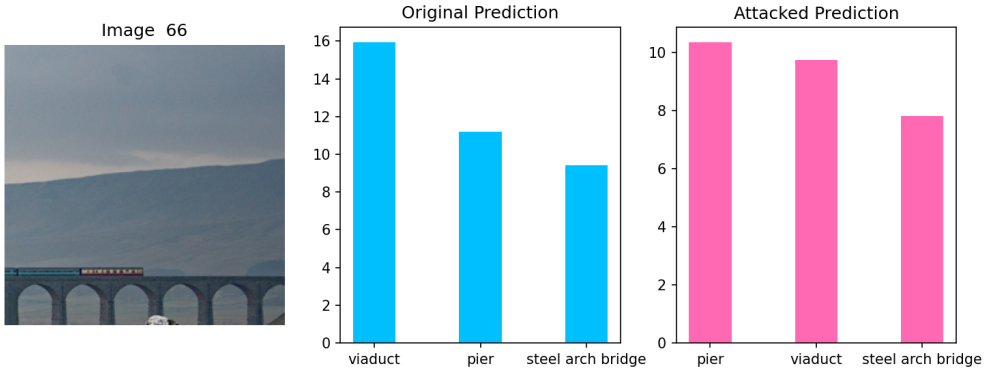
   - Proxy model: DenseNet 121
   - 方法：基本上best還是基於FGSM。我主要做的變化在於ϵ值的調整。在原本的FGSM裡面，每一張圖都使用相同的ϵ，我認為這是可以調整的，因此對於每一張圖片，我都使用$\epsilon = \frac{n}{256}, n = 0, 1, \ldots, 255$，從小的$n$開始往上，直到攻擊成功。但因為不希望L-inf太大，因此如果嘗試到$n = 255$還是沒成功，就放棄這次的攻擊。
   - 與FGSM的差異：將原本的FGSM的ϵ固定設為0.01，在judge boi上可以做到0.805的success rate跟1.000的L-inf。在Best model裡，大概有80%以上的圖片在$n = 1$就成功攻擊，少部分需要做到$n = 4, 5$，另外有一些圖片需要80以上甚至超過100。最後的success rate是0.990，L-inf是2.2950。這邊的L-inf比原本還要高我認為是那些n比較大的圖片造成的，不過因為他們只是少數，所以平均下來L-inf還是落在合格範圍內。

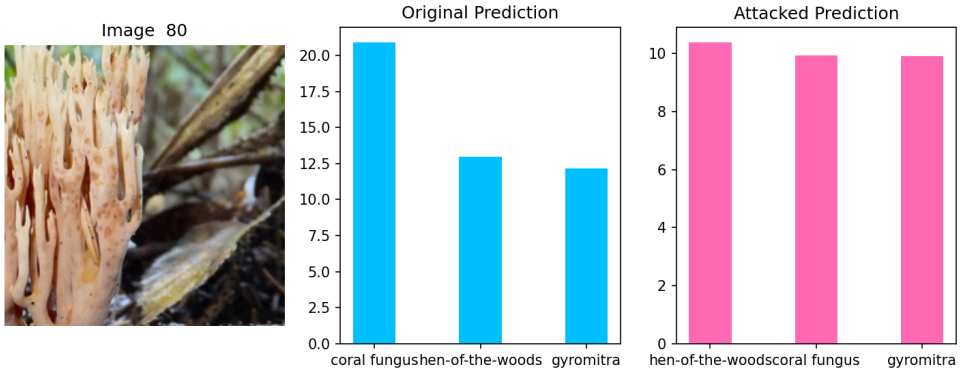
2. (1%) 請嘗試不同的 proxy model，依照你的實作的結果來看，背後的 black box 最有可能為哪一個模型？請說明你的觀察和理由。

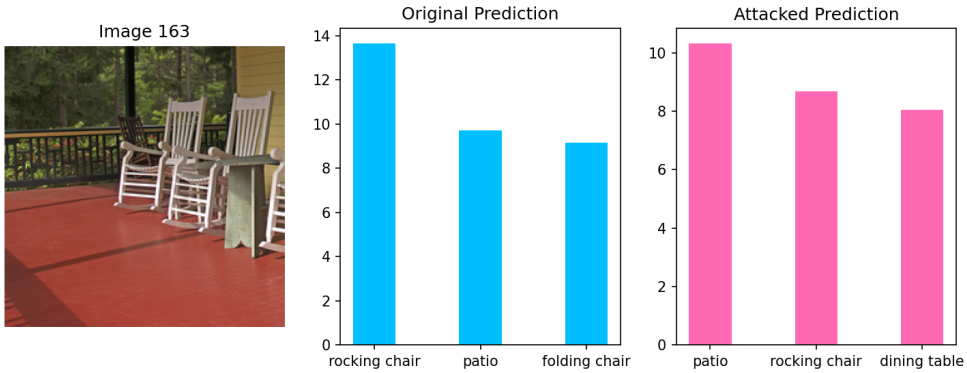
   經過與同學的討論以及使用FGSM的方式實驗不同proxy model後，發現使用DenseNet 121會在judge boi上得到最好的表現。其他proxy model，例如VGG-16與VGG-19，在程式碼相同，只更改proxy model的情況下，在judge boi上的success rate不到0.1，遠低於DenseNet 121 0.9以上的表現。另外經過同學的提點，改變計算success rate與L-inf的算法後發現DenseNet 121的表現跟judgeboi是完全吻合的，因此我認為背後的black box是DenseNet 121。

3. (1%) 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）。

   因為有些類別的名稱太長會破壞排版，因此這些圖片的categories名稱有被我縮減過（取第一個單字）







4. (2%) 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。

   我使用了python的PIL套件的BoxBlur來對圖片進行模糊化，在讀進圖片後先進行模糊化再讓model判斷。

```
from PIL import Image
from PIL import ImageFilter
img = Image.open('...')
img = img.filter(ImageFilter.BoxBlur(1))
```

   使用radius = 1讓每個pixel受到周圍各個方向1個pixel（形成一個3x3的方形，包含自己共9個pixel）的影響，讓他的數值變成這些pixel的平均。

   我的實驗結果如下：

```
Success rate:
攻擊圖檔 = 0.805
攻擊圖檔 + BoxBlur = 0.39
```

```
Accuracy:
原圖 = 0.925
原圖 + BoxBlur = 0.86
```

   加上Blur能夠使success rate降低的原因是因為每個pixel會受到周圍pixel的數值影響而變化。我們在攻擊的時候通常是改動一些pixel去影響model判斷，但在經過模糊化之後，這些改動的影響會因為周圍其他pixel的數值而被淡化，降低了攻擊的效果並增加model預測成功的機率。

但這個模糊的效果也影響了model本身的正確率。對原圖進行模糊後，正確率下降了約5%。不過我認為這個代價是可以接受的。

我有嘗試讓模糊化更強烈。當radius = 2時，我得到如下結果：

```
Success rate:
攻擊圖檔 = 0.805
攻擊圖檔 + BoxBlur = 0.36
```

```
Accuracy:
原圖 = 0.925
原圖 + BoxBlur = 0.745
```

增加模糊化雖然又略為降低攻擊的success rate，但原圖的分類成功率也降低了將近20%，我認為這個代價就太高了。在經過幾次調整以及換不同smoothing方式後，我認為在這次的情況，取BoxBlur加上radius = 1是個不錯的平衡點。