Machine Learning HW5 Report

學號：B06902066　系級：資工二　姓名：蔡秉辰

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)

在 hw5_best 中，使用的 proxy model 為---pytorch 上的 ResNet50、攻擊方式為---iterative FGSM，每次都從 epsilon = 1 開始做 FGSM，若 model 的判定和原圖不同則輸出，反之則將 epsilon + 1，直到 model 判定和原圖不同為止。

此方法可以使多數圖片用更小的 epsilon 就攻擊成功，不需要使用到原本(epsilon = 4)那麼大的 epsilon。幾乎有約莫七成的圖片使用 epsilon = 1 就攻擊成功了。而剩餘的圖片多半 epsilon < 5。最後幾張難以攻擊成功的就可以使用大很多的 epsilon，且 L-inf 因平均的關係，也不會太大。

結果方面，攻擊成功率可以來到 100%。當然前提是因為已經先找到 black box 使用的 model。和使用同樣 proxy model 但僅單純使用 FGSM 的方法(epsilon = 4)相比(成功率：92%, L-inf norm = 4)，成功率更高，L-inf norm 亦更小。

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)。

	Proxy model	Success rate	L-inf norm.
hw5_fgsm	ResNet50 on Keras	0.385	3.96
hw5_best	ResNet50 on Pytorch	1.000	2.65

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。

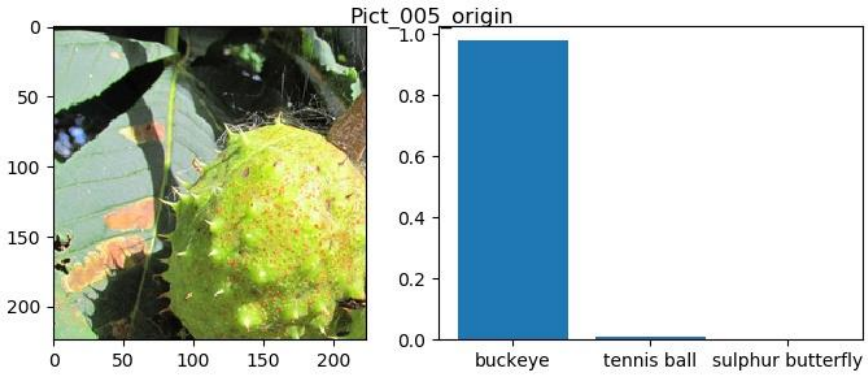
當初尋找 model 時，依序使用：Keras 上的 VGG16、VGG19、ResNet50 以及 Pytorch 上的 ResNet50。而使用到 Pytorch 的 ResNet50 這個 model 時，其攻擊成功率較其它 model 高上非常多，且正確率和在自己本機上測得幾乎一樣，故判定 black box 背後的 model 就是這個；也因此，不需要再去做剩下未試過的 model。

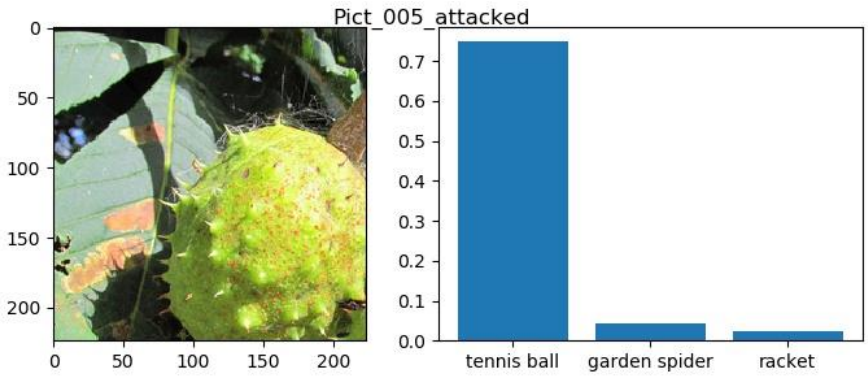
測試時所使用的方式、變數、攻擊 black box 的結果：

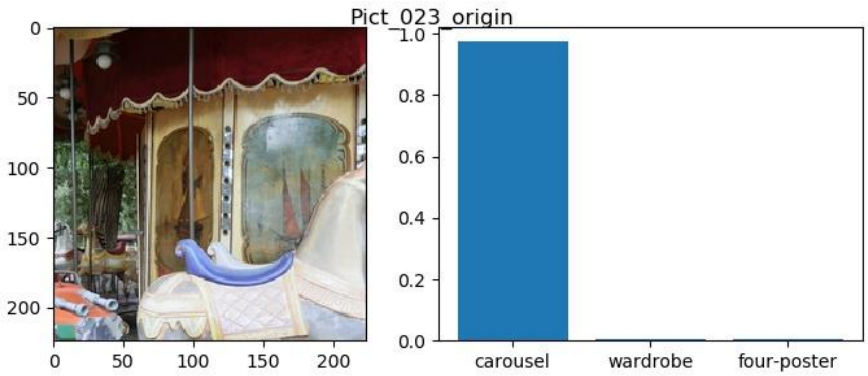
	VGG16	VGG19	ResNet50 (Keras)	ResNet50 (Pytorch)
Success rate	0.340	0.350	0.385	0.920

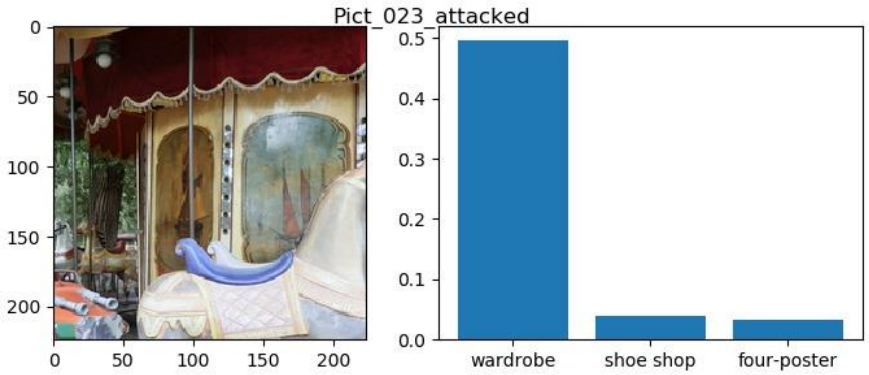
使用之方式均為 FGSM，epsilon 設為 4。

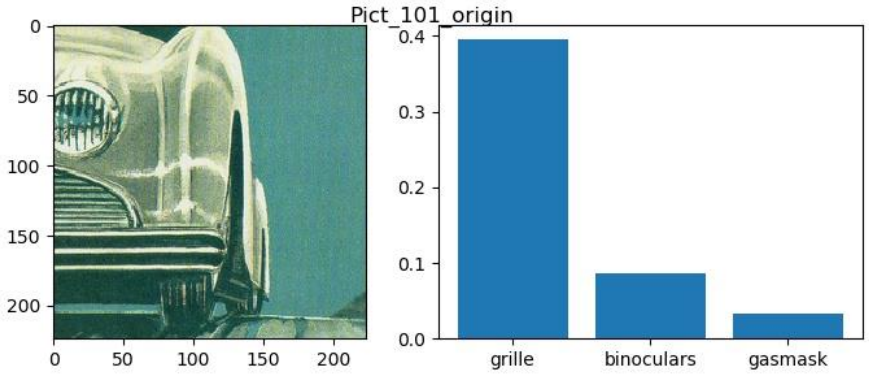
4. (1%) 請以 hw5_best.sh 的方法，visualize 任意三張圖片攻擊前後的機率圖 (分別取前三高的機率)。

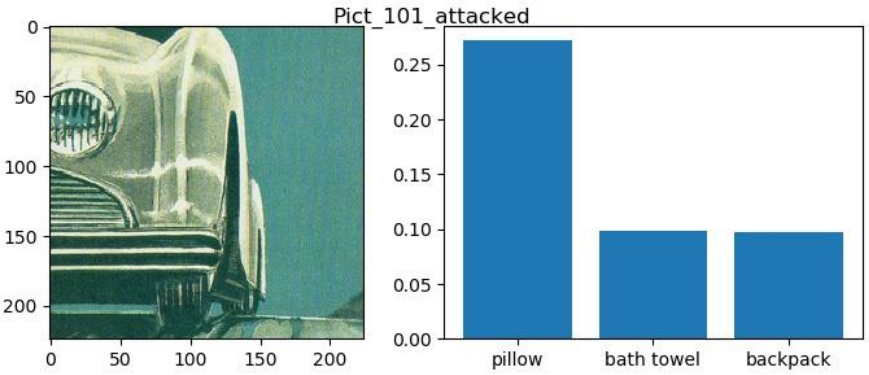












可以看出因為是使用 iterative FGSM，故 attack 成功的很多是使 model 判斷成和原圖較相近的東西，舉例像是將 buckeye 判成 tennis。並沒有觀察到像課堂上所說之將貓判成鍵盤這種差很多的東西。

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。

使用的防禦方式為 Gaussian Blur，利用 opencv 套件來實行；其中，使用之 ksize(模板大小) = (7, 7)。以下為有無實行攻擊/防禦的結果：

	Not attack	Attack
Not defense	0	1
Defense	0.215	0.330

以上表格的數字代表辨識和原圖不同的比例(attack 成功的比例)

由結果可知，使用 Gaussian Blur 的確可使攻擊成功率下降，達到防禦的目的。但同時對原圖而言，辨識錯誤率也會提升(以上述例子而言是 21.5%)。

而此項操作會使原圖變得模糊，以”016.png”為例：(左為原圖，右為 Defense 後)