學號：b06502158　　系級：機械三　　姓名：陳柏元

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。(依內容完整度給分)**

   在 hw5_best 中我是使用 densenet121 當作 Proxy model，並使用 Basic Iterative Method 來做 Adversarial attack。Basic Iterative Method 和 FGSM 都是基於 gradient 產生對抗樣本的方法，但是，FGSM 只會在原圖加上一次 noise，導致常常發生無法準確擬合模型參數，讓誤判成功率無法上升。而 Basic Iterative Method 會基於前一次加上 noise 的圖片，再去產生新的 gradient，再累加上新的 noise，且每次加上的 noise 會由一個參數 alpha 控制更新速度，照上述方法迭迨下去，求得最後結果。而從我的 Success rate 中也可以發現 Basic Iterative Method 透過多步增添 noise 的方法，更能擬合模型參數，可以用較小的 L-inf norm 達到較高的 Success rate。在 Basic Iterative Method 中，我將 epsilon 設為 0.02、alpha =0.005、n_iter = 10。

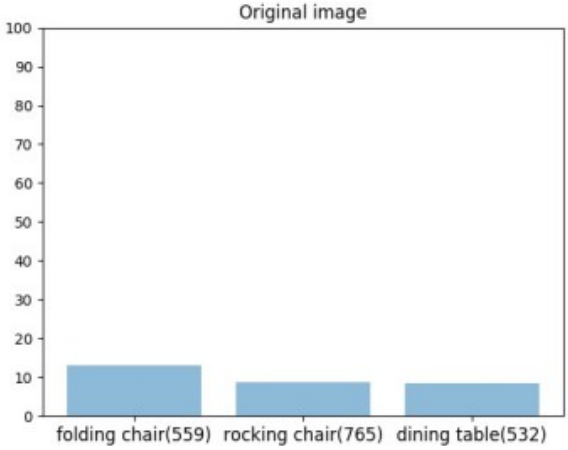
2. **(1%) 請嘗試不同的 proxy model，依照你的實作的結果來看，背後的 black box 最有可能為哪一個模型？請說明你的觀察和理由。**

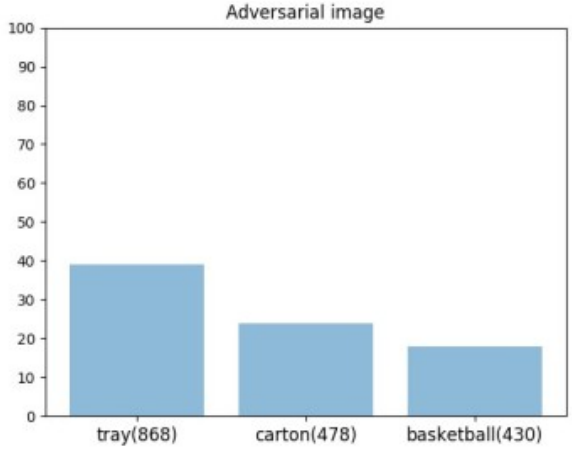
Model	VGG16	VGG19	Resent50	Resnet101	Densenet121	Densenet169
Success rate(%)	33.5%	28.5%	26.5%	24%	100%	29%

從表格中可以發現，在同樣的參數，只有更改 Model 之下，只有 Densenet121 的 success rate 和其他 model 有顯著差異，因此我認為 Black box 最有可能為 Densenet121。

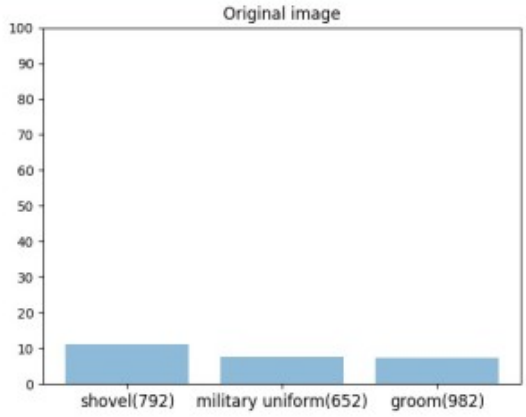
3. **(1%) 請以 hw6_best.sh 的方法，visualize 任意三張圖片攻擊前後的機率圖 (分別取前三高的機率)。**

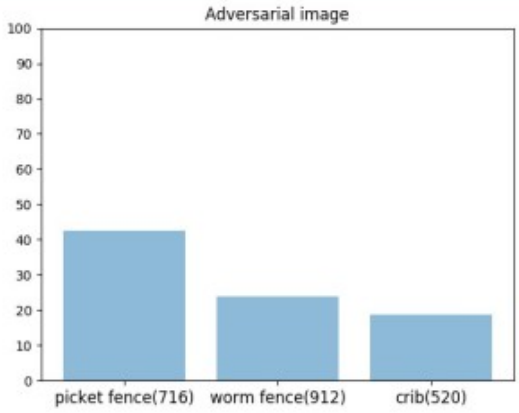
Highest: Origin->folding chair, Adversarial→tray



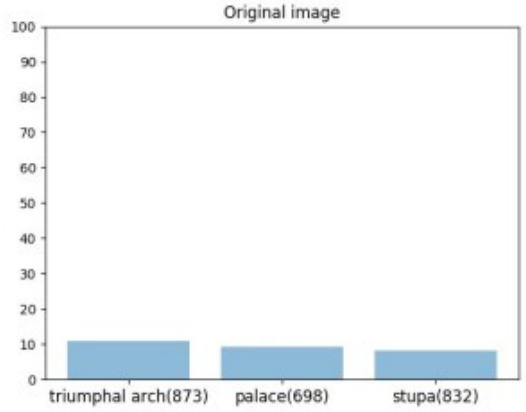


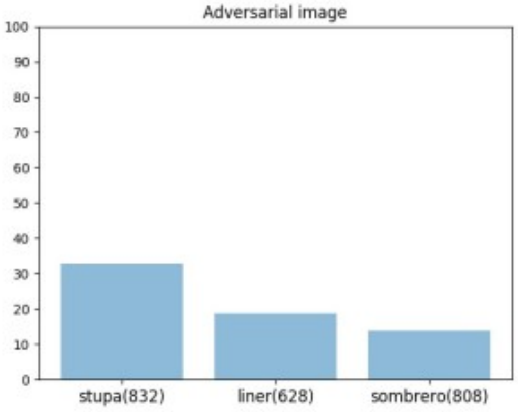
Highest: Origin→shovel, Adversarial→picket fence





Highest: Origin → triumphal arch, Adversarial → stupa





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。**

我是使用 Gaussian Filter 的方式濾除雜訊，達到 Passive defense 的效果。將 Adversarial image 的圖片和 Gaussian Filter 進行 Convolution 之後，可以將圖片的雜訊過濾，但也會讓圖片變得模糊許多，儘管圖片變得模糊，但並不會讓圖片看 起來跟原圖有太多差距，這是因為 Gaussian Filter 的矩陣最大值在中心點，捲積後會強化圖片中心點，並弱化邊角的權重。

	Success rate	L-inf.norm
Before Defense	1	6
After Defense	0.74	78.51

從表格可以發現，將 Adversarial image 通過 Gaussian Filter 過濾之後，success rate 雖然下降許多，但還是偏高。而原本的圖片在通過 Gaussian Filter 後 success rate 為 0.08。我認為是因為 Gaussian Filter 的原理是讓圖片變得較圓 滑，濾除雜訊，仍保有原圖的特徵，因此防禦對原始圖片的影響並不大，誤判的比例不高。而相比 Adversarial image，防禦後影響較大，可能是因為經由 Adversarial attack 所產生的雜訊主要加在圖片的重要特徵上，但是因為通過 Gaussian Filter，讓這些重要特徵變得較圓滑，因此讓誤判成功率降低。