

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)

proxy model	Resnet50
method	Iterative LEAST-LIKELY CLASS Method
Eps(可以決定 L-infinity 的大小)	18
Alpha(更新 image 的 step size)	1
Number of Iteration	22

我使用的方法是 ICLR2017 年提出的一篇 paper “ADVERSARIAL EXAMPLES IN THE PHYSICAL WORLD” 其中的 Iterative LEAST-LIKELY CLASS Method。其實這個演算法跟 FGSM 蠻像的，差別在於，我針對 gradient 更新了 input image 很多次，不只更新一次而已。並去計算 proxy model predict 出來 confidence 最低的那個 class 和 model 的 output 之間的 loss，再去做 back propagation 算出梯度值，根據梯度的方向去更新 image。希望最後生成的攻擊影像可以讓 model 辨識為原本 confidence 最低的 class。(FGSM 則是希望更新完的影像可以讓 ground truth 那個 class confidence 越低越好)

詳細的公式列在下方：

$$y_{LL} = \arg \min_y \{p(y|\mathbf{X})\}.$$

$$\mathbf{X}_0^{adv} = \mathbf{X}, \quad \mathbf{X}_{N+1}^{adv} = \text{Clip}_{X,\epsilon} \{ \mathbf{X}_N^{adv} - \alpha \text{sign}(\nabla_{\mathbf{X}} J(\mathbf{X}_N^{adv}, y_{LL})) \}$$

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)。

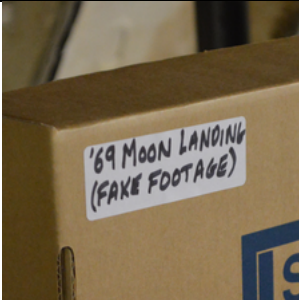
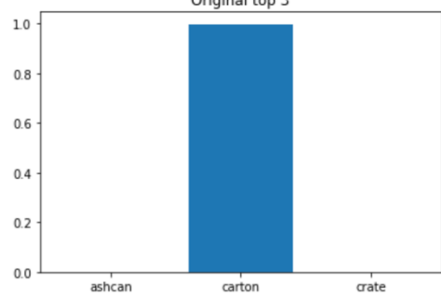
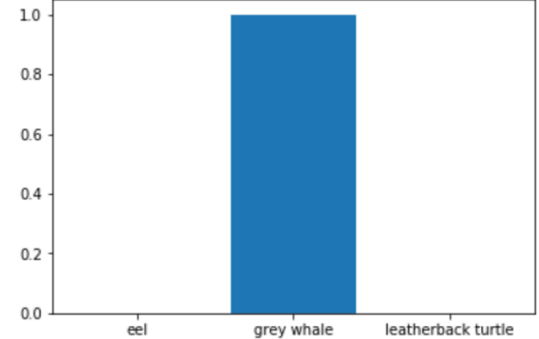
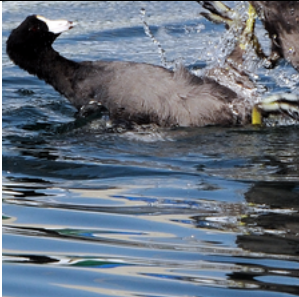
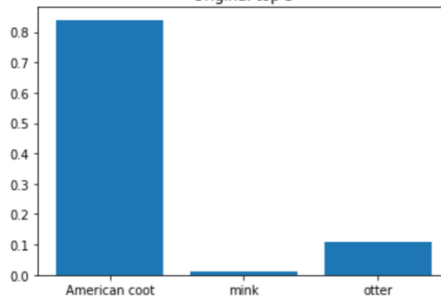
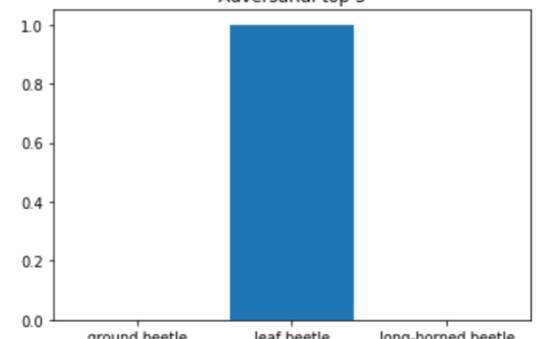

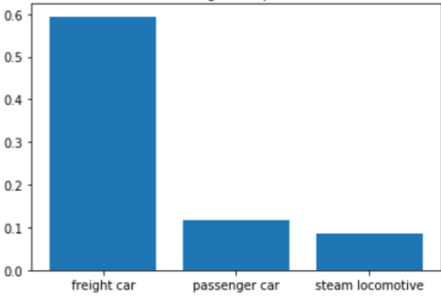
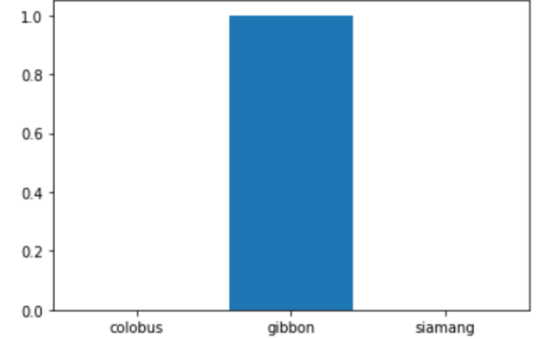
	FGSM (hw5_fgsm)	Iterative LEAST-LIKELY CLASS Method (hw5_best)
Proxy model	Resnet 50	Resnet 50
Success rate	0.32	0.99
L-inf. norm	23	5

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。

Proxy model	Success rate	L-inf. norm
vgg16	0.060	5
vgg19	0.070	5
resnet50	0.99	5
resnet101	0.060	5
densenet121	0.090	5
densenet169	0.055	5

根據上表，推測背後的模型為 resnet50，因為 success rate 明顯好非常多！

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)。

Original Image	攻擊前機率圖	攻擊後機率圖																
 <p>035.png</p>	<p>Original top 3</p>  <table><tr><th>Category</th><th>Probability</th></tr><tr><td>ashcan</td><td>0.0</td></tr><tr><td>carton</td><td>1.0</td></tr><tr><td>crate</td><td>0.0</td></tr></table>	Category	Probability	ashcan	0.0	carton	1.0	crate	0.0	 <table><tr><th>Category</th><th>Probability</th></tr><tr><td>eel</td><td>0.0</td></tr><tr><td>grey whale</td><td>1.0</td></tr><tr><td>leatherback turtle</td><td>0.0</td></tr></table>	Category	Probability	eel	0.0	grey whale	1.0	leatherback turtle	0.0
Category	Probability																	
ashcan	0.0																	
carton	1.0																	
crate	0.0																	
Category	Probability																	
eel	0.0																	
grey whale	1.0																	
leatherback turtle	0.0																	
 <p>159.png</p>	<p>Original top 3</p>  <table><tr><th>Category</th><th>Probability</th></tr><tr><td>American coot</td><td>0.85</td></tr><tr><td>mink</td><td>0.02</td></tr><tr><td>otter</td><td>0.12</td></tr></table>	Category	Probability	American coot	0.85	mink	0.02	otter	0.12	<p>Adversarial top 3</p>  <table><tr><th>Category</th><th>Probability</th></tr><tr><td>ground beetle</td><td>0.0</td></tr><tr><td>leaf beetle</td><td>1.0</td></tr><tr><td>long-horned beetle</td><td>0.0</td></tr></table>	Category	Probability	ground beetle	0.0	leaf beetle	1.0	long-horned beetle	0.0
Category	Probability																	
American coot	0.85																	
mink	0.02																	
otter	0.12																	
Category	Probability																	
ground beetle	0.0																	
leaf beetle	1.0																	
long-horned beetle	0.0																	
	<p>Original top 3</p>  <table><tr><th>Category</th><th>Probability</th></tr><tr><td>freight car</td><td>0.6</td></tr><tr><td>passenger car</td><td>0.12</td></tr><tr><td>steam locomotive</td><td>0.09</td></tr></table>	Category	Probability	freight car	0.6	passenger car	0.12	steam locomotive	0.09	<p>Adversarial top 3</p>  <table><tr><th>Category</th><th>Probability</th></tr><tr><td>colobus</td><td>0.0</td></tr><tr><td>gibbon</td><td>1.0</td></tr><tr><td>siamang</td><td>0.0</td></tr></table>	Category	Probability	colobus	0.0	gibbon	1.0	siamang	0.0
Category	Probability																	
freight car	0.6																	
passenger car	0.12																	
steam locomotive	0.09																	
Category	Probability																	
colobus	0.0																	
gibbon	1.0																	
siamang	0.0																	

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。

我使用的防禦方法是 3x3 Median filter，此方法非常適合拿來解決 salt and pepper 的雜訊問題。此方法是以一個  $n \times n$  的 mask 遮罩原始影像，並排序 mask 遮罩中的所有 pixel 值，並取中間值當作該點 pixel 的輸出。防禦前及防禦後的 success rate 如下表所示：

	Success rate	L-inf.
Before smoothing	0.99	5
After smoothing	0.295	131.8450



Before smoothing



After smoothing

可以發現做完 smoothing(3x3 median filter)以後，success rate 確實降低很多，但同樣的卻使得原始影像變得更模糊，L-inf.大幅上升。