Amnesia as a Catalyst for Enhancing Black Box Attacks in Image Classification and Object Detection

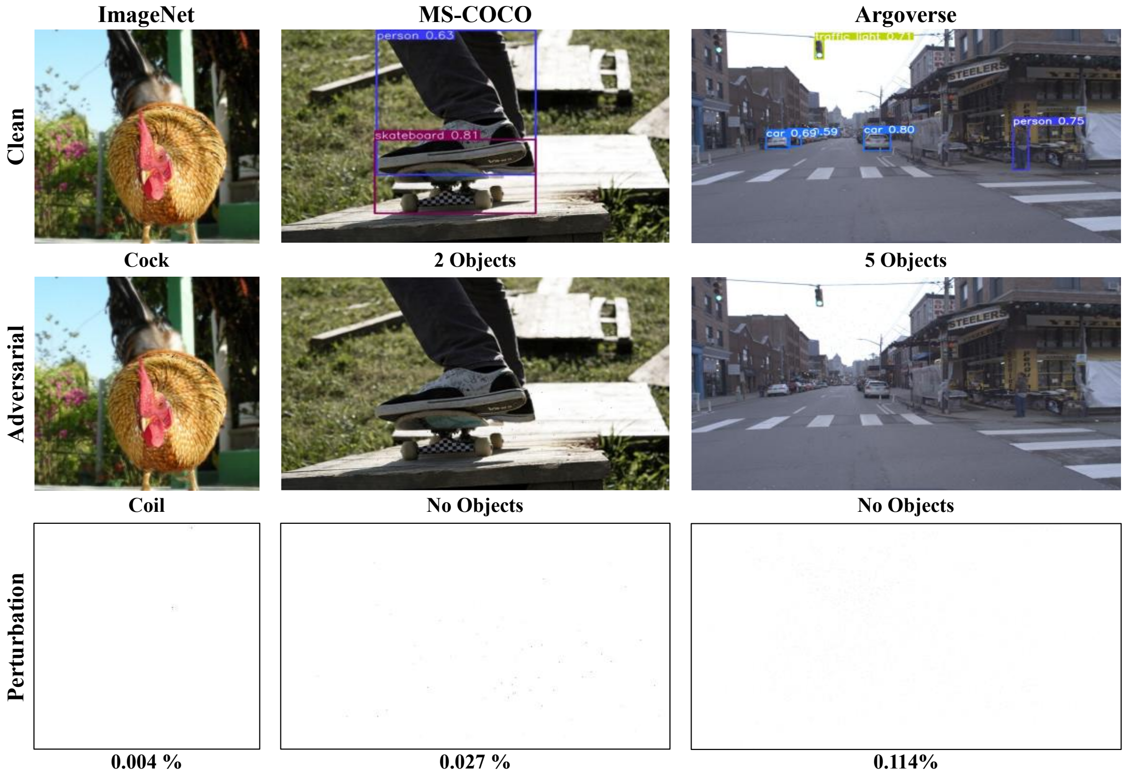
Dongsu Song, Daehwa Ko, Jay Hoon Jung

Korea Aerospace University

Neurips 2024

Introduction

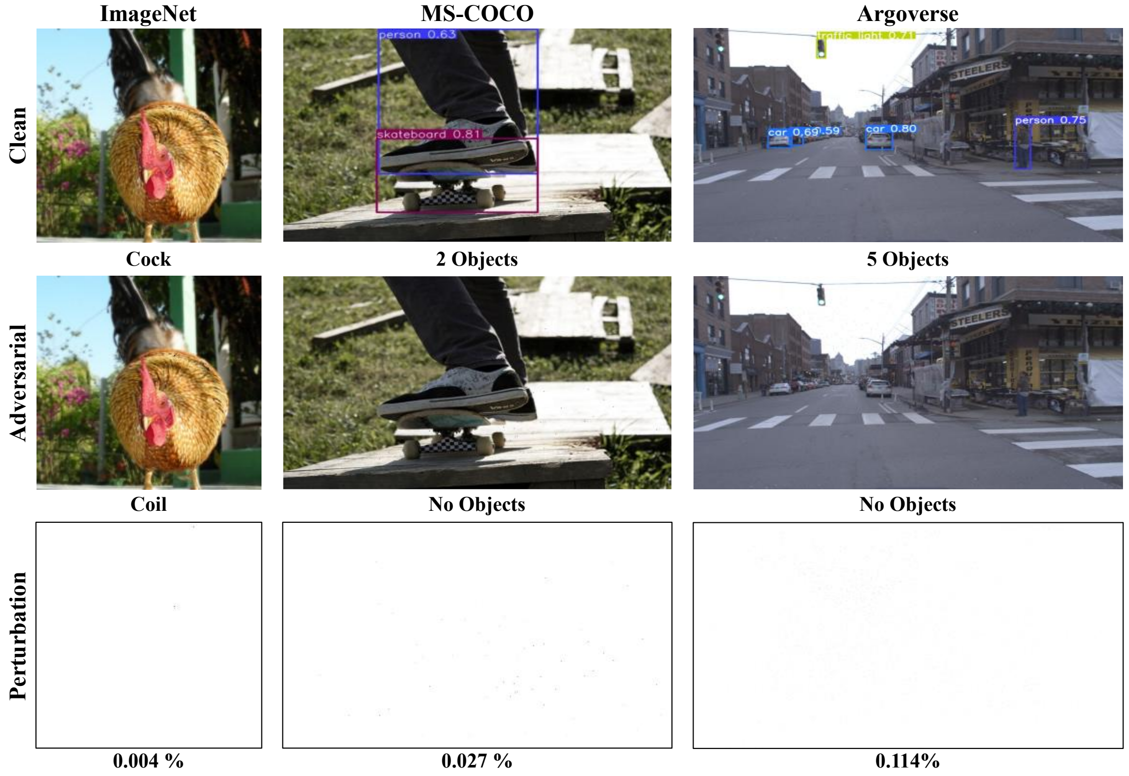
- Adversarial Attack



- Adversarial attacks are adding imperceptible noise to clean samples for misleading Deep Neural Networks(DNNs).

Introduction

- Adversarial Attack



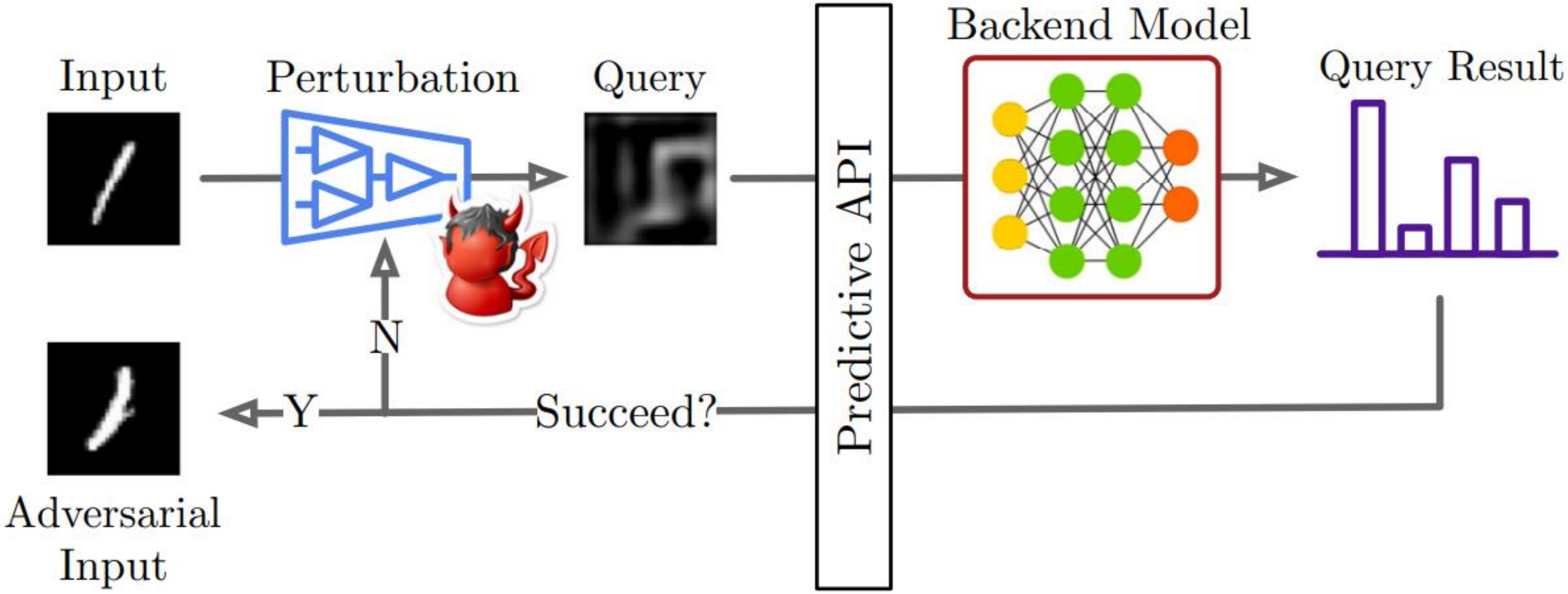
How to evaluate imperceptibility

$$\|x\|_0 = |x_1|^0 + |x_2|^0 + \cdots + |x_n|^0$$

- Adversarial attacks are adding imperceptible noise to clean samples for misleading Deep Neural Networks(DNNs).

Introduction

- Query-based attack in Black box



- A query-based attack approach receives limited information (e.g., confidence scores) to generate perturbations in a black-box setting.

Motivations

- Many pixel attacks rely on patches with fixed shapes, leading to increased pixel perturbations. Therefore, we shift our focus from **patch-based** methods to **individual pixels**.

Motivations

- Many pixel attacks rely on patches with fixed shapes, leading to increased pixel perturbations. Therefore, we shift our focus from **patch-based** methods to **individual pixels**.
- Some studies generate adversarial attacks by training Reinforcement Learning (RL) models. However, fully training RL is **inefficient** for queries. Therefore, we tackle this issue by focusing on reward convergence in **Memory**, thereby improving the query efficiency of adversarial example generation.

Motivations

- Many pixel attacks rely on patches with fixed shapes, leading to increased pixel perturbations. Therefore, we shift our focus from **patch-based** methods to **individual pixels**.
- Some studies generate adversarial attacks by training Reinforcement Learning (RL) models. However, fully training RL is **inefficient** for queries. Therefore, we tackle this issue by focusing on reward convergence in **Memory**, thereby improving the query efficiency of adversarial example generation.
- We consider not only **adversarial attack scenarios** but also **real-world scenarios** by simulating the pixel defect issues found in cameras.

Contributions

- **RFPAR**: We introduce the **R**emember and **F**orget **P**ixel **A**ttack using **R**einforcement learning, which enhances query efficiency and achieves low l_0.

Contributions

- **RFPAR**: We introduce the **R**emember and **F**orget **P**ixel **A**ttack using **R**einforcement learning, which enhances query efficiency and achieves low l_0.

- **Extension task:** We extends pixel attacks from image classification to object detection.

Contributions

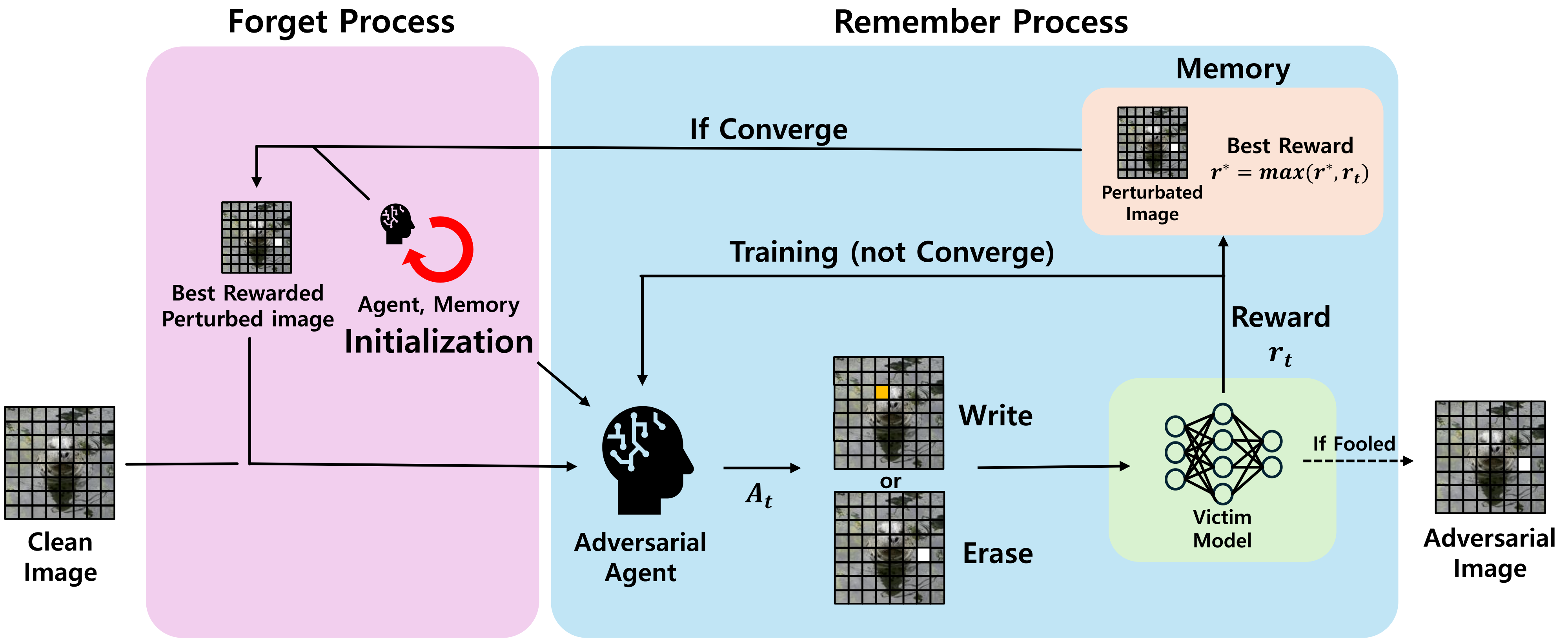
- **RFPAR**: We introduce the **R**emember and **F**orget **P**ixel **A**ttack using **R**einforcement learning, which enhances query efficiency and achieves low l_0.

- **Extension task:** We extends pixel attacks from image classification to object detection.

- **Resolution Enhancement:** RFPAR supports attacks on high-resolution images(up to 1920x1200).

RFPAR

- RFPAR: Remember and Forget Pixel Attack using Reinforcement learning



Results in Image classification

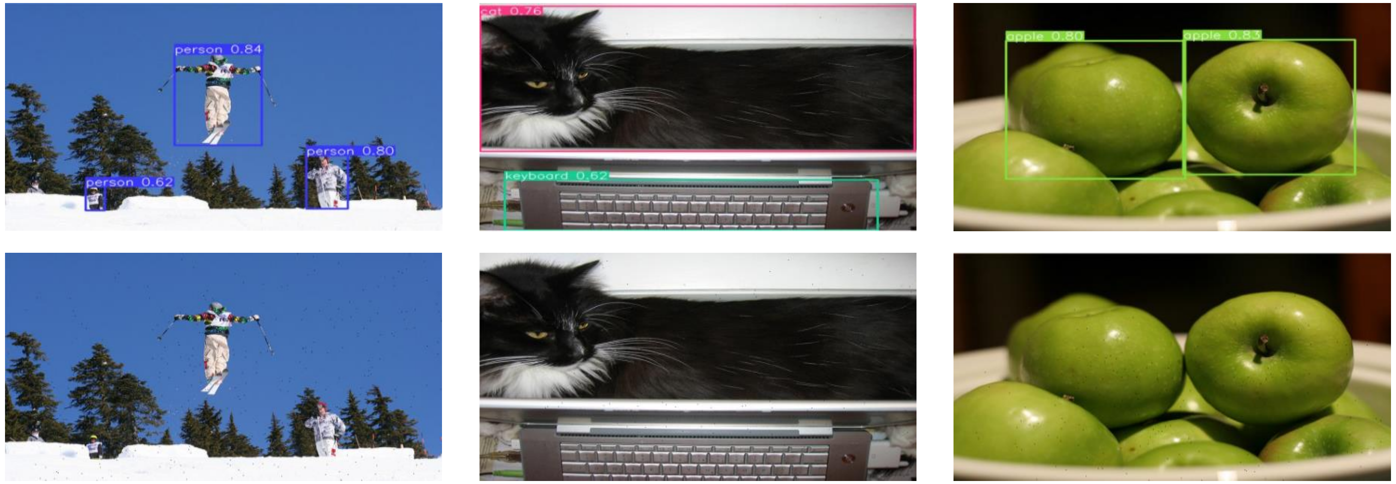
Table 1: **The results of adversarial attacks on the ImageNet dataset.** Each score represents the mean success rate of the attack, mean L_0 norm and mean the number of queries. In terms of the success rate, a higher value signifies better performance, whereas for the L_0 norm and the number of queries, lower values are indicative of superior performance. The best method is highlighted in bold.

Model	Test accuracy	Attack	Succes rate ↑	L_0 ↓	Query ↓
VIT-B[24]	81.07 %	OnePixel[8]	9.3 %	15	1453
		ScratchThat[9]	40.9 %	420	9418
		Pixle[11]	51.4 %	286	728
		RFPAR(Ours)	**64.1 %**	211	613
ResNeXt50[25]	77.62 %	OnePixel[8]	8.1 %	15	5100
		ScratchThat[9]	38.1 %	95	1400
		Pixle[11]	89.1 %	538	663
		RFPAR(Ours)	**95.3 %**	138	442
RegNetX-32GF[26]	80.62 %	OnePixel[8]	12.3 %	15	1358
		ScratchThat[9]	60.6 %	427	8653
		Pixle[11]	73.7 %	276	705
		RFPAR(Ours)	**88.4 %**	164	484
DenseNet161[27]	77.14 %	OnePixel[8]	14.1 %	15	1248
		ScratchThat[9]	60.6 %	425	8367
		Pixle[11]	82.3 %	243	625
		RFPAR(Ours)	**91.7 %**	152	464
MNASNet[28]	73.46 %	OnePixel[8]	14.2 %	15	1128
		ScratchThat[9]	65.3 %	425	8828
		Pixle[11]	83.7 %	240	607
		RFPAR(Ours)	**95.0 %**	150	442
MobileNet-V3[29]	74.04 %	OnePixel[8]	8.1 %	15	1461
		ScratchThat[9]	51.8 %	420	9293
		Pixle[11]	69.6 %	306	769
		RFPAR(Ours)	**86.6 %**	213	596

Results in Objective detection

Table 2: **Attack Results on Object Detection Models.** The subscripts after RFPAR denote a pixel attack rate, α. RM indicates the average percentage of objects removed from the clean image. L_0 represents the average $\|\delta\|_0$. Query denotes the average number of queries made to the victim model. Higher RM, lower mAP, lower L_0, and lower Query values indicate better performance.

Attacks	YOLOv8[22]				DDQ[33]			
	RM $\uparrow$	mAP $\downarrow$	L_0 $\downarrow$	Query $\downarrow$	RM $\uparrow$	mAP $\downarrow$	L_0 $\downarrow$	Query $\downarrow$
clean	-	0.398	-	-	-	0.376	-	-
RFPAR$_{0.01}$	0.65	0.218	521	1403	0.60	0.125	391	1450
RFPAR$_{0.02}$	0.70	0.187	955	1427	0.73	0.103	787	1690
RFPAR$_{0.03}$	0.75	0.151	1459	1374	0.76	0.075	1074	1512
RFPAR$_{0.04}$	0.76	0.150	1814	1348	0.80	0.061	1429	1457
RFPAR$_{0.05}$	0.91	0.111	2043	1254	0.83	0.054	1780	1528



Email: raister2873@gmail.com

Thank you