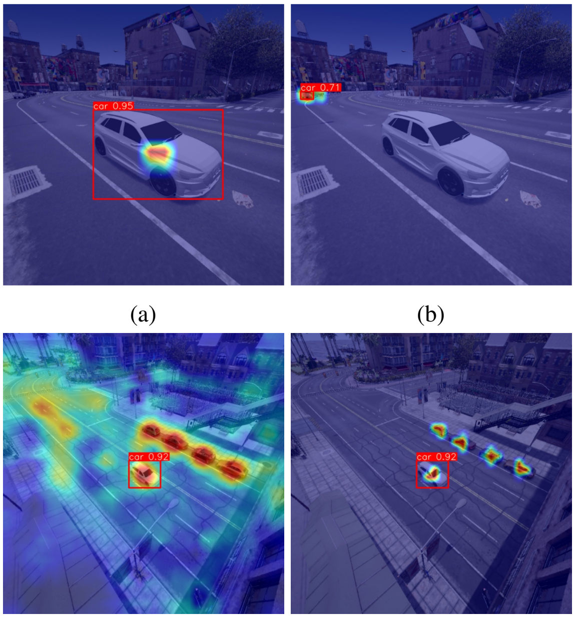
MFA: Multi-layer Feature-aware Attack for Object Detection

Wen Chen[1] **Yushan Zhang**[3] **Zhiheng Li**[1] **Yuehuan Wang**[1,2*]

[1]School of Artificial Intelligence and Automation , Huazhong University of Science and Technology, Wuhan, China
[2]National Key Lab of Science and Technology on Multi-spectral Information Processing , Wuhan, China
[3]Shanghai Institute of Satellite Engineering , Shanghai, China

Abstract

Physical adversarial attacks can mislead detectors in real-world scenarios and have attracted increasing attention. However, most existing works manipulate the detectors final outputs as attack targets while ignoring the inherent characteristics of objects. This can result in attacks being trapped in model-specific local optima and reduced transferability. To address this issue, we propose a *Multi-layer Feature-aware Attack* (MFA) that considers the importance of multi-layer features and disrupts critical object-aware features that dominate decision-making across different models. Specifically, we leverage the location and category information of detector outputs to assign attribution scores to different feature layers. Then, we weight each feature according to their attribution results and design a pixel-level loss function in the opposite optimized direction of object detection to generate adversarial camouflages. We conduct extensive experiments in both digital and physical worlds on ten outstanding detection models and demonstrate the superior performance of MFA in terms of attacking capability and transferability. Our code is available at: `https://github.com/ChenWen1997/MFA`.



(a) (b)

(c) (d)

Figure 1: Attention maps on yolov3 obtained by performing feature attribution on the target object marked by the red box [Redmon and Farhadi, 2018]. **(a)** and **(b)** show attention maps of different feature layers when the detector detects objects of different scales in the same picture. **(c)** shows an attention map attributed with category information. **(d)** shows an attention map attributed with both location and category information.

1 INTRODUCTION

Deep neural networks (DNNs)have achieved impressive performance in object detection [Ren et al., 2015, He et al., 2017, Tian et al., 2019, Liu et al., 2021]. However, they are found to be vulnerable to adversarial examples [Szegedy et al., 2013],which are elaborately crafted to fool DNNs. The effectiveness of adversarial examples has also been proved in object detection [Lu et al., 2017].

Many adversarial attack methods have been proposed for object detection, which can generally be divided into two categories: **1) Digital attacks**, which modify the pixels of input images directly in the digital space [Xie et al., 2019, Wang et al., 2021b, Zhang et al., 2022], and **2) Physical attacks**, which perform attacks on physical objects before camera imaging [Athalye et al., 2018, Chen et al., 2019b, Wang et al., 2022b]. Physical attacks typically generate adversarial perturbations in the digital world and then apply

[*] Corresponding Author

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

them to real objects through painting or direct creation of perturbed objects. In this paper, we focus on physical attacks as they have more practical significance for deployed deep learning applications.

However, existing works always ignore the inherent characteristics of objects, resulting in subpar attack ability and transferability. In particular, These limitations can be summarized as follows: **(1)** Existing works have yet to attempt to disrupt multi-layer features, which play a significant role in object detection. As illustrated in Figures 1a and 1b, the detection of different objects with significant scale differences on the same image is performed on different feature layers. **(2)** Most current methods directly take the final outputs of the model as attack targets[Thys et al., 2019, Wang et al., 2022b, Du et al., 2022], which can easily overfit the source model and reduce transferability. The Dual Attention Suppression (DAS) Attack[Wang et al., 2021a] exploits attention maps to generate perturbations but has certain limitations. Firstly, DAS only utilizes category information to attack a detector, which cannot accurately assign attribution scores to non-target regions, especially when the target is relatively small in the image. As shown in Figure 1c, the non-target regions of the attention map have a relatively strong response. Secondly, DAS will compromise the accuracy of importance estimation since using a method similar to Grad-CAM[Selvaraju et al., 2017] to average each channel as weight will not be able to distinguish the attribution scores within each channel.

To address the mentioned issues, we propose the Multilayer Feature-aware Attack (MFA) by distorting critical object-aware features at different layers. Specifically, we first attempt to attribute the location and category information of outputs to different feature layers. As shown in Figure 1d, the attribution with location and category would assign high attribution scores to the target regions but low attribution scores to the non-target regions. The attribution results can accurately reflect the attribution of each activation of the feature map to the outputs. Subsequently, we take into consideration both the polarity and magnitude of the attribution results and weight each feature accordingly. Finally, we model the generation of the camouflage texture as an optimization problem and optimize in the opposite direction of object detection. Comprehensive experiments confirm that our proposed MFA outperforms state-of-the-art methods.

In summary, our main contributions list as follows.

1. We leverage the location and category information of the detectors outputs to assign attribution scores to features, capturing critical multi-layer object-aware features of target objects.
2. We propose taking multi-layer features as attack object and disrupt them at the pixel level, improving the attacking ability and transferability of attacks.
3. Extensive experiments demonstrate the superior attacking ability and transferability of adversarial examples generated by the proposed MFA compared to state-of-the-art physical attack methods.

2 RELATED WORK

Physical Attacks on Object Detection Physical attack aims to generate adversarial perturbations by modifying the visual characteristics of the real object in the physical world. A simple method is adversarial patch[Brown et al., 2017], which is often stuck to a planar object. Chen et al. [2019b] generated a planar adversarial stop sign to fool detectors. Thys et al. [2019] trained a printable patch that can successfully hide a person from a person detector. Huang et al. [2020] adopted a set of transformations to generate adversarial camouflage for non-rigid or non-flat objects. However, these methods can only attack at certain viewing angles. The more recent approaches involve manipulating the color texture patterns of target 3D objects. Zhang et al. [2019] proposed CAMOU to hide vehicles from detectors by training a clone network that simulates applying camouflage to vehicles. Wu et al. [2020a] generated an adversarial patch and then repeated and enlarged the patches until they covered the vehicle surface. Besides, there is a rising trend of leveraging differentiable neural renderers for adversarial camouflage generation. Wang et al. [2021a] proposed DAS to generate natural adversarial camouflage using a neural renderer[Kato et al., 2018] by suppressing the model and human attention. Suryanto et al. [2022] proposed DTA to learn the expected transformation of a rendered object, which can gain both the advantages of the various physical-world transformations and white-box access. Wang et al. [2022b] and Duan et al. [2022] achieved more robust attacks in multi-view, long-distance, and partial occlusions situations by utilizing a renderer to generate full-coverage camouflage texture.

Feature-level Attacks Since the most critical features are shared among different DNNs[Ganeshan et al., 2019, Wu et al., 2020b], feature-level attacks have shown promise in synthesizing more transferable adversarial samples. Zhou et al. [2018] maximized the feature distance between clean images and adversarial examples in the intermediate layers. Huang et al. [2019] improved the transferability of black-box attacks by increasing the perturbation strength of the feature layer. Ganeshan et al. [2019] used the average activation values to distinguish the positive and negative polarity of the feature. Wang et al. [2021b] introduced aggregated gradients to suppress the model-specific features and preserve important features. Zhang et al. [2022] weighed neurons using neuron attribution, considering the importance of different neurons. The above feature-level attacks are digital attacks for classification tasks and are difficult to implement in the physical world.

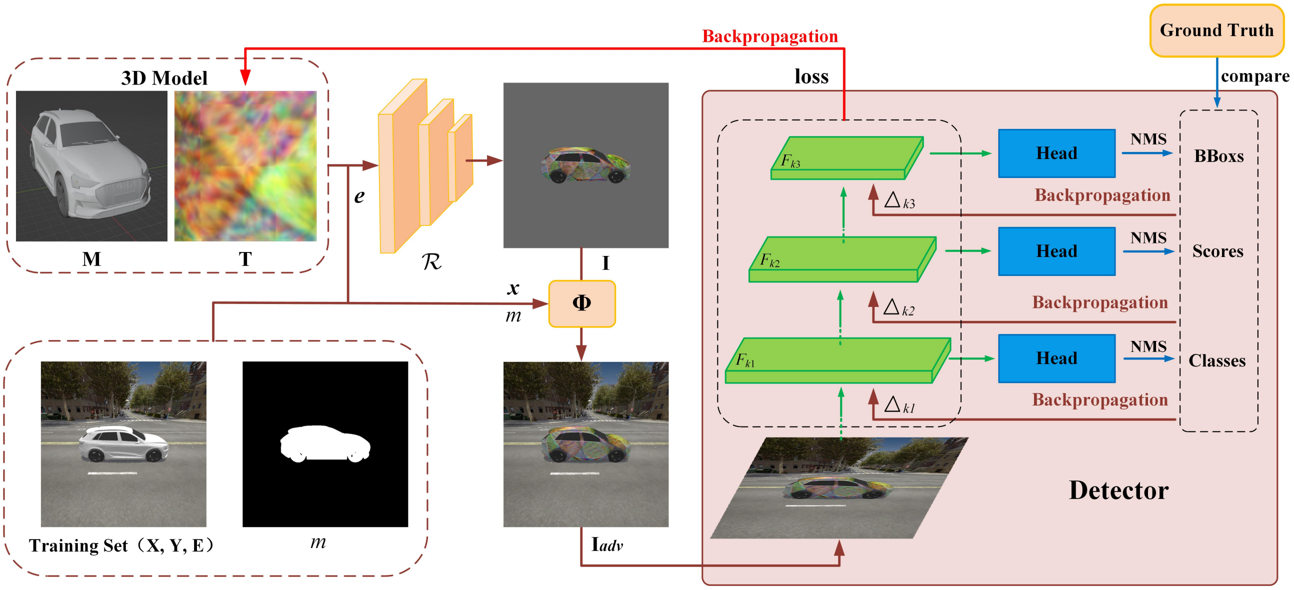


Figure 2: Overview of MFA. Given a training set(**X**, **Y**, **E**) with corresponding binary mask m and a 3D vehicle model (**M**, **T**). The camouflaged vehicle image **I** is the rendered result with environmental condition e from a renderer $\mathcal{R}$. Next, we use a physical transformation function $\mathbf{\Phi}$ to transfer the camouflaged vehicle into the different physical scenarios and feed it into the detector. Then, we can obtain the importance of multi-layer features by backpropagating the ultimate outputs filtered by post-processing (NMS and comparison with ground truth). Finally, the adversarial camouflage is updated through backpropagation with our devised loss function.

Feature Attribution Methods Feature attribution methods are popular in interpretable machine learning[Zhou et al., 2022]. These methods accept model inputs and assign attribution scores to input features based on the feature's contribution to the model outputs. There is no consensus on the definition of "attribution". In various works, the notion of attribution has been defined as sensitivity[Simonyan et al., 2013], relevance[Bach et al., 2015], local influence[Ribeiro et al., 2016], Shapley values[Lundberg and Lee, 2017], or filter activations[Selvaraju et al., 2017].

3 APPROACH

3.1 PRELIMINARIES

Given a training set (**X**, **Y**, **E**), where **X**, **Y**, and **E** are the sampled images, the ground truth and the sampling environmental condition (e.g., transformation and location, etc.), let (**M**, **T**) denote a 3D real object with a mesh tensor **M** and a texture tensor **T**, The image **I** is the rendered result of the real object (**M**, **T**) with environmental condition $e \in \mathbf{E}$ from a renderer $\mathcal{R}$ by $\mathbf{I} = \mathcal{R}((\mathbf{M}, \mathbf{T}), e)$. To perform physical attacks, we use a transformation function $\mathbf{\Phi}$ to transfer the rendered image to different environment scenarios. The physical transformation will be discussed in depth in section 3.3. Then, we generate the input image of the detector $\mathbf{I}_{adv} = \mathbf{\Phi}(\mathcal{R}((\mathbf{M}, \mathbf{T}_{adv}), e), m, x)$ by replacing the original **T** with an adversarial texture tensor $\mathbf{T}_{adv}$. Now we can obtain the detector's outputs $O = \mathcal{F}(\mathbf{I}_{adv}; \theta_f)$, where $\mathcal{F}$ is the detector with parameters θ_f. Take the yolov3 detector for example, each anchor point in the output grid contains a vector $[x_{\text{offset}}, y_{\text{offset}}, w, h, p_{\text{obj}}, p_{\text{cls1}}, \cdots, p_{\text{clsn}}]$ with bounding boxes containing different aspect ratios. x_{offset} and y_{offset} are the positions of the center of the bounding box compared to the current anchor point, w and h are the width and height of the bounding box, p_{obj} is the probability that this anchor point contains an object, and p_{cls1} through p_{clsn} is the class probability score of the object.

Our attack object is to generate the adversarial camouflage texture, which can be painted on the surface of the 3D object and hide the object from being detected. We treat the adversarial texture generation as an optimization problem, and our objective function is expressed as follows

$$\arg\max_{\mathbf{T}_{adv}} J(\mathcal{F}(\mathbf{\Phi}(\mathcal{R}((\mathbf{M}, \mathbf{T}_{adv}), e); \theta_f), \mathbf{Y}) \qquad (1)$$

where $J(,)$ measures the distance between ground truth and predicted results of the model. We can obtain the adversarial camouflage texture by solving the above optimization problem.

3.2 MULTI-LAYER FEATURE-AWARE ATTACK

The key to craft feature-level attacks is to find a proper way of measuring the importance of each feature. Let $\mathcal{F}$ denote the source model with parameters θ_f. The feature map from the k-th layer is expressed as $\mathcal{F}_k(\mathbf{I}_{adv};\theta_f)$ for the input image $\mathbf{I}_{adv}$. Since the attribution scores reflect how the features contribute to the final decision, an intuitive strategy is obtaining the gradient commonly used in feature attribution methods[Selvaraju et al., 2017]. So the attribution scores as written in the following.

$$\Delta_k^{\mathbf{I}_{adv}} = \frac{\partial \mathcal{P}(O, y)}{\partial \mathcal{F}_k(\mathbf{I}_{adv};\theta_f)} \tag{2}$$

where y is the ground truth and $\mathcal{P}(,)$ is post-processing which includes NMS[Neubeck and Van Gool, 2006] and comparison with ground truth.

The original outputs O contain many more predicted bounding boxes than the actual number of targets. Attributing all predicted bounding boxes is not meaningful due to the high redundancy between them and would result in extensive computation consumption. Therefore, we employ NMS to eliminate redundant predicted bounding boxes and then compare them with the ground truth to filter out the attacked objects.

Utilizing the aforementioned attribution scores $\Delta_k^{\mathbf{I}_{adv}}$ as the measurement to weight each feature, reflecting the feature real influence on the output, we design the loss function to guide the generation of the adversarial camouflage texture. Intuitively, the essential features will yield relatively higher intensity, indicating the efforts of correcting the features to approach the true label, and the sign provides the correcting direction. In the object detection task, the positive will be corrected in the positive direction and the negative will be corrected in the negative direction. The objective of generating transferable adversarial examples is exactly the opposite of the correction direction of the object detection task. In other words, we aim to guide the positive to be corrected in the negative direction and the negative to be corrected in the positive direction. Therefore, our objective function should be designed to manipulate the features in the opposite direction of the object detector's correction direction. Therefore, our attack loss function can be written as

$$L_{adv} = \sum^{H_k} \sum^{W_k} |\Delta_k^{\mathbf{I}_{adv}} \odot \mathcal{F}_k(\mathbf{I}_{adv};\theta_f)| \tag{3}$$

Where $\odot$ means pixel-wise multiplication, and H_k and W_k denote the height and width of the k-th layer feature map.

Additionally, empirical studies from most DNN-based detectors have shown that low-level features have high resolution and contain more location and detail information, and high-level features have a lower resolution but more robust semantic information[Lin et al., 2017]. Detectors often use multi-scale features to achieve better performance. Therefore, adversarial attacks should also consider the destruction of multi-scale features. To sum up, the Eq. 3 can be rewritten as

$$L_{adv} = \sum_{k \in \mathrm{K}} \sum^{H_k} \sum^{W_k} |\Delta_k^{\mathbf{I}_{adv}} \odot \mathcal{F}_k(\mathbf{I}_{adv};\theta_f)| \tag{4}$$

Where K is the set of target feature layers to attack.

To suppress high-frequency noise to ensure the smoothness of the the generated adversarial camouflage, we utilize the smooth loss[Mahendran and Vedaldi, 2015] to reduce the the difference square between adjacent pixels. For a rendered vehicle image $\mathbf{I}$, the calculation of smooth loss L_{smooth} can be written as

$$L_{smooth} = \sum_{i,j} (x_{i,j} - x_{i+1,j})^2 + (x_{i,j} - x_{i,j+1})^2 \tag{5}$$

where $x_{i,j}$ is the pixel value of image at coordinate (i, j).

3.3 PHYSICAL TRANSFORMATION

To bridge the gap between the digital and physical world, we follow the transformation function $\mathbf{\Phi}$ of [Wang et al., 2022b] to transfer the rendered vehicle to different environment scenarios. However, we discovered that despite preserving the location and rotation information during the sampling stage of the photo-realistic images, the rendered vehicle cannot fully cover the vehicle in the sampled image, resulting in an unnatural appearance. As shown in Figure 3a, the upper outline of the vehicle has a unnatural black edge, and the lower outline is not fully displayed. To address this issue, we introduce a simple but effective method. Specifically, the binary mask is obtained by segmentation from the original photo-realistic image, so we extract the outline of the vehicle in the mask and rendered image, scale and shift the rendered images to align the car's outline, and then feather the binary mask for softer boundaries and more realistic transformation. The visualization of our physical transformation can be seen in Figure 3b.

$$\mathbf{I}_{adv} = \mathbf{\Phi}(\mathbf{I}, m, x) = m \cdot \mathcal{T}(\mathbf{I}) + (1 - m) \cdot x \tag{6}$$

where $\mathcal{T}$ represents the scaling, translation and other operations performed on the rendered image.

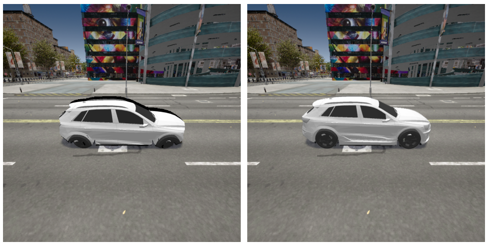

(a) FCA (b) Ours

Figure 3: The result of different physical transformation.

Algorithm 1 Multi-layer Feature-aware Attack (MFA)

```
Input: training set(X, Y, E) with corresponding binary mask $m$, 3D object model(M, T), neural renderer $\mathcal{R}$, physical transformation function $\mathbf{\Phi}$, object detector $\mathcal{F}$
Output: adversarial texture $\mathbf{T}_{adv}$
Initial $\mathbf{T}_{adv}$ with random noise $\mathbf{T}_0 \sim U(0,1)$
for $i = 1$ to $maxiteration$ do
    select minibatch sample $(x, y, e)$ from training set(X, Y, E)
    $\mathbf{I} \leftarrow \mathcal{R}((\mathbf{M}, \mathbf{T}_{adv}), e)$
    $\mathbf{I}_{adv} \leftarrow \mathbf{\Phi}\ (\mathbf{I}, m, x)$
    $O \leftarrow \mathcal{F}(\mathbf{I}_{adv}; \theta_f)$
    calculate $\Delta_k^{\mathbf{I}_{adv}}$ by Eq. 2
    calculate $L_{smooth}$ and $L_{adv}$ by Eq. 4, 5
    optimize the $\mathbf{T}_{adv}$ by Eq. 7
end for
```

3.4 OVERALL OPTIMIZATION PROCESS

Overall, we generate the adversarial camouflage by jointly optimizing the multi-layer feature attack loss L_{adv} and smooth loss L_{smooth}. substitute the Eq. 4 and Eq. 5 into Eq. 1, we get the proposed objective for MFA.

$$\underset{\mathbf{T}_{adv}}{\arg\min}\, L_{adv} + \lambda L_{smooth} \tag{7}$$

where λ controls the contribution of the term L_{smooth}. The overall training algorithm for the generation of adversarial camouflage can be described as Algorithm 1.

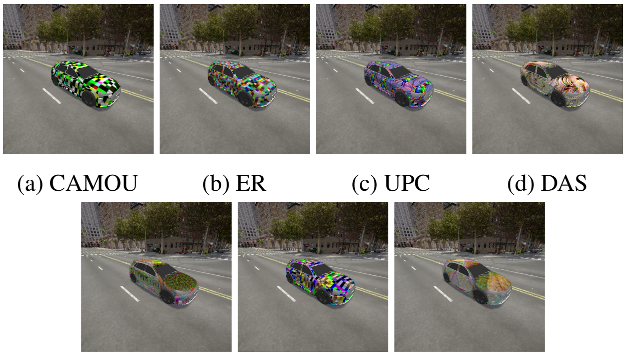
4 EXPERIMENTS

4.1 EXPERIMENTAL SETTINGS

Datasets To compare with previous works, we use the same dataset provided by Wang et al. [2021a], which were sampled from CARLA[Dosovitskiy et al., 2017], a prevalent opensource simulator for autonomous driving research. The CARLA simulator provides a variety of high-fidelity digital scenarios (e.g., modern urban) based on Unreal Engine 4. The training set consists of 12,500 high-resolution images, and the testing set has 3,000 high-resolution images sampled from different angles and distances. The dataset also provides corresponding masks of the vehicle targets for the training and testing set.

Evaluation Metrics To evaluate the performance of our proposed method, we select the commonly used Attack Success Rate (ASR)[Wu et al., 2020c] as our first evaluation metric, which is defined as the percentage of the target vehicles detected before perturbation and not detected or falsely detected after perturbation. Further, we average the attack success rate of multiple models and called it the *mean Attack Success Rate* (mASR) to better evaluate the cross-model transferability. In addition, we adopt the P@0.5 following [Duan et al., 2022, Wang et al., 2022b] as our second evaluation metric, which is defined as the percentage of the correctly detected when the detection IoU threshold is set to 0.5.

Compared methods We choose several state-of-the-art works in the 3D attack and physical attack literature, including CAMOU[Zhang et al., 2019], ER[Wu et al., 2020a], UPC[Huang et al., 2020], DAS[Wang et al., 2021a], FCA[Wang et al., 2022b], and DTA[Suryanto et al., 2022]. Note that UPC and DAS paint the adversarial camouflage only on the part of the vehicle model. In order to fairly compare, we reimplement them with full-coverage camouflage. The adversarial examples of different methods as shown in Figure 4



(a) CAMOU (b) ER (c) UPC (d) DAS

(e) FCA (f) DTA (g) MFA

Figure 4: Adversarial examples of different methods

Target models We select ten different commonly used model architectures for experiments. Specifically, SSD[Liu et al., 2016], Faster RCNN[Ren et al., 2015], Mask RCNN[He et al., 2017], Cornernet[Law and Deng, 2018], FCOS[Tian et al., 2019], Swin Transformer[Liu et al., 2021], TOOD[Feng et al., 2021], VFNet[Zhang et al., 2021], yolov5[1] and yolov7[Wang et al., 2022a]. In our experiments, all models are the official implementation version of MMDetection[Chen et al., 2019a], except for yolov5 [1] and yolov7[2].

Implementation details We train adversarial camouflage texture on the yolov3[Redmon and Farhadi, 2018]. All experiments are under black-box settings. We adopt an Adam optimizer with a learning rate of 0.01. We empirically set the $\lambda = 10^{-4}$ and a maximum of 5 epochs. The other hyperparameters are set as provided by the yolov3 implementation. We conduct the experiment on an NVIDIA RTX 1080Ti 12GB GPU, and all codes are implemented in PyTorch. For all the models, we use the pre-trained version on COCO.

[1] https://github.com/ultralytics/yolov5

[2] https://github.com/WongKinYiu/yolov7

Table 1: The comparison result of adversarial attacks in the digital space.

Method	ASR(%)										
	SSD	Faster	Mask	Corner	FCOS	Swin	TOOD	VFNet	yolov5	yolov7	**mASR**
UPC	49.11	71.93	56.77	42.88	63.11	42.76	55.67	45.60	40.91	26.40	49.51
DTA	44.32	82.08	72.21	42.20	71.16	47.84	69.48	52.65	37.91	31.09	55.09
ER	45.68	88.18	63.92	48.39	60.89	43.84	74.89	65.03	44.17	43.06	57.81
CAMOU	49.76	81.72	76.02	47.61	72.45	49.75	70.51	60.81	49.10	38.88	59.66
DAS	90.89	87.00	78.26	61.01	82.35	64.56	81.23	81.60	46.19	51.36	72.45
FCA	86.98	75.77	81.13	62.77	88.29	71.71	73.37	64.28	75.30	72.29	75.19
MFA	**96.39**	**92.69**	**92.98**	**85.62**	**98.86**	**87.81**	**94.01**	**82.64**	**95.99**	**89.96**	**91.7**

Table 2: The comparison result of adversarial attacks in the physical space.

Method	P@0.5(%)										
	SSD	Faster	Mask	Corner	FCOS	Swin	TOOD	VFNet	yolov5	yolov7	**Average**
RAW	90.28	99.31	100.00	92.36	98.61	97.22	97.92	100.00	93.75	99.31	96.88
CAMOU	70.14	33.33	69.44	68.06	31.94	70.83	72.22	77.78	56.25	71.53	62.15
DAS	47.22	48.61	54.86	34.72	32.64	70.14	66.67	61.11	57.64	65.28	53.89
FCA	45.14	47.92	56.94	36.11	22.22	54.86	71.53	64.58	48.61	57.64	50.56
MFA	**22.92**	**24.31**	**36.81**	**22.22**	**9.03**	**40.28**	**46.53**	**47.92**	**34.72**	**43.06**	**32.78**

4.2 DIGITAL WORLD ATTACK

In this section, we evaluate the performance of our generated adversarial camouflages on the vehicle in the digital world under black-box settings. We report the ASR for the detection of the target vehicle. More experimental results can be found in the Supplementary Material.

The comparison results are outlined in Table 1. Our adversarial camouflage outperforms other methods across all the detectors. Specifically, our adversarial camouflage achieves the highest mASR at **91.7%**, and the ASR of each detector exceeds 80%. Six detectors (SSD, Faster-RCNN, Mask-RCNN, FCOS, TOOD and yolov5) are easily vulnerable by our proposed MFA with ASR surpassing 90%. The ASRs of the other four detectors (Cornernet, Swin Transformer, VFNet and yolov7) range between 80% and 90%, which may be due to the special design that makes it more robust against adversarial attacks on object detection. For example, the backbone of Cornernet comes from the Hourglass Network of pose estimation, and the backbone of Swin Transformer is a novel vision Transformer.

In addition, our proposed MFA improves the mASR by **19.25%** against DAS, indicating that our attack can more accurately capture inherent conducive characteristics of objects and successfully paralyze the vehicle detection system. The mASR of MFA is **16.51%** higher than that of FCA, which suggests that attacking intermediate features is more transferable than directly attacking the final output layer.

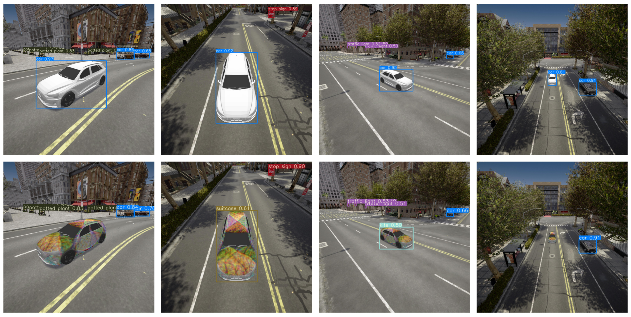


Figure 5: The detection result of the vehicle before and after our attack in the digital world.

We provide some adversarial camouflage vehicle examples in different scenarios. As illustrated in Figure 5, we select yolov7 as the detector, the vehicle before painted with adversarial camouflage is correctly detected as a car with high detection confidence. However, after being painted with our adversarial camouflage texture, the target vehicle turns out to be incorrectly detected or undetected, while the vehicles not painted with our adversarial camouflage texture are correctly detected.

4.3 PHYSICAL WORLD ATTACK

As for the physical world attack, we conduct several experiments to validate the practical effectiveness of our generated adversarial camouflage. Because it is difficult to guarantee that all other elements except the adversarial camouflages are preserved consistently before and after the attack, We report the P@0.5 for the detection of the target vehicle.

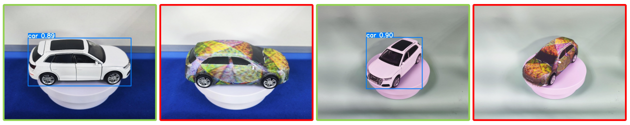


Figure 6: The detection result of the toy cars before and after our attack in the physical world.

For simplicity, we compare three attack methods that are more robust in digital adversarial attacks (i.e., CAMOU, DAS, FCA). Due to the limitation of funds and conditions, we follow Wang et al. [2021a] and Wang et al. [2022b] to print adversarial camouflages by a Xerox Color 550 printer and crop the camouflage parts, then stick them on a 1:32 scale model of an Audi Q5 with different backgrounds to mimic the real car painting in the physical world. To show the efficiency of our adversarial camouflage under various scenarios, we captured 144 pictures of the painted car in different settings (i.e., 8 directions {left, right, front, back and their corresponding intersection directions}, 2 angles {0ř and 45ř}, 3 distances {long, middle, and short distance} and 3 surroundings) with a Xiaomi 12S phone. The visualization of our generated adversarial camouflages can be found in Figure 6.

The experiment results are shown in Table 2. Each detector correctly detects almost all raw toy cars, with their P@0.5 over 90%. Compared with other methods, the MFA shows competitively transferable attacking ability in the physical world. Its average P@0.5 is the lowest at **32.78%**, significantly better than the compared baselines (e.g., 62.15% on CAMOU, 53.89% on DAS, and 50.56% on FCA, respectively). VFNet is the most robust against adversarial attacks, and Swin Transformer, TOOD and yolov7 also exhibit strong robustness. The conclusion is consistent with the results for digital attacks except for TOOD. TOOD is more robust in the physical world than in the digital world which is worth further study. Besides, FCOS is the most vulnerable with a maximum drop of 89.58%. This may be because FCOS is an anchor-free, one-stage model with a relatively simple structure.

We provide some detection result examples of attacking toy cars in the physical world on yolov7. As shown in Figure 6, the toy cars painted with our adversarial camouflage texture are hidden and undetected.

To sum up, the experimental results demonstrate that our adversarial camouflages have strong transferable attacking ability in the physical world.

4.4 EFFECT OF DIFFERENT LAYER FEATURE TO ATTACK

We take different feature layers as attack objects and observe the attack effect of various layer features. The source model yolov3 has three detection layers, which are called low layer, medium layer and high layer in this paper for convenience.

First, we compare the impact of the attack detection layer and the non-detection layer. We take the previous layer of the yolov3's detection layers as the non-detection layer to carry out comparative experiments. The results are shown in Table 3. For the adjacent detection layer and non-detection layer, the mASR of detection layer is higher than that of non-detection layer. The main reason may be that the detection layers fuse different features, which is more conducive to object detection. So we select the detection layers as the attack target in the rest of this paper.

Furthermore, we evaluate single-layer and multi-layer feature attacks. As shown in Figure 7, multi-layer attacks significantly outperform single-layer attacks (e.g., For the average mASR, 76.24% on a single layer, 84.98% on two layers and 91.7% on three layers, respectively). On the other hand, The attack on the middle layer is better than on the low or high layer. The same conclusion can be drawn from Table 3. The reason might be that low layers have not learned salient features and semantic concepts, and high layers are model-specific and it is easily to get trapped in soure model local optimum. By contrast, middle layers have well-separated representations and they are not highly correlated to the model architecture.

Table 3: The mASR performance of MFA under different target layer settings.

low		middle		high	
non-Det	Det	non-Det	Det	non-Det	Det
57.07	79.78	75.28	84.06	63.65	64.88

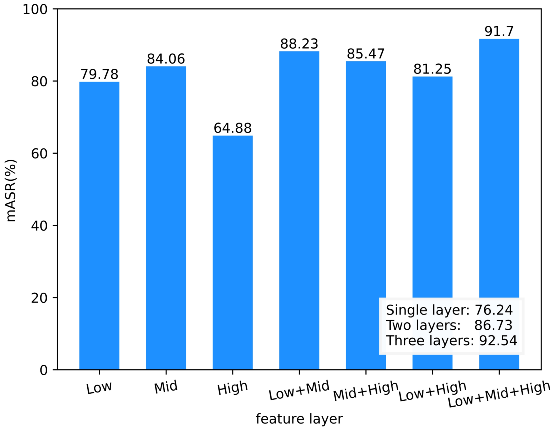


Figure 7: The mASR performance for attacks at different layer features.

4.5 EFFECT OF DIFFERENT OUTPUTS FOR ATTRIBUTION

Object detection is a multi-output task, and we investigate the influence of using different outputs of the object detection model for attribution in this part. As mentioned in Section 3.1, there are two scores for each anchor point in yolov3: the object score, which can reflect the location information, and the class score, which is the probability of the most likely category of the object.

Figure 8 shows the results of attribution using different outputs. The OBJ-CLS approach utilizes the product of object and class scores for attribution, while OBJ only uses the object score and CLS only the class score. The MFA using both object and class scores for attribution yields better results than using either object or class score alone for each model. Specifically, the ASR of OBJ-CLS is higher than that of CLS or OBJ for almost every detector, and the mASR is 91.7% for OBJ-CLS, 83.92% for CLS, and 84.64% for OBJ. This confirms our earlier analysis that only using category information to attribute will cannot accurately assign attribution scores to features. The object-aware/important features can be captured by using category and location information to attribute, guiding the generation of more transferable adversarial camouflage.

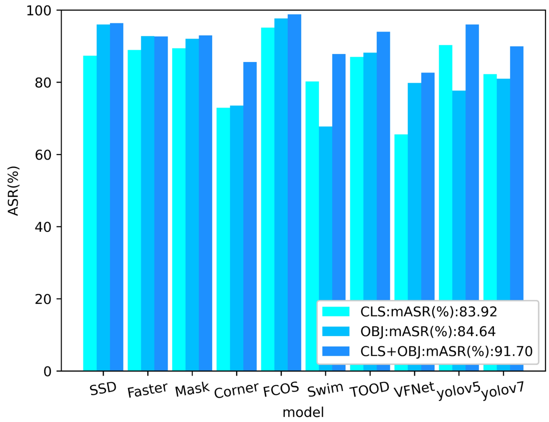


Figure 8: The mASR performance of attribution using different outputs.

4.6 EFFECT OF HYPERPARAMETERS

In this section, we conduct several experiments to further investigate the effect of loss function items and the confidence thresholds of NMS.

The effect of hyper-parameter λ The hyper-parameter λ controls the contribution of the term L_{smooth}. As we can observe from Table 4, When λ is between 0 and 10^{-4}, L_{adv} dominates the optimization direction and and L_{smooth} is negligible, resulting in strong attack ability. In particular, the highest mASR is achieved 91.83% when $\lambda = 10^{-5}$, but the adversarial camouflage appears unnatural. As λ continues to increase, L_{smooth} will dominate the optimization direction , thus reducing the attacking ability(e.g., 78.64% when $\lambda = 10^{-3}$, 53.27% when $\lambda = 10^{-2}$, respectively).

Table 4: The mASR performance for hyper-parameter λ

λ	0	10^{-5}	10^{-4}	10^{-3}	10^{-2}
mASR	90.98	**91.83**	91.70	78.64	53.27

Table 5: The mASR performance for thresholds of NMS

threshold	0.05	0.15	0.25	0.35	0.45
mASR	90.46	89.21	**91.70**	86.53	85.70

The effect of NMS confidence thresholds As mentioned in Section 3.2, the NMS will be used to filter the outputs to remove redundant predicted bounding boxes. We compare the effect of different confidence thresholds of NMS on attacks in this part. As we can observe from Table 4, The mASR performance is optimal when the threshold is 0.25, which is the default threshold of yolov3. When the threshold increases, the mASR will decrease, primarily due to the exclusion of certain targets. For instance, when the threshold is set to 0.35, targets with confidence scores between 0.25 and 0.35 are discarded, even though they are meaningful positive targets. Conversely, when the threshold decreases, the mASR decreases slightly, possibly due to the introduction of negative targets that disrupt the optimization direction.

4.7 INTERPRETABILITY OF THE ADVERSARIAL CAMOUFLAGE

In this part, we adopt model attention visualization to conduct qualitative analysis to further validate our MFA attack's effectiveness. The regions the models pay attention to can be deemed discriminative. Because the vehicle with adversarial camouflage texture will not be detected correctly when it is sent to the detector, which leads to the attention maps of the vehicle cannot be obtained, we follow [Wang et al., 2021a] and [Wang et al., 2022b] to generate the attention maps of the vehicle with different viewpoints on ResNet50[He et al., 2016] model by the commonly used model-agnostic attention maps technique[Selvaraju et al., 2017]. Figure 9 shows the original vehicle, virtual adversarial vehicle, and their attention maps for the "car" class label. We can observe that the MFA attack distracts the attention maps from the vehicle body to other uncamouflaged regions, suggesting that the model's decision evidence has been changed.

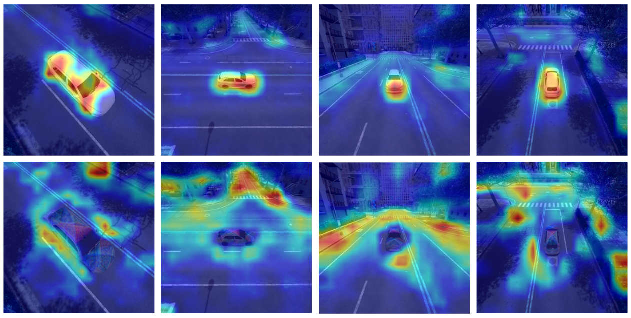


Figure 9: The detection result of the vehicle under different view angles before and after our attack in the digital space.

5 CONCLUSION AND FUTURE WORK

In this paper, we investigate the problem of generating robust adversarial examples in the physical world for object detectors. We propose the Multi-layer Feature-aware Attack(MFA) method, which improves the attacking ability and transferability of adversarial attacks by distorting important features at different layers. Specifically, we first use location and category information to assign attribution scores to different feature layers and utilize their amplitude and polarity to weight each feature. Finally, we optimize the generation problem of the camouflage texture in the opposite direction of the object detection. Comprehensive experiments confirm the superiority of our method.

In the future, we are interested in investigating the attack abilities of our adversarial camouflage using a real vehicle in a real-world scenario, we could paint our camouflage on a real-world vehicle by projection or 3D printing. Additionally, we would also like to investigate the appearance of our generated camouflage to be more visually unsuspicious and natural.

References

Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *International conference on machine learning*, pages 284–293. PMLR, 2018.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017.

Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019a.

Shang-Tse Chen, Cory Cornelius, Jason Martin, and Duen Horng Chau. Shapeshifter: Robust physical adversarial attack on faster r-cnn object detector. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2018, Dublin, Ireland, September 10–14, 2018, Proceedings, Part I 18*, pages 52–68. Springer, 2019b.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017.

Andrew Du, Bo Chen, Tat-Jun Chin, Yee Wei Law, Michele Sasdelli, Ramesh Rajasegaran, and Dillon Campbell. Physical adversarial attacks on an aerial imagery object detector. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1796–1806, 2022.

Yexin Duan, Jialin Chen, Xingyu Zhou, Junhua Zou, Zhengyun He, Jin Zhang, Wu Zhang, and Zhisong Pan. Learning coated adversarial camouflages for object detectors. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 891–897, 2022.

Chengjian Feng, Yujie Zhong, Yu Gao, Matthew R Scott, and Weilin Huang. Tood: Task-aligned one-stage object detection. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3490–3499. IEEE Computer Society, 2021.

Aditya Ganeshan, Vivek BS, and R Venkatesh Babu. Fda: Feature disruptive attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8069–8079, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

Lifeng Huang, Chengying Gao, Yuyin Zhou, Cihang Xie, Alan L Yuille, Changqing Zou, and Ning Liu. Universal physical camouflage attacks on object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 720–729, 2020.

Qian Huang, Isay Katsman, Horace He, Zeqi Gu, Serge Belongie, and Ser-Nam Lim. Enhancing adversarial example transferability with an intermediate level attack. In

Proceedings of the IEEE/CVF international conference on computer vision, pages 4733–4742, 2019.

Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3907–3916, 2018.

Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*, pages 734–750, 2018.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

Jiajun Lu, Hussein Sibai, and Evan Fabry. Adversarial examples that fool detectors. *arXiv preprint arXiv:1712.02494*, 2017.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5188–5196, 2015.

Alexander Neubeck and Luc Van Gool. Efficient non-maximum suppression. 3:850–855, 2006.

Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Naufal Suryanto, Yongsu Kim, Hyoeun Kang, Harashta Tatimma Larasati, Youngyeo Yun, Thi-Thu-Huong Le, Hunmin Yang, Se-Yoon Oh, and Howon Kim. Dta: Physical camouflage attacks using differentiable transformation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15305–15314, 2022.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Simen Thys, Wiebe Van Ranst, and Toon Goedemé. Fooling automated surveillance cameras: adversarial patches to attack person detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 0–0, 2019.

Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019.

Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696*, 2022a.

Donghua Wang, Tingsong Jiang, Jialiang Sun, Weien Zhou, Zhiqiang Gong, Xiaoya Zhang, Wen Yao, and Xiaoqian Chen. Fca: Learning a 3d full-coverage vehicle camouflage for multi-view physical adversarial attack. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2414–2422, 2022b.

Jiakai Wang, Aishan Liu, Zixin Yin, Shunchang Liu, Shiyu Tang, and Xianglong Liu. Dual attention suppression attack: Generate adversarial camouflage in physical world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8565–8574, 2021a.

Zhibo Wang, Hengchang Guo, Zhifei Zhang, Wenxin Liu, Zhan Qin, and Kui Ren. Feature importance-aware transferable adversarial attacks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7639–7648, 2021b.

Tong Wu, Xuefei Ning, Wenshuo Li, Ranran Huang, Huazhong Yang, and Yu Wang. Physical adversarial attack on vehicle detector in the carla simulator. *arXiv preprint arXiv:2007.16118*, 2020a.

Weibin Wu, Yuxin Su, Xixian Chen, Shenglin Zhao, Irwin King, Michael R Lyu, and Yu-Wing Tai. Boosting the transferability of adversarial samples via attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1161–1170, 2020b.

Zuxuan Wu, Ser-Nam Lim, Larry S Davis, and Tom Goldstein. Making an invisibility cloak: Real world adversarial attacks on object detectors. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 1–17. Springer, 2020c.

Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2730–2739, 2019.

Haoyang Zhang, Ying Wang, Feras Dayoub, and Niko Sunderhauf. Varifocalnet: An iou-aware dense object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8514–8523, 2021.

Jianping Zhang, Weibin Wu, Jen-tse Huang, Yizhan Huang, Wenxuan Wang, Yuxin Su, and Michael R Lyu. Improving adversarial transferability via neuron attribution-based attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14993–15002, 2022.

Yang Zhang, Hassan Foroosh, Philip David, and Boqing Gong. Camou: Learning physical vehicle camouflages to adversarially attack detectors in the wild. In *International Conference on Learning Representations*, 2019.

Wen Zhou, Xin Hou, Yongjun Chen, Mengyun Tang, Xiangqi Huang, Xiang Gan, and Yong Yang. Transferable adversarial perturbations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 452–467, 2018.

Yilun Zhou, Serena Booth, Marco Tulio Ribeiro, and Julie Shah. Do feature attribution methods correctly attribute features? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 9623–9633, 2022.