Defending from Physically-Realizable Adversarial Attacks through Internal Over-Activation Analysis

Giulio Rossolini, Federico Nesti, Fabio Brau, Alessandro Biondi and Giorgio Buttazzo

Department of Excellence in Robotics and AI, Scuola Superiore Sant'Anna, Pisa, Italy
{giulio.rossolini, federico.nesti, fabio.brau, alessandro.biondi, giorgio.buttazzo}@santannapisa.it

Abstract

This work presents *Z-Mask*, an effective and deterministic strategy to improve the adversarial robustness of convolutional networks against *physically-realizable* adversarial attacks. The presented defense relies on specific *Z-score* analysis performed on the internal network features to detect and mask the pixels corresponding to adversarial objects in the input image. To this end, spatially contiguous activations are examined in shallow and deep layers to suggest potential adversarial regions. Such proposals are then aggregated through a multi-thresholding mechanism. The effectiveness of *Z-Mask* is evaluated with an extensive set of experiments carried out on models for semantic segmentation and object detection. The evaluation is performed with both digital patches added to the input images and printed patches in the real world. The results confirm that *Z-Mask* outperforms the state-of-the-art methods in terms of detection accuracy and overall performance of the networks under attack. Furthermore, *Z-Mask* preserves its robustness against defense-aware attacks, making it suitable for safe and secure AI applications.

Introduction

Nowadays, deep neural networks (DNNs) yield impressive performance in computer vision tasks such as semantic segmentation (SS) and object detection (OD). These remarkable results have encouraged the use of deep learning models also in *cyber-physical systems* (CPS) as autonomous cars. However, the trustworthiness of neural networks is often questioned by the existence of adversarial attacks (Huang et al. 2020), especially those performed in the physical world (Athalye et al. 2018; Wu et al. 2020; Rossolini et al. 2022; Braunegg et al. 2020; Kong et al. 2020), which are most relevant to CPS. Such attacks are usually crafted by means of adversarial objects, most often in the form of patches (Brown et al. 2018), which are capable of corrupting the model outcome when processed as a part of the input image and can also be printed to perform *physically-realizable* attacks.

To defend DNNs from these adversarial objects, several techniques were proposed in the literature based on specialized learning modules or robust training. However, such approaches are often expensive, do not transfer well in realistic scenarios, and are still susceptible to specific attacks.

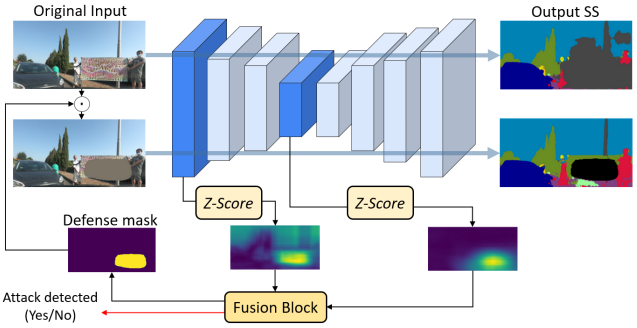


Figure 1: Illustration of the proposed approach.

Differently from those strategies, the defense method proposed in this paper leverages the evidence that physical adversarial attacks yield anomalous activation patterns in the internal network layers. Such anomalous activations caused by adversarial patches have been observed in several works (Yu et al. 2021; Co et al. 2021; Rossolini et al. 2022), but to the best of our records, no studies deepen this phenomenon from a spatial perspective. To this end, recalling the spatial propagation effect of CNNs (Krizhevsky, Sutskever, and Hinton 2012), we noticed that a set of shallow layers contains high/medium over-activations in the spatial image areas corresponding to adversarial objects, while in deeper layers such over-activations grow in magnitude while referring to lower spatial resolutions (further illustrations in the supplementary material). Based on such evidences, this paper proposes *Z-Mask*, a novel defense mechanism that combines the analysis of multiple layers to precisely detect and mask potential adversarial objects.

Figure 1 illustrates the proposed defense approach for the case of SS. To extract preliminary adversarial region proposals, *Z-Mask* runs an over-activation analysis on a set of selected layers. This analysis exploits a *Spatial Pooling Refinement* (SPR) to filter out high-frequency noise in over-activated regions. For each of these layers, the analysis produces an adversarial region proposal expressed through a heatmap. Then, all the heatmaps are aggregated into a *shallow heatmap* $\mathcal{H}^S$ and a *deep heatmap* $\mathcal{H}^D$, which summarize the over-activation behavior at two different depth levels. Finally, these two heatmaps are processed by a *Fusion and De-*

tection Block that flags the presence of an adversarial object and generates the corresponding defense mask.

A set of experimental results highlights the effectiveness and the robustness of the proposed defense approach on digital and real-world scenarios, which outperforms the state-of-the-art methods both in adversarial objects detection and masking. Furthermore, the experiments show the robustness of *Z-Mask* against defense-aware attacks, which is a property inherited from the clear relation between over-activations and adversarial patches. In summary, this paper provides the following contributions:

- It proposes *Z-Mask*, a novel robust adversarial defense method designed to detect and mask the regions belonging to adversarial objects;
- It shows the effectiveness of a *Z-score*-based defense by improving a naive neuron-wise approach with a *Spatial Pooling Refinement*, which removes high-frequency noise and helps extract proper contiguous masks;
- It provides an activation-aware patch optimization to strengthen the relation between over-activations and adversarial effects induced from physical attacks.

The remainder of the paper is organized as follows: first, it introduces the related work, then it presents the *Z-Mask* pipeline, it reports the experimental results, and finally states the conclusions.

Related Work

Physical adversarial attacks. Adversarial attacks are widely studied methods capable of easily fooling neural models by adding imperceptible input perturbations (Nakka and Salzmann 2020; Metzen et al. 2017; Xie et al. 2017; Szegedy et al. 2014; Rony et al. 2019; Brau et al. 2022). However, in recent years, particular interest has been devoted to adversarial attacks aimed at controlling the output of DNNs through physical adversarial objects or patches.

In this context, Athalye et al. (2018) presented the Expectation Over Transformations (EOT) paradigm, which allows crafting adversarial objects robust against real-world transformations, as scaling, translation, orientation, and illumination changes. Later, Brown et al. (2018) proposed an attack method based on adversarial patches, which achieved great success as a means to study the real-world robustness of DNNs and generate new effective physical attacks (Braunegg et al. 2020; Wu et al. 2020; Nesti et al. 2022; Lee and Kolter 2019; Hu et al. 2021).

Defense methods. To tackle the problem of physical attacks and digital adversarial patches, several defense methods have been proposed in the literature. For the sake of clarity, we divide defense methods in two main categories: *adversarial training* and *external tools*. The former aims at making a model more robust by re-training the network including attacked images and regularization terms (Saha et al. 2020; Metzen, Finnie, and Hutmacher 2021; Rao, Stutz, and Schiele 2020; Wu, Tong, and Vorobeychik 2020). These approaches significantly increase the training and testing efforts (especially when dealing with adversarial patches).

Conversely, the methods based on external tools preserve the original model parameters and complement the model output with additional information that typically consists in an attack detection flag (Co et al. 2021; Rossolini et al. 2022; Xiang and Mittal 2021; Xu, Yu, and Chen 2020) and/or defense masks (Chiang, Chan, and Wu 2021; Naseer, Khan, and Porikli 2019; Chou, Tramer, and Pellegrino 2020; Liu et al. 2022; Xiang et al. 2021; Zhou et al. 2020) that remove the adversarial parts of the image. Although all such methods do not alter the model parameters, only a few of them are task-agnostic (e.g., capable of working for both OD and SS models) or address comprehensive evaluations on large datasets and realistic scenarios.

The role of internal activations. Among the large plethora of methods that study the internal behavior of DNNs under adversarial perturbations, some works (Rossolini, Biondi, and Buttazzo 2022; Yu et al. 2021) noticed that adversarial patches cause large activations in the internal network layers. In particular, (Co et al. 2021; Rossolini et al. 2022) exploited this fact to detect adversarial patches by computing the cumulative sum of the neurons activation in a certain layer. Such a score is deemed as *safe* or *unsafe* by comparing it to a threshold. Although this approach achieves good performance in detecting adversarial patches, it is applied to a single layer only using a neuron-wise over-activation analysis, which may leave room for effective defense-aware attacks. Furthermore, it is limited to detection purposes only, without addressing the fact through a spatial analysis.

This work faces the over-activation phenomenon also from a spatial perspective to derive an effective and straightforward defense that performs a multi-layer and a multi-neuron analysis. First, a *spatial pooling refinement* based on the *Z-Score* values is introduced in the analysis, which helps better identify the image regions that cause the over-activations. Second, shallow and deep analysis are combined to generate an aggregated defense mask. These steps make *Z-Mask* a fully task-agnostic defense that outputs both a precise pixel mask and an attack detection flag. It works on top of any pretrained convolutional model in the context of a large-scale evaluation that also targets realistic attacks (i.e., physically-printed patches) and preserves its robustness against defense-aware attacks.

Proposed Defense

This section presents the *Z-Mask* defense strategy, which is formulated to be task agnostic, i.e., applicable on any convolutional model. In this work, we consider the case of SS and OD models. In both cases, the input consists of an image with $H \times W$ pixels and C channels, denoted by $\mathbf{x} \in [0,1]^{C \times H \times W}$, while the form of the output $f(\mathbf{x})$ depends on the task. For a semantic segmentation model with N classes, the output $f(\mathbf{x}) \in [0,1]^{N \times H \times W}$ is an image that encodes the semantic context of each pixel. For an OD model, the output $f(\mathbf{x})$ is a tensor encoding the class and the bounding box of each detected object. Without loss of generality, a task-specific loss function $\mathcal{L}(f(\mathbf{x}), \mathbf{y})$ is used to

quantify the quality of a prediction $f(\mathbf{x})$ against the ground-truth output $\mathbf{y}$.

A real-world adversarial attack can be simulated by applying an *adversarial patch* in a specific region of the input image $\mathbf{x}$. A patch $\boldsymbol{\delta}$ is a $\tilde{H} \times \tilde{W}$ image within C channels, where $\tilde{H} \leq H$ and $\tilde{W} \leq W$. Crafting an adversarial patch requires solving an optimization problem that aims at minimizing a specific attack loss function while making patch features more robust against real-world transformations in the input image (Athalye et al. 2018).

In detail, given an input image $\mathbf{x}$ and a patch $\boldsymbol{\delta}$, an additional function γ is randomly sampled from a set $\boldsymbol{\Gamma}$ of compositions of appearance-changing and placement transformations. The appearance-changing transformations include brightness, contrast change and noise addition; the patch placement transformations include random translation and scaling for defining the position of the patch in the image. Then, a patch $\boldsymbol{\delta}$ is applied to $\mathbf{x}$, according to γ, through a patch application function $g_\gamma(\mathbf{x}, \boldsymbol{\delta})$. Formally, an adversarial patch $\hat{\boldsymbol{\delta}}$ can be crafted by solving the following optimization:

$$\hat{\boldsymbol{\delta}} = \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \; \mathbb{E}_{\mathbf{x}\sim\mathbf{X},\gamma\sim\boldsymbol{\Gamma}} \, \mathcal{L}_{Adv}(f(g_\gamma(\mathbf{x}, \boldsymbol{\delta})), \mathbf{y}_{Adv}), \quad (1)$$

where $\mathbf{X}$ is a set of known inputs, $\mathbf{y}_{Adv}$ is the adversarial target, and $\mathcal{L}_{Adv}$ is the adversarial loss that specifies the objective of the attacker. In the case of untargeted attacks, the adversarial target is the regular ground truth $\mathbf{y}$ and the adversarial loss function is $-\mathcal{L}(f(\tilde{\mathbf{x}}), \mathbf{y})$, to maximize the task-specific loss. To enhance the physical realizability of the patches, the adversarial loss includes additional terms that are described in the supplementary material.

A defense masking strategy obscures a portion of the input image (supposedly containing the adversarial patch) through a pixel-wise product $\odot$ with a binary mask having the same size of the image. Formally, for each perturbed image $\tilde{\mathbf{x}} = g_\gamma(\mathbf{x}, \hat{\delta})$, a binary mask $M(\tilde{\mathbf{x}})$ is computed with the intent of satisfying the following property:

$$\mathcal{L}(f(\tilde{\mathbf{x}} \odot M(\tilde{\mathbf{x}})), \mathbf{y}) \approx \mathcal{L}(f(\mathbf{x}), \mathbf{y}). \quad (2)$$

Equation (2) states that the objective of a masking defense is to mitigate the effectiveness of a physical adversarial perturbation while preserving a correct behavior outside the region of the mask. In this work, Mask $M(\tilde{\mathbf{x}})$ is generated by leveraging multiple over-activation analysis, which are then aggregated through a *Fusion Block* mechanism.

Layer-Wise Over-Activation Analysis

Let $\boldsymbol{h}^{(l)} \in \mathbb{R}^{C^{(l)} \times H^{(l)} \times W^{(l)}}$ be the output features of layer l, obtained during the forward pass of $f(\mathbf{x})$, where $H^{(l)}$ and $W^{(l)}$ are its spatial dimensions. The heatmap $\mathcal{H}^{(l)}$ is obtained by applying the following operations to $\boldsymbol{h}^{(l)}$ (illustrated in Figure 2). First, for a layer l, the channel-wise *Z-score* $\boldsymbol{z}^{(l)} = \frac{\boldsymbol{h}^{(l)} - \mu^{(l)}}{\sigma^{(l)}}$ of $\boldsymbol{h}^{(l)}$ is computed, where $\mu^{(l)}$ and $\sigma^{(l)}$ are the channel-wise mean and standard deviation of the output features, respectively, obtained from a dataset $\boldsymbol{X}$ that does not include attacked images. The *Z-score* is then

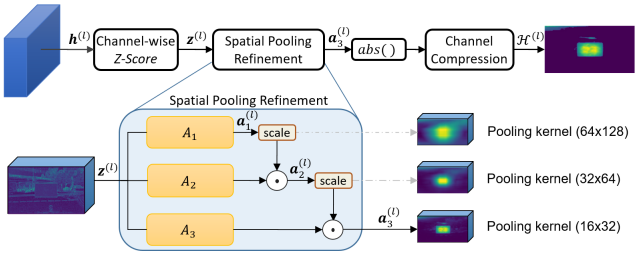


Figure 2: Over-activation pipeline performed by *Z-Mask* on a given layer with $m = 3$ Average-Pooling stages. The *scale* blocks refer to the ∞-norm used in Equation 3. Resizing operations are omitted in the figure.

processed in cascade by a sequence of m Average-Pooling operations $(A_1, \ldots, A_i, \ldots A_m)$ as follows:

$$\begin{cases} \boldsymbol{a}_i^{(l)} = \mathcal{R}(A_i(\mathcal{R}(\boldsymbol{z}^{(l)}))) \odot \frac{\boldsymbol{a}_{i-1}^{(l)}}{\|\boldsymbol{a}_{i-1}^{(l)}\|_\infty}, \quad i = 1, \ldots, m \\ \boldsymbol{a}_0^{(l)} \equiv 1, \end{cases} \quad (3)$$

where each A_i has kernel size k_i and $\mathcal{R}$ is an operator that resizes (by interpolation) the spatial dimensions of a given tensor to a configurable size $H^{\mathcal{R}} \times W^{\mathcal{R}}$. Note that the i^{th} kernel is larger than the $(i+1)^{\text{th}}$ one. Also, the resize operation is performed before and after each A_i to enable the use of the pixel-wise product and the same sequence of Average-Pooling operations on different network layers.

The rationale for using such Average-Pooling operations is the following. Observe that the *Z-score* itself provides a pixel-wise metric capable of highlighting the over-activated pixels (i.e., pixels with internal activation values that are significantly far from $\mu^{(l)}$ in terms of $\sigma^{(l)}$). However, since we aim at masking adversarial patches, we are interested in highlighting *contiguous* over-activated portions of the image rather than spurious over-activated pixels (i.e., pixels whose neighbors have activation values close to $\mu^{(l)}$). To do that, the SPR implements a cascade filtering (Satti, Sharma, and Garg 2020) that reduces the effects of spurious over-activated pixels. The process is iteratively refined: first larger kernels identify macro-regions that include over-activated contiguous pixels and then smaller kernels refine the analysis within such macro-regions. Finally, to obtain the desired heatmap $\mathcal{H}^{(l)}$ (of size $1 \times H^{\mathcal{R}} \times W^{\mathcal{R}}$), the absolute values of $\boldsymbol{a}_m^{(l)}$ are averaged across the channels. As shown in the experimental section, this process yields a heatmap of the over-activated region with sharper areas (Figure 5).

Fusion and Detection Mechanism

This section explains how the mask $M(\mathbf{x})$ is generated by merging the information of two sets of heatmaps, $\mathcal{S}$ and $\mathcal{D}$. The set $\mathcal{S}$ contains $N_{\mathcal{S}}$ heatmaps belonging to the selected shallow layers only, while $\mathcal{D}$ contains $N_{\mathcal{D}}$ heatmaps belonging to deeper layers and possibly to shallow layers. Leveraging these sets of heatmaps, we reduce the analysis to two aggregated heatmaps $\mathcal{H}^{\mathcal{S}} = \mathcal{F}(\mathcal{S})$ and $\mathcal{H}^{\mathcal{D}} = \mathcal{F}(\mathcal{D})$, where $\mathcal{F}(\cdot)$ is an operator that merges multiple heatmaps belong-

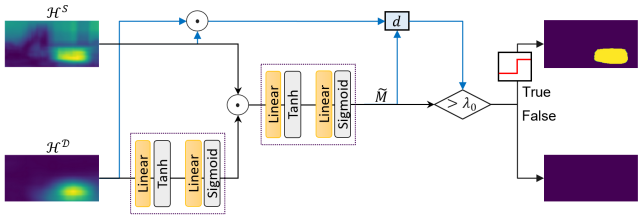


Figure 3: Fusion and Detection Block.

ing to a given set. In practice, a pixel-wise *max* function is used for $\mathcal{F}(\cdot)$. $\mathcal{H}^{\mathcal{S}}$ and $\mathcal{H}^{\mathcal{D}}$ summarize the over-activation behavior at different depths in the model: $\mathcal{H}^{\mathcal{S}}$ represents the over-activated regions in the shallow layers, while $\mathcal{H}^{\mathcal{D}}$ takes into consideration also deep layers. The reason for using these two heatmaps emerged after a series of experimental observations. From a practical perspective, $\mathcal{H}^{\mathcal{S}}$ allows highlighting the over-activated portions of the image (i.e., the regions that may contain adversarial objects): it provides a high spatial accuracy, but a limited capability of discriminating adversarial and non-adversarial regions. Conversely, $\mathcal{H}^{\mathcal{D}}$ provides a high accuracy in identifying adversarial over-activations, but with a much lower spatial accuracy. In fact, experiments showed that over-activations coming from non-adversarial regions do not propagate their effect to deeper layers (a more detailed analysis of this effect is provided in the supplementary material). Hence, $\mathcal{H}^{\mathcal{D}}$ can be used to filter out the regions highlighted by $\mathcal{H}^{\mathcal{S}}$ that are not adversarial, yielding a more accurate heatmap.

Figure 3 illustrates the operations performed by the Fusion and Detection Block. The merging process leverages two *soft-thresholding blocks*. The first block extracts a region of interest from $\mathcal{H}^{\mathcal{D}}$, which is then multiplied by $\mathcal{H}^{\mathcal{S}}$ to pose attention only to over-activated areas in the deeper layers. The second block extracts $\tilde{M}$, a soft version of the final mask with real pixel values in $[0, 1]$. Each *soft-thresholding block* consists of two sequential linear layers (both with 1-dimensional weight and bias), activated by a *tanh* and a *sigmoid* function, respectively.

Finally, to apply the masking only when an adversarial region is detected, we measure the over-activation as $d = \frac{\|\mathcal{H}^{\mathcal{S}} \odot \mathcal{H}^{\mathcal{D}} \odot \tilde{M}\|_1}{\|\tilde{M}\|_1}$ and compute the mask $M(\mathbf{x})$ as follows:

$$M(\mathbf{x}) = \begin{cases} 1 - \hbar(\tilde{M}), & d > \lambda_0 \\ 1, & \text{otherwise,} \end{cases} \tag{4}$$

where $\hbar$ is the Heaviside function centered in 0.5 and λ_0 is a given threshold. The soft-thresholding parameters (eight in total) are fitted by supervised learning, while the threshold λ_0 is configured through an ROC analysis. The main motivation of using this module is its high deterministic behavior, since it mimics a soft-threshold operation in a differentiable manner. This allows testing against gradient-based defense-aware attacks and offers a transparent robustness by constraining an attacker to reduce the over-activation values to fool the defense (see experimental part).

Experimental Evaluation

This section presents a set of experiments carried out on several convolutional models for OD and SS to evaluate the effectiveness of the proposed defense. All the experiments were implemented using PyTorch (Paszke et al. 2019) on a server with 8 NVIDIA-A100 GPUs. For both SS and OD tasks, the effectiveness of an adversarial attack was measured by evaluating the drop of the model performance with a task-dependent metric. For SS models, the mIoU was used on the subset of pixels not belonging to the applied patch, as done by (Rossolini et al. 2022). For OD models, the performance was measured by the COCO mAP.

Datasets and Models. Three state-of-the-art models were selected for the SS task: ICNet (Zhao et al. 2018), DDRNet (Hong et al. 2021), and BiseNet (Yu et al. 2018), using pre-trained weights provided by their authors. For the OD task, SSD (Liu et al. 2016), RetinaNet (Lin et al. 2017), and Faster R-CNN (Ren et al. 2017) were selected from the PyTorch model zoo. More details are in the supplementary material.

Several datasets were used for the experiments. The Cityscapes dataset (Cordts et al. 2016) is a canonical dataset of driving images for SS. It contains 2975 and 500 1024×2048 images for training and validation, respectively. For OD, we considered the COCO 2017 dataset (Lin et al. 2014), containing 112k and 5k images for training and validation, respectively. Being COCO a dataset of common images, pictures have different sizes, hence a network-specific resizing is required. To assess the proposed approach on real-world scenarios, we considered APRICOT (Braunegg et al. 2020), which is a COCO-like dataset including more than 1000 images, each containing a physical adversarial patch for one between Faster R-CNN, RetinaNet, and SSD.

Attack and defense strategies. Different attack methodologies were used to craft adversarial patches. For SS models, we leveraged the untargeted attack pipeline used in (Rossolini et al. 2022), while, for OD models, we performed an untargeted attack on the classes, similarly to (Chen et al. 2019). The patches contained in the APRICOT dataset rely on a false-detection attack (Braunegg et al. 2020). More details are provided in the supplementary material.

Concerning defense strategies, we compared *Z-Mask* against different approaches for both adversarial pixel masking and detection. For the masking task, we re-implemented the Local Gradient Smoothing method (LGS) (Naseer, Khan, and Porikli 2019) and MaskNet (Chiang, Chan, and Wu 2021), both with the original settings described by the authors. For the adversarial detection, we considered for comparisons FPDA (Rossolini et al. 2022) and HN (Co et al. 2021). Details are in the supplementary material.

Activation-aware patch optimization. We crafted adversarial patches while controlling the over-activation to understand its relation with the induced adversarial effect, as well train the defense to properly scale into real-world scenarios. To do that, we proposed the following optimization:

$$\hat{\boldsymbol{\delta}}_\beta = \underset{\boldsymbol{\delta}}{\operatorname{argmin}}\, \mathbb{E}_{\mathbf{x}\sim\mathbf{X},\gamma\sim\boldsymbol{\Gamma}} \big[(1-\beta)\cdot \mathcal{L}_{OZ}(f, g_\gamma(\mathbf{x}, \boldsymbol{\delta})) + \beta \cdot \mathcal{L}_{Adv}(f(g_\gamma(\mathbf{x}, \boldsymbol{\delta})), \mathbf{y})\big], \tag{5}$$

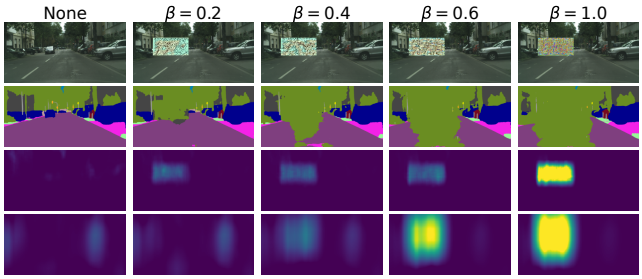


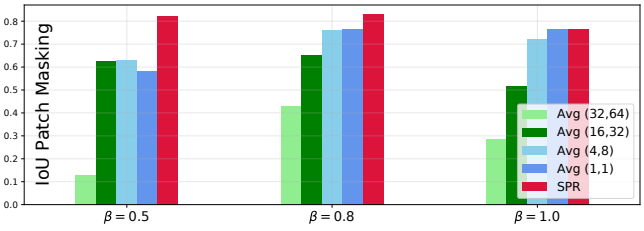
Figure 4: Visualization of the predictions and SPR heatmaps obtained from several β activation-aware patches. The rows report x, $f(x)$, $\mathcal{H}^{\mathcal{S}}$ and $\mathcal{H}^{\mathcal{D}}$, respectively.

where $\beta \in [0, 1]$ is a control parameter and $\mathcal{L}_{OZ}$ is a loss function that measures the magnitude of over-activation of internal layers (details are available in the supplementary material). The rationale behind this optimization problem is that a low value of β reduces the importance assigned to the adversarial effect, while forcing the attack to generate less over-activation in the internal layers, hence simulating real-world patches. Figure 4 illustrates the over-activation of these patches (computed with the SPR) both in shallow and deep layers, remarking the relation with the induced adversarial effect. Furthermore, Figure 7 (discussed later) provides a measure of the adversarial effect as a function of β.

***Z-Mask* settings and training.** For SS models, the heatmaps in $\mathcal{S}$ were generated with a SPR composed of four pooling operations, with kernel sizes $k_1 = (64, 128)$, $k_2 = (32, 64)$, $k_3 = (16, 32)$, $k_4 = (8, 16)$. Instead, the heatmaps in $\mathcal{D}$ were generated using two pooling operations with kernel sizes $k_1 = (64, 128)$, $k_2 = (32, 64)$. After each pooling operation, the heatmaps were resized to $(H^{\mathcal{R}} \times W^{\mathcal{R}}) = (150 \times 300)$. Please note that all the resulting heatmaps have a 1:2 aspect ratio, keeping the same aspect ratio of the input images. For OD models, the SPR used $k_1 = (40, 40)$, $k_2 = (25, 25)$, $k_3 = (10, 10)$ to build $\mathcal{S}$, and $k_1 = (80, 80)$, $k_2 = (40, 40)$ to build $\mathcal{D}$. The resizing dimension was set to (400×500). For all the tests, pooling operations were applied with stride 1. These kernel settings were motivated by extensive preliminary tests performed to analyse the internal activations. To illustrate the benefits of the SPR, Figure 5 provides the results of ablation studies by comparing the performance of the Fusion and Detection block with different pooling settings and patches crafted with different β. The SPR block always achieves a better IoU Patch Masking, which is computed as the IoU between the predicted mask and its ground truth.

The description of the layers selected for extracting $\mathcal{D}$ and $\mathcal{S}$ in each model is reported in the supplementary material.

The parameters of the *soft-thresholding* operations inside the Fusion and Detection block were trained in a supervised fashion by considering a set of patches crafted with Equation 5 and minimizing the pixel-wise binary cross-entropy loss $\mathcal{L}_{\text{BCE}}(\tilde{M}, \bar{M})$, where $\bar{M}$ is the ground-truth binary mask. To this end, we collected a set of adversarial patches $\Delta = \{\hat{\boldsymbol{\delta}}_\beta : \beta \in [\beta_0, 1]\}$, where we set $\beta_0 = 0.5$ to avoid generating



(a) IoU Patch Masking comparison

(b) No Avg (1,1) (c) Avg (16,32) (d) SPR

Figure 5: Ablation studies on the masking accuracy with different pooling strategies on 100 images of the Cityscapes validation set and BiseNet (a). Bottom figures are the predicted masks of a same input, using a patch with $\beta = 0.5$.

patches with scarce adversarial effect. These patches were used to craft the set $\tilde{\boldsymbol{X}}$, which was obtained by adding the patches in Δ to each image of $\boldsymbol{X}$. Set $\tilde{\boldsymbol{X}}$ was used to train the Fusion and Detection Block and make it robust to a wide spectrum of over-activations.

In our tests, $\boldsymbol{X}$ contained 500 images randomly sampled from the original training dataset. The ADAM optimizer (Kingma and Ba 2015) was used for this purpose, with a learning-rate of 0.01 and training for 15 epochs. The channel-wise std and mean of each selected layer was computed on a different subset of the training set containing 500 clean (i.e., non-patched) images. The detection threshold λ_0 was deduced after the soft-thresholding training as the *cut-off* threshold of the ROC curve. The ROC was generated by computing the measure d on each input of a dataset, including the clean set $\boldsymbol{X}$ and the patched set $\tilde{\boldsymbol{X}}$, labeled as negative and positive samples, respectively.

Evaluation for Digital Attacks

Masking performance. The benefits of the proposed defense mechanism were evaluated by attacking the validation sets with different adversarial patch sizes. For Cityscapes, we used patches with size 600x300 (L), 400x200 (M) and 300x150 (S) pixels, whereas for COCO, due to the different image aspect ratio, we used 200x200 (L), 150x150 (M), and 100x100 (S). Also, an L-size random patch was evaluated to test the case in which a portion of the image is occluded without the intent of generating an adversarial attack.

As shown in Table 1, *Z-Mask* outperformed the other defense strategies, achieving scores similar to the random case, when tested against adversarial attacks, and close to the original model without applying patches, meaning that does not affect the nominal model performance. Figure 6 illustrates the benefits of *Z-Mask*: attacked areas are identified and covered without affecting other portions.

Detection performance. All the adversarial patches evaluated in Table 1 were perfectly detected by both *Z-Mask*,

Net	Patch	Defense Method (mAP Val)			
		Z-Mask	MaskNet	LGS	None
FR-CNN	None	**0.357**	0.353	0.350	0.357
	Rand	0.301	0.295	**0.320**	0.308
	S	0.335	0.333	**0.354**	0.337
	M	**0.302**	0.289	0.246	0.140
	L	**0.300**	0.289	0.244	0.164
SSD	None	**0.253**	0.180	0.243	0.264
	Rand	**0.208**	0.132	0.198	0.215
	S	**0.237**	0.159	0.233	0.245
	M	**0.202**	0.125	0.144	0.065
	L	**0.205**	0.113	0.163	0.072
RetinaNet	None	**0.355**	0.269	0.337	0.359
	Rand	0.305	0.227	**0.312**	0.308
	S	**0.339**	0.245	0.339	0.335
	M	**0.326**	0.222	0.306	0.304
	L	**0.305**	0.212	0.297	0.283

(a)

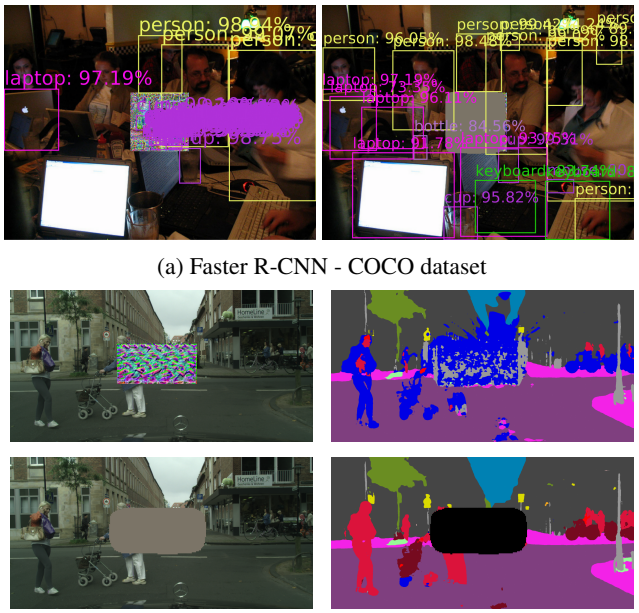
Net	Patch	Defense Method (mIoU Val)			
		Z-Mask	MaskNet	LGS	None
DDRNet	None	**0.778**	0.739	0.777	0.778
	Rand	0.731	0.710	**0.769**	0.761
	S	**0.741**	0.701	**0.741**	0.702
	M	**0.723**	0.699	0.719	0.663
	L	**0.691**	0.689	0.642	0.532
BiseNet	None	0.684	0.622	**0.685**	0.687
	Rand	0.650	0.569	**0.668**	0.653
	S	**0.663**	0.560	0.522	0.475
	M	**0.653**	0.550	0.413	0.323
	L	**0.621**	0.535	0.320	0.220
ICNet	None	**0.785**	0.783	0.782	0.785
	Rand	**0.768**	0.736	0.764	0.746
	S	**0.748**	0.737	0.657	0.625
	M	**0.729**	0.718	0.593	0.549
	L	**0.747**	0.725	0.528	0.430

(b)

Table 1: Robustness performance evaluated for different patch sizes for OD-COCO (a) and SS-Cityscapes (b).

HN, and FPDA. To better assess the performance of these adversarial detection methods, we used the optimization described in Equation (5) to generate a set of patches with a wider range of over-activation values, selecting the values of $\beta \in \{0.1, 0.2, ..., 0.9, 1.0\}$. Please note that $\beta = 1.0$ corresponds to a regular adversarial attack, while lower β values decrease the importance of adversarial effect to reduce the magnitude of over-activation. An L-sized patch was generated for each β. Figure 7 shows the detection and masking accuracy against this set of patches as a function of β for DDRNet. The top part of the figure shows the detection accuracy, evaluated using the AUC of ROC on a dataset, including both the clean and the attacked validation set (as negative and positive samples, respectively). Note that *Z-Mask* achieved better results than the other adversarial patch detectors, providing good detection performance also to patches that do not retain much adversarial effect.

The bottom part of the Figure 7 reports the performance of *Z-Mask*, MaskNet, LGS, and the original model (without defense). Again, our method achieved higher mIoU among



(a) Faster R-CNN - COCO dataset

(b) BiseNet - Cityscapes dataset

Figure 6: *Z-Mask* effects (comparison w/ and w/o defense)

almost all the β values. Similar results were obtained for other models in the supplementary material.

Evaluation for Physical Attacks

The masking and detection performance of *Z-mask* was evaluated in real-world scenarios with images containing printed adversarial patches. For this test, we adopted the same *Z-mask* settings and parameters used for digital attacks on COCO, which generalize well also for real-world patches. The detection performance was assessed with the APRICOT dataset, as positive samples, and 1000 images of the COCO validation set, as negative samples. Figure 8 (a) reports the corresponding ROC, where *Z-Mask* obtained the best AUC with respect to FPDA and HN on both RetinaNet and Faster R-CNN. The analysis on SSD was omitted since the large rescaling factor on the input image required by the pretrained network restrained APRICOT patches to just a few pixels, thus neutralizing their adversarial effect.

Figure 8 (b) illustrates the effect of *Z-Mask* on a sample of APRICOT. We also provide additional illustrations of real-world attacked datasets in the supplementary material.

Defense-Aware Attacks

Since the *Z-Mask* pipeline is fully differentiable up to $\tilde{M}$, an attacker might exploit that knowledge to craft defense-aware attacks, i.e., optimize patches that are adversarial for the model and the defense together. To this end, we propose two different defense-aware attacks.

The first attack, denoted as *Mask-Attack*, is designed to induce errors in the mask output to yield an incorrect input masking operation. This would allow the adversarial patch to pass without being masked or induce additional occlusion

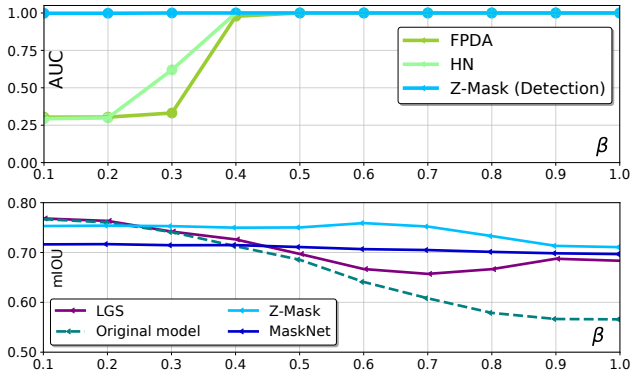


Figure 7: Comparison of the Detection accuracy (top plot) and task mIoU performance (bottom plot) using DDRNet on the validation set of Cityscapes.

in the image. This attack is obtained by solving the following problem with $\alpha \in \{0, 0.1, 0.2, \ldots, 1.0\}$:

$$\hat{\boldsymbol{\delta}}_\alpha = \underset{\boldsymbol{\delta}}{\operatorname{argmin}}\, \mathbb{E}_{\mathbf{x}\sim\mathbf{X},\gamma\sim\boldsymbol{\Gamma}}\big[(1-\alpha)\cdot(-\mathcal{L}_{BCE}(\tilde{M},\bar{M})) + \alpha\cdot\mathcal{L}_{Adv}(f(g_\gamma(\mathbf{x},\boldsymbol{\delta})),\mathbf{y})\big]. \quad (6)$$

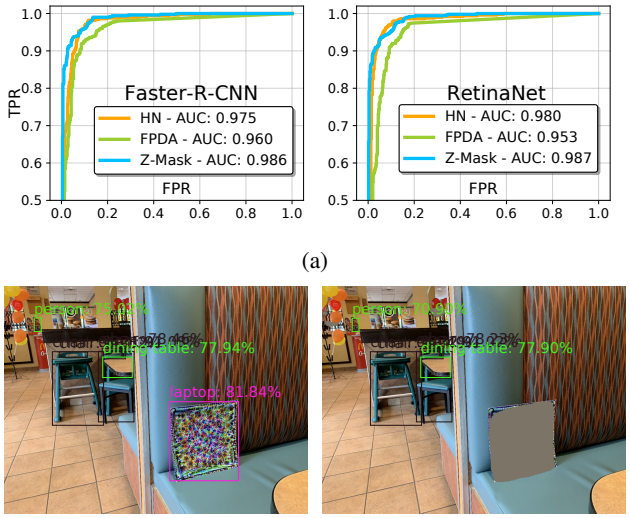
Recall that $\mathcal{L}_{\text{BCE}}(\tilde{M},\bar{M})$ is the pixel-wise binary cross-entropy loss between the defense mask $\tilde{M}$ and the ground-truth patch mask $\bar{M}$ (which is known).

A second attack formulation, denoted as *Flag-Attack*, targets the detection flag aiming at causing false negatives in the detector. This attack is performed by replacing $\mathcal{L}_{\text{BCE}}(\tilde{M},\bar{M})$ with $\mathcal{L}_{\text{BCE}}(Sigmoid(d-\lambda_0),1)$. This is done to force $d < \lambda_0$ in the optimization, hence resulting in a mask $M(\mathbf{x}) = 1$. Figure 9 shows the results of *Z-Mask* against these attacks on DDRNet (results on other networks in the supplementary material). A mask defense-aware attack was also tested on *MaskNet* to provide a comparison, while the results of LGS are not reported, since other works already addressed its weaknesses under defense-aware attacks (Chiang, Chan, and Wu 2021).

Note that, even exploiting the knowledge of the defense, the proposed attacks were not able to reduce the performance of *Z-Mask* more than what obtained for the digital evaluation, as reported in Table 1. Indeed, observe from Figure 9 that, when *Z-Mask* does not detect the attack (TPR=0), the attack is not effective (maximum mIoU). Practically speaking, the robustness of *Z-Mask* comes from the fact that it directly exploits the over-activation values. In fact, recalling that physical attacks are strictly related to over-activations, the attacker is required to reduce their magnitude to bypass the defense, thus inevitably yielding less effective attacks. Conversely, for MaskNet, certain values of α induce larger performance degradation.

Conclusions

This paper presented *Z-Mask*, a method for masking and detecting physically-realizable adversarial examples. This is accomplished by leveraging specific processing modules, such as the Spatial Pooling Refinement and the Fusion and



(a)

(b)

Figure 8: (a) ROC analysis performed on the dataset including APRICOT images and 1000 COCO images. (b) The effect of *Z-Mask* on an APRICOT image.

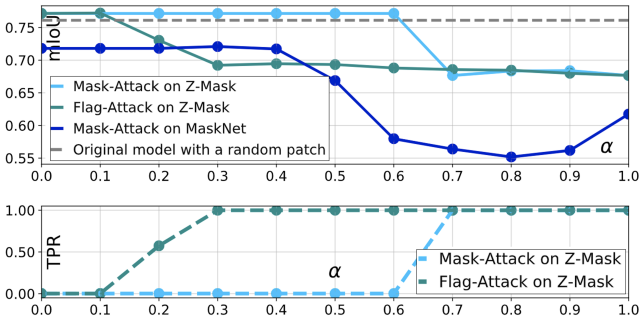


Figure 9: Evaluation and comparison of defense benefits (mIoU) and detection performance (TPR) against defense-aware attacks as a function of α. The results refer to DDRNet evaluated on the validation set of Cityscapes.

Detection Block. *Z-Mask* is task-agnostic and was tested with OD and SS models, obtaining state-of-the-art results for both adversarial masking and detection on large datasets, as COCO and Cityscapes, and in real-world scenarios. Furthermore, we strengthened the robustness of *Z-Mask* by underlining the relation between over-activation and adversarial effect through an activation-aware patch optimization.

As a future work, we plan to address an automatic selection of the shallow and deep layers involved in the over-activation analysis. Although the relation between over-activation and physical adversarial attacks is evident, it is less clear why certain model layers are more affected than others by this phenomenon. Addressing this task from a more theoretical perspective is not straightforward and requires further investigations.

References

Athalye, A.; Engstrom, L.; Ilyas, A.; and Kwok, K. 2018. Synthesizing Robust Adversarial Examples. In *Proceedings of the 35th International Conference on Machine Learning*, Proceedings of Machine Learning Research, 284–293.

Brau, F.; Rossolini, G.; Biondi, A.; and Buttazzo, G. 2022. On the Minimal Adversarial Perturbation for Deep Neural Networks With Provable Estimation Error. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–15.

Braunegg, A.; Chakraborty, A.; Krumdick, M.; Lape, N.; Leary, S.; Manville, K.; Merkhofer, E.; Strickhart, L.; and Walmer, M. 2020. Apricot: A dataset of physical adversarial attacks on object detection. In *European Conference on Computer Vision*, 35–50. Springer.

Brown, T. B.; Mané, D.; Roy, A.; Abadi, M.; and Gilmer, J. 2018. Adversarial Patch. *arXiv:1712.09665 [cs]*.

Chen, S.-T.; Cornelius, C.; Martin, J.; and Chau, D. H. P. 2019. ShapeShifter: Robust Physical Adversarial Attack on Faster R-CNN Object Detector. In *Machine Learning and Knowledge Discovery in Databases*, 52–68. Springer.

Chiang, P.-H.; Chan, C.-S.; and Wu, S.-H. 2021. Adversarial Pixel Masking: A Defense against Physical Attacks for Pre-trained Object Detectors. In *Proceedings of the 29th ACM International Conference on Multimedia*, MM '21, 1856–1865. Association for Computing Machinery.

Chou, E.; Tramer, F.; and Pellegrino, G. 2020. Sentinet: Detecting localized universal attacks against deep learning systems. In *2020 IEEE Security and Privacy Workshops (SPW)*, 48–54. IEEE.

Co, K. T.; Muñoz-González, L.; Kanthan, L.; and Lupu, E. C. 2021. Real-time Detection of Practical Universal Adversarial Perturbations. *arXiv:2105.07334*.

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *Conference on Computer Vision and Pattern Recognition CVPR*, 3213–3223. IEEE Computer Society.

Hong, Y.; Pan, H.; Sun, W.; and Jia, Y. 2021. Deep dual-resolution networks for real-time and accurate semantic segmentation of road scenes. *arXiv:2101.06085*.

Hu, Y.-C.-T.; Kung, B.-H.; Tan, D. S.; Chen, J.-C.; Hua, K.-L.; and Cheng, W.-H. 2021. Naturalistic Physical Adversarial Patch for Object Detectors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE.

Huang, X.; Kroening, D.; Ruan, W.; Sharp, J.; Sun, Y.; Thamo, E.; Wu, M.; and Yi, X. 2020. A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability. *Computer Science Review*, 37: 100270.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA*.

Kong, Z.; Guo, J.; Li, A.; and Liu, C. 2020. PhysGAN: Generating Physical-World-Resilient Adversarial Examples for Autonomous Driving. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA*, 14242–14251. IEEE.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In Pereira, F.; Burges, C.; Bottou, L.; and Weinberger, K., eds., *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

Lee, M.; and Kolter, Z. 2019. On Physical Adversarial Patches for Object Detection. *arXiv:1906.11897*.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision (ICCV)*, 2980–2988.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, 740–755. Springer.

Liu, J.; Levine, A.; Lau, C. P.; Chellappa, R.; and Feizi, S. 2022. Segment and Complete: Defending Object Detectors against Adversarial Patch Attacks with Robust Patch Detection. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; and Berg, A. C. 2016. Ssd: Single shot multibox detector. In *European conference on computer vision ECCV*, 21–37. Springer.

Metzen, J. H.; Finnie, N.; and Hutmacher, R. 2021. Meta adversarial training against universal patches. In *ICML 2021 Workshop on Adversarial Machine Learning*.

Metzen, J. H.; Kumar, M. C.; Brox, T.; and Fischer, V. 2017. Universal Adversarial Perturbations Against Semantic Image Segmentation. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy*, 2774–2783. IEEE Computer Society.

Nakka, K. K.; and Salzmann, M. 2020. Indirect Local Attacks for Context-Aware Semantic Segmentation Networks. In Vedaldi, A.; Bischof, H.; Brox, T.; and Frahm, J., eds., *16th European Conference Computer Vision ECCV, Glasgow, UK*, volume 12350, 611–628. Springer.

Naseer, M.; Khan, S.; and Porikli, F. 2019. Local Gradients Smoothing: Defense Against Localized Adversarial Attacks. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*.

Nesti, F.; Rossolini, G.; Nair, S.; Biondi, A.; and Buttazzo, G. 2022. Evaluating the Robustness of Semantic Segmentation for Autonomous Driving against Real-World Adversarial Patch Attacks. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2826–2835. IEEE Computer Society.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Kopf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.;

and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Rao, S.; Stutz, D.; and Schiele, B. 2020. Adversarial training against location-optimized adversarial patches. In *European Conference on Computer Vision ECCV*, 429–448. Springer.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6): 1137–1149.

Rony, J.; Hafemann, L. G.; Oliveira, L. S.; Ayed, I. B.; Sabourin, R.; and Granger, E. 2019. Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4322–4330.

Rossolini, G.; Biondi, A.; and Buttazzo, G. 2022. Increasing the Confidence of Deep Neural Networks by Coverage Analysis. *IEEE Transactions on Software Engineering*, 1–14.

Rossolini, G.; Nesti, F.; D'Amico, G.; Nair, S.; Biondi, A.; and Buttazzo, G. 2022. On the Real-World Adversarial Robustness of Real-Time Semantic Segmentation Models for Autonomous Driving. *arXiv:2201.01850*.

Saha, A.; Subramanya, A.; Patil, K.; and Pirsiavash, H. 2020. Role of Spatial Context in Adversarial Robustness for Object Detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 3403–3412. IEEE.

Satti, P.; Sharma, N.; and Garg, B. 2020. Min-max average pooling based filter for impulse noise removal. *IEEE Signal Processing Letters*, 27: 1475–1479.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I. J.; and Fergus, R. 2014. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada*.

Wu, T.; Tong, L.; and Vorobeychik, Y. 2020. Defending Against Physically Realizable Attacks on Image Classification. In *8th International Conference on Learning Representations ICLR*.

Wu, Z.; Lim, S.; Davis, L. S.; and Goldstein, T. 2020. Making an Invisibility Cloak: Real World Adversarial Attacks on Object Detectors. In Vedaldi, A.; Bischof, H.; Brox, T.; and Frahm, J., eds., *16th European Conference on Computer Vision ECCV, Glasgow, UK*, volume 12349, 1–17. Springer.

Xiang, C.; Bhagoji, A. N.; Sehwag, V.; and Mittal, P. 2021. {PatchGuard}: A Provably Robust Defense against Adversarial Patches via Small Receptive Fields and Masking. In *30th USENIX Security Symposium (USENIX Security 21)*, 2237–2254.

Xiang, C.; and Mittal, P. 2021. DetectorGuard: Provably Securing Object Detectors against Localized Patch Hiding Attacks. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 3177–3196. New York, NY, USA: Association for Computing Machinery.

Xie, C.; Wang, J.; Zhang, Z.; Zhou, Y.; Xie, L.; and Yuille, A. L. 2017. Adversarial Examples for Semantic Segmentation and Object Detection. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy*, 1378–1387. IEEE Computer Society.

Xu, Z.; Yu, F.; and Chen, X. 2020. LanCe: A comprehensive and lightweight CNN defense methodology against physical adversarial attacks on embedded multimedia applications. In *2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 470–475. IEEE.

Yu, C.; Chen, J.; Xue, Y.; Liu, Y.; Wan, W.; Bao, J.; and Ma, H. 2021. Defending Against Universal Adversarial Patches by Clipping Feature Norms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 16434–16442.

Yu, C.; Wang, J.; Peng, C.; Gao, C.; Yu, G.; and Sang, N. 2018. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 325–341. Springer.

Zhao, H.; Qi, X.; Shen, X.; Shi, J.; and Jia, J. 2018. Icnet for real-time semantic segmentation on high-resolution images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 405–420. Springer.

Zhou, G.; Gao, H.; Chen, P.; Liu, J.; Dai, J.; Han, J.; and Li, R. 2020. Information Distribution Based Defense Against Physical Attacks on Object Detection. In *2020 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, 1–6.