2024 ACM/IEEE 15th International Conference on Cyber-Physical Systems (ICCPS) | 979-8-3503-6927-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICCPS61052.2024.00009

Attention-Based Real-Time Defenses for Physical Adversarial Attacks in Vision Applications

Giulio Rossolini
Dept. of Excellence in Robotics & AI
Scuola Superiore Sant'Anna
Pisa, Italy
giulio.rossolini@santannapisa.it

Alessandro Biondi
Dept. of Excellence in Robotics & AI
Scuola Superiore Sant'Anna
Pisa, Italy
alessandro.biondi@santannapisa.it

Giorgio Buttazzo
Dept. of Excellence in Robotics & AI
Scuola Superiore Sant'Anna
Pisa, Italy
giorgio.buttazzo@santannapisa.it

***Abstract*—Deep neural networks exhibit excellent performance in computer vision tasks, but their vulnerability to real-world adversarial attacks, achieved through physical objects that can corrupt their predictions, raises serious security concerns for their application in safety-critical domains. Existing defense methods focus on single-frame analysis and are characterized by high computational costs that limit their applicability in multi-frame scenarios, where real-time decisions are crucial.**

To address this problem, this paper proposes an efficient attention-based defense mechanism that exploits adversarial channel-attention to quickly identify and track malicious objects in shallow network layers and mask their adversarial effects in a multi-frame setting. This work advances the state of the art by enhancing existing over-activation techniques for real-world adversarial attacks to make them usable in real-time applications. It also introduces an efficient multi-frame defense framework, validating its efficacy through extensive experiments aimed at evaluating both defense performance and computational cost.

***Index Terms*—adversarial attacks, real-world adversarial defense, neural network monitoring, robust and secure AI**

I. Introduction

In recent years, *deep neural networks* (DNNs) have demonstrated remarkable performance in several computer vision tasks. At the same time, they have been shown to be quite vulnerable to adversarial attacks [1], where small perturbations of input data can cause a model to output wrong predictions. To address this problem, an increased research effort has been devoted to make DNNs more reliable, robust, and secure, to be adopted in *cyber-physical systems* (CPS), as autonomous vehicles and robots [2]–[4].

Although adversarial perturbations represent a concrete security threat for DNNs, they raised significant discussions in the CPS community, mainly questioning the practical relevance of these attacks. It is indeed not entirely realistic to consider threat models in which the attacker has access to the digital representation of the frames captured by a vision system, to run adversarial attacks against DNNs, while not having the capability of compromising other software components in the system that could be even easier to attack. In response to this argument, research efforts have been shifted towards *real-world adversarial attacks* [5], which are deployed through *physical* objects, such as billboards and patches, that are specifically crafted and strategically placed in the external environment to fool DNNs [6], [7].

To enhance the robustness of DNNs against such real-world adversarial attacks, various techniques have been proposed in the literature (discussed in Section II). A common paradigm that can be found in previous work consists in producing at run-time a mask to cover adversarial objects, thereby preserving the predictions of the DNNs under attack.

Although recent defense methods have shown a promising performance to contrast such types of attacks, even on complex real-world scenarios, previous work mainly focused on single-image (i.e., single-frame) cases and without paying particular attention at the computational cost of the proposed defense method, resulting in more inference passes or additional expensive neural models. These limitations make state-of-the-art approaches inadequate for CPS, where efficient solutions capable to operate in real time on video streams (i.e., multiple frames) are required.

This work. To face these challenges, we take inspiration from recent studies [10], [11] that assess strong and provable connections between anomalous over-activations in convolutional network layers and real-world adversarial effects on the model output. In particular, this work delves deep into understanding the over-activation phenomenon by observing the presence of specific channels even in the first layers of DNNs, which are predominantly targeted by the real-world attack for propagating adversarial effects. We systematically identified this attack pattern through channel-wise weights, denoted as *adversarial trace*, that enable a significantly faster and more accurate identification of attacks by means of a proper attention strategy. This allows for the immediate removal of adversarial features before their spatial propagation in the deep layers, hence detecting and masking attacks in a single inference pass.

After presenting the results of our analysis and providing insights into the nature of the adversarial trace, we propose a defense algorithm for multi-frame vision applications named *Adversarial-Channel Attention Tracing* (ACAT). To enable the efficient tracing of adversarial physical objects in a video stream, ACAT requires to know a starting spatial position of the objects, which can be extracted using a single inference pass of state-of-the-art single-frame defense methods. As witnessed by the experimental results reported in the paper,

979-8-3503-6927-4/24/$31.00 ©2024 IEEE
DOI 10.1109/ICCPS61052.2024.00009

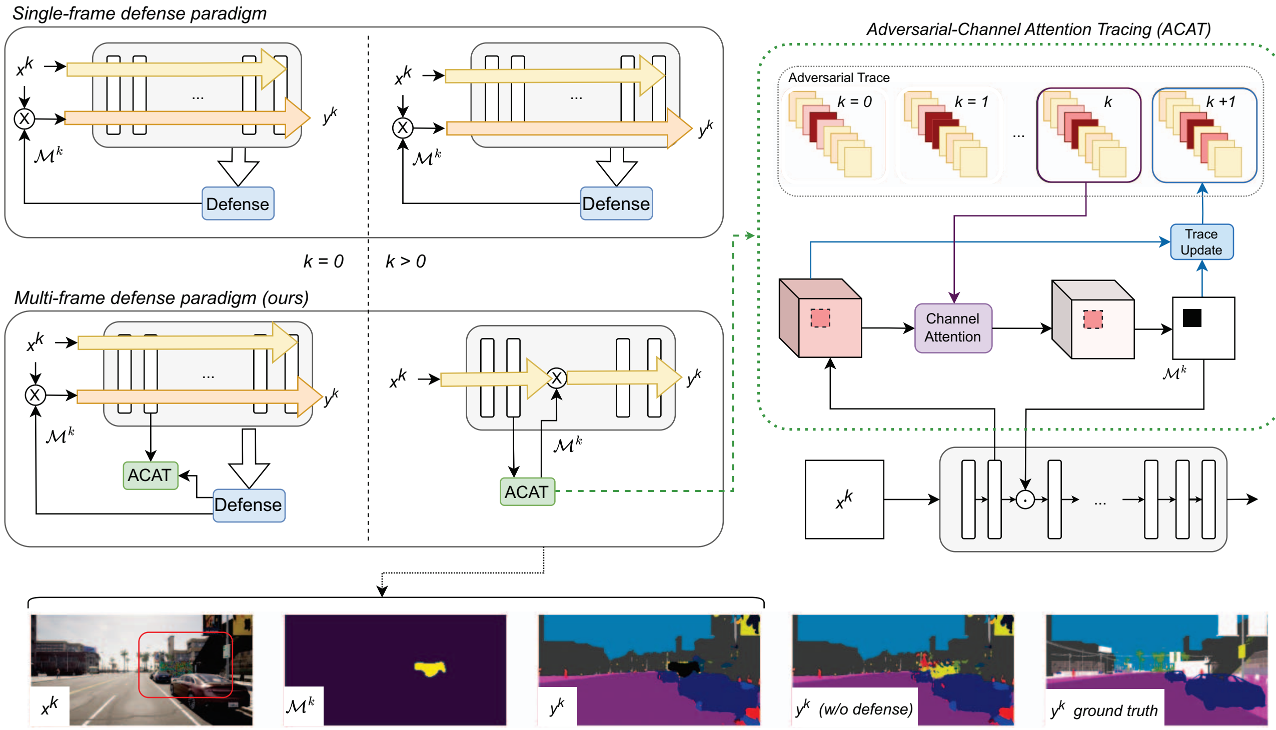


Fig. 1: Schematic and simplified overview of the proposed multi-frame defense paradigm compared to state-of-the-art, single-frame defense paradigms. At frame $k = 0$, with a first inference pass (yellow arrow in the figure), a single-frame defense mechanism extracts a mask to inhibit the detected attack. Another inference pass is required at frame $k = 0$ to apply the defense mask (orange arrow in the figure). With the proposed approach, the mask is used to implement pattern analysis in the shallow layers for the next frames $k > 0$, which is the core task performed by ACAT. This allows extracting an *adversarial trace* that allows for a quick identification of the shape of the adversarial object, hence efficiently generating and applying defense masks (right side of the figure). At the bottom, we show illustrations of the defense mechanism in a simulated attacked Carla driving scenario [8] with the BiseNet model [9], where the adversarial object is highlighted in the red area. For completeness, we also report the output of the same frame without any defense mechanism and the ground truth.

this improves both efficiency in terms of running times and computing load, as well as the attack detection effectiveness. The proposed approach is illustrated in Figure 1.

In summary, this work makes the following contribution:

- It advances the understanding of adversarial over-activations in shallow convolutional layers when aiming at detecting real-world adversarial attacks, hence introducing the concept of the *adversarial trace*.
- It proposes *ACAT*, an algorithm for multi-frame applications based on a channel-attention mechanism to make more computationally efficient and more effective the defense from real-world adversarial attacks.
- It presents extensive experiments and ablation studies to show the benefits of the proposed approach in terms of defense performance and computational costs, focusing on autonomous driving scenarios.

II. Related Work

a) Real-world adversarial attacks: In the context of the analysis of adversarial perturbations [12], real-world (RW) attacks have received particular interest from the secure AI community, due to their capability of fooling the model outcomes from the physical environment in which they operate. Indeed, from the standpoint of the attacker, the RW attack paradigm ideally avoids injecting adversarial features digitally, thus circumventing the need for compromising a computing system. To this end, different use cases addressed in the literature illustrate how physical attacks pose significant threats to AI systems. These include deceiving intrusion detection systems [13]–[15], manipulating the identification of pedestrians or cars in driving scenarios [6], [16], and fooling steering angle predictor [17]. From an architectural point of view, all the vision models can be susceptible to physical attacks, as image classification [5], [18], semantic segmentation [6], object detection [19], depth estimation [20].

To comprehensively assess the model's robustness against these threats, recent studies have also emphasized the necessity of proposing proper benchmarks to evaluate model robustness against RW attacks [8], [21].

b) Defense methods: To enhance the robustness of vision models against these attacks, various defense mechanisms have been proposed. While some focus on flagging the presence of attacks only, thus allowing to just reject the attacked frames [6], [22]–[25], more sophisticated mechanisms aimed at mitigating the attack effects at run-time, providing an attack-free DNN output. The main idea involves segmenting the position of the adversarial object within the image with the purpose of generating a pixel mask, which is capable of inhibiting the attack effects in the input space or directly in the network layer.

Some techniques [26]–[28] used a secondary encoder-decoder model to compute the mask. The mask is then used to eliminate the adversarial attack from the image before it is passed to the DNN of the target vision application. These approaches significantly increase the overall computational cost for each input (even for the non-attacked ones). More classic approaches instead, as LGS [29], aim at filtering out adversarial features from the image using gradient-based filters, assuming the adversarial features of objects have high frequency.

Other strategies, instead, are based on internal analysis of DNN layers to identify anomalous over-activations at run time [10], [11], which proved to be highly correlated with real-world adversarial effects in any targeted vision tasks. Specifically, these defense mechanisms extract masks by addressing the spatial over-activation of deep features. These approaches tend to exhibit a more predictable and robust behavior compared to those based on a secondary model. Nevertheless, they require two inference (i.e. forward) passes as the attacks are detected when analyzing deep layers of the model, where the effects of the attacks are not anymore recoverable. The second pass is hence needed to process the input image with the pixel mask applied.

Overall, although all the approaches presented in previous work to defend from RW attacks have shown promising performance and the capability to generalize among different vision tasks, little to no efforts have been made in addressing their usage in real time within CPS applications.

III. Background and Preliminaries

This section concisely provides background and preliminary concepts for the rest of the paper. We consider vision models that take as input an image with dimensions $H \times W$ pixels and C channels, denoted by $\mathbf{x} \in \mathbb{R}^{3\times H\times W}$. The model output, denoted as $f(\mathbf{x})$, depends on the specific vision task under consideration.

For simplicity, the notation is introduced by referring to a simple feed-forward DNN, with a list of layers $\{L^1, ..., L^{N_L}\}$, where the input is forwarded sequentially. To this end, we use the operational notation $f^{i\to j}$ to denote the processing flow of features from layer L^i to layer L^j. The layer index 0 is used to refer to the input of the model. For instance, $f(\mathbf{x})$ is equivalent to $f^{0\to N_L}(\mathbf{x})$ or $f^{j\to N_L}(f^{0\to j}(\mathbf{x}))$.

A. Real-World Adversarial Attacks

Real-world adversarial attacks can be generated by introducing *adversarial physical objects* into specific regions of the scene captured by the input image $\mathbf{x}$. Following previous work, we can model these objects as rectangular patches denoted by $\boldsymbol{\delta}$, where $\boldsymbol{\delta}$ is an image of size $\tilde{H} \times \tilde{W}$ with C channels, where $\tilde{H} \leq H$ and $\tilde{W} \leq W$. Crafting an adversarial patch involves solving an optimization problem aimed at minimizing a specific attack loss function, while enhancing the robustness of the patch features against realistic transformations that can occur while filming the patch in the physical world [5].

Formally, we craft an adversarial patch $\boldsymbol{\delta}$ by solving the following optimization problem:

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

where $\mathbf{X}$ is a set of images to attack, $\tilde{\boldsymbol{x}} = \boldsymbol{x} + \gamma(\boldsymbol{\delta})$ is the attacked image, $\boldsymbol{\Gamma}$ is a set of appearance and placement transformations that can be randomly selected to apply a patch, $\mathbf{y}_{\text{Adv}}$ is the adversarial output target, and $\mathcal{L}_{\text{Adv}}$ is the adversarial loss function that specifies the objective of the attacker, the lower $\mathcal{L}_{\text{Adv}}$ the more adversarial effect. Please refer to [5], [6], [18] for further details.

B. Defense Mechanisms and Internal Analysis

As discussed in Section 2, several defense strategies have been proposed in the literature to mitigate real-world adversarial attacks, particularly in single-frame applications. In this context, our approach aligns closely with works that perform internal analysis of neural models during inference.

Following this paradigm, we denote by $\boldsymbol{h}^l \in \mathbb{R}^{C^l\times H^l\times W^l}$ the features produced by any layer L^l, where C^l, H^l, W^l are the corresponding tensor dimensions, i.e., $\boldsymbol{h}^l = f^{0\to l}(\mathbf{x})$. The notation $(T)_{c,i,j}$ is used to denote a single element of any 3D tensor T, where c, i, and j are the indices for the channel, height, and width dimensions. Given an attacked input $\tilde{\mathbf{x}}$, a defense mechanism based on internal analysis studies one or more deep features to extract a heatmap $\mathcal{H} \in \mathbb{R}^{H^l\times W^l}$, which can in turn allow to highlight the position of the adversarial object within the input image.

Then, the heatmap can then be binarized, using a threshold, to obtain a mask $\mathcal{M}_\delta \in \{0,1\}^{H^l\times W^l}$, where the elements set to 0 are deemed adversarial while those set to 1 are not. Formally speaking, these steps can be summarized by means of a function $\Lambda^\xi : \mathbb{R}^{C^l\times H^l\times W^l} \to \{0,1\}^{H^l\times W^l}$, which takes as input the features $\boldsymbol{h}^l$ from a given layer and produces a mask based on a pre-determined threshold ξ. The resulting mask can then be applied at any layer L^z (e.g., even the input image itself, $z = 0$) to filter out the adversarial object, thereby aiming at making the attack ineffective, i.e.,

$$f^{z\to N_L}(f^{0\to z}(\tilde{\mathbf{x}}) \odot r^{l\to z}(\mathcal{M}_\delta)) \approx f(\mathbf{x}), \tag{2}$$

where $\odot$ is the Hadamard product operator on the spatial dimensions and $r^{l\to z}$ is a spatial resizing function to apply a mask extracted at the l-th layer to the z-th layer (clearly not needed when $l = z$). In general, function $r^{l\to z}$ consists of a simple interpolation that resizes the mask to comply with the spatial dimensions of the desired target layer.. Finally, it is also convenient to define the complementary mask $\bar{\mathcal{M}}_\delta = \mathbb{1} - \mathcal{M}_\delta$.

Since this work proposes a defense mechanism for multi-frame cases, from Section IV-B on we adopt a discrete-time notation with the superscript k to refer to the symbols introduced above when related to the k-th frame, where $k \in \{0, \dots, K\}$.

IV. Adversarial-Channel Attention

Inspired by prior research on RW attacks and internal over-activation analysis for convolutional neural networks, we observed that attacks can be detected by even analyzing shallow network layers only (as opposed to deep layers as done by previous work). In the following, we provide insights into the existence of over-activation patterns within shallow layers and then address the definition of *adversarial trace*, which is later used to enable the implementation of an adversarial attention mechanism for multi-frame scenarios.

A. Single-frame Adversarial Attention Analysis

We start by providing insights that link abnormal activations induced by adversarial objects with a particular pattern of channels in the shallow layers.

Observation 1: An adversarial object δ is designed to minimize a specific adversarial loss function by influencing certain network features (see Eq.(1)). As for instance shown in Figure 2(a), we argue that in any layer L^l with activations $\boldsymbol{h}^l$, there exists a subset of channels targeted by the adversarial object. The channels can be identified by leveraging some channel weights $\boldsymbol{\sigma} \in [0,1]^{C^l}$ that, if applied to $\boldsymbol{h}^l$, amplify the adversarial features, i.e.,

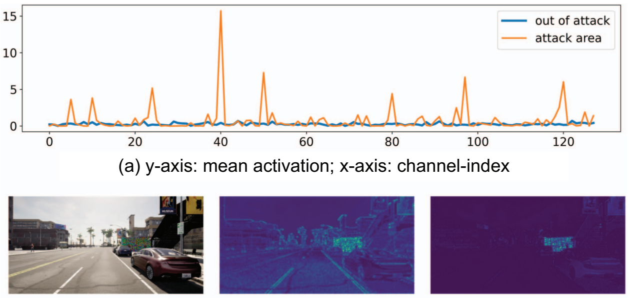
$$\mathcal{L}_{\text{Adv}}(f^{l \to N_L}(\boldsymbol{\sigma} \cdot \boldsymbol{h}^l), \mathbf{y}_{\text{Adv}}) \;\leq\; \mathcal{L}_{\text{Adv}}(f^{l \to N_L}(\boldsymbol{h}^l), \mathbf{y}_{\text{Adv}}).$$

Observation 2: As known from previous work [30], the adversarial features introduced by physical attacks are characterized by over-activations. Therefore, from the perspective of an internal analysis (as introduced in Section III-B), a proper definition of $\boldsymbol{\sigma}$ also allows focusing on the channels that are more subject to over-activation within the attacked area. This results in a better separation of the attacked area from all the others in the heatmap $\mathcal{H}$, which can be interpreted as a more accurate computation of defense masks. Formally, this observation can be formulated by means of the intersection-over-union (IoU) [31], which provides the amount of overlap between two masks. If $\mathcal{M}_{GT}$ is the ground-truth mask capable of perfectly masking the attacked area in the input image, this observation can be written as a function of the IoU between the predicted attacked area and the ground-truth mask, i.e.,

$$IoU\left(\Lambda^{\xi^{'}}(\boldsymbol{\sigma} \cdot \boldsymbol{h}^l),\ \mathcal{M}_{GT}\right) \geq IoU\left(\Lambda^{\xi^{''}}(\boldsymbol{h}^l),\ \mathcal{M}_{GT}\right),$$

where $\xi^{'}$ and $\xi^{''}$ are two thresholds used to extract the masks from activation values ($\boldsymbol{\sigma} \cdot \boldsymbol{h}^l$) and ($\boldsymbol{h}^l$), respectively.[1]

Channel weights $\boldsymbol{\sigma}$ play a pivotal role in efficiently identifying over-activated areas associated with adversarial features, even in shallow layers. In fact, while the over-activation phenomenon may look straightforward to detect, our preliminary experiments revealed that simple operations directly applied to all channels, e.g., a channel-wise sum compared to a threshold, do not allow detecting the presence of adversarial objects.



(a) y-axis: mean activation; x-axis: channel-index

(b) Heatmaps with (right) and without (center) the channel-attention

Fig. 2: (a) Mean channel-wise activation from the first spatial BiSeNet [9] layer during the inference of the attacked image; (b) representation of the heatmap w/ and w/o the attention mechanism.

B. Computing the Adversarial Trace

Following the above observations, we propose a practical usage and update of channel weights $\boldsymbol{\sigma}$, which are used to track an adversarial object over time. As anticipated in the paper introduction (see Figure 1), the proposed implementation is conceived to be complemented with a defense method capable of providing a starting mask $\mathcal{M}_{\boldsymbol{\delta}}^0$. When and how this starting mask needs to be computed will be discussed in Section V, where the complete defense framework is presented.

The *adversarial trace* is defined as a sequence of weights that highlight the channels over-activated by adversarial attacks. Formally, given a layer L^l, the adversarial trace at time k, denoted by $\boldsymbol{\sigma}^k$, is a vector of C^l elements in $[0,1]$ that enables the computation of an accurate heatmap $\mathcal{H}^k$ at time $k > 0$ as follows:

$$\mathcal{H}^k = \sum_{c=1}^{C^l} (\boldsymbol{\sigma}^k)^\tau \cdot \boldsymbol{h}^{l,k}, \tag{3}$$

where parameter τ is introduced to amplify the attention pattern within the heatmap. Figure 2(b) shows the benefits of using attention based on the adversarial trace. Once the heatmap is obtained, a threshold parameter ξ^k (defined below) can be used to devise the binary mask $\mathcal{M}_{\delta}^k$.

In this work, we proposed a per-frame update of the adversarial trace, so that the next element for time $k+1$ can be computed as a function of the mask and activations computed at time k:

$$\boldsymbol{\sigma}^{k+1} = \mathcal{N}\left(\frac{\sum_{i,j=1}^{H,W}\left(\boldsymbol{h}^{l,k} \odot \bar{\mathcal{M}}_{\boldsymbol{\delta}}^{k}\right)_{c,i,j}}{|\bar{\mathcal{M}}_{\boldsymbol{\delta}}^{k}|} - \frac{\sum_{i,j=1}^{H,W}\left(\boldsymbol{h}^{l,k} \odot \mathcal{M}_{\boldsymbol{\delta}}^{k}\right)_{c,i,j}}{|\mathcal{M}_{\boldsymbol{\delta}}^{k}|}\right), \tag{4}$$

where $\mathcal{M}_{\boldsymbol{\delta}}^k$ is the predicted mask at time k, $\bar{\mathcal{M}}_{\boldsymbol{\delta}}^k$ denotes a complementary mask to address all other tensor values not

[1] Note that the thresholds must be different because $\boldsymbol{\sigma}$ scales $\boldsymbol{h}^l$.

interested by $\mathcal{M}_{\boldsymbol{\delta}}^k$, and $\mathcal{N}$ represents a normalization function that scales the values to the $[0, 1]$ range. In our experiments, we implemented $\mathcal{N}$ as a ReLU function followed by a channel-wise min-max normalization.

In particular, the first fractional term in Eq. (4) provides attention to over-activated patterns within the area of the adversarial object, while the second term provides negative attention to activations outside the same.

The effectiveness of using information obtained from the current frame to compute the next adversarial trace element $\boldsymbol{\sigma}^{k+1}$ was verified by means of experiments (see Section VI).

Summary of the approach. Figure 3 provides a schematic representation that illustrates the use and update of the adversarial trace for a frame at time k. Note that a noise filter (e.g., a Gaussian filter) can be introduced into the pipeline for computing the heatmap. As highlighted in [11], noise filters help mitigate the effects of small spurious activations.

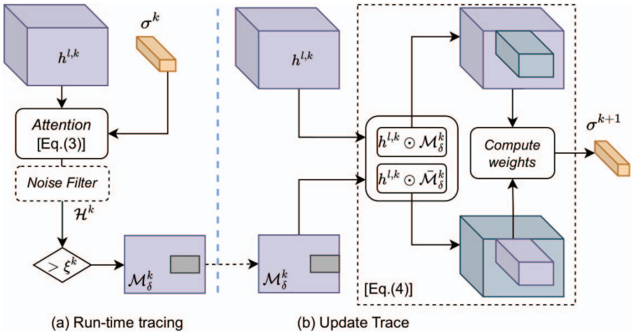


Fig. 3: Illustration of the operations to implement adversarial attention mechanism performed at time k. The resulting output is the next element $\boldsymbol{\sigma}^{k+1}$ of the adversarial trace.

Threshold definition. Differently from previous work, which used a static threshold computed offline on a calibration dataset [11], [22], this work adopts an adaptive threshold that is dynamically computed frame by frame. This is necessary due to the attack-specific channel weighting of the attention mechanism, which makes not effective thresholds computed a priori. In ACAT, the threshold is updated at each frame as follows:

$$\xi^{k+1} = \max(\bar{\mathcal{H}}^k) + \psi(\bar{\mathcal{H}}^k). \tag{5}$$

In the above equation, $\bar{\mathcal{H}}^k = \mathcal{H}^k \odot D(\bar{\mathcal{M}}_{\boldsymbol{\delta}}^k)$, where $D(\cdot)$ is an operator that expands $\bar{\mathcal{M}}_{\boldsymbol{\delta}}^k$ by means of a unitary kernel convolution. This expansion is designed to account for *uncertainty* in the areas around the mask, coping with potential spurious activations close to its border. After applying a unitary convolution, non-integer values can be obtained: hence, the operator $D(\cdot)$ eventually binarizes all values using a threshold equal to 0.5.

To further reduce false positives, an extra safety margin $\psi(\bar{\mathcal{H}}^k)$ is included in Eq. (5). It is computed as the difference between the v-th percentile of the values in $\bar{\mathcal{H}}^k$ and the mean value of the same. In our experiments, we used $v = 70$, which proved to offer effective resilience to uncertainty.

V. ACAT Framework

This section shows how to integrate adversarial-channel attention within the continuous processing loop of vision applications. Algorithm 1 reports the pseudocode of the operations to be performed at each frame (retrieved with function capture_frame()). To improve readability, the discrete-time notation with the superscript k is omitted in the pseudocode, as all variables are updated to be used at the next cycle.

Algorithm 1 Adversarial-Channel Attention Tracing

```
1:  $\boldsymbol{\sigma} \leftarrow$ None
2:  while True do
3:      $x \leftarrow$ capture_frame()
4:      if $\boldsymbol{\sigma}$ is None then
5:          $(y, \mathcal{M}_\delta, \boldsymbol{h}^l) \leftarrow$ inference_with_SoA_method$(x, f)$
6:          if $\mathcal{M}_\delta$ is not None then
7:              #Attack notified
8:              $\boldsymbol{\sigma} \leftarrow$ ACAT_update($\boldsymbol{h}^l$, $\mathcal{M}_\delta$)    #Eq. (4)
9:              $\xi \leftarrow$ compute_threshold($\boldsymbol{h}^l$, $\mathcal{M}_\delta$)    #Eq. (5)
10:             $y = f(x \odot \mathcal{M}_\delta)$    # Inference with masked input
11:         end if
12:         Continue    #Wait for next frame
13:     end if
14:     $\boldsymbol{h}^l = f^{[0 \to l]}(x)$
15:     $\mathcal{H} =$ noise_filter$\left(\sum_{c=1}^{C^l} (\boldsymbol{\sigma})^\tau \cdot \boldsymbol{h}^l\right)$    #Eq. (3)
16:     $\mathcal{M}_\delta \leftarrow \Lambda^\xi(\mathcal{H})$    #Apply threshold to get mask
17:     if $|\bar{\mathcal{M}}_\delta| < \lambda_\mathcal{M}$ then
18:         $\boldsymbol{\sigma} \leftarrow$ None    #Stop adv. tracing
19:         $y = f^{[l \to L]}(\boldsymbol{h}^l)$
20:     else
21:         $\boldsymbol{\sigma} \leftarrow$ ACAT_update$(\boldsymbol{h}^l, \mathcal{M}_\delta)$    #Eq. (4)
22:         $\xi \leftarrow$ compute_threshold($\boldsymbol{h}^l$, $\mathcal{M}_\delta$)    #Eq. (5)
23:         $y = f^{[l \to L]}(\boldsymbol{h}^l \odot \mathcal{M}_\delta)$ #Inference with masked layer
24:     end if
25: end while
```

For each frame, it checks if the adversarial trace $\boldsymbol{\sigma}$ exists. If not, it means no adversarial attack was detected at the previous frame. In this case, a state-of-the-art attack detection method, e.g., [11], is executed (line 5) with a single inference pass. If the latter detects an attack, the algorithm initializes the adversarial trace $\boldsymbol{\sigma}$, computes the threshold ξ, and leverages the mask compute by the state-of-the-art method to defend from the attack (lines 8-10). The processing of the current frame can hence end.

Otherwise, when the adversarial trace $\boldsymbol{\sigma}$ is available from the previous frame, the algorithm leverages it to compute the defense mask following the results of Sec. IV-B (lines 14-16). If the mask is meaningful (details provided next), it also computes the next adversarial trace and threshold (lines 21-22), still based on Sec. IV-B, and continues the inference process by applying the mask at the inner layer L^l to defend from the attack (line 23).

a) Reset criterion: Knowing about the connection between the mask size and the induced adversarial effect by the masked attack [11], we use the number of pixels detected in the predicted complementary mask to decide whether to reset adversarial tracing or not. This could mean that the adversarial object is either too small or far away from the camera. Specifically, we disable adversarial tracing when the computed mask has less than $\lambda_{\mathcal{M}}$ pixels (line 17), where the latter is a configurable parameter.

b) Timing performance: As discussed in Section II, state-of-the-art approaches introduce a non-negligible delay to defend from adversarial attacks: either they require an extra inference pass or they must wait for the inference of an external model specialized to detect an attack. Differently, as it can clearly be seen from Algorithm 1, the proposed ACAT method allows defending from an attack with just *one* inference pass (completed in two stages at lines 14 and 23, respectively) for the next frame. This holds until tracing is active, i.e., the reset criterion is not reached. Once a new attack will be detected the same will hold for the next frames, and so on and so forth. Overall, ACAT allows significantly improving the timing performance of the defense mechanism (quantitative results provided in the next section) by halving inference times in general, except for the very first frame in which the attack manifests.

VI. Experimental evaluation

The experimental evaluation is focused on semantic segmentation models designed for autonomous driving, which have recently garnered attention due to the need to address real-world adversarial attacks in outdoor scenarios [6], [32]. Please note however that defense mechanisms based on over-activation also work for different computer vision tasks, where the connection between over-activation and adversarial effect persists [6], [30].

Next, we first provide details on the experimental settings. Then, we discuss different tests and ablation studies conducted to validate the design and benefits of the proposed defense algorithm. All the experiments were implemented using PyTorch [33] and a platform with one NVIDIA-A100 GPU.

A. Experimental settings

Complete multi-frame benchmarks to evaluate the effectiveness of defense methods against real-world adversarial attacks are not available from previous work.

For this reason, we addressed two evaluation approaches:
(i) attack scenarios generated with the CARLA simulator [35], used to test the attention mechanism of ACAT only, and
(ii) digitally attacked video generated with Cityscapes [36], which instead allow testing the whole ACAT framework.

a) Attacks in CARLA-simulated scenarios: With the intent of facing with realistic settings, we utilized the CarlaGear framework [8], which offers 9 photo-realistic scenarios (50 test images each) collected in areas of Carla-town 10 [35], integrating adversarial billboards specifically designed for each model in use. Please note that the framework only provides random viewpoints of the area next to the adversarial billboards, which are not sequential videos. For this reason, this setting allows evaluating the benefits of adopting adversarial-channel attention only, i.e., improving the capabilities of state-of-the-art defense mechanisms when used on a single frame, while not enabling meaningful tests to evaluate ACAT as a whole.

b) Digitally attacked video datasets: To address the lack of a dedicated video dataset featuring attacked driving scenes, we generated custom videos that include digital adversarial attacks. Three extended sequences from Cityscapes [36] videos[2] were utilized with images sized at 2048x1024 pixels. Within each video, a dynamic adversarial patch was digitally introduced in the frames, which, at every frame, changes its position and scaling factor, following a sinusoidal trend. The patch position and scale were computed as follows:

$$\begin{bmatrix} x_{pos} \\ y_{pos} \\ s \end{bmatrix} = \begin{bmatrix} c_x + A_x \sin(\alpha_x \cdot k + \omega_x) \\ c_y + A_y \sin(\alpha_y \cdot k + \omega_y) \\ 1 + A_s \sin(\alpha_s \cdot k + \omega_s) \end{bmatrix}, \qquad (6)$$

where k is the frame index, x_{pos} and y_{pos} are coordinates of the position of the patch, c_x, c_y are the center coordinates of the frame, and s is the scaling factor of the patch. In our experiments we set $(A_x, A_y, A_s, \alpha_x, \alpha_y, \alpha_s) = (500, 300, 0.3, 0.05, 0.05, 0.05)$. The ω values represent a phase used to randomize tests among different initial positions. With these settings, the patch can partially go beyond the image boundaries while holding a size that is sufficient for producing an adversarial effect [6], [19]. The α values provide a smooth trend of the patch among subsequent frames.

The attack mechanism used to generate the patch was the Over-Activation-aware Expectation Over Transformation (EOT) optimization [5], [11], where a parameter $\beta \in [0, 1]$ is used to regulate the over-activation level of the patch within the internal layers while reducing the adversarial effect (the lower the β the lower the over-activation, and so the adversarial effect). This approach is particularly useful for conducting exhaustive evaluations of the proposed defense, as the attacker can deliberately attempt to minimize over-activation to execute attacks that are challenging to detect.

c) Network models and defense methods: Following related work on semantic segmentation [8], we considered real-time high-performance DNN models: DDRNet-Slim23 version [34] and BiSeNetX39 [9]. We use the pre-trained versions available from [8].

We compared our method ACAT with two lightweight single-frame approaches designed to mask real-world attacks. The first is LGS [29], which applies gradient-based filtering of the image to mask adversarial pixels. The second is ZMask [11], which, as for ACAT, is based on over-activation but necessitates of two forward passes at any frame, since

[2] https://www.cityscapes-dataset.com, *leftImg8bit_demoVideo.zip*

	Scene 1	Scene 2	Scene 3	Scene 4	Scene 5	Scene 6	Scene 7	Scene 8	Scene 9
No Attack	28.35	29.66	28.88	24.71	25.10	26.51	20.56	24.46	23.68
No Def	-10.2	-4.32	-2.81	-6.4	-5.06	-9.5	-7.00	-2.93	-6.75
ACAT$_{GT}$	**-2.9** [1]	**+0.59** [1]	**-2.67** [1]	**-2.11** [2]	**-0.9** [1]	-5.4 [1]	**-2.87** [1]	**-0.63** [1]	**-0.78** [1]
ACAT$_{ZM}$	**-2.9** [2]	-3.4 [1]	-3.15 [12]	-2.82 [2]	-1.1 [7]	**-5.33** [1]	-2.89 [21]	-5.92[1]	-5.12[1]
ZMask	-5.32	-2.35	-2.81	-3.01	-1.7	-5.87	-4.84	-3.0	-5.89
LGS	-8.63	-5.09	-4.09	-6.43	-4.04	-6.65	-6.01	-1.60	-5.83
No Attack	33.04	34.11	34.39	34.22	31.01	33.26	30.73	31.86	37.41
No Def	+0.65	-1.74	-0.6	-6.53	-1.01	-2.31	-12.69	-3.03	-1.11
ACAT$_{GT}$	**+1.59** [1]	**-1.58** [1]	**+0.66** [3]	**-3.29** [1]	**+0.4** [1]	**-0.83** [1]	**-1.52** [1]	**-1.11** [1]	**-0.86** [1]
ACAT$_{ZM}$	+0.59 [2]	-1.73 [3]	-0.67 [15]	-3.52 [6]	+0.35 [5]	-2.85 [1]	-4.25 [33]	-2.22 [16]	-1.01 [5]
ZMask	-2.27	-3.2	-1.32	-5.2	-0.97	-5.55	-3.58	-0.98	-1.01
LGS	+0.45	-3.51	+1.01	-8.30	-0.87	-4.06	-11.97	-1.22	-1.25

TABLE I: Variation of the multi-class mIoU w/ and w/o defense mechanisms across the 9 driving scenarios of CarlaGear [8]. Results for both BiseNet [9] (top) and DDRNet [34] (bottom) are reported. The values inside square brackets denote the number of times ACAT required to be re-initialized (reset criterion). The results of ACAT are averaged across 5 random shuffling of each scene dataset.

addresses also deep network layers. For both, we utilized the original settings provided by the respective authors.

For ACAT, we set $\tau = 2$, and the kernel size to $5, 3$ and $31, 11$ for the Gaussian filter and dilatation operators for Bisenet and DDRNet, respectively. The different sizes are due to the different spatial dimensions of the features. The layers analyzed by ACAT are in the shallower blocks of the considered model, specifically the output of the second block of DDRNet and the output of the first context layer of BiSeNet. Ablation studies were also performed to understand the selection process of these layers (see Sec. VI-E).

d) Metrics: Different metrics were used to assess the performance of the addressed mechanisms. Given the unavailability of annotations for the Cityscapes videos, we use the binary Intersection-over-Union (IoU), referred to as Mask-IoU, to measure the overlap between the predicted complementary mask $\bar{\mathcal{M}}^k_{\delta}$ (whose values equal to 1 denote the predicted adversarial region) and the corresponding ground-truth mask $\bar{\mathcal{M}}^k_{GT}$. Intuitively, Mask-IoU quantifies the quality of the predicted defense mask: the higher the better.

For the tests conducted on the Carla-Gear dataset, as indicated in the benchmark, we measured the effectiveness of adversarial attacks by addressing the original multi-class MIoU [8], [36] of the task, since annotations are available.

B. Performance Evaluation on Carla

Table I highlights the advantages of our approach across nine scenarios of the Carla-Gear dataset on BiSeNet (top part) and DDRnet (bottom). Regarding ACAT, which is designed to integrate with state-of-the-art defenses, we conducted analyses under two settings: *ACAT$_{ZM}$* and *ACAT$_{GT}$*. The former utilizes ZMask [11], reflecting a realistic scenario built upon an already available approach. In the second setting, *ACAT$_{GT}$* assumes the knowledge of an ideal, ground-truth mask at first frame in which the attack is detected. While this setting depicts a less realistic scenario, it serves to highlight the intrinsic performance of ACAT, independently from the defense method with which it is integrated.

In the table, the first line for each scenario depicts the task MIoU without an adversarial billboard, while subsequent lines show the drop in MIoU with the adversarial billboard and/or without the related defenses. The value between the brackets for the ACAT results depicts the number of times that the reset criterion takes effect, necessitating the extraction of a new starting mask. As also mentioned in [8], there are instances where certain attacks can be particularly challenging for a specific model and scenario, leading to a poor reduction in the MIoU (or even a small average improvement). To assist the reader, in Table I we highlighted in gray the scenarios that have resulted in a more pronounced adversarial effect.

As it can be noted from the table, ACAT consistently outperforms the other methods, significantly reducing the number of extra inference passes, reaching the reset conditions only a few times. Note also that *ACAT$_{ZM}$* generally improves the performance of ZMask. However, when ZMask fails to return an accurate mask, it may jeopardize the initialization of ACAT, resulting in lower performance (e.g., scene 8 - BiSeNet and scene 7 on DDRNet). This is not the case for *ACAT$_{GT}$*, confirming that the lower performance is not due to ACAT. Concerning LGS, as known from previous work, it loses accuracy in real-world scenarios [11], [27].

Please note that, in these tests only, we did not update the trace and threshold of ACAT (lines 21-22 in Algorithm 1). As anticipated above, this is because the tested images do not pertain to sequential video. The whole ACAT framework is instead addressed by the following experiments.

It is however interesting to also observe the number of times ACAT required a re-initialization (reset criterion) during these tests, even if updates are disabled. As one may expect, we found scenarios in which the mask provided by ZMask was frequently required (e.g., note the numbers between square brackets in Table I for Scenes 3 and 7), while surprisingly, in other cases, it was not at all. This means that the attention mechanism offered by ACAT is sometimes effective even with sporadic updates (see also the other experiments below). Conversely, in the former case, we found that the reset criterion was prominently triggered because the mask provided by

ZMask was not particularly accurate, as $ACAT_{GT}$ almost never requires to be re-initialized.

C. Performance Evaluation on Digital Attacks

In Figure 4, we studied the Mask-IoU for the digital attacked video. To show that ACAT provides high robustness even when the adversarial trace and thresholds are not updated at every frame as mandated by Alg. 1, we measured the average Mask-IoU under $ACAT_{ZM}$ on attacked video streams from Cityscapes, varying the period with which the trace and thresholds are updated. The period is expressed in number of frames and is reported on the x-axis of the figure (e.g., value 1 on the x-axis means that the update occurs at each frame). In the analysis, we tested two digital adversarial patches, with $\beta = 0.6$ and $\beta = 0.8$, to better investigate on the robustness of ACAT. We also evaluated ZMask, which achieves $(0.66, 0.75)$ and $(0.70, 0.72)$ of Mask-IoU with ($\beta = 0.6$, $\beta = 0.8$) for Bisenet and DDRNet, respectively. The results for LGS are not reported since it does not provide a binary defense mask, but rather a soft filtering of the input image, for which it is not possible to compute the Mask-IoU.

The figure shows that ACAT surprisingly works well even with sporadic updates of the adversarial trace and thresholds. This was also due to the fact that the Cityscapes videos are related to rather static scenarios. In fact, despite some changes in the appearance of adversarial patches and their background, the over-activated pattern of the patch in these cases continuously insist on a similar set of channels to induce the adversarial effect. An update of the parameters is anyway required in more dynamic scenarios with more frequent changes of the background and appearance of the adversarial object. The figure also shows that sporadic updates always provide better performance than ZMask.

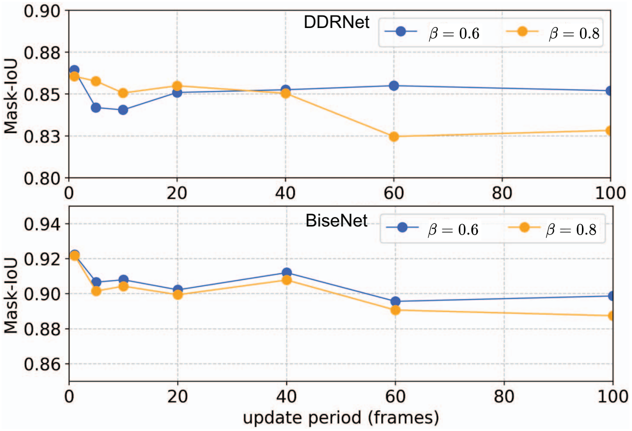


Fig. 4: Mask-IoU performance by varying the update period (in frames, x-axis of the figure) of the adversarial trace and thresholds. The upper plot refers to the DDRNet architecture, while the second one pertains to BiseNet. The tests evaluate the performance of $ACAT_{ZM}$ for two distinct digital patches ($\beta = 0.6$ and $\beta = 0.8$). The results are the average of five different initializations of the ω parameters in Eq. (6).

D. Ablation Studies

To better understand the contribution of each operation performed by ACAT to its overall performance, Table II reports the Mask-IoU of $ACAT_{GT}$ on the attacked videos under different settings. With the aim of acquiring a deeper understanding about the attention mechanism of ACAT, we independently examined the two fractional terms defined in Equation (4) to update the adversarial trace. The first term provides positive attention within the attacked area, which is the most important part of the attention mechanism. We hence introduce a flag Att^{+} to indicate a setting of ACAT that uses this term. The second term refines the previous operation by introducing negative attention to the elements outside the attacked area. Another flag Att^{-} is also introduced to denote if this second term is used by ACAT.

As shown in the table, it is clear that using both Att^{+} and Att^{-} leads to better results, hence motivating the construction of Equation (4). In general, it is evident that the use of the attention mechanism significantly improves the Mask-IoU when compared to not using attention (both Att^{+} and Att^{-} disabled, first rows of the table). Its benefits are especially notable in the results obtained with DDRNet, where adversarial over-activations in the shallow layers proved to be very difficult to detect without attention. These observations are also illustrated with an example frame in Figure 5. The ablation studies also tested ACAT with and without the update of the adversarial trace and the threshold of Eq. (5) (flag *Upd* in Table II), and with and without the noise filter (flag NF).

				Bisenet		DDRNet	
Att^{+}	Att^{-}	*Upd*	NF	$\delta_{0.6}$	$\delta_{0.8}$	$\delta_{0.6}$	$\delta_{0.8}$
				11.9	16.2	0.00	0.01
			✓	89.0	88.7	7.2	0.6
✓			✓	90.7	90.2	76.91	84.90
✓		✓	✓	91.3	90.8	72.05	83.05
✓	✓		✓	**92.24**	91.9	85.56	84.22
✓	✓	✓	✓	92.23	**92.18**	**86.12**	**86.42**

TABLE II: Experimental results of ablation studies with respect to the different components used to update the adversarial trace. The results are in terms of Mask-IoU and related to the digitally-attacked Cityscape videos using $ACAT_{\text{GT}}$ as a defense mechanism. Two model-specific patches were utilized, one with $\beta = 0.6$ and another with $\beta = 0.8$.. In the table, Att^{+} and Att^{-} denote two flags to enable the two attention terms of Equation 4, respectively, while *Upd* and the NF are other two flags to enable the update of the trace and threshold, and the usage of the noise filter, respectively.

E. Layer-wise Ablation

Figure 6 reports the Mask-IoU by varying the layer of the DDRNet model with which ACAT operates (parameter l in Alg. 1). As observed, the more shallow the layer the better the performance. In fact, if addressing deeper layers, the mask based on the over-activation extends beyond the ground-truth position (in the figure, only the yellow parts denote a complete overlap of the ground-truth and the predicted mask). This is attributed to the fact that the features of shallow layers are

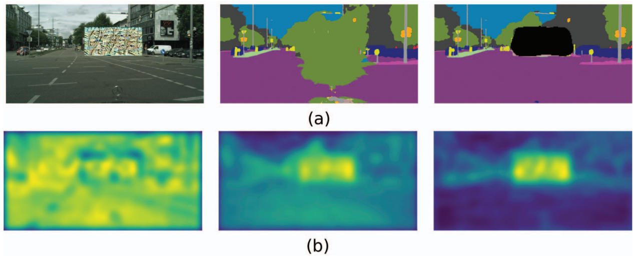


Fig. 5: (a) Comparison of the adversarial effect of a patch with $\beta = 0.6$ (left), with ACAT_{GT} mechanism (right), and without the ACAT_{GT} mechanism (middle). (b) Illustration of the heatmap among different settings, from left to right: **(i)** only NF enabled, **(ii)** only Att^+ and NF enabled, **(iii)** Att^+, Att^-, NF, and *Upd* enabled.

less spatially compressed (i.e., they have a higher spatial size) than those in deeper layers.

Note that, for fair comparisons, in layer $l = 3$, we used the same kernel size as layer $l = 2$ (i.e., 3), which provided better performance than kernel size 1 (i.e., no Gaussian filter). While, for layer $l = 5$, we did not use the Gaussian filter due to the high compression of the spatial dimension. These results highlight how ACAT allows focusing on shallow layers so that attacks can be masked within a single inference pass, as opposed to previous work that analyzes deep layers and hence requires another inference pass to mask attacks.

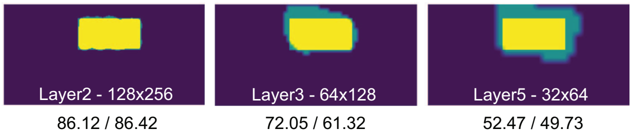


Fig. 6: Mask-IoU (in black) for the digital adversarial patch with $\beta = 0.6$ and 0.8 on attacked cityscapes video using the $ACAT_{GT}$ on different layers of DDRNet. The figures show the overlapping between the predicted mask and the ground truth for $\beta = 0.6$, with the highest color indicating the degree of overlap. The depth of the DDRNet layer and the spatial dimension are denoted in white.

F. Timing Evaluation

To demonstrate the improvements provided by ACAT in terms of running times, we measured the inference times when testing the attacked Cityscapes videos. Figure 7 reports the overall inference time required on average to process a frame by the tested defense mechanisms, with the baseline labeled by *No_Def*, denoting the original model without defenses. Two inference times are reported for ZMask: when an attack is not detected and when an attack is detected, which are separated by a slash in the figure. As expected, when ZMask detects an attack, its inference time is approximately twice the one of the baseline model. Conversely, when no adversarial attacks are detected, ZMask is particularly efficient and hence represents an excellent choice to work in conjunction with ACAT, which activates only when an attack is first detected (see Alg. 1).

The figure also reports the results for another state-of-the-art defense mechanism, named MaskNet [27], which incorporates a secondary model. It is relatively more expensive due to the necessity of always running an encoder-decoder model in tandem with the original model.

Note that LGS exhibits comparable timing performance with respect to ACAT, since it focuses on specific filters that are directly applied to the input image. However, as shown by the results in Table 1 and other studies in previous work [11], [27], LGS tends not to perform well in detecting adversarial attacks that can be carried out in real world, i.e., by means of physical adversarial objects.

In summary, these results remark on how ACAT provides a well-balanced trade-off between defense performance and overall inference time.

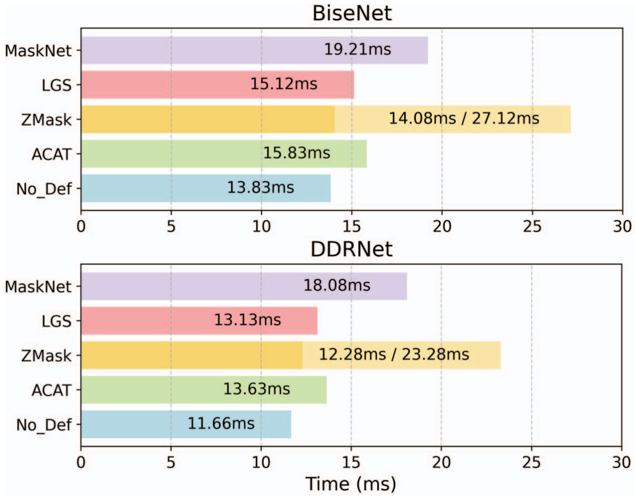


Fig. 7: Overall inference time with and without defense mechanisms for DDRNet and BiseNet.

VII. Discussion and Conclusion

This work established a novel understanding of the feature over-activation induced by physical adversarial objects in modern neural networks. Differently from previous work, this work proposed an approach that allow identifying physical adversarial attacks by analyzing the first layers of the network, enabling the implementation of efficient defenses for multi-frame vision applications that mostly require just an inference pass to inhibit attacks. Based on these findings, we proposed *Adversarial-Channel Attention Tracing* (ACAT), a framework based on the concept of *adversarial trace* that focuses on specific channels (within a given layer) that are primarily responsible for propagating the adversarial effect. ACAT is used to extend single-frame defense mechanisms from previous work, which instead may require two inference passes to defend from attacks. Experimental results demonstrated that ACAT allows both improving the defense capabilities of state-of-the-art defense methods, even when used for a single frame, as well as providing a lower computational cost by detecting and defending attacks in a single inference pass.

Despite the achieved results, we acknowledge possible limitations of the approach, such as the reliance on a reference defense mechanism to identify the presence of physical attacks in the first frame. This implies that the robustness of the approach is somewhat dependent on the effectiveness of the reference method used. Another anticipated challenge, as mentioned in the experimental section, is the need for additional benchmarks to evaluate the effectiveness of physical attacks in multi-frame applications. Future work will address these issues, also providing a more comprehensive integration of the approach into complex AI-based vision systems.

We believe that, beyond the presentation of the ACAT framework, our findings and analyses will also contribute to gaining a deeper understanding of the nature of these physical attacks and, consequently, the development of even more effective defense strategies.

ACKNOWLEDGEMENTS

This work was partially supported by project SERICS (PE00000014) under the MUR (Ministero dell'Università e della Ricerca) National Recovery and Resilience Plan funded by the European Union - NextGenerationEU.

REFERENCES

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *2nd International Conference on Learning Representations, ICLR*, 2014.

[2] C. Shea-Blymyer and H. Abbas, "Algorithmic ethics: Formalization and verification of autonomous vehicle obligations," *ACM Transactions on Cyber-Physical Systems (TCPS)*, 2021.

[3] S. Mohan, S. Bak, E. Betti, H. Yun, L. Sha, and M. Caccamo, "S3a: Secure system simplex architecture for enhanced security and robustness of cyber-physical systems," in *Proceedings of the 2nd ACM international conference on High confidence networked systems*, 2013.

[4] L. Sun, M. Tan, and Z. Zhou, "A survey of practical adversarial example attacks," *Cybersecurity*, 2018.

[5] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," in *in 35th International Conference on Machine Learning*, 2018.

[6] G. Rossolini, F. Nesti, G. D'Amico, S. Nair, A. Biondi, and G. Buttazzo, "On the real-world adversarial robustness of real-time semantic segmentation models for autonomous driving," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2023.

[7] D. Wang, W. Yao, T. Jiang, G. Tang, and X. Chen, "A survey on physical adversarial attack in computer vision," *preprint arXiv:2209.14262*, 2022.

[8] F. Nesti, G. Rossolini, G. D'Amico, A. Biondi, and G. Buttazzo, "Carla-gear: a dataset generator for a systematic evaluation of adversarial robustness of vision models," *arXiv preprint arXiv:2206.04365*, 2022.

[9] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Bisenet: Bilateral segmentation network for real-time semantic segmentation," in *European Conference on Computer Vision*. Springer, 2018.

[10] T. Wu, L. Tong, and Y. Vorobeychik, "Defending against physically realizable attacks on image classification," in *8th International Conference on Learning Representations ICLR*, 2020.

[11] G. Rossolini, F. Nesti, F. Brau, A. Biondi, and G. Buttazzo, "Defending from physically-realizable adversarial attacks through internal over-activation analysis," in *AAAI Conference on Artificial Intelligence*, 2023.

[12] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *5th International Conference on Learning Representations*, 2017.

[13] Z. Wu, S.-N. Lim, L. S. Davis, and T. Goldstein, "Making an invisibility cloak: Real world adversarial attacks on object detectors," in *6th European Conference on Computer Vision, 2020*. Springer, 2020, pp. 1–17.

[14] Z. Hu, S. Huang, X. Zhu, F. Sun, B. Zhang, and X. Hu, "Adversarial texture for fooling person detectors in the physical world," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022.

[15] K. Xu, G. Zhang, S. Liu, Q. Fan, M. Sun, H. Chen, P. Chen, Y. Wang, and X. Lin, "Adversarial t-shirt! evading person detectors in a physical world," in *16th European Conference*, vol. 12350. Springer, 2020.

[16] Y. Zhang, H. Foroosh, P. David, and B. Gong, "CAMOU: Learning physical vehicle camouflages to adversarially attack detectors in the wild," in *International Conference on Learning Representations*, 2019.

[17] T. Wu, X. Ning, W. Li, R. Huang, H. Yang, and Y. Wang, "Physical adversarial attack on vehicle detector in the carla simulator," *CoRR*, vol. abs/2007.16118, 2020.

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

[19] M. Lee and J. Z. Kolter, "On physical adversarial patches for object detection," *CoRR*, vol. abs/1906.11897, 2019.

[20] Z. Cheng, J. Liang, H. Choi, G. Tao, Z. Cao, D. Liu, and X. Zhang, "Physical attack on monocular depth estimation with optimal adversarial patches," in *European conference on computer vision*. Springer, 2022.

[21] A. Braunegg, A. Chakraborty, M. Krumdick, N. Lape, S. Leary, K. Manville, E. Merkhofer, L. Strickhart, and M. Walmer, "Apricot: A dataset of physical adversarial attacks on object detection," in *European Conference on Computer Vision*, 2020.

[22] K. T. Co, L. Muñoz-González, L. Kanthan, and E. C. Lupu, "Real-time detection of practical universal adversarial perturbations," *arXiv:2105.07334*, 2021.

[23] C. Xiang, A. N. Bhagoji, V. Sehwag, and P. Mittal, "PatchGuard: A provably robust defense against adversarial patches via small receptive fields and masking," in *30th USENIX Security Symposium*, 2021.

[24] C. Xiang, S. Mahloujifar, and P. Mittal, "{PatchCleanser}: Certifiably robust defense against adversarial patches for any image classifier," in *31st USENIX Security Symposium*, 2022.

[25] E. Chou, F. Tramer, and G. Pellegrino, "Sentinet: Detecting localized universal attacks against deep learning systems," in *2020 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2020, pp. 48–54.

[26] J. Liu, A. Levine, C. P. Lau, R. Chellappa, and S. Feizi, "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)*, 2022.

[27] P.-H. Chiang, C.-S. Chan, and S.-H. Wu, "Adversarial pixel masking: A defense against physical attacks for pre-trained object detectors," in *Proceedings of the 29th ACM International Conference on Multimedia*, ser. MM '21. ACM, 2021.

[28] K. Xu, Y. Xiao, Z. Zheng, K. Cai, and R. Nevatia, "Patchzero: Defending against adversarial patch attacks by detecting and zeroing the patch," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 4632–4641.

[29] M. Naseer, S. Khan, and F. Porikli, "Local gradients smoothing: Defense against localized adversarial attacks," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019.

[30] C. Yu, J. Chen, Y. Xue, Y. Liu, W. Wan, J. Bao, and H. Ma, "Defending against universal adversarial patches by clipping feature norms," in *in IEEE/CVF International Conference on Computer Vision*, 2021.

[31] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 658–666.

[32] J. Zhang, Y. Lou, J. Wang, K. Wu, K. Lu, and X. Jia, "Evaluating adversarial attacks on driving safety in vision-based autonomous vehicles," *IEEE Internet of Things Journal*, vol. 9, no. 5, pp. 3443–3456, 2021.

[33] A. Paszke et al., "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc., 2019.

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

[35] A. Dosovitskiy, G. Ros, F. Codevilla, A. M. López, and V. Koltun, "CARLA: an open urban driving simulator," in *1st Annual Conference on Robot Learning*, vol. 78. PMLR, 2017.

[36] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.