DOI:
https://doi.org/10.1145/3555375

LanCeX: A Versatile and Lightweight Defense Method against Condensed Adversarial Attacks in Image and Audio Recognition

ZIRUI XU, FUXUN YU, and XIANG CHEN, George Mason University, USA
CHENCHEN LIU, University of Maryland, Baltimore County, USA

Convolutional Neural Networks (CNNs) are widely deployed in various embedded recognition applications. However, they demonstrate a considerable vulnerability to adversarial attacks, which leverage the well-designed perturbations to mislead the recognition results. Recently, for easier perturbation injection and higher attack effectiveness, the adversarial perturbations are concentrated into a small area with various types and different data modalities. When defending such condensed adversarial attacks on the embedded recognition scenarios, most of the existing defense works show two critical issues: First, they are particularly designed for each individual condensed attack scenario, lacking enough versatility to accommodate attacks with different data modalities. Second, they rely on computation-intensive pre-processing techniques, which is impractical for time-sensitive embedded recognition scenarios. In this paper, we propose *LanCeX* – a versatile and lightweight CNN defense solution against condensed adversarial attacks. By examining CNN's intrinsic vulnerability, we first identify the common attacking mechanism behind condensed adversarial attacks across different data modalities. Based on this mechanism, *LanCeX* can defend against various condensed attacks with the optimal computation workload in different recognition scenarios. Experiments show that *LanCeX* can achieve an average 91%, 85%, and 90% detection success rate and optimal adversarial mitigation performance in three recognition scenarios, *e.g.* image classification, object detection, and audio recognition. Moreover, *LanCeX* is at most 3× faster compared with the state-of-the-art defense methods, making it feasible to resource-constrained embedded systems.

CCS Concepts: • **Computing methodologies** → **Artificial intelligence**; • **Security and privacy** → *Systems security*.

Additional Key Words and Phrases: Convolutional Neural Networks (CNNs), Physical Adversarial Attack, Image Classification, Voice Recognition, Object Detection

1 INTRODUCTION

In the past few years, due to the advanced algorithms and complex mobile usage demands, Convolutional Neural Networks (CNNs) have been widely applied in various embedded recognition applications, *e.g.* image classification on smartphones, object detection on autonomous vehicles, and speech recognition on voice assistants [25, 42, 43]. However, recent research has revealed that CNN-powered recognition applications are facing a critical challenge – adversarial attacks. By injecting well-designed perturbations into input data, adversarial attacks can mislead recognition results. Recently, in order to achieve an easier injection and better attack effectiveness, the adversarial perturbations can be aggressively concentrated into a small patch area with various types and even different data modalities *e.g.* image, audio, which we define as condensed adversarial attacks. Fig. 1 illustrates a condensed adversarial attack example in a typical object detection scenario. The detector on the camera can successfully

Authors' addresses: Zirui Xu, trovato@corporation.com; Fuxun Yu; Xiang Chen, George Mason University, USA; Chenchen Liu, University of Maryland, Baltimore County, USA.

1539-9087/2022/8-ART $15.00
https://doi.org/10.1145/3555375

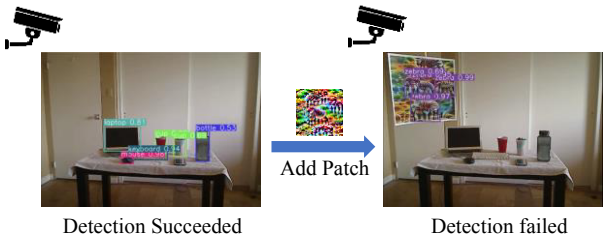

Fig. 1. Condensed Adversarial Attack Examples in Real World [18]

detect the objects in the original frame. However, when placing an adversarial patch [18] in front of the camera, the detector will totally neglect the original objects, causing a detection failure.

Be aware of the significant impacts on the recognition scenarios, many works have been proposed to detect and defend against condensed adversarial attacks [12, 21, 22, 27, 39, 44, 46, 48, 50]. However, when applying them on the practical embedded recognition scenarios, three main issues arise: First, most of these methods are particularly designed for each individual attack scenario (*e.g.* image classification or object detection) without investigating the attacking mechanism under the condensed adversarial attacks across various application scenarios. Therefore, these methods lack enough versatility to accommodate condensed attacks with different data modalities. Second, most of these methods highly rely on computation-intensive data pre-processing techniques, *e.g.* filtering [46], pixel-level entropy calculation [50] and even human efforts [39], which conflicts with the fact that embedded recognition scenarios are time-sensitive. Third, some of these methods require can only be implemented with specific settings. For example, [26] is designed only for vision transformers. [44] generally requires a small kernel size to enable a minimal upper bound of corrupted feature numbers. Therefore, considering the above three issues, the ideal defense methodology in the embedded recognition scenarios is expected to have two important characteristics: ***versatility*** and ***computation-efficiency***.

In this paper, we propose *LanCeX*, a versatile and lightweight defense solution against condensed adversarial attacks in various embedded recognition scenarios. By interpreting CNN's vulnerability and analyzing the condensed adversarial attack process, we identify the common attack mechanism behind various condensed adversarial attacks across different data modalities. Inspired by such a mechanism, we investigate the *inference inconsistencies* between the condensed adversarial attacks and the natural recognition process. Such *inference inconsistencies* further benefit the defense methodology derivation. The proposed methodology can protect the recognition application from different types of condense adversarial attacks. Even with different data modalities, it still can guide us to apply a similar defense strategy against condensed attacks.

Specifically, we have the following main contributions in this work:

- Firstly, we interpret CNN's intrinsic vulnerability and analyze the attack processes in the multiple recognition scenarios. We find that condensed adversarial attacks with distinct data modalities inherently share a common attacking mechanism.
- By reviewing such a common mechanism, we build the metrics to evaluate the *inference inconsistencies* between the condensed adversarial attack and the natural recognition process.
- We propose a versatile and lightweight defense solution – *LanCeX*, which includes a detection stage to identify the potential adversarial patterns. Based on the detection result, a corresponding data recovery methodology is used to mitigate the adversarial perturbations in the input data.
- In order to show the versatility of *LanCeX*, we apply the proposed defense methodology into three practical recognition scenarios, namely, image classification, object detection, and audio recognition. In each scenario, we quantitatively analyze method's computational complexity to illustrate the lightweight computation cost.

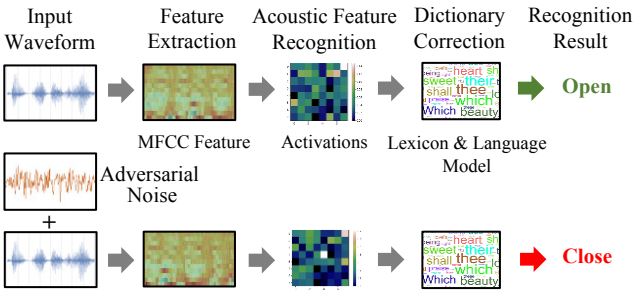

Fig. 2. Condensed Adversarial Attack in Audio Recognition

Experiments show that our method can achieve an average 91%, 85%, and 90% detection successful rate for attack detection and optimal accuracy recovery in three computation scenarios, respectively. Moreover, our method is at most 3× faster than the state-of-the-art defense methods, which is feasible to various resource-constrained and time-sensitive embedded systems, such as mobile devices.

2 BACKGROUND AND RELATED WORKS

2.1 Condensed Adversarial Attacks

Adversarial attacks started to arouse researchers' general concern with adversarial examples, which were first introduced by [10]. The adversarial examples were designed to project prediction errors into input space to generate noises, which can perturb digital input data (*e.g.*, images and audio clips) and manipulate prediction results. Since then, various adversarial attacks were proposed, such as L-BFGS [37], FGSM [10], CW [4], *etc.* Most of these adversarial attacks share a similar mechanism, trying to cause the most error increment within model activation and meanwhile regulate the imperceptible noises within the input space. Though the generated noises are invisible to human perception, these methods are less impractical for attacking real-world recognition applications due to two main reasons: 1) the environmental noises in the physical world such as light may disrupt the generated adversarial perturbations [17, 33, 36]; 2) this kind of adversarial perturbation need to manipulate the entire image and usually can only be effective for a single target image. However, in the physical world, the attacker cannot directly manipulate input data of the prediction model (such as the images collected by on-board camera of an autonomous car).

Recently, with the more aggressive methods, the generated adversarial noises no longer restrict themselves to imperceptible changes. Instead, an image-independent patch is generated which is extremely salient to CNN. Therefore, they are easily implemented in the various recognition applications from either the digit domain or the physical world, which is referred to as the condensed adversarial attack.

Attack in Image Classification: In the image classification scenario, [9] first leveraged a masking method to concentrate the adversarial perturbations into a small area and implement the attack on real traffic signs with taped graffiti. [15] focused on creating localized attacks covering as little as 2% of the image area instead of generating an image-size noise pattern. [3] extended the scope of condensed attacks with adversarial patches. With more aggressive and concentrated patterns, these patches can be attached to physical objects arbitrarily and have strong model transferability.

Attack in Object Detection: When extending from image classification scenario to object detection, condensed adversarial attacks are also massively existed. There are two types of condensed adversarial attacks in object detection scenarios: targeted and untargeted. A targeted attack can mislead the detector to only locate the

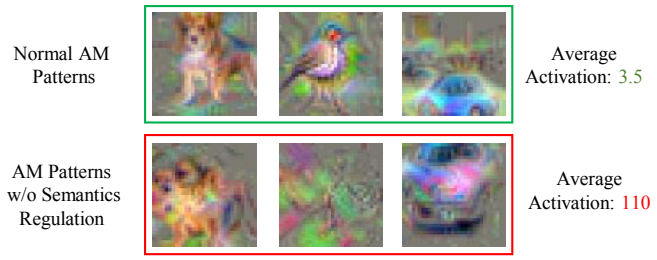

Fig. 3. Visualized Neuron's Input Patterns by Activation Maximization Visualization

adversarial patch and make a targeted class prediction while an untargeted attack will disrupt the detector to output any results. Several related works have been proposed to prevent the detectors from correct detection. Song *et al.* proposed a disappearance and creation attack to fool Yolov2 [32] in traffic scenes [36]. Chen *et al.* adopt the expectation over transformation method to create more robust adversarial stop signs, which misled faster R-CNN [34] to output errors [5]. Liu *et al.* proposed DPatch, which explicitly created patches that do not overlap with the objects of interest but can eliminate the original object localization and achieve both targeted and untargeted condensed attack [19].

Attack in Audio Recognition: Beyond the aforementioned image-related cases, some condensed adversarial attacks also have been proposed in audio recognition areas. Yakura *et al.* proposed an audio condensed adversarial attack that can still be effective after playback and recording in the physical world [45]. [47] first generated audio adversarial commands and further embedded them in a normal song that can be played through the air.

Compared to the traditional imperceptible noise-based adversarial attacks, condensed adversarial attacks reduce the attack difficulty and further impair the practicality and reliability of CNN-powered recognition applications.

2.2 Condensed Adversarial Attack Defense

Defense in Image Classification: There are several works that have been proposed to defense such condensed adversarial attacks in the image classification process [12, 21, 22, 26, 27, 30, 39, 44]. Naseer *et al.* proposed a local gradients smoothing scheme against condensed adversarial attacks [27]. By regularizing gradients in the estimated noisy region before feeding images into CNN inference, their method can eliminate the potential impacts from adversarial attacks. McCoyd *et al.* proposed a cross-verification-based defense method [22]. It iteratively occludes different portions of the input image and analyzes the prediction results. Since the adversarial patch will not be occluded at most times, they considered the minority of predictions as the true result. Rao *et al.* leveraged adversarial training to increase the robustness of networks which can defense the adversarial patch attack [30]. tao *et al.* proposed an interpretability-based adversarial detection method which can identify the neurons in CNN that correspond to human perceptible attributes [39]. However, this method is only for detection and is specifically designed for human face recognition. Ma *et al.* developed a novel adversarial detection method which leverage neural network invariant checking [21]. Specifically, they define two types of invariants: value invariants and provenance invariants in the model inference process. However, in order to obtain the two invariants, the proposed method needs to calcualte the activation distribution of each layer and derive sub-model for each layer, which introduce a lot of extra computation cost. Although these methods can achieve effective defense performance, most of them either introduce a tedious preprocessing or training process [22, 30] or lack enough flexibility to different adversarial patch settings (*e.g.* sizes) [26, 44]. Moreover, they are designed for solving specific adversarial attacks which are not integrated for different adversarial attack scenarios. Recently, Mu *et al.* proposed a defense method which measures the statistic heterogeneity between adversarial patch and benign image patches and replace the adversarial patch with average values from other patches [26]. Although this

method requires zero additional parameters or extra training effort, it can be only applied on vision transformer which has limitation on application scenarios. Moreover, [44] proposes a provable defense framework. In [44], the authors claim that a network with a small kernel size can impose a bound on the number of features that can be corrupted due to adversarial patches. In order to correctly mask the corrupted features, it requires most of the features are benign. In our paper, we have a similar finding: a small area of adversarial patch can generate extremely high activation values and will gradually overwhelm the benign values during the network inference. However, our method doesn't depend on kernel size, and we still use the default kernel size for each model (i.e. 3x3).

Defense in Object Detection: Compared to image classification, attacking an object detector is more complicated. The reason resides in the detector architecture: detectors first locate the objects with different sizes at different locations on the image and then conduct classification. In that case, the number of the attacked objects in this case is much larger than the ones in the image classification scenario.

Currently, only a few works have been proposed for defending the adversarial attack in object detection [6, 50]. Zhou *et al.* proposed a defense method which consists of an entropy-based proposal component and a gradient-based filtering component. The first component is used to identify the potential patch location and then the second component can remove the patch by using filtering techniques. However, this method highly depends on the sliding window and entropy calculation across the entire input image, which introduces a huge computation workload. Therefore, it is inefficient to deploy this method on the embedded recognition scenarios. Chiang *et al.* proposed a pixel masking method to eliminate condensed adversarial path. However, the path localization step in this method needs to introduce a *U-Net* inference, which is computation-intensive [6].

Defense in Audio Recognition: Compared with images, the audio data requires extra processing efforts for recognition. Fig. 2 shows a typical audio recognition process and the corresponding condensed adversarial attack. The audio waveform is first extracted as Mel-frequency Cepstral Coefficient (MFCC) features. Then we leverage a CNN to achieve acoustic feature recognition, which can obtain the candidate phonemes. Finally, a lexicon and language model is applied to obtain the recognition result "open". When the adversarial noise is injected into the original input waveform, the final recognition result is misled to "close".

Several works have been proposed to detect and eliminate such adversarial attacks [29, 46, 48]. Zeng *et al.* leveraged multiple Automatic Speech Recognition (ASR) systems to detect audio condensed adversarial attack based on a cross-verification methodology [48]. However, their method lacks certain versatility which cannot detect the adversarial attacks with model transferability. Yang *et al.* proposed an audio adversarial attack detection and defense method by exploring the temporal dependency in audio adversarial attacks [46]. However, their method requires multiple CNN recognition inferences which is time-consuming. Rajaratnam *et al.* leveraged the random noise flooding to defense audio adversarial attacks [29]. Since the ASR systems are relatively robust to natural noise while the adversarial noise is not, by injecting random noise, the functionalities of adversarial noise can be destroyed. However, this method doesn't achieve a good defense performance.

In this paper, we will reveal that most condensed adversarial attacks across different data modalities actually share a common attacking mechanism, thus provides opportunities to establish a unified defense methodology to accommodate attacks in multiple application scenarios.

3 INTERPRETATION ORIENTED CONDENSED ADVERSARIAL ATTACKS ANALYSIS

In this section, we first interpret CNN vulnerability and identify that neurons can be activated by condensed adversarial patterns with abnormal distinguished activation magnitudes. Furthermore, we leverage the attention mechanism to reveal that all condensed adversarial attacks across different data modalities inherently share a common attacking mechanism. Finally, inspired by this mechanism, we propose metrics to measure the inference

Fig. 4. Visualizing the Most Activated Neuron's Class Preference

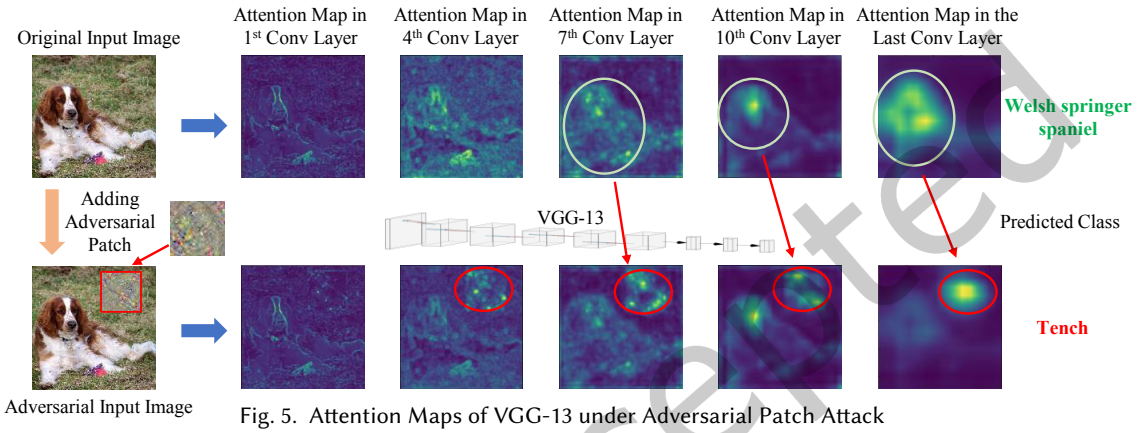

Fig. 5. Attention Maps of VGG-13 under Adversarial Patch Attack

inconsistencies between the condensed adversarial attack and natural recognition process, which provides the design motivation for the following defense methodology development.

3.1 CNN Vulnerability Interpretation

Interpretation and Assumption: In a typical image or audio recognition process, CNN extracts features from the original input data and gradually derives a prediction result. However, when injecting condensed adversarial perturbations into the original data, CNN will be misled to a wrong prediction result. In Fig. 1, by comparing with the original input, we find that an adversarial patch usually has no constraints in color, pattern content, *etc.* However, by sacrificing the semantic structures, such adversarial patches can cause abnormal activations during CNN recognition process. *Therefore, we make an* ***assumption*** *that CNN lacks qualitative semantics distinguishing ability which can be significantly activated by the non-semantic adversarial patch during CNN inference.* To better interpret this vulnerability, we major focus on a typical image condensed adversarial attack – adversarial patch attack as an example.

Assumption Verification: To verify our assumption, two sub-questions need to be addressed: 1) Whether the non-semantic input patterns will lead to abnormal activations while the semantic input patterns generate normal activations. 2) Whether the condensed adversarial patches are such non-semantic patterns that activate the neurons with different preferred classes.

Since the above two questions are highly related to neurons in CNNs, we adopt a visualized CNN semantic analysis method – Activation Maximization Visualization (AM) [7] to investigate the semantic of each neuron in CNN. AM provides the intuitive CNN interpretation by visualizing each neuron's most sensitive activation pattern. With AM, we can identify neurons' preferred features and corresponding semantics, and infer their activation exclusiveness to particular classes. The generation process of pattern $V(N_i^l)$ can be considered as synthesizing an input image to a CNN model that delicately maximizes the activation of the ith neuron N_i^l in the

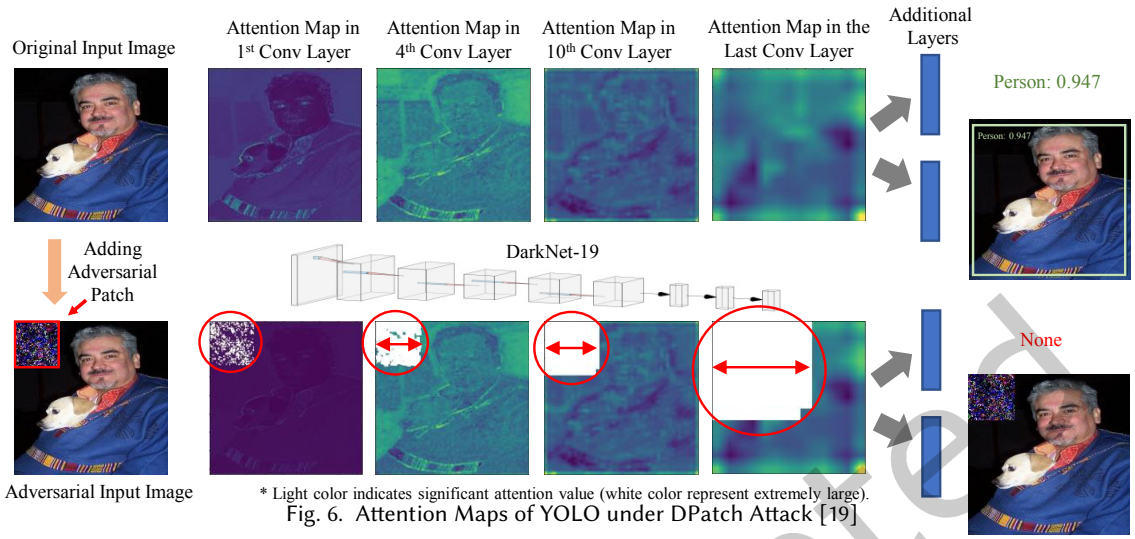

Fig. 6. Attention Maps of YOLO under DPatch Attack [19]

l^{th} layer. Specifically, this process can be defined as:

$$V(N_i^l) = \arg\max_X A_i^l(X), \qquad X \leftarrow X + \eta \cdot \frac{\partial A_i^l(X)}{\partial X} \tag{1}$$

where, $A_i^l(X)$ is the activation of N_i^l from an input image X, η is the gradient ascent step size.

Fig. 3 shows the visualized semantic input patterns by using AM. As the original AM method is designed for semantics interpretation, many feature regulations and hand-engineered natural image references are involved in generating interpretable visualization patterns. Therefore we can get three AM patterns with an average activation magnitude value of 3.5 in Fig. 3 (a). The objects in the three patterns indicate they have clear semantics. However, when we remove these semantics regulations in the AM process, we obtain three different visualized patterns as shown in Fig. 3 (b). We can find that these three patterns are non-semantic, but they have significant abnormal activation with an average magnitude value of 110. This phenomenon addresses the first question that CNN neuron lacks the semantic distinguishing ability, therefore ***non-semantic*** inputs patterns will generate abnormal and significant activation.

To further identify the second question, we investigate the CNN neuron activation status during the condensed adversarial attack process. Fig. 4 shows the AM patterns of neurons with the largest activation at the last convolutional layer of VGG-13 [35]. The salient object in the natural image is a dog, and the neurons that prefer dog class are highly activated. However, with the adversarial patch injected, the deception-class (bird) related neurons are significantly activated and quickly overwhelm the original neurons, therefore mislead the prediction results.

Combining the above two phenomenons, we can verify our assumption that *CNN prediction process is mainly based on indiscriminate quantitative activation, lacking necessary qualitative semantics distinguishing ability.* Therefore, adversarial patches can discard semantics' original natural input patterns and cater to overwhelming activation with condensed non-semantic patterns.

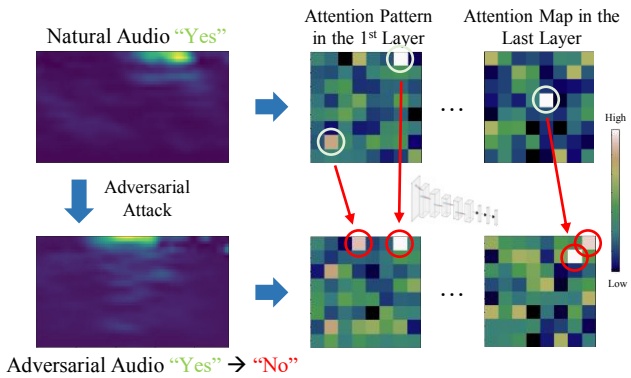

Fig. 7. Attention Maps of AlexNet under Audio Adversarial Attack

3.2 Attack Process Analysis with Attention Mechanism

In order to reveal how condensed adversarial patches utilize such CNN vulnerability to manipulate the recognition process, we further examine the adversarial attack processes in three computation scenarios (*i.e.* image classification, object detection, and audio recognition) via attention mechanism[1].

In our analysis, we calculate the attention values of the feature map from the spatial aspect. Given an intermediate input feature map $F \in R^{C*H*W}$ (here, C, H, and W is the channel depth, height, and width of the feature maps), the spatial attention map is defined as:

$$A_s(F, x, y) = \frac{1}{C} \sum_{i}^{C} F_{x,y}(i), \tag{2}$$

where i represents the ith feature map and (x, y) indicates the spatial location. Therefore, a feature map F would have a spatial attention map with $H \times W$ size. Higher attention value indicates the higher activation generated by the object on the corresponding location.

Attention Analysis in Image Classification: Fig. 5 shows the layer-level attention maps in a typical image classification process with VGG-13 [35]. The original image class is "Welsh springer spaniel" and the significant patterns (indicated by the higher brightness) are gradually focusing on the dog's head and body (surrounded by the green circles) from shallow layers to deep layers. However, when attaching an adversarial patch on the top right corner of the original image, we can find the brightness level decreases at the dog's location. Moreover, from the shallow layers to deep layers, the significant attention patterns shifted from dog to patch location (surrounded by the red circles). Eventually, at the last convolutional layer, the most significant attention patterns are totally concentrated on the patch and the final prediction result is manipulated to "Tench". Through this example, we can find the adversarial patch can gradually draw higher attention from CNN and finally change the prediction result.

Attention Analysis in Object Detection: In image classification, only a single object needs to be classified. Therefore, if the attention patterns at the patch location are more significant than the original object, the adversarial attack can manipulate the prediction result. However, in the object detection scenario, the network usually needs to locate and classify multiple objects. Therefore, the adversarial patches are required to affect the entire image detection process. We first examine the attack process of untargeted adversarial attacks in object

[1]Notably, our attention mechanism doesn't have trainable parameters similar to [14]. Instead, we calculate it as the feature map activation patterns.

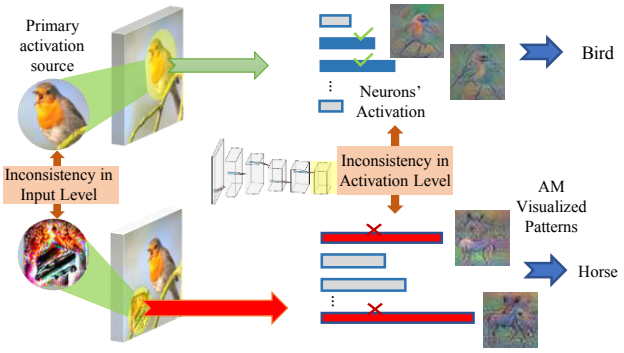

Fig. 8. Image Adversarial Patch Attack

detection. Fig. 6 illustrates the attention maps of YOLO [31] under a well-known condensed adversarial attack (DPatch) [19]. The light colors indicate significant attention patterns (white color means the attention values are extremely large). We can easily find that the adversarial patch directly has the most significant attention patterns at the first layer of the neural network. Moreover, when the layer becomes deeper, the size of the attention pattern of the adversarial patch constantly increases and covers more than half of the feature map at the last convolutional layer in the feature extraction network. This characteristic illustrates why the adversarial patch can affect object detection processes on the entire image. We also investigate the attention maps under targeted condensed adversarial attacks in object detection scenario and can obtain a same conclusion.

Attention Analysis in Audio Recognition: As aforementioned, different from the image-related recognition process, the raw audio waveform needs to be pre-processed and converted into a frequency spectrum before feeding into the neural network to conduct recognition. Fig. 7 shows the attention maps comparison of AlexNet [16] when feeding original audio and adversarial audio. The original audio is recognized as "Yes" while the adversarial attack manipulates it to "No". The patterns surrounded by the white circles have the most significant attention values. However, by adding the adversarial attack, the most significant attention locations are shifted (red circles) and the attention distribution on the entire attention maps also changed.

According to the above attention-based attack analysis, we can find that all the condensed adversarial attacks across different data modalities actually share a ***common attacking mechanism***: *Since the condensed adversarial patterns can introduce abnormal activation and even overwhelm the natural input patterns, they will gradually manipulate the neural networks' attention from the location of the original salient objects to the patch's spatial location.*

3.3 Inference Inconsistency between Condensed Adversarial Attack and Natural Input Recognition

The common attacking mechanism reveals that the condensed adversarial patterns will cause different recognition process compared to the original natural inputs. In order to identify the adversarial input, it is necessary to investigate such difference by evaluating the inference inconsistencies between adversarial attacks and natural recognition process.

Inference Inconsistency Identification: Fig. 8 shows a typical adversarial patch based condensed attack. The patterns in the left circles are the primary activation sources from the input images, and the bars on the right are the neurons' activations at the last convolutional layer. From the perspective of the input patterns, we identify the first inference inconsistency: the significant difference between the adversarial patch and primary activation source on the original image, which is referred to as ***Input Semantic Inconsistency***. From the aspect

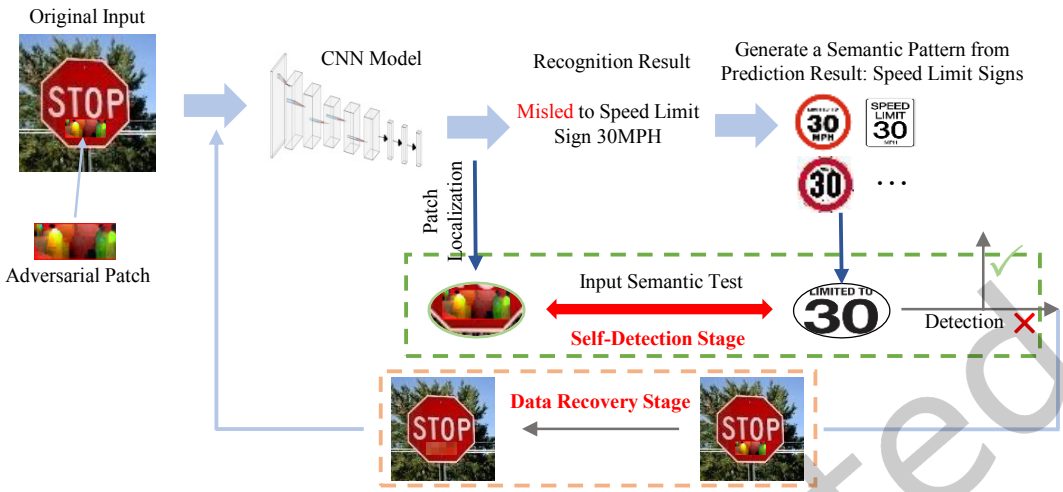

Fig. 9. Condensed Adversarial Attack Defense for Traffic Sign Example

of prediction activation magnitudes, we observe another inference inconsistency, namely, ***Prediction Activation Inconsistency***.

Inconsistency Metrics Formulation: We further define two metrics to indicate above two inconsistencies' degrees.

1) Input Semantic Inconsistency Metric: This metric measures the input semantic inconsistency between the non-semantic adversarial patches and the semantic local input patterns from the natural image. It can be defined as:

$$D(P_{pra},P_{ori})=1-S(P_{pra},P_{ori}),P_{pra}\xleftarrow{\Re}\Phi{:}A_i^l(p),P_{ori}\xleftarrow{\Re}\Phi{:}A_i^l(o), \tag{3}$$

where P_{pra} and P_{ori} represent the input patterns from the adversarial input and the original input. $\Phi : A_i^l(p)$ and $\Phi : A_i^l(o)$ represent the set of neurons' activation produced by the adversarial patch and the original input, respectively. $\Re$ maps neurons' activation to the primary local input patterns. S represents a similarity metric.

2) Prediction Activation Inconsistency Metric: The second inconsistency is on the activation level, which reveals the activations' magnitude distribution inconsistency in the last convolutional layer between the adversarial input and the original input. We also use a similar metric to measure it as:

$$D(f_{pra},f_{ori})=1-S(f_{pra},f_{ori}),f_{pra}\sim\Phi{:}A_i^l(p),f_{ori}\sim\Phi{:}A_i^l(o), \tag{4}$$

where f_{pra} and f_{ori} represent the magnitude distribution of activation in the last convolutional layer generated by the adversarial input and the original input data.

For the above two inconsistency metrics, we can easily obtain P_{pra} and f_{pra} since they come from the input data. However, P_{ori} and f_{ori} are not easily to get because of the variety of the natural input data. Therefore, we need to synthesize the standard input data which can provide the semantic input patterns and activation magnitude distribution. The synthesized input data for each prediction class can be obtained from a standard dataset. By feeding CNN with a certain number of input from the standard dataset, we can record the average activation magnitude distribution at the last convolutional layer. Moreover, we can locate the primary semantic input patterns for each prediction class.

4 *LANCEX* DEFENSE METHODOLOGY

In this section, guided by the two inconsistency metrics, we propose our versatile defense methodology which consists of a self-detection stage and a data recovery stage in the CNN decision-making process.

4.1 Self-Detection for Condensed Adversarial Attack

Since both the input patterns and the prediction activations are directly calculated from CNN inference, a recognition application can easily achieve self-detection for condensed adversarial attacks. The entire self-detection stage can be summarized as following 4 main steps:
Step 1: The input data are fed into CNN for one inference and we can obtain the prediction class; *Step 2:* During the inference process, CNN can locate the primary activation sources from the practical input and calculate the activations in the last convolutional layer. *Step 3:* CNN leverages the proposed metrics to measure the two inconsistencies $D(P_{pra}, P_{ori})$ and $D(f_{pra}, f_{ori})$ between the practical input and the synthesized data with the predicted class. *Step 4:* Once any inconsistency exceeds the given threshold, CNN will consider the input as an adversarial input.

Fig. 9 illustrates the method flow of CNN self-detection in a traffic sign recognition example. For one input image, CNN will first conduct inference process and get the prediction result (*e.g.* 30 MPH). Then a synthesized traffic sign pattern with 30 MPH is retrieved. After the inference, the significant activation source (green circle) will be located. Next, the detection stage will calculate the input semantic inconsistency with the expected semantic patterns (right circle) according to the prediction result. If the inconsistency exceeds a pre-defined threshold, CNN will identify the input data as an adversarial input and further conduct the data recovery process to mitigate the adversarial perturbation from the input image.

4.2 Adversarial Input Mitigation via Data Recovery

As aforementioned, after a condensed adversarial attack has been detected by the self-detection stage, the data recovery method is further applied to mitigate the adversarial perturbation and thereby recover the attacked input data. Due to different characteristics between the image and audio data, we leverage image inpainting method to recover the input image while using the activation denoising method to restore the input audio. We will derive specific defense methods from such methodology for three embedded recognition scenarios, *i.e.* image classification, object detection, and audio recognition in the following sections.

4.3 Computational Complexity Analysis

Computation cost is critical to the adversarial defense approaches for embedded recognition scenarios. Therefore, we leverage computational complexity to evaluate the methodology's total computation cost. A low computational complexity indicates a small computation workload, proving the proposed methodology is lightweight. In our defense methodology, the computational complexity is mainly contributed by the inner stages such as the CNN inference, inconsistency metrics calculation, and data recovery. In the following three scenarios, we will specifically analyze their computation complexity.

5 SCENARIO 1: DEFENSE AGAINST CONDENSED ADVERSARIAL ATTACKS IN IMAGE CLASSIFICATION

5.1 Defense Process in Image Classification

Primary Activation Pattern Localization: For the image condensed adversarial attacks defense, we mainly depend on the ***input semantic inconsistency*** in input pattern level. Therefore, we need to locate the primary activation source from the input image by adopting a CNN activation visualization method – Class Activation

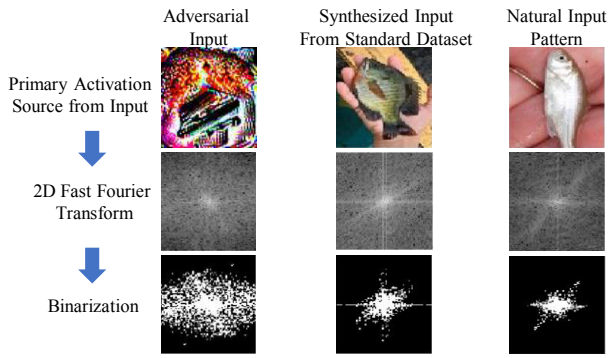

Fig. 10. The Results after 2D Fast Fourier Transform

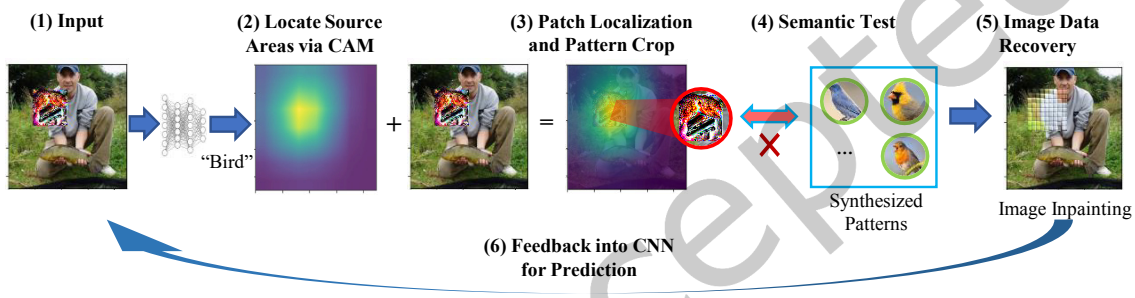

Fig. 11. Adversarial Patch Attack Defense in Image Classification

Mapping (CAM) [49]. Let $A_k(x, y)$ denotes the value of the kth activation in the last convolutional layer at spatial location (x, y).

We can compute a sum of all activations at the spatial location (x, y) in the last convolutional layer as:

$$A_T(x, y) = \sum_{K}^{1} A_k(x, y), \tag{5}$$

where K is the total number of activations in the last convolutional layer. The larger value of $A_T(x, y)$ indicates the activation source in the input image at the corresponding spatial location is more important for classification result. For a natural input, it is the object pattern's location while it is the adversarial patch's location for an adversarial input.

In order to conduct further self-detection and data recovery, we need to determine the specific size of such primary activation pattern area. During this step, we can first identify the location (x_m, y_m) with the highest $A_T(x, y)$ on the input image. Next, if the adversarial patch size and shape are given, we can easily select the areas with the same size and shape based on the location (x_m, y_m). However, when patch size and shape are not prior knowledge, we need to first calculate the average activation value A_a across the entire image. Then, start from (x_m, y_m), the surrounding locations whose values are higher than A_a are considered in the pattern area.

Inconsistency Derivation: According to our preliminary analysis, the input adversarial patch contains much more high-frequency information than the natural semantic input patterns. We first leverage 2D Fast Fourier Transform (2D-FFT) [41] to transfer the patterns from the temporal domain to the frequency domain and thereby concentrate the low-frequency components together. Then we convert the frequency-domain pattern to a binary pattern with an adaptive threshold. Fig. 10 shows a conversion example, including adversarial patterns, expected

synthesized patterns with the same prediction result, and natural input patterns. For binary patterns, we can observe the significant difference between adversarial input and semantic synthesized input. Therefore, we replace $S(I_{pra}, I_{ori})$ with Jaccard Similarity Coefficient (JSC) [28] and propose our image inconsistency metric which is formulated as:

$$D(P_{pra},P_{exp})=1-JSC(P_{pra},P_{exp})=\frac{|P_{pra}\bigcup P_{exp}|-|P_{pra}\bigcap P_{exp}|}{|P_{pra}\bigcup P_{exp}|}, \tag{6}$$

where P_{exp} is the synthesized semantic pattern with predicted class. $P_{pra} \bigcap P_{exp}$ means the numbers of pixels where the pixel value of P_{pra} and P_{exp} both equal to 1. For image classification, the input semantics patterns from expected prediction results can be referred by the ground-truth dataset. By testing a CNN model with a certain number of data for once, we can record model's preferred natural semantic input pattern by leveraging the CAM and size determination methods which are discussed before.

With the above inconsistency metric, we propose our specific defense methodology which contains self-detection and image recovery, and is described in Fig. 11.

Self-Detection: For each input image, we apply CAM to locate the source location of the largest model activations. Then we crop the image to obtain patterns with maximum activations. During the semantic test, we calculate the inconsistency between P_{pra} and P_{exp}. If it is higher than a predefined threshold T_{ic}, we consider an adversarial input detected. The threshold value T_{ic} is determined by the prepossessing works. Specifically, for a given dataset (*e.g. ImageNet-10*), we first generate the synthesized semantic patterns for each class (*e.g.* 100 patterns in our experiment). Then we calculate the inconsistence value across patterns in each class and assign the average value as $D_{avg}^{ground}(i)$, where i indicates the ith class. Next, we generate a certain number of adversarial patches for each class (10 in our experiment) and calculate the inconsistency value between them and the target synthesized semantic patterns. We consider the average inconsistency values as $D_{avg}^{adv}(i)$. Based on above settings, the value range of threshold T_{ic} for each class is between $D_{avg}^{ground}(i)$ and $D_{avg}^{adv}(i)$.

Data Recovery: After the patch is detected and located, we conduct image data recovery by directly removing the patch from the original input data. In our case, considering the requirement of lightweight computation workload, two potential image inpainting methods are adopted: *Zero Mask* and *Telea* method [40] (shown in Fig. 12).

Zero Mask directly sets all pixel values inside the patch area as 0, which achieves the smallest computation workload and has been already applied in the recent adversarial patch defense work [50]. As Fig. 12 shows, the masked area will not affect the image classification results if the patch location is outside the object. However, when the attack is inside the object, directly masking the pattern with black color will degrade the further prediction performance. On the other hand, *Telea* method achieves better inpanting performance while slightly sacrificing computation efficiency. We will evaluate the recovery performance of the two methods in terms of effectiveness and efficiency in Section 8.

5.2 Computational Complexity Analysis

The total computation complexity of the defense process in the image classification scenario is contributed by the following four steps: CNN inference, maximum activation pattern localization, inconsistency metric calculation, and image interpolation. We model each step's computational complexity as following:

CNN Inference: When the input image is first fed into CNN, the inference computational complexity C_C is:

$$C_C \sim O(\sum_{i=1}^{L}\sum_{j=1}^{n_i}(r_i^j)^2 n_{i-1} h_i^j w_i^j), \tag{7}$$

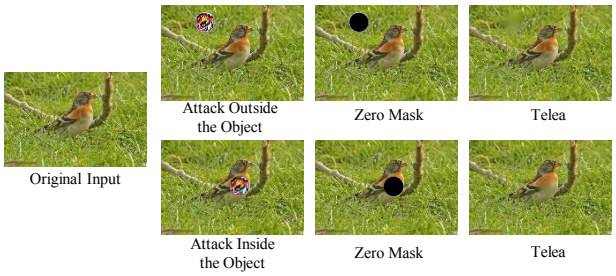

Fig. 12. The Image Data Recovery Performance by Two Inpainting Methods

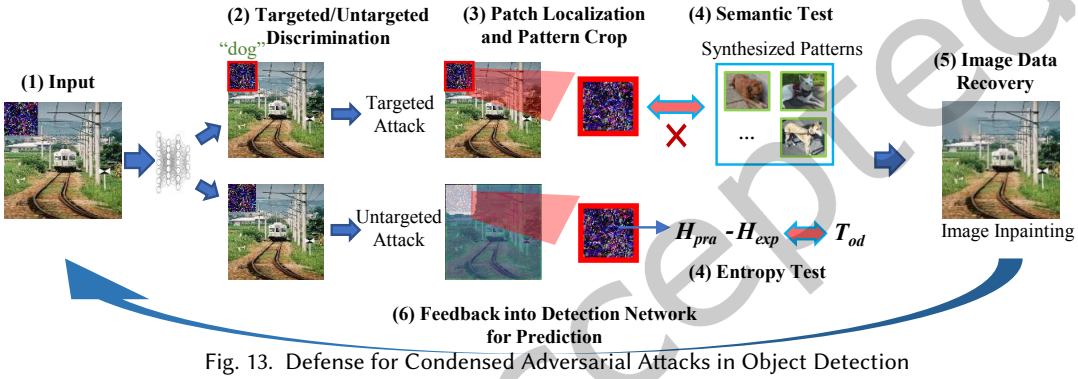

Fig. 13. Defense for Condensed Adversarial Attacks in Object Detection

where $(r_i^j)^2$ represents the jth filter's kernel size in the ith layer, $h_i^j w_i^j$ denotes the corresponding size of output feature map, L is the total layer number and n_i is the filter numbers in the ith layer.

Primary Activation Pattern Localization: Since computation complexities of other operations such as cropping are negligible, we consider CAM contributes the primary computational complexity in this step. In CAM, each spatial location (x, y) in the last convolutional layer is the weighted sum of K activations. Therefore, the total computational complexity is: $C_M \sim O(Kh_L^{n_L} w_L^{n_L})$, where $h_L^{n_L} w_L^{n_L}$ is the size of the feature map at the last convolutional layer.

Inconsistency Metric Derivation: This step consists of 2D-FFT calculation and JSC calculation. According to the analysis in [20], the computational complexities of the above two processes can be approximated to $C_F \sim O(NlogN)$ and $C_J \sim O(n_a logn_a)$, where N and n_a represents N pixel number in the input image and maximum activation pattern, respectively.

Image Inpainting: For *Zero Mask*, the total operation number is $C_z \sim O(n)$, where n is the pixel number inside the patch. For *Telea* method, the total computation complexity is $C_t \sim O(3bn)$, where b represents the total operation number when inpainting each pixel.

Comparing with activation localization, metric derivation, and image inpainting, the computational complexity of CNN inference dominates the entire computational complexity in the image scenario. Since our methodology only involves one CNN inference, it has the same-order computation workload as the normal CNN prediction.

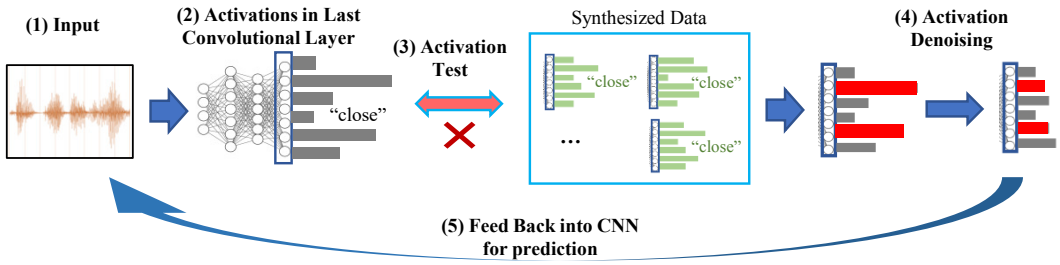

Fig. 14. Defense for Condensed Adversarial Attacks in Audio Recognition

6 SCENARIO 2: DEFENSE AGAINST CONDENSED ADVERSARIAL ATTACK IN OBJECT DETECTION

6.1 Defense Process in Object Detection

Potential Adversarial Patch Localization: Due to different attack results, the specific defense methods for targeted adversarial attacks and untargeted adversarial attacks need to be discussed separately.

1): For *targeted attack*, since only the adversarial patch can be located by the bounding box, the size and location information of the adversarial patch can be directly obtained. Then we depend on ***input semantic inconsistency*** and leverage Eq. 6 to calculate the inconsistency. Next, the neural network can further conduct the self-detection process.

2): For *untargeted attack*, both adversarial patch and normal objects will not be located and classified by the detection network. However, according to the attention map analysis, the adversarial patch can still attract neural network's attention from original salient objects during the feature extraction process. We can leverage CAM method at the shallow layers (*e.g.* 1st or 2nd convolutional layers) to locate the spatial location of the adversarial patch on the input image. The reason why we apply CAM at the shallow layers is that the activation pattern of the adversarial patch will constantly become larger when layers go deeper. Therefore, the spatial location calculated at the last convolutional layer is not accurate. Since an untargeted attack doesn't output a particular prediction class, it is impossible to generate the synthesized patterns from the dataset. Fortunately, since adversarial patch contains higher information than the background and other objects [50], we can define an ***input information entropy inconsistency***. Specifically, we first leverage Discrete Entropy [11, 23] to define the information entropy H_{pra} of a potential adversarial patch area:

$$H_{pra} = -\frac{1}{3}\sum_{i=1}^{3}\sum_{j=0}^{255}(n_{ij}/N)log_2(n_{ij}/N) \tag{8}$$

where i represents one of the RGB color channels and j represents the specific pixel value. n_{ij} indicates the number of pixels with j gray value in channel i and N means the total pixel number on the potential area. Then we compare the calculated entropy H_{pra} with a synthesized H_{exp}. Specifically, H_{exp} can be obtained by randomly selecting some images from dataset and calculated their average information entropy.

Based on the above localization and inconsistency analysis, we further propose our defense methodology for condensed adversarial attacks in object detection scenario, which is described in Fig. 13.

Self-Detection: For a targeted attack, the located pattern conducts a semantic test. If the computed input semantic inconsistency value $D(P_{pra}, P_{exp})$ is higher than a predefined threshold T_{odt}, the network considers it has an adversarial input. For an untargeted attack, we calculate the inconsistency between the H_{pra} and H_{exp} and compare it to another predefined threshold T_{odu}.

Data Recovery: Since the input data in the object detection scenario is the image as well, we can leverage the same image inpainting methods proposed in Section V. Different from the image classification scenario that usually only one object needs to be classified at the center of the input image, there are multiple objects in the object detection scenario. Moreover, some of these objects have small scales and locate at the border of the input image. Therefore, the adversarial patch will easily cover part of small-scale objects. If we directly mask the patch area with black by using *Zero Mask*, the detection performance will significantly degrade. Due to this consideration, we only apply *Telea* method during data recovery in the object detection scenario.

6.2 Computational Complexity Analysis

The computational complexity in the object detection scenario is mainly determined by the CNN inference, inconsistency metric calculation, and data recovery. Therefore, we model the computational complexity as following:

CNN Inference: In object detection scenario, the computational complexity C_C can still be calculated via Eq. 7. However, the layer number L is usually larger than the image classification since object detection includes two sub-steps: feature extraction and detection.

Inconsistency Metric Derivation: For a targeted attack, the computation complexity is the same as the image classification scenario. However, for an untargeted attack, the computational complexity introduced by information entropy calculation is $C_I \sim O(3nlog_2n)$, where n represents the pixel number in the adversarial patch area.

7 SCENARIO 3: DEFENSE AGAINST CONDENSED ADVERSARIAL ATTACK IN AUDIO RECOGNITION

7.1 Defense Process in Audio Recognition

Inconsistency Derivation: Different from images, the audio data requires more processing efforts. As Fig. 2 shows, during the audio recognition, the input waveform needs to pass Mel-frequency Cepstral Coefficient (MFCC) conversion to be transferred from the time domain into the time-frequency domain. In that case, the original input audio data will loss semantics after the MFCC conversion. Therefore, we leverage the **prediction activation inconsistency** to detect the audio condensed adversarial attacks.

Specifically, we measure the activation magnitude distribution inconsistency between the practical input and the synthesized data with the same prediction class. We adopt a popular similarity evaluation method - Pearson Correlation Coefficient (PCC) [2] and the inconsistency metric is:

$$D(f_{pra},f_{exp})=1-PCC(f_{pra},f_{exp})=1-\frac{E[(f_{pra}-\mu_{pra})(f_{exp}-\mu_{exp})]}{\sigma_{pra}\sigma_{exp}}, \tag{9}$$

where f_{pra} and f_{exp} represent activations in the last convolutional layer for both practical input and synthesized input. μ_a and μ_o are mean values of f_{pre} and f_{exp}, σ_{pra} and σ_{exp} are standard derivations, and E is the overall expectation.

Self-Detection: With the established inconsistency metric, we further apply the self-detection stage to CNN for the audio condensed adversarial attack. The detection flow is described as following: We first obtain activations at the last convolutional layer for every possible input word by testing CNN with a standard dataset. Then we calculate the inconsistency value $D(I_{pra}, I_{exp})$. If the model is attacked by the audio adversarial attack, $D(I_{pra}, I_{exp})$ will exceed a pre-defined threshold T_{ar}. For T_{ar}, we leverage the similar determination method. First, in a given audio dataset (*e.g.* Voice Command dataset [24]), we generate the synthesized activation distributions for each class (50 in our experiment). Then the average inconsistency value $D_{avg}^{ground}(i)$ can be calculated, where i represents the ith class. Second, we generate a certain number of adversarial audio for each class (30 in our experiment) and compute

Table 1. Detection Evaluation with Other Baseline Methods

Method	Inception-V3	VGG-16	ResNet-18
PM [12]	88%	89%	85%
NIC [21]	90%	88%	**90%**
PatchGuard [44]	90%	90%	89%
LanCeX	**91%**	**90%**	89%

Table 2. Detection Evaluation under Different Attack Settings

Setting	Inception-V3	VGG-16	ResNet-18
medium & random	91%	90%	89%
medium & fixed	93%	92%	92%
small & random	85%	83%	82%
large & random	94%	91%	92%

Table 3. Image Data Recovery Performance Evaluation

	Inception-V3		VGG-16		ResNet-18	
	Acc	Time	Acc	Time	Acc	Time
Original	9.8%	N/A	9.5%	N/A	9.8%	N/A
PM [12]	88.1%	233ms	88.7%	315ms	90.3%	461ms
PatchGuard [44]	86.8%	203ms	90.1%	**220ms**	89.6%	338ms
LanCeX(Zero Mask)	88.5%	**188ms**	87.6%	233ms	89.5%	**315ms**
LanCeX(Telea)	**91.2%**	211ms	**91.8%**	268ms	**91.3%**	357ms

*: 1. Patch Masking (PM) [12]. 2. Original means the model without applying any defense methods.

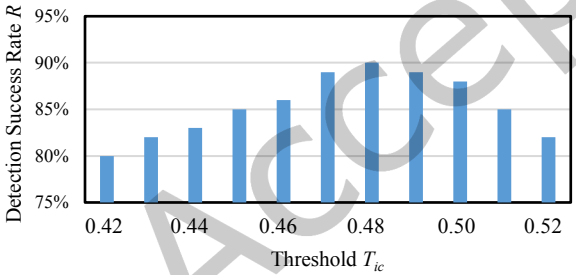

Fig. 15. The Detection Performance *w.r.t.* Different Thresholds in Image Classification Scenario

the average inconsistency value $D_{avg}^{adv}(i)$ between them and the target synthesized activation distributions. Finally, the value range of the threshold T_{ar} is between $D_{avg}^{ground}(i)$ and $D_{avg}^{adv}(i)$.

Data Recovery: After identifying the adversarial input audio, simply denying it can cause undesired consequences. Therefore, attacked audio recovery is considered as one of the most acceptable solutions. We propose a new solution - "activation denoising" as our defense method, which targets ablating adversarial effects from the activation level[2]. The activation denoising takes advantages of the aforementioned last layer activation patterns, which have stable correlations with determined predication labels.

Our adversarial audio recovery method is shown in Fig. 14: Based on detection results, we can identify the wrong prediction label, and obtain the standard activation patterns of the wrong class in the last layer. (For the best performance, we locate the top-k activation index.) Then we can find the activations with the same index. These activations are most potentially caused by the adversarial noises and supersede the original activations. Therefore, we suppress these activations to resurrect original ones.

[2]It should be noted that the proposed defense method here could not recover the original input. We name it "recovery" to keep inconsistency with the previous sections.

7.2 Computational Complexity Analysis

The computational complexity in the audio scenario is mainly determined by the CNN inference and the inconsistency metric calculation, since other steps directly manipulate limited activation values with negligible computation workload involved. Therefore, we model the computational complexity as following:

CNN Inference: Since the audio has same inference process in CNN, we can use the same model in image scenario to measure the computation complexity in the audio scenario.

Inconsistency Metric Derivation: The computation complexity of this step is contributed by the PCC calculation, which can be formulated as $C_P \sim O({n_L}^2)$, where n_L is the activation number in the last layer.

8 EXPERIMENT AND EVALUATION

In this section, we evaluate *LanCeX* in terms of effectiveness and efficiency against condensed adversarial attacks in three computation scenarios: image classification, object detection, and audio recognition.

8.1 Defense Evaluation for Image Classification

Experiment Setup: The condensed adversarial attack method we defense in this section is adversarial patch attack [3]. We iteratively conduct attack process on Inception-V3 [38] with 1000 randomly selected images from ImageNet training set to generate "cat" target adversarial patch. The generated patch with high transferability are utilized to attack other two models: VGG-13 [35] and ResNet-18 [13]. During the evaluation, we attach the generated patch on 500 images as adversarial examples and combine with other 500 natural images. All these evaluation images are chosen from ImageNet validation set. Specifically, in order to simulate different attack setting, the adversarial patches are generated with three different sizes (small: 40×40, medium: 60×60, large: 80×80) and two positions (fixed position and random position). For detection, we introduce three state-of-the-art detection methods as baselines, which are *PM* [12], *NIC* [21], and *PatchGuard* [44]. We reproduce *PatchGuard* [44] with official code and reproduce *PM* [12] and *NIC* [21] with our own implementation on Pytorch. Notably, since it is not clear how to derive the sub-models in *NIC* [21], we generate sub-models by using all the previous layers before target layer l.

Detection Effectiveness: We first formulate the detection success rate R_d as:

$$R_d = \frac{N_a + N_n}{N_t}, \tag{10}$$

where N_a and N_n represents the number of correctly detected adversarial inputs and natural inputs, respectively. N_t means the total test image number. The higher R_d indicates the method has higher detection effectiveness. Then we apply our defense method on all three models and test their detection success rates. Table 1 shows the overall detection performance with same attacking setting (medium patch size and random position). On all three models, R_d of *LanCeX* are from 89% to 91% while *PM* and *PatchGuard* are from 85% to 90%. *NIC* shows slightly higher R_d than our method for *ResNet-18*. Table 2 demonstrates our method's detection results under different attacking settings. We can find that detecting a fixed patch is easier than a random patch since its location is prior known, which introduces a more precise patch location identification. A larger patch size will introduce more significant inconsistency between the patch and the normal pattern, thereby can increase the detection performance. However, the larger the patch is, the higher possibility that it will cover the original objects, decreasing the recognition performance after the data recovery.

Threshold Selection: As aforementioned, threshold value T_{ic} is critical since inappropriate threshold will decrease either N_a or N_n, thereby hurt the entire detection performance. According to the method in Section V, we can calculate the threshold range for "cat" class as $D_{avg}^{ground}(i) = 0.41$ and $D_{min}^{adv}(i) = 0.53$. We evaluate the R_d for "cat" class patch when threshold T_{ic} are selected from 0.41 to 0.53 and the results are shown in Fig. 15. We can find the detection success rate in terms of different threshold values show a normal-like distribution, both smaller and

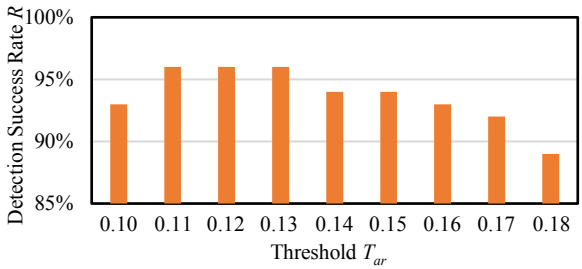

Fig. 16. The Detection Performance *w.r.t.* Different Thresholds in Audio Recognition Scenario

Table 4. Adversarial Patch Attack Detection Evaluation

	Attacked		*Information-based [50]*		*LanCeX*	
	Tar	Untar	Tar	Untar	Tar	Untar
Detection (R_d%)	-	-	79	80	83	86
Recovery (mAP)	1.24	0.07	56.11	56.23	**58.92**	**58.23**
Time Cost (ms)	-	-	455	461	**321**	**328**

*: Tar/Untar represents targeted/Untargeted adversarial attack respectively.

larger T_{ic} will degrade the detection success rate. When T_{ic} equals to 0.48, the corresponding R_d can achieve an optimal performance.

Data Recovery Effectiveness: We also evaluate image prediction accuracies under two image data recovery methods (*Zero Mask* and *Telea*) and compare their results with *Patch Masking*. Table 3 shows the specific image data recovery performance. We can find that *Telea* significantly help to correct predictions, resulting in 90.0%~91.5% accuracy recovery improvement on different models while *Patch Masking* achieves 88.0%~90.0% accuracy recovery improvement. On the other hand, the performance of *Zero Mask* is lower than *Telea* since it will remove some import informations from the original salient objects.

Time Cost: We leverage the process time cost to represent the method's computational complexity. In Table 3, we can find that the process time cost of our two defense methods for one condensed adversarial attack is from $188ms$~$357ms$ while the *Patch Masking* is from $233ms$~$461ms$. Among the two methods, *Zero Mask* shows faster computation process than the *Telea* while also slightly sacrifices the recovery performance in terms of prediction accuracy.

Based on the above comparison, we show that our defense method has better defense performance than *Patch Masking* with respect to both effectiveness and efficiency.

8.2 Defense Evaluation for Object Detection

Experiment Setup: We evaluate our proposed defense method based on DPatch [19] with two 120×120 adversarial patches (targeted and untargeted). The detection model and dataset used here are YOLO [31] and PASCAL VOC 2007 [8], respectively. We re-implement a state-of-the-art work, *Information-based Defense* [50] and use it as the baseline method. According to the threshold evaluation method proposed in Section VI, we can determine the thresholds for targeted attack and untargeted attack as 0.41 and 8.1. The entropy threshold of *Information-based Defense* is set as 7.5 which is same to the original paper.

Detection and Data Recovery Effectiveness: Table 4 shows the overall detection and image data recovery performance, which are represented by detection ratio rate R_d and mean Average Precision (mAP). From the table we can find: 1) In both targeted attack and untargeted attack settings, *LanCeX* achieves 84.5% detection

Table 5. Detection Evaluation for Audio Recognition

	FGSM	BIM	CW	Genetic
Dependency [6]	91%	89%	90%	88%
LanCeX	**96%**	**94%**	**93%**	**91%**

*:Dependency Detection (Dependency) [46]

Table 6. Audio Data Recovery Performance Evaluation

Method	FGSM	BIM	CW	Genetic	Time Cost (ms)
No Recovery	10%	5%	4%	13%	-
Dependency Detection [46]	85%	83%	80%	80%	1813
Noise Flooding [29]	62%	65%	62%	59%	1246
LanCeX	**87%**	**88%**	**85%**	**83%**	**521**

successful rate in average, which is higher than *Information-based Defense* method. 2) As for the image recovery performance, *LanCeX* can improve the attacked mAP from original 1.24 and 0.07 to 58.92 and 58.23, respectively. However, *Information-based* defense method can only achieve 56.11 mAP and 56.23 mAP.

Time Cost: Similarly, we measure the time cost of the entire defense process to reflect the method's computational complexity. Table 4 describes the time cost of *LanCeX* and *Information-based Defense*. The time cost of *LanCeX* for one condensed adversarial attack is from $321ms$~$328ms$ while *Information-based Defense* is from $455ms$~$466ms$. This is because *Information-based Defense* needs to calculate the entropy across the entire image while *LanCeX* just needs conduct computation on the patch area, which significantly saves the computation workload.

Based on the above comparison, we show that our defense method has better defense performance than the state-of-the-art method with respect to both effectiveness and efficiency.

8.3 Defense Evaluation for Audio Recognition

Experiment Setup: For audio recognition scenario, we use Command Classification Model [24] on Google Voice Command dataset [24]. For comparison, we re-implement another state-of-the-art defense methods: *Dependency Detection* [46]. Four works: FGSM [10], BIM [17], CW [4], and Genetic [1], are used as attacking methods to prove the generality of our defense method. We randomly select 1000 samples from Voice Command dataset as attacked audio.

Detect Effectiveness: Tab. 5 shows the overall detection performance comparison. *LanCeX* always achieves more than 92% detection success rate R_d for all audio adversarial attacks. By contrast, *Dependency Detection* achieves 89% detection success rate R_d in average. Therefore, *LanCeX* demonstrates the best detection performance.

Threshold Choosing Discussion: Fig. 16 shows the detection success rate R_d under different threshold T_{ar} settings. We achieves an optimal detection success rate when T_{ar} equals to 0.11, 0.12 and 0.13 while other threshold settings degrades the detection performance.

Data Recovery Effectiveness: Then we evaluate *LanCeX*'s data recovery performance. The k value in the top-k index is set as 6. We re-implement another method, *Noise Flooding* [29] and add it into evaluation as comparison. And we use the original vulnerable model without data recovery as the baseline. Table 6 shows the overall audio recovery performance evaluation. After applying our recovery method, the prediction accuracy significantly increases from average 8% to average 85.8%, which achieves 77.8% accuracy recovery. On the contrary, *Dependency Detection* and *Noise Flooding* have lower recovery rate, which are 74% and 54%, respectively.

Time Cost: For defense efficiency, the computational complexity of *LanCeX* is much lower than other methods according to our previous analysis. As the result, the time cost of our method is $521ms$ while other two methods usually cost more than $1813ms$ for a single condensed adversarial attack. Thus, our defense method is 2∼3× faster.

9 CONCLUSION

In this paper, we propose a versatile and lightweight CNN defense solution against condensed adversarial attacks and apply it in three practical recognition scenarios (image classification, object detection, and audio recognition). Leveraging the comprehensive CNN vulnerability visualization, attention-base attack process analysis, and two novel CNN inference inconsistency metrics, our method can effectively and efficiently detect and eliminate the condensed adversarial attacks in the above three recognition scenarios. Experiments show that our methodology can achieve optimal detection successful rate and data recovery performance. Moreover, due to the light computation consideration during the method design, our method is feasible to resource-constrained embedded systems, such as mobile devices.

REFERENCES

[1] M. Alzantot and *et al.* 2018. Did You Hear that? Adversarial Examples Against Automatic Speech Recognition. *arXiv preprint arXiv:1801.00554* (2018).

[2] J. Benesty and *et al.* 2009. Pearson Correlation Coefficient. In *Noise reduction in speech processing*. Springer, 1–4.

[3] T. Brown and *et al.* 2017. Adversarial Patch. *arXiv preprint arXiv:1712.09665* (2017).

[4] N. Carlini and *et al.* 2017. Towards Evaluating the Robustness of Neural Networks. In *Proc. of SP*. 39–57.

[5] Shang-Tse Chen, Cory Cornelius, Jason Martin, and Duen Horng Polo Chau. 2018. Shapeshifter: Robust physical adversarial attack on faster r-cnn object detector. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 52–68.

[6] Ping-Han Chiang, Chi-Shen Chan, and Shan-Hung Wu. 2021. Adversarial Pixel Masking: A Defense against Physical Attacks for Pre-trained Object Detectors. In *Proceedings of the 29th ACM International Conference on Multimedia*. 1856–1865.

[7] D. Erhan and *et al.* 2009. Visualizing Higher-layer Features of A Deep Network. *University of Montreal* 1341, 3 (2009), 1.

[8] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. 2007. The PASCAL visual object classes challenge 2007 (VOC2007) results. (2007).

[9] K. Eykholt and *et al.* 2017. Robust Physical-world Attacks on Deep Learning Models. *arXiv preprint arXiv:1707.08945* (2017).

[10] I. Goodfellow and *et al.* 2014. Explaining and Harnessing Adversarial Examples. *arXiv preprint arXiv:1412.6572* (2014).

[11] Robert M Gray. 2011. *Entropy and information theory*. Springer Science & Business Media.

[12] J. Hayes. 2018. On Visible Adversarial Perturbations & Digital Watermarking. In *Proc. of CVPR Workshops*. 1597–1604.

[13] K. He and *et al.* 2015. Deep Residual Learning for Image Recognition. In *Proc. of CVPR*. 770–778.

[14] Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7132–7141.

[15] Danny Karmon, Daniel Zoran, and Yoav Goldberg. 2018. Lavan: Localized and visible adversarial noise. *arXiv preprint arXiv:1801.02608* (2018).

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.

[17] A. Kurakin and *et al.* 2016. Adversarial Examples in the Physical World. *arXiv preprint arXiv:1607.02533* (2016).

[18] Mark Lee and Zico Kolter. 2019. On physical adversarial patches for object detection. *arXiv preprint arXiv:1906.11897* (2019).

[19] Xin Liu, Huanrui Yang, Ziwei Liu, Linghao Song, Hai Li, and Yiran Chen. 2018. Dpatch: An adversarial patch attack on object detectors. *arXiv preprint arXiv:1806.02299* (2018).

[20] M. Lohne. 2017. The Computational Complexity of the Fast Fourier Transform. https://folk.uio.no/mathialo/texts/fftcomplexity.pdf

[21] Shiqing Ma and Yingqi Liu. 2019. Nic: Detecting adversarial samples with neural network invariant checking. In *Proceedings of the 26th Network and Distributed System Security Symposium (NDSS 2019)*.

[22] Michael McCoyd, Won Park, Steven Chen, Neil Shah, Ryan Roggenkemper, Minjune Hwang, Jason Xinyu Liu, and David Wagner. 2020. Minority Reports Defense: Defending Against Adversarial Patches. *arXiv preprint arXiv:2004.13799* (2020).

[23] Byong Seok Min, Dong Kyun Lim, Seung Jong Kim, and Joo Heung Lee. 2013. A novel method of determining parameters of CLAHE based on image entropy. *International Journal of Software Engineering and Its Applications* 7, 5 (2013), 113–120.

[24] S. Morgan and *et al.* 2001. Speech Command Input Recognition System for Interactive Computer Display with Term Weighting Means Used in Interpreting Potential Commands from Relevant Speech Terms. US Patent 6,192,343.

[25] Todd F Mozer and Pieter J Verneulen. 2015. Background speech recognition assistant. US Patent 8,996,381.

[26] Norman Mu and David Wagner. [n.d.]. Defending against Adversarial Patches with Robust Self-Attention. ([n.d.]).
[27] M. Naseer and *et al.* 2019. Local Gradients Smoothing: Defense against localized adversarial attacks. In *Proc. of WACV*. 1300–1307.
[28] S. Niwattanakul and *et al.* 2013. Using of Jaccard Coefficient for Keywords similarity. In *Proc. of IMECS*, Vol. 1. 380–384.
[29] K. Rajaratnam and *et al.* 2018. Noise Flooding for Detecting Audio Adversarial Examples Against Automatic Speech Recognition. In *Proc. of ISSPIT*. 197–201.
[30] Sukrut Rao, David Stutz, and Bernt Schiele. 2020. Adversarial training against location-optimized adversarial patches. In *European Conference on Computer Vision*. Springer, 429–448.
[31] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 779–788.
[32] Joseph Redmon and Ali Farhadi. 2017. YOLO9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7263–7271.
[33] Huali Ren, Teng Huang, and Hongyang Yan. 2021. Adversarial examples: attacks and defenses in the physical world. *International Journal of Machine Learning and Cybernetics* (2021), 1–12.
[34] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*. 91–99.
[35] K. Simonyan and *et al.* 2014. Very Deep Convolutional Networks for Large-scale Image Recognition. *arXiv preprint arXiv:1409.1556* (2014).
[36] Dawn Song, Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramer, Atul Prakash, and Tadayoshi Kohno. 2018. Physical adversarial examples for object detectors. In *12th {USENIX} Workshop on Offensive Technologies ({WOOT} 18)*.
[37] C. Szegedy and *et al.* 2013. Intriguing Properties of Neural Networks. *arXiv preprint arXiv:1312.6199* (2013).
[38] C. Szegedy and *et al.* 2015. Going Deeper with Convolutions. In *Proc. of CVPR*. 1–9.
[39] Guanhong Tao, Shiqing Ma, Yingqi Liu, and Xiangyu Zhang. 2018. Attacks Meet Interpretability: Attribute-steered Detection of Adversarial Samples. *Advances in Neural Information Processing Systems* 31 (2018), 7717–7728.
[40] Alexandru Telea. 2004. An image inpainting technique based on the fast marching method. *Journal of graphics tools* 9, 1 (2004), 23–34.
[41] Yusuke Tsuzuku and Issei Sato. 2019. On the structural sensitivity of deep convolutional networks to the directions of fourier basis functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 51–60.
[42] F. Wang and *et al.* 2017. Residual Attention Network for Image Classification. In *Proc. of CVPR*. 3156–3164.
[43] Bichen Wu, Forrest Iandola, Peter H Jin, and Kurt Keutzer. 2017. Squeezedet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 129–137.
[44] Chong Xiang, Arjun Nitin Bhagoji, Vikash Sehwag, and Prateek Mittal. 2020. PatchGuard: Provable Defense against Adversarial Patches Using Masks on Small Receptive Fields. *arXiv preprint arXiv:2005.10884* (2020).
[45] H. Yakura and *et al.* 2018. Robust Audio Adversarial Example for A Physical Attack. *arXiv preprint arXiv:1810.11793* (2018).
[46] Z. Yang and *et al.* 2018. Characterizing Audio Adversarial Examples Using Temporal Dependency. *arXiv preprint arXiv:1809.10875* (2018).
[47] X. Yuan and *et al.* 2018. CommanderSong: A Systematic Approach for Practical Adversarial Voice Recognition. *arXiv preprint arXiv:1801.08535* (2018).
[48] Q. Zeng and *et al.* 2018. A Multiversion Programming Inspired Approach to Detecting Audio Adversarial Examples. *arXiv preprint arXiv:1812.10199* (2018).
[49] B. Zhou and *et al.* 2016. Learning Deep Features for Discriminative Localization. In *Proc. of CVPR*. 2921–2929.
[50] Guangzhi Zhou, Hongchao Gao, Peng Chen, Jin Liu, Jiao Dai, Jizhong Han, and Ruixuan Li. 2020. Information Distribution Based Defense Against Physical Attacks on Object Detection. In *2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 1–6.