Real-Time Robust Video Object Detection System Against Physical-World Adversarial Attacks

Husheng Han®, Xing Hu®, Member, IEEE, Yifan Hao®, Kai Di Xu, Pucheng Dang, Ying Wang®, Member, IEEE, Yongwei Zhao, Zidong Du®, Member, IEEE, Qi Guo®, Member, IEEE, Yanzhi Wang®, Member, IEEE, Xishan Zhang, and Tianshi Chen

Abstract—DNN-based video object detection (VOD) powers autonomous driving and video surveillance industries with rising importance and promising opportunities. However, adversarial patch attack yields huge concern in live vision tasks because of its practicality, feasibility, and powerful attack effectiveness. This work proposes Themis, a software/hardware system to defend against adversarial patches for real-time robust VOD. We observe that adversarial patches exhibit extremely localized superficial feature importance in a small region with nonrobust predictions, and thus propose the adversarial region detection algorithm for adversarial effect elimination. Themis also proposes a systematic design to efficiently support the algorithm by eliminating redundant computations and memory traffics. Experimental results show that the proposed methodology can effectively recover the system from the adversarial attack with negligible hardware overhead.

Manuscript received 29 August 2022; revised 13 February 2023 and 20 June 2023; accepted 29 July 2023. Date of publication 16 August 2023.

I. INTRODUCTION

POWERED by deep neural network (DNN) techniques, video recognition achieves tremendous success and starts to boost existing industries, such as autonomous driving, surveillance systems, drones, and robots. For example, autonomous driving based on video recognition, whose market is predicted to leap to $77 billion (25% of the whole automotive market) by 2035 [1], has attracted the attention of giants, including Tesla, Audi, and Waymo [2], [3], [4], [5].

Despite the promising opportunities and rising importance of DNN-powered video recognition, the vulnerability of DNNs emerges as an important problem in video recognition tasks, especially in life-critical scenarios. DNN techniques have been shown to be vulnerable to adversarial attacks. For example, wearing the T-shirt with adversarial patch printing on it, which effectively fools DNN-based person detectors in physical environments even under diverse scenarios like people walking, sitting, and running [6], [7]. Such attacks are malicious in the surveillance and autonomous vehicle application scenarios, which evade the video detectors in the physical world and incur life-or-death problems. Therefore, robust video recognition that defends against such adversarial attacks and eliminates the adversarial effects is urgent and important.

For live vision scenarios, the defensive methodology should meet two requirements.

1) Effectively recover (more than detection only) the system from the adversarial attacks considering video recognition is usually adopted in real-time decision-making scenarios.

2) The proposed defensive methodology should introduce lightweight performance overhead to achieve the goal of real-time object detection. Effectively recovering the video object detection (VOD) system from adversarial attacks in real time is a highly challenging task.

Existing pioneering studies for robust image classification fail to meet these two requirements: 1) they either detect abnormal inputs only without recovery [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], works poorly under adaptive attack [18], or 2) introduce too much overhead that cannot be born in real-time VOD systems [19], [20]. Local gradient smoothing (LGS) [18] regards regions with high gradients as patches with the assumption that patch pixels are not smooth but work poorly with smooth patches. MRD introduces extremely large
overhead (costs about 1446 s for one ImageNet-class image), which is not feasible in real-time scenario [19].

To this end, focusing on the important live vision scenario that is widely adopted in autonomous driving and surveillance systems, this work proposes the real-time robust VOD system, Themis, to defend the adversarial patch attacks that practically introduce damaging consequences in video recognition tasks. We propose localized important superficial feature (LISF)-based defensive methodology, which not only effectively identifies the adversarial regions but is also able to leverage temporal and spatial redundancy of video data for efficient robust object detection.

Themis Algorithm1: We draw the key observation that adversarial inputs induce VOD to be overshadowed by the localized but inductive superficial features, and the effect of adversarial patches can be eliminated facilely without aggravating the prediction accuracy of benign images when moving out LISFs. Hence, we propose LISF-based detection and recovery method based on prediction stability testing by moving out LISFs. Results show that the Themis algorithm effectively locates the adversarial regions and eliminates adversarial effects.

Themis Architecture: Although the Themis algorithm is effective at detecting the adversarial regions in input data, it also introduces challenges due to the additional multiple rounds of inference during the prediction stability testing by occluding LISFs. To reduce the performance overhead to support real-time robust VOD, we propose the Themis architecture to eliminate both the interframe and intraframe redundant computations.

1) Interframe: Themis leverages the spatial and temporal redundancy of video data to eliminate unnecessary computations in nonkey frames. Specifically, for key frames of video data, the complete defensive algorithm is performed to locate adversarial regions or features. For nonkey frames, by leveraging the temporal and spatial locality in video data, Themis approximately predicts the adversarial region location or features in nonkey frames by estimating the motion movement of adversarial regions and features (i.e., optical flow in this work).

2) Intraframe: In observing the redundant computations of benign features during LISF-based detection and recovery, we propose an efficient hardware design for benign feature computation reuse and eliminating redundant unnecessary computation and memory traffic. Themis architecture can be easily integrated with existing DNN accelerators to support state-of-the-art performance-oriented (PO) or accuracy-oriented (AO) video recognition methodologies.

In summary, this work has the following contributions.

1) We propose the LISF-based methodology to accurately identify the adversarial region locations in input data. The detection methodology can effectively work under adaptive attack when an adversary has white-box knowledge of defensive approaches.

2) We propose a defensive framework that significantly reduces defending overhead in nonkey frames by leveraging temporal and spatial locality in video data, which can be integrated with state-of-the-art VOD frameworks.

3) We propose lightweight hardware customization to efficiently support the defensive framework and fully exploit the computation reuse of benign features, which can be easily adapted to existing deep learning accelerators.

4) We make an extensive experimental evaluation and the results show that Themis system can effectively and efficiently defend against adversarial attacks in real time. Compared to the system without defensive mechanisms, Themis improves the average mean average precision (mAP) of real-time object detection from 0.03 to 0.66, with 1.08% of hardware overhead.

II. BACKGROUND

A. Video Object Detection Preliminary

VOD, recognizing instances of visual objects (e.g., humans, cars, animals) and their locations in digital videos, is a fundamental important computer vision task. It forms the basis of many other computer vision tasks, such as instance segmentation [21], object tracking [22], image captioning [23], etc.

Object Detection in a Single Image: Image object detection is the foundation of detecting objects in videos. Image object detector solves the following two subtasks:

1) Predicting how many objects are in the images.
2) Classifying these objects and estimating their locations with bounding boxes.

The most important evaluation metric for object detector prediction accuracy is (mAP, defined in Section V-A) which considers both precision and recall rate. Recently, VOD capability has been largely boosted by deep learning techniques with milestones of CNN-based image detectors, such as RCNN, YOLO [24], SSD [25], RetinaNet [26], etc.

Object Detection in Video Data: Video data consists of many time-sequential images. Hence, VOD can be achieved by performing image object detection in every image (referred to as AO schema). For better-computing efficiency, prior VOD methodologies are proposed to leverage temporal redundancy across frames in time-sensitive applications [27], [28], [29] (referred to as PO schema). The key design concept of PO schema is to undergo the precise computation of key frames, while approximately computing nonkey frames based on key frame features and the motion, trajectory, or optical flow information.

Optical Flow: Optical flow is widely used to utilize the temporal redundancy in video data and to approximate the nonkey frames. Optical flow describes the apparent motion of image objects in consecutive frames caused by the movement of objects or the camera. Specifically, as shown in Fig. 1, optical flow(c) is a 2-D vector field where each vector is a displacement (dx, dy), showing the movement of pixels from first frame (a) to the second (b). Once we obtain the optical flow, the pixels or feature map of nonkey frames can be estimated by warping the pixels or feature map of their predecessor key frame with the optical flow

\[ V(x + dx, y + dy, k + 1) \defeq V(x, y, k) \] (1)

where V can be the pixel values or feature activation, x and y are the 2-D coordinates, k is the frame index, dx and dy are the pixel displacements, and \( \defeq \) can be implemented by various interpolation methods like linear interpolation.

---

1 THEMIS, (Greek: “Order”) in Greek religion, personification of justice, a blindfolded goddess holding a pair of scales.
Optical flow can be calculated based on pixel matching [30] with the assumption that objects maintain the same intensity (brightness) between consecutive frames or end-to-end CNN framework [31].

AO and PO VOD Details: Fig. 2 illustrates AO and PO VOD frameworks. For AO framework, each frame of video is input to the full video object detector for precise computation. For PO framework, the DNN-based video object detector is divided into two parts: 1) DNN-prefix for more generic feature extraction with much heavier computation and 2) DNN-suffix for final prediction with low-computation overhead [29], [32]. AO framework first calculates the optic flow between nonkey frame and last key frame.

Then, warps the key-frame features with the resized and scaled optic flow information to approximately compute the nonkey-frame features. The predicted nonkey-frame feature map is input to the DNN-suffix for result computation. Due to the large gap between the computation of DNN-prefix and DNN-suffix, the computation overhead of nonkey frames is significantly reduced. These two VOD frameworks (image-based and feature-based) are adopted in different scenarios that are optimized for accuracy or performance. It is important to support effective and efficient defensive mechanisms in both these two cases. We propose Themis framework to achieve this goal.

B. Adversarial Attacks in Video Recognition

Attack Formalization: Adversarial patch can largely damage video recognition tasks by evading video object detectors. It manipulates the victim model to output malicious results by adding patch perturbation in the object of video data.

Formally, the goal of adversarial patch attack is to generate the adversarial patch, \( \hat{p} \), to maximize the expectation of possibility for classifier \( h \) to output targeted malicious label \( y' \) with all adversarial inputs \( x' \) derived from dataset \( X \)

\[
\hat{p} = \arg \max_p E_{x' \sim A(p,x)} [\log Pr(h(x') = y')].
\]  

(2)

Patched image \( x' \) is generated by applying patch \( p \) to \( x \) in the input dataset \( X \), which can be formalized as

\[
x' = A(p, x), \quad x \in X
\]  

(3)

where \( p \) is the adversarial patch, \( x \) is the clean image, and \( A \) is the transformation function (environmental noises, resizing, rotations, and deformations) applied on the adversarial patch when attaching the patch to the clean image. An example of an attack image is shown in Fig. 3(a), where the patch is a clock-like pattern on the chest. The adversarial patch determines the prediction results with a very small region of pixels (adversarial region) for a relatively broad range of input images. By moving out the adversarial region, the adversarial effects are eliminated and the prediction results are recovered. Hence, autonomously identifying the adversarial region in the video data is the key foundation of robust object detection.

Patch Attacks Versus Example Attacks: Compared to adversarial example attacks [33] that have been largely studied in image classification tasks, patch attacks have the following advantages.

1) Better universality, because the adversarial patch is independent of input images, which enables physical-world attack without prior knowledge of the scene. Adversarial example attacks, on the other hand, generate perturbation noises highly dependent on the input images, which hinders their deployment in the physical world.

2) Better robustness to environmental noises and geometric distortion. For example, a human being wearing a T-shirt printed with the adversarial patch can be ignored in the object detection systems in different environments and body gestures [6].

More studies prove that adversarial patches are robust to not only environmental noises but also geometric distortions. Adversarial example attacks, however, are scene-dependent and transfer poorly in different inputs. In real cases, the adversary neither can obtain the attack scene in advance nor compute the adversarial examples for every frame in real time. Therefore, adversarial example attack is not a practical attack.
model in physical environments for video recognition tasks. In this following, we do not consider the adversarial example attacks.

Limitation of Existing Defenses: Existing defenses fail to meet the requirement of recovering the system with effective defense performance and lightweight overhead. Adversarial example detection methods [15], [34] cannot be applied in patch attacks due to the differences between the adversarial patches and the adversarial example noises. Certified defenses for patch attacks [19], [20] could locate the patch certifiably but with unacceptable overhead. For example, over 100 inferences are needed for one patched image in [20]. Empirical defense methods [18], [35], [36] locate the patch based on empirical observations with lightweight overhead but are proved to be invalidated when attackers have the white-box knowledge of the defense [37]. For example, LGS [18] locates the patch on the image gradient map with the assumption that patch pixels are not smooth and have high gradients but work poorly with smooth patches. Under adaptive attack, the detection rate of LGS drops to 19.8%, which is not acceptable. More adversarial defense comparisons are introduced in Section VII-B. This work aims to propose a high-efficient and effective detection and recovery system.

III. ADVERSARIAL VIDEO PATCH CHARACTERIZATION

Intuitively, patched images rely on the extremely localized important neurons in the adversarial regions to deceive and induce the object detector to output incorrect results, while benign images perform stable object detection without relying on extremely localized important neurons. Hence, we perform the LISF characterization and have the following observations.

1) LISFs are Good Candidates for Detecting Adversarial Regions in a Single Frame: LISF Distribution: The important neurons in feature maps of superficial layers exhibit a localized pattern in patched images while scattered in benign images. The superficial important features refer to the neurons that contribute significantly to the feature map value in the superficial layers (in this article, we use the first layer). Specifically, we take the superficial important neurons as the Top-K ones having the biggest value in the output feature map of the first layer. We visualize these important neurons of a patched image example in Fig. 3(a). Intuitively, the important neurons are extremely localized in the patch region (clock-like pattern located on the chest). Further, we make statistical counting about the distribution of Top-200 neurons of 12K images randomly selected from FLIC dataset [38] based on two metrics: 1) cluster distance and 2) cluster number. We compare the standard deviation of the distance between the highlighted neurons to the central neuron in benign images and patched images. The patched images’ distance deviation is much smaller than benign images, as shown in Fig. 3(b). We also cluster the Top-K neurons with the classic MeanShift clustering algorithm. As shown in Fig. 3(c), for patched images, about 86% of cases have only one cluster and 97% of cases have no more than two clusters. While for benign images, about 80% of the cases have more than three clusters. Both of these two metrics show the localized distribution of superficial important neurons in patched images.

Prediction Stability by moving out LISFs: When occluding the LISFs, patched images can be recovered mostly without affecting the prediction accuracy of benign images. In Fig. 4, we compare the detection rate of benign and patched objects before and after moving out the localized important features. The detection rate of benign objects decreases to 97.4% slightly, while the detection rate of patched data increases from 17.9% to 72.8% significantly, which means that the prediction results of benign objects are much more stable than patched objects. For benign objects, the detection rate is not sensitive to localized features. For the patched objects, the patch effect would be effectively eliminated after moving the LISFs.

2) LISFs Exhibit Temporal Association in Video Data, Which Enables Us to Leverage Temporal Redundancy to Eliminate the Recovery Overhead in Nonkey Frames: Superficial feature computing is very close to the input, hence the important superficial features exhibit the similar temporal association as the image frames in video data. The bottom two subfigures in Fig. 3(a) show the LISFs marked within the solid red boxes in two consecutive frames. The LISF location in the following frames can be predicted by warping the optical flow with the LISF location in the previous key frame (predicted LISF location marked in the green box).

In summary, LISF-based methodology not only effectively detects adversarial regions and recovers the object detection in key frames but is also able to leverage temporal redundancy in videos and eliminate defensive overhead in nonkey frames. Based on these two observations, we then propose the LISF-based robust VOD system.

IV. THEMIS SYSTEM ARCHITECTURE

A. Overview

The overall Themis system is shown in Fig. 5 with algorithm (Section IV-C), framework (Section IV-B), and hardware designs (Section IV-D).

Algorithms: LISF-based methodology effectively targets the adversarial regions in input images and recovers the prediction results by moving out those regions.

Framework: Although the Themis algorithm already reduces the adversarial region searching space with LISF-based methodology, multiple inferences are introduced in the occluding testing stage and incur large overhead. To reduce the detection overhead, Themis proposes the interframe and intraframe optimizations to reduce redundant computations.

1) Interframe Optimization: By leveraging the spatial and temporal locality between frames, Themis only performs complete adversarial detection in key frames. For nonkey frames, Themis framework either predicts the adversarial region locations based on regions detected in
key frames (image-based warping) or reuses the clean features in key frames after eliminating the adversarial effects (feature-based warping). These two warp strategies can be easily integrated with existing AO and PO VOD frameworks.

2) Intraframe Optimization: The algorithm introduces a large volume of redundant calculations of benign features during occluding prediction stage for the key frames. Hence, Themis scheduler reduces the computing overhead with computation reuse of benign features (Section IV-B).

Hardware: Although the Themis algorithm can be implemented in pure software, it is inefficient because of the following reasons.

1) Searching the heat-map of the input activation for the adversarial candidates is inefficient in the DNN accelerator due to the lack of the computing parallelism between the searching process and typical inference process.

2) During the voting stage, multiple patch candidate regions will be occluded to calculate the inference results. In this process, a significant volume of benign features is computed repeatedly and introduces large unnecessary performance overhead.

To address these issues, we propose the Themis hardware architecture for efficient defense. The top-level block diagram of hardware architecture is shown in Fig. 5(c). Apart from the typical DNN accelerators with PE array, scalar function unit (SFU), global buffer, and control logic, Themis is augmented with the LISF searching logic to search the candidate regions according to the heat map of the first superficial layer feature map, masked Neuron buffer (MNB) to optimize the data and computation reuse of benign features, and the voting logic to decide the final prediction results.

B. Framework

Interframe Optimization Based on Tempo-Spatial Redundancy: The overview of Themis framework is shown in Fig. 5(a) and (b) with support for AO and PO VOD frameworks. Under both AO and PO scenarios, the defensive mechanisms for key frames are the same: the Themis algorithm detects the adversarial region of frames and masks them to eliminate the adversarial effect. For nonkey frames, Themis proposes two warp strategies with optical flow information which are integrated into existing AO and PO VOD frameworks, as shown in Fig. 5(a) and (b).

1) Image-Based Warping in AO Frameworks: For AO framework, we approximately estimate the patch location in nonkey frames by warping the patch location in key-frame images with the optical flow. Specifically, every frame will be forwarded to the object detector for a complete inference. Themis only performs the adversarial patch detection in key frames and obtains the adversarial region location by the Themis algorithm which will be introduced in Section IV-C and Algorithm 1. Then warps the detected adversarial region in key-frames with the optical flow information to estimate the adversarial region in nonkey frames. The corresponding area of nonkey frames is masked and the masked nonkey frames are forwarded to the object detector for inference. The rationality is that the adversarial regions in the input data also exhibit temporal and spatial redundancy in live vision. Moreover, compared to the object to be segmented, the adversarial region is much less sensitive to the derivation brought by the inaccurate optical flow information (more validation in Section VI-A).

2) Feature-Based Warping in PO Frameworks: For PO framework, we directly warp the clean features in key-frames with the resized and scaled optical flow to compute
the interfaces to configure the following design knobs that are introduced in Section IV-D. The detailed computing process and dataflow optimization are introduced, which incurs a large execution overhead. When the prefix is close to the classification layers, the feature map is too small and may introduce larger accuracy degradation during optical flow estimation. We test extensive datasets and set the splitting spot when the feature map size is smaller than 56 × 56.

Adversarial Region Detection Parameters: In the adversarial patch detection stage, the more adversarial candidates, the more redundant computations are introduced, which incurs a large execution overhead. Hence, we propose the scheduling methodology with computation reuse to alleviate the execution overhead and eliminate redundant computations. As shown in Fig. 7, the masked images and the original image share the same pixels, with only the differences in masked regions. Thus, we compute the masked regions separately and splice the masked region back to the complete feature map for the final prediction results. Under both AO and PO cases, the defensive overhead of nonkey frames is minimized.

Intraframe Optimization for Benign Feature Computation Reuse: With the obtained coordinates of the patch candidates, Themis then makes the prediction decisions by masking them from the original image individually. During this process, multiple inference rounds of inputs with different mask locations are introduced, which incurs a large execution overhead. Hence, we propose the scheduling methodology with computation reuse to alleviate the execution overhead and eliminate redundant computations. As shown in Fig. 7, the masked images and the original image share the same pixels, with only the differences in masked regions. Thus, we compute the masked regions separately and splice the masked region back to the complete feature map for the final prediction results, to eliminate the most redundant computations. The detailed mask region computing is as follows: given an image (224 × 224 pixels) with the masked regions (50 × 50 pixels). The pixels in the masked region are set as 0 to occlude their effect on the results. However, the features of the masked region through neural network layers are not simply set to 0, because the existence of weight bias and the kernels that are larger than 1. So we need to take the padding number into consideration when computing the features of the masked region. For example, when computing the C1 layer in Fig. 7 for masked images, the complete region (masked region + padded region) for recomputation is 53 × 53, with an additional purple part.

Reconfigurable Design Knobs: Themis framework provides the interfaces to configure the following design knobs that affect overall performance efficiency: the key frame proportion, the strategy of dividing the object detector to feature extraction DNN prefix and classification DNN suffix, and the candidate numbers during adversarial patch detection.

Key Frame Selection: We set the key frame rate as 10% with fixed length. Themis framework can support the adaptive key frame strategies proposed by prior VOD studies [29], [40]. However, how to choose the key frame is out of the scope of this work.

PO Framework Splitting: Designing DNN prefix and suffix in PO frameworks is the tradeoff between performance and detection accuracy. When the prefix is close to the classification layers, the feature map is too small and may introduce larger accuracy degradation during optical flow estimation. We test extensive datasets and set the splitting spot when the feature map size is smaller than 56 × 56.

C. Adversarial Detection and Recovery

In observing that important superficial features exhibit different spatial distribution characteristics and exert distinct influence on the prediction results in benign and adversarial input data, we propose the LISF-based patch candidate searching methodology and then detect and eliminate the adversarial effect by occluding testing (see Algorithm 1).

LISF-Based Patch Candidate Searching: We perform the LISF searching in the output feature map of the first layer. The size of the searching window is the upper limit of the patch sizes and the searching stride is 1. When the number of important neurons in one searching window is larger than the threshold (θ), it is recognized as an important window and marked as the patch candidate. When several important windows are overlapped, the central important window will be retained as a patch candidate with all the others deleted. The detailed searching logic implementation is described in Section IV-D.

Occluding Testing and Recovery: We recover the prediction results with the following two steps.

1) Masked Image Inference: After obtaining the candidate locations of adversarial patches, we generate masked images by occluding the patch candidate locations individually from the original images. These masked images are taken into the victim model to produce the prediction decisions.

2) Monopolist-Occluded Voting: Themis performs the prediction decision analysis to detect patched images by examining whether there is a monopolist patch candidate that determines the prediction results. The prediction result of attacked image is manipulated by the local adversarial patch. Only when the patch is occluded, the output will be recovered. In other cases, the prediction results are consistent to be the wrong label that is controlled by the patch. Therefore, the monopolist label is the correct prediction where the patch effect is eliminated.

The detailed patch candidate searching and voting mechanisms are introduced as follows. There are k candidates: \( P_1, P_2, \ldots, P_k \). The prediction result of the original image is \( L_0 \),
and the corresponding prediction results of masked images: \(L_1, L_2, \ldots, L_k\). We first detect the patch by checking whether there is a \(L_i\) distinct from others, while all the other labels are the same: if positive, the \(P_i\) is the monopolist that dominates the prediction results, which is the adversarial patch. Only when tearing it off the image, the classifier predicts the robust and benign label \(L_i\).

If there is only one candidate, i.e., \(k = 1\), we compare \(L_1\) and \(L_0\) to determine whether there is an adversarial patch. If they are different, \(P_1\) is the adversarial patch and \(L_1\) would be the recovered label.

If there is no such particular label (either all the labels are the same, or several labels are different), no monopolist is detected. Then the image is recognized as a benign image. Themis then performs the majority voting to obtain the predicted label.

D. Hardware Design

The Themis architecture can be integrated into the typical DNN hardware accelerator to support real-time detection with small overhead.

LISF Searching Logic: The LISF searching logic outputs the coordinates of clustered important neurons, which infers the possible candidate locations in the input image. LISF searching consists of the following three steps:

1) Obtain the binary important feature map of the first layer. Computing Top-K neurons in the feature map is time-consuming and introduces large hardware overhead. Therefore, we make the estimation of the adaptive threshold according to the maximum value of the feature map. All neurons with activation values larger than the threshold \(\beta * \text{feature}_{\text{max}}\) are selected as important neurons. The important neuron map is stored in the buffer of LISF searching logic.

2) Identify the important windows. We slide the fixed-size window to make statistic counting about the important neuron numbers in one window. When the important neuron numbers occupy more than the threshold (\(\theta\)) of the total neurons, we mark this window as an important neuron window, which will be highlighted as the patch candidates. To reduce the hardware overhead, we make the incremental accumulation of the important neurons in one window. As shown in Fig. 6(a), with the number of important neurons in Window\([i, j, i + s, j + s]\), to compute the Window\([i, j + 1, i + s, j + 1 + s]\), we simply subtract the important neurons in column \(j + 1\) and add the important neurons in column \(j + 1 + s\) between row \(i\) and row \(i + s\).

3) Delete the overlapped important windows. When two sliding important windows have more than 30% of the area overlapped, we take them as one single patch candidate.

Noted, the LISF searching is not on the critical path of DNN inference, which is parallelized with the processing of the original images.

MNB: To support efficient computation reuse of benign features and optimize the data accesses of the adversarial features, the MNB is proposed to buffer the padded bounding area for feature computation of candidate adversarial regions.

The data reuse flow of computing candidate adversarial regions is as follows (Fig. 8): with the coordinates of potential adversarial region candidates, the padded bounding areas through all the neural network layers in Fig. 7(a) are determined. Through the typical inference, all the activation values of the padded bounding areas are stored in MNB. Then, during the inferences of masked images, all the masked regions are batched for computation. For every layer, the PE array reads the candidate adversarial features from global buffers and the padded bounding areas from MNBs, as the red box and purple box shown in Fig. 8(c). The PE array combines these neurons by padding the red box with the purple box, takes it as input, and computes the candidate adversarial feature of next layer. After completing the computation of this layer, the adversarial features are stored back in global buffers. Because the padded bounding area is very small compared to the full activation map, we configure the mask neuron buffer with a size of 8 KB for every PE array.

Voting Logic: The basic voting logic is as shown in Fig. 6(b). We use a comparer array to perform the pairwise comparison between the prediction labels of masked images \((L_0, L_1, \ldots, L_k)\). If there is an orphan label \(L_i\), this image is identified as a patched image and the recovered label is \(L_i\). Otherwise, it is a benign image and Themis performs the majority voting to obtain the recovered label.

For one input image, searching and voting operations are only performed once, while the inference rounds are determined by the searching candidates. Among these stages, multiple inferences introduce the most performance cost. Voting is simple and performed within several cycles. Although the searching algorithm is time-consuming when offloaded to the CPU platform, its customized hardware largely boosts the performance and the overhead is less than 0.5% of the multiple inferences. The details of our experimental evaluation for comparing different hardware platforms are shown in Section VI-B.

V. EXPERIMENTAL METHODOLOGY

In the following sections, we will evaluate the defensive effectiveness and architecture efficiency of Themis, which is complementary to enable robust and real-time VOD.

A. Validation on Algorithm Accuracy

We test the attack success rate and the defensive effectiveness in both single-frame object detection tasks and video object recognition tasks. For the previous scenario, we focus on exploring and validating the adversarial patch detection capability of the Themis algorithm on static images. In the latter scenario, we focus on exploring the defensive effectiveness
on the nonkey frames when Themis framework leverages the temporal and spatial information in video data.

**Attack Methodologies:** For single-frame testing scenario, we adopt the digital-synthesized adversarial methodology that randomly attaches the digital adversarial patches onto the bounding box regions of the objects in MS COCO, FLIC, LSP datasets and random locations of images in ImageNet dataset with random rotated angles. The patch size is scaled with the area of bounding boxes ranging from 33 × 33 to 130 × 130 pixels. Since we focus on the defensive effectiveness of Themis, we only perform attacks on the objects or images that can be correctly identified by the detector. For video data testing scenarios, we adopt physical attack videos in the state-of-the-art attack methodology [6] (Adv T-shirt), which significantly damages the functionality of YOLOv2 object detector. The patch size is variable and the adversarial patch has been significantly deformed during human movement.

**Optical Flow Methodologies:** Themis framework is compatible with different optical flow methodologies. We use both the CV-based and DNN-based optic flow methodologies: 1) DIS [30] and 2) SpyNet [31], to validate the defensive effectiveness and architecture efficiency. The SpyNet is customized with scaled input image sizes and reduced pyramid levels.

**Evaluation Metrics:** We adopt the commonly used metrics, the detection rate and the standard mAP score, to measure the accuracy of VOD. Precision measures true positives (TPs) rate out of all positive predictions. average precision (AP) is calculated as the area under the precision-recall curve. mAP is the average AP of each class

\[
    mAP = \frac{1}{n} \sum_{k=1}^{n} AP_k
\]

(4)

where AP\(_k\) is the AP of class \(k\). In terms of performance efficiency, the frame per second (FPS) is used to indicate how fast the framework processes the video data.

**Hyperparameters:** We assume the upper limit of the patch area is 3% of the image and use it to determine the size of the sliding window, which is a common assumption in the adversarial patch field [41], [42]. We set LISP selection threshold \(\beta = 0.75\) and candidate window selection threshold \(\theta = 0.85\) by parameter searching to balance the defense accuracy and the system overhead. Despite being empirical, the searched hyperparameters are robust for all selected datasets. For new datasets, hyperparameters can be fine-tuned with generated patched datasets by adopting the digital-synthesized attack in new datasets as discussed in above **Attack Methodologies** paragraph without the need for manual labeling.

**B. Validation on Architecture Efficiency**

**NN Accelerator Hardware Implementation:** Our methodology can be generalized and adopted in diverse neural network accelerators. In this work, we evaluate our methodology based on the classic Eyeriss accelerator. Specifically, we augment the Eyeriss accelerator with the patch detector consisting of the LISP searching, MNB, and voting logic. To prove the generality and its low overhead, we test both the server and edge accelerators with different configurations, as shown in Table I.

We implement the Themis hardware design in Verilog RTL. To obtain the area and power, we synthesize and place and route the RTL code with Synopsys under TSMC 28-nm technology. We use CACTI 7 and DESTINY to model the DRAM memory and on-chip SRAM buffers. Due to the unbearable long duration of silicon simulation, we also tailor an open-sourced simulator, NN-dataflow [43] to support Themis for the total execution latency simulation. The simulator also reports the exact memory traces and module activities, which are then used to calculate dynamic energy consumption.

Besides, we deploy Themis system on an off-the-shelf FPGA, Zynq UltraScale+ MPSoC ZCU104 board, to show the real performance. Specifically, we implement the proposed architecture components like Adv Candidate Search Logic and integrate them with a DPU (a soft IP for NN inference) using Vitis-ai framework.

**VI. EXPERIMENTAL RESULTS**

We first show the defensive effectiveness of Themis in terms of single frame and video scenarios (Section VI-A). Then we show the performance and energy efficiency of Themis (Section VI-B), which introduces negligible overhead to real-time VOD. Finally, we show that Themis adds negligible area overhead to the baseline DNN accelerator (Section VI-B4).

**A. Defensive Effectiveness Evaluation**

We validate the defensive effectiveness of Themis under the following two scenarios: 1) single-frame data and 2) video data with sequential frames.

1) **Single-Frame Defensive Effectiveness:** For single-frame testing, we use FLIC [38], LSP [44], MS COCO [45] and ImageNet [46] datasets that are commonly used in object detection domain. We first evaluate the adversarial patch detection accuracy based on the metric of overlapped area proportion in Fig. 9(a). The predicted patch area and the actual patch area have an average of 72%, 81%, 50%, and 88% overlapped region compared to the actual patch area size. The results show that LISP-based searching methodology can accurately identify the location of adversarial patches. We then show the object detection rate before and after Themis defense in Fig. 9(b) compared with LGS [18] and inpainting with Laplacian prior (ILP) [35]. LGS locates the patch using the local gradient of image pixels with the basic assumption that patch pixels are not smoothing. The ILP method is similar.

**TABLE I**

| HARDWARE PLATFORM CONFIGURATIONS |
|----------------------------------|
| Eyeriss accelerator with the patch detector consisting of the LISP searching, MNB, and voting logic. To prove the generality and its low overhead, we test both the server and edge accelerators with different configurations, as shown in Table I. |  |
| 1** MN**B denotes Masked Neuron Buffer. |
| (a) Predicted Patch ACC | (b) Detection Rate |
| FLIC | LSP | COCO | ImageNet |
| 0.791 | 0.812 | 0.504 | 0.464 |
| 0.962 | 0.819 | 0.871 | 0.640 |
| 0.199 | 0.233 | 0.288 | 0.140 |
| 0.505 | 0.705 | 0.365 | 0.591 |

Fig. 9. Detection effectiveness. (a) Prediction accuracy of patch region. (b) Detection rate comparison.
to LGS but replaces the first-order gradient with second-order (Laplacian) gradient. The detailed attack methodology is as follows: the adversary randomly attaches the adversarial patch in the person bounding box of the images in the datasets, so that the object detectors are evaded to ignore the persons. With the adversarial patch attack, the object detection rate is 11.9%, 6.8%, 22.8%, and 14.0% for FLIC, LSP, MS COCO, and ImageNet. As shown in Fig. 9(b), compared to LGS and ILP, our Themis defensive mechanism works better and improves them to 96.2%, 91.3%, 91.4%, and 85.8%. The results show that Themis can eliminate the adversarial patch effect effectively.

**Defensive Effectiveness Against Adaptive Attacks:** In the further step, we evaluate the defensive effectiveness under the strong adaptive attack [47] that the adversary gets known the full knowledge of the defensive strategies. To steer clear of the detection of Themis, the adversary trains the adversarial patch with (5) that a penalty loss for the superficial activation value of patch is considered compared to (2). In this way, the adversary aims to build the adversarial patch with good poisoning effects, but also tries to escape from the adversary candidate searching. \( \alpha \) is the parameter to control the scale of the penalty loss in superficial activation value

\[
\text{loss} = -\log P(h(x) = y) + \alpha \sum (w_i \cdot \hat{p}).
\]

We perform the adaptive attacks on ImageNet dataset and the results are shown in Fig. 10. Adversary detection rate refers to the rate that Themis correctly identifies the adversarial region in the adversarial inputs or the benign input with no adversarial region. Recovery object detection rate refers to the rate that Themis correctly identifies the object in the image. The results show that, it is indeed that both the adversary detection rate and the recovery object detection rate decrease to 82.1% and 82.0%, respectively, when \( \alpha \) increases from 0 to 0.01. However, compared to the gentle slope of adversary detection rate and the recovery object detection rate, the attack success rate decreases much more drastically. When \( \alpha \) is set to 0.01, the adversarial attack success rate is dropped to 7.9%. These results indicate that the adversary cannot maintain the two goals of high-attack success rate and good stealthiness simultaneously. Themis can work effectively even under adaptive attack. As a comparison, the detection rate of LGS drops from 59.1% to 19.8% after adding total variation loss and the detection rate of ILP drops from 64.0% to 17.3% after adding the 2nd gradient penalty loss while attack success rate maintains high, which indicates that LGS and ILP are both vulnerable to adaptive attack.

2) **Video Frame Defensive Effectiveness:** For the video frames, we adopt the adversarial video benchmarks in the state-of-the-art adversarial attack study [6], where the people wearing the adversarial T-shirt move in indoor and outdoor scenarios and perform the practical attacks in the physical environment. The video object detector is based on YOLOv2. We evaluate the object detection rate and mAP under the following scenarios.

1) **AO-ND:** AO framework with no defensive mechanisms.
2) **AO-Full:** Defensive AO framework that examines every frame.
3) **AO-Dis:** Defensive AO framework that completely examines key frames, but CV-based methodology to predict the adversarial patch locations in nonkey frame.
4) **AO-Spynet:** Defensive AO framework that completely examines key frames, but DNN-based optical flow information to predict the adversarial patch locations in nonkey frame.
5) **PO-ND:** PO framework with no defensive mechanisms.
6) **PO-Spynet:** Defensive PO framework with DNN-based optical flow (spynet).

**Detection Rate:** Fig. 11 shows the object detection rate under different scenarios. With adversarial attacks, both AO and PO object detectors have significantly low-object detection rates of 4.4% and 4.8%, which indicates that adversarial attacks can essentially damage the integrity and functionality of object detectors even in physical environments. With the Themis defensive algorithm, the detection rate is significantly improved above 93.8% for different defensive strategies. Compared to AO-Full which examines every frame, other approximate defensive methods achieve relatively good object detection rates within a gap of less than 5%.

Detection rate is only a coarse-grained metric that intuitively indicates the detection recall rate of object detectors. In the further step, we evaluate the mAP that considers both the prediction accuracy, recall rate, and the predicted bounding box accuracy in the following.

**mAP:** Fig. 12 shows the mAP results under IoU = 0.5 (mAP@0.5), IoU = 0.75 (mAP@0.75), and average mAP value where IoU ranges from 0.5 to 0.95 with the step of 0.05. Intersections over union (IoU) is the metric to determine
Fig. 13. Performance and energy comparison among different defensive strategies. (a) Performance comparison. (b) Energy comparison.

whether it is an accurate prediction of the bounding box, which is calculated as the rate of dividing the area of overlap by area of union. A larger IoU indicates a more strict criterion of mAP prediction accuracy. From the plot, specifically, we have the following observations.

1) Consistently, it is observed that adversarial attacks can effectively fool the object detector to ignore the human being with mAP as low as 0.03, 0.05 of AO-ND and PO-ND. With the guard of Themis, the functionality of object detector is recovered and the average mAP is improved to the range of (0.53, 0.68) under different defensive mechanisms.

2) When IoU is low (IoU = 0.5), the mAP of defensive PO frameworks is equally good to the defensive approaches that examine every frame (AO-Full). When IoU is high, PO series may introduce the reduction of mAP due to the shift and deviation of the optical flow information. Specifically, the gap between mAP@0.75 and the average mAP of PO-SpyNet and AO-Full is 0.09 and 0.15.

3) Adversarial patch location prediction is less sensitive to the deviation of optical flow. Although optical flow information also introduces the deviation between the predicted and actual adversarial regions, such deviation does not markedly hurt the defensive effectiveness of Themis (less than 0.03), because of the prediction instability of adversarial regions, as analysis in Section III.

B. Architecture Efficiency Evaluation

1) Overall Performance and Energy Comparison: We also evaluate the performance and energy of AO and PO frameworks with different defensive strategies. Fig. 13(a) shows the object detection throughput for four commonly used video detectors: 1) YOLOv2; 2) YOLOv3; 3) RetinaNet; and 4) FasterRCNN. From the plot, we have the following conclusions.

1) PO frameworks significantly boost performance when their model architectures can be divided into heavy NN-prefix and light NN-suffix.

2) Performing adversarial detection in every frame incurs heavy overhead. Compared to AO-ND, AO-Full incurs 3.3 × execution latency, which remarkably reduces the FPS in all four object detectors.

3) Eliminating the unnecessary recomputing for the adversarial region locations in nonkey frames improves the performance and reduces the performance gap between defensive approaches with original approaches while maintaining the defensive effectiveness. Specifically, AO-SpyNet introduces 25.0%, 17.7%, 15.5%, and 22.9% overhead compared to AO-ND. PO-SpyNet introduces 28.6%, 20.4%, 14.6%, and 26.4% overhead compared to PO-SpyNet-ND.

In summary, Themis can effectively defend against adversarial attacks in video tasks, while still maintaining the throughput of about 36 FPS and 59 FPS for real-time object detection in AO and PO frameworks.

2) Scheduling Optimization: We first evaluate the effectiveness of scheduling optimization for computing masked images in four typical datasets: 1) MS COCO; 2) FLIC; 3) LSP; and 4) T-shirt [6]. Fig. 14 shows the latency reduction ratio with benign feature computation reuse for different object detection models. Patches on MS COCO, FLIC, LSP, and T-shirt account for 2.81%, 1.47%, 3.60%, and 2.01% of the whole image pixels on average, respectively. We have the following observations.

1) Overall, Themis reduces the latency for masked images effectively with scheduling optimization. The average latency reduction ratio for MS COCO, FLIC, LSP, and T-shirt are 51.93%–53.77%, which indicates that the scheduling optimization in Themis is applicable to various scenarios.

2) The scheduling optimization strategy in Themis is more efficient in object detectors with a shallower depth. The average latency reduction in FasterRCNN is 73.75%, which is significantly higher than 29.13% in RetinaNet. The intrinsic reason is that the computation reuse ratios in the first several layers are higher than the latter. Specifically, for the first layer in YOLOv3, the computation reuse ratio reaches 98.38%.

Themis can efficiently reduce latency of the detection process for masked images with scheduling optimization. With an average latency reduction of 53.31%, which significantly reduces the overhead incurred by the defense scheme.

3) Comparison With Different Architectures: We compare the performance and energy of Themis with CPU (Intel...
Core i7-4770K at 3.50 GHz and GPU (NVIDIA TITAN V) platforms. Fig. 15 shows the normalized latency and energy for CPU, GPU, and Themis architectures running four video detectors: 1) YOLOv2; 2) YOLOv3; 3) RetinaNet; and 4) FasterRCNN with MS COCO dataset. Themis hardware components can be integrated with any DNN architectures and we choose Eyeriss-like architecture as the basic DNN accelerator. Specifically, Themis refers to the cases of adopting candidate searching hardware components and “Themis-Opt” refers to the Themis architecture with computation reuse optimization. For each case, we break down the system into inference procedure and searching procedure (CPU data bars are too large in this figure and are rescaled with the value marking on it). We draw the following conclusions:

1) Themis is much more efficient compared to CPU and GPU platforms. Compared with CPU, Themis achieves speedup from 16.3 × to 60.5 ×. Compared with Titan V, Themis reduces energy from 9.47 × to 29.7 ×.

2) Candidate searching consumes non-negligible overhead in GPU platform. The customized searching logic achieves 745.7 × speedup.

3) With the computation reuse optimization, Themis-Opt achieves speedup from 1.20 × to 1.75 ×.

4) Hardware Area Overhead: The baseline Eyeriss-like DNN accelerator has an area of 12.60 mm². On top of the basic accelerator design, Themis only introduces a total area overhead of 0.136 mm², which incurs only 1.08% of hardware overhead. Specifically, the LISF searching logic occupies 0.0746% of the hardware overhead, the MNB occupies 1.005%, and the rest is attributed to the voting logic.

5) Implementation on FPGA: The experimental FPGA device (Xilinx ZCU104 evaluation board) consists of two ARM processor cores and 16nm FinFET+ programmable logic. The ARM processors are used to run Linux systems and control the operation flow, while the programmable logic is reconfigured to accelerate the user applications. We use the Vitis-ai framework to deploy Themis with a customized DPU IP. Compared to the officially provided Yolov2 demo that runs at 24 FPS, Themis achieves 22 FPS at AO defense mode and 35 FPS at PO defense mode. The resource consumption of our FPGA implementation can be seen in Table II.
proposes a high-efficient and effective detection and recovery system to defend the adversarial attacks that practically introduce damaging consequences in video scenarios.

C. Feature Importance Analysis

Neuron importance has been widely used for abnormal input detection [17], [69], [70], [71] in previous studies. However, the metric (superficial feature importance) in our methodology is distinct from previous work. Previous studies focus more on the neurons that contribute significantly to the inference output (deep feature importance). We envision that deep feature importance is not a good candidate from the following two aspects.

1) Both the benign and adversarial images have deep feature importance. Its discrimination in the benign images and adversarial images is not straightforward so it is hard to identify the benign and adversarial images.

2) Calculating such deep feature importance is time-consuming, which demands gradient information and a complete backward propagation process.

We propose the superficial input feature importance as the metric for discrimination analysis based on the intuition that in order to efficiently manage the output prediction results with a very small region of the input data, the adversarial patch must incur large activation from the first place instead of the accumulation of the deep feature extraction.

VIII. CONCLUSION

Themis efficiently and accurately recovers the DNN systems from the adversarial attacks with both algorithmic framework and the architectural support. At the algorithmic level, Themis prevents the classifier from being overshadowed by the trivial but extremely biased parts by tearing the patch off the original images. At the architectural level, Themis not only proposes efficient searching and voting logic but also proposes the scheduling methodology to accelerate the masked image execution by eliminating the redundant computations and memory traffics. The results show that the proposed methodology can effectively recover the VOD system from the adversarial effect in real time.

REFERENCES
Kaidi Xu received the B.S. degree from Sichuan University, Chengdu, China, in 2015, the M.S. degree from the Department of Computer Science, University of Florida, Gainesville, FL, USA, in 2017, and the Ph.D. degree from Northeastern University, Boston, MA, USA, in 2021. He is currently an Assistant Professor with the Department of Computer Science, Drexel University, Philadelphia, PA, USA. His primary research interest is the robustness of machine learning, including physical adversarial attacks, rigorous robustness verification, and certified defenses.

Pucheng Dang received the B.S. degree from the Harbin Institute of Technology, Harbin, China, in 2019. He is currently pursuing the Ph.D. degree with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, and the School of Computer Science, University of Chinese Academy of Science, Beijing. He also works with the Department of Architecture Algorithm, Cambricon Technologies, Beijing. His current research interests include machine learning security and computer vision.

Ying Wang (Member, IEEE) received the B.S. and M.S. degrees from the School of Astronautics, Harbin Institute of Technology, Harbin, Heilongjiang, China, in 2007 and 2009, respectively, and the Ph.D. degree from the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing, China, in 2014. He is currently an Associate Professor with the State Key Laboratory of Computer Architecture, ICT, CAS. His research interests primarily focus on the area of reliable computer architecture and VLSI design.

Yongwei Zhao received the bachelor’s degree in computer science and technology from the Huazhong University of Science and Technology, Wuhan, China, in 2015, and the Ph.D. degree from the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing, China, in 2020. He is currently an Assistant Professor with ICT, CAS. His research interests primarily focus on the area of accelerator architecture.

Qi Guo (Member, IEEE) received the B.E. degree in computer science from Tongji University, Shanghai, China, in 2007, and the Ph.D. degree from the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing, China, in 2012. From 2012 to 2014, he was a Staff Researcher with IBM Research, Beijing. From 2014 to 2015, he was a Postdoctoral Researcher with Carnegie Mellon University, Pittsburgh, PA, USA. He is currently a Professor with ICT, CAS. His research interests include computer architecture, programming models, and machine learning.

Zidong Du (Member, IEEE) received the B.E. degree from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 2011, and the Ph.D. degree from the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing, in 2016. He is currently an Associate Professor with Intelligent Processor Research Center, ICT, CAS. He has published over 20 top-tier computer architecture research papers, including ASPLOS, MICRO, ISCA, IEEE TRANSACTIONS ON COMPUTERS, Theory of Computing Systems, and IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS. His research interests mainly focus on novel architecture for artificial intelligence, including deep learning processors, inexact/approximate computing, neural network architecture, and neuromorphic architecture.

Xishan Zhang received the B.S. degree from the Department of Information Security, Shanghai Jiao Tong University, Shanghai, China, in 2011, and the Ph.D. degree from the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing, China, in 2017. She is currently an Associate Professor with ICT, CAS, and a Team Leader with the Department of Architecture Algorithm, Cambricon Technologies, Beijing. Her current research interests include deep learning system and reinforcement learning.

Tianshi Chen received the bachelor’s degree in mathematics from the Special Class for the Gifted Yong, University of Science and Technology of China (USTC), Hefei, China, in 2005, and the Ph.D. degree in computer science from the Department of Computer Science and Technology, USTC, in 2010. He is currently a Professor with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. He is also serving as the CEO of a startup called Cambricon Technologies Corporation Ltd., Beijing, whose commercial processor products are named “Cambricon.” Prof. Chen received the China Computer Federation Distinguished Doctoral Dissertation Award in 2011 and the Chinese Academy of Sciences Distinguished Doctoral Dissertation Award in 2011 for his Ph.D. work on computational complexity analysis of evolutionary algorithms.