Detecting Localized Adversarial Examples: A Generic Approach using Critical Region Analysis

Fengting Li*†, Xuankai Liu*†, Xiaoli Zhang*†‡, Qi Li*†, Kun Sun§, and Kang Li¶
*Institute for Network Sciences and Cyberspace & Department of Computer Science and Technology, Tsinghua University
†Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University
‡Gemini Lab, Alibaba Group §Department of Information Sciences and Technology, CSIS, George Mason University ¶Baidu
{lft18, liuxk18}@mails.tsinghua.edu.cn, xiaoli.z@outlook.com, qli01@tsinghua.edu.cn, ksun3@gmu.edu, kangli.ctf@gmail.com

***Abstract*—Deep neural networks (DNNs) have been applied in a wide range of applications, e.g., face recognition and image classification; however, they are vulnerable to adversarial examples. By adding a small amount of imperceptible perturbations, an attacker can easily manipulate the outputs of a DNN. Particularly, the localized adversarial examples only perturb a small and contiguous region of the target object, so that they are robust and effective in both digital and physical worlds. Although the localized adversarial examples have more severe real-world impacts than traditional pixel attacks, they have not been well addressed in the literature. In this paper, we propose a generic defense system called TaintRadar to accurately detect localized adversarial examples via analyzing critical regions that have been manipulated by attackers. The main idea is that when removing critical regions from input images, the ranking changes of adversarial labels will be larger than those of benign labels. Compared with existing defense solutions, TaintRadar can effectively capture sophisticated localized partial attacks, e.g., the eye-glasses attack, while not requiring additional training or fine-tuning of the original model's structure. Comprehensive experiments have been conducted in both digital and physical worlds to verify the effectiveness and robustness of our defense.**

I. Introduction

With the rapid development of deep learning techniques, neural networks have been applied in various applications, e.g., face recognition [1] and object detection [2], to achieve a better accuracy than traditional machine learning methods. However, neural networks are vulnerable to adversarial examples [3], where attackers can generate perturbations against neural network models to raise misclassification. For example, an adversarial attack can generate imperceptible pixel-level perturbations in an image so that the image looks the same as the original one for humans, but easily deceives the classifier into generating an entirely different label [4], [5], [6], [7], [8]. A number of defenses have been proposed to detect such pixel-level attacks [9], [10], [11], [12]. One limitation of this imperceptible perturbation is that it can hardly be applied in the real world due to the difficulty on manipulating the images of the real world in the granularity of pixels.

As one type of adversarial examples, the localized adversarial examples focus on perturbing a small and contiguous region that is visible to human eyes and can be printed out to launch attacks in the real world. By adding physical penalty constraints, localized adversarial examples become robust to various physical conditions such as locations, sizes, and even different background patterns. Localized adversarial examples can be classified into two categories, namely, *localized universal attacks* and *localized partial attacks*. The localized universal attacks aim to raise false predictions on arbitrary inputs by using one adversarial object. For example, adversarial patches [13] can be printed to perform generalized attacks when it appears in the scene and is robust under different real-world environments. In contrast, the localized partial attacks focus on manipulating the predictions on one or a small subset of labels. For example, an adversary can easily fool the face recognition system by wearing a decoration such as a pair of glasses [14].

Though it is well known that localized adversarial examples can cause severe threats in the real world, a generic and deployable defense system is still missing. In general, model robustness approaches [15], [16] require extra training with high training overhead and are difficult to work at ImageNet scale [15], [10]. Also, they cannot be applied to guard existing models. Several defense solutions have been proposed to defeat localized adversarial examples [17], [18], [19]; however, as shown in Table I, they all have significant limitations, Hayes et al. [17] propose a digital watermarking (DW) mechanism based on the observation that the density of salient pixels with respect to the final output is larger in adversarial objects than that in benign objects. Based on a similar observation that pixel values change more drastically inside the perturbed area, Local Gradient Smoothing (LGS) mechanism [18] can mitigate the adversarial effects of perturbations by smoothing the gradient inside that area. These two input transformation approaches are vulnerable under the threat of adaptive attacks [15]. Also, LGS is sensitive to the sizes and shapes of patches and not attack-agnostic. SentiNet [19] can detect adversarial objects by testing the behavior of adversarial objects with benign test samples [20]; however, its detection effectiveness heavily relies on the generalization of the adversarial objects. Overall, none of existing defenses can detect the stealthy partial attacks including eye-glasses attacks [14].

In this paper, we develop TaintRadar, a generic defense system that can detect different types of localized adversarial examples in both digital world and physical world scenarios. TaintRadar accurately detects attacks by identifying critical regions that have been manipulated, where a critical region is the region that supports the final prediction. When removing a critical region, the ranking of the predicted label will be

TABLE I: Comparison with existing defenses against adversarial objects. ○ denotes that the attack cannot be detected; ● denotes that the defense can defeat the attacks; ✓/× illustrates whether the defense can achieve the corresponding property.

	Localized Adversarial Example		Desired Properties of Defenses				
Defenses	Partial Attack	Universal Attack	Genericity	Attack-agnostic	Agility	Lightweight	Robustness
LGS [18]	○	○	×	×	✓	✓	×
DW [17]	○	○	×	✓	✓	×	×
SentiNet [19]	○	●	×	✓	✓	×	✓
TaintRadar	●	●	✓	✓	✓	✓	✓

lowered along with a corresponding probability. However, after removing critical regions of the same small sizes from benign and adversarial images, the ranking changes of predicted labels on adversarial images will be larger than benign images. To accurately detect localized adversarial examples, TaintRadar first estimates the critical regions of input images and then measures the ranking changes of the input images before and after removing the critical regions. In particular, we utilize the top-K changes of logits to refine the estimated region, which can effectively enlarge the difference between benign and malicious images and increase the accuracy on identifying the attack.

TaintRadar provides several critical properties on detecting localized adversarial examples. First, it is a generic detection system that can detect various localized adversarial examples including both localized universal attacks and localized partial attacks. Second, it is attack-agnostic, since it does not require any prior knowledge of the attacks. Third, it offers good agility for deployment as a plug-and-play mechanism, since it can work with various neural networks without modifying the models. Fourth, it is lightweight since it does not require training any extra model and can detect attacks in real time. Finally, it is robust against different attack variants including the adaptive attacks.

We systematically evaluate the performance of TaintRadar using two typical scenarios, i.e., scene classification and face recognition scenarios. In the scene classification scenario, TaintRadar can achieve more than 93% True Positive Rate (TPR) with a 6% False Positive Rate (FPR) when we use different attack variables to construct various partial and universal attacks. Moreover, TaintRadar can effectively detect partial attacks in the face recognition scenario and achieve more than 73% TPR in detecting partial attacks. In contrast, SentiNet [19] can only achieve a 0.7% TPR on the same task. In addition, we can achieve similar detection performance in the physical environment. Furthermore, we validate the robustness of TaintRadar under various advanced attack settings. The experimental results confirm that TaintRadar can achieve comparable detection performance even under advanced attacks such as adaptive attacks.

In summary, we make the following contributions:

- We propose a defense solution to defeat localized adversarial examples. The basic idea is to measure the difference before and after removing the critical regions that may be leveraged by the attackers. It is the first generic defense that can be applied to detect various localized adversarial examples (including partial and universal attacks) in different scenarios, without requiring any prior knowledge of the attacks.
- We evaluate the effectiveness and overhead of TaintRadar under different attacking settings in both digital and physical worlds. The experimental results show that TaintRadar can effectively detect both partial attacks and universal attacks. Since it requires no extra training and only incurs around 80 ms extra processing delay, it is promising to be deployed in time-bounded real systems.
- We conduct experiments to demonstrate the robustness of TaintRadar against advanced attacks with different adversarial variants, e.g., the adaptive attacks that have the white-box knowledge of TaintRadar.

II. Background on Neural Networks

A neural network consisting of multiple layers can be expressed as a function $F_\theta(x) = y$, where θ is the model parameters, x is the input (e.g., an image), and y is the output (e.g., the result of face recognition). In this paper, we focus on convolutional neural networks (CNN) that can be used as m-class image classifiers. One CNN is mainly composed of convolutional layers, fully connected layers, and a softmax layer. A convolutional layer consists of multiple convolutional kernels, and each kernel can extract specific high-level visual features called feature maps. The fully connected layers map these features to classification labels. The softmax layer, as the last layer, takes the results of the prior layer (called logits $Z(x)$) as input and calculate the output vector as a probability distribution, which meets $0 \leq y_i \leq 1$ and $y_1 + y_2 + \cdots + y_m = 1$. The final classification label l for the input x is the one with the highest probability y_l. We use r_l to denote the ranking of the probability corresponding to a label y_l among all probabilities.

III. Attack Model

Attackers can generate various adversarial examples to fool a pre-trained CNN model. There are effective attacks [3], [21] that produce imperceptibly global perturbations for misclassification purpose; however, it is challenging to apply them in the physical world due to the difficulty of adding global pixel-level perturbations. In this paper, we focus on *localized adversarial examples* that construct visible but inconspicuous perturbations in a small and contiguous region. For simplicity, we call such regions as *adversarial objects*. The localized attacks can be more easily launched in the physical world by directly putting or wearing adversarial objects in the scene. With the presence of adversarial objects in an input image, the classification model can be hijacked to generate a completely different output, e.g., in facial biometric systems [14] or object recognition systems [13]. In this paper, we study both localized universal attacks and localized partial attacks.

Localized Universal Attack. It aims to raise false predictions on arbitrary inputs using one adversarial object. The adversarial object is generated against multiple images with different source labels to achieve either targeted or untargeted attacks. Size variations and printability of adversarial objects are two main factors to be considered when launching robust attacks in varying physical environments. [13] is an representative example of localized universal attacks against benign CNN models in scene classification scenario.

Localized Partial Attack. It focuses on manipulating prediction results of a model on one or a small set of labels. For example, Sharif et al. [14] successfully generate a pair of eyeglasses as an adversarial accessory to achieve source-label-specific dodging or impersonation attacks. This type of attack is stealthier than the localized universal attacks, since they only need to compromise a small set of specific labels and those misclassification results may not be easily discovered.

IV. Overview of TaintRadar

The goal of TaintRadar is to detect localized attacks by identifying critical regions that have been manipulated. A critical region is the region that contributes heavily to the final prediction result of neural networks. When removing a critical region, the probability of the output label of original image (i.e., the predicted label) decreases and the corresponding ranking drops. Meanwhile, when the size of the removed region increases, the ranking will decrease more.

We conduct experiments to measure the ranking changes of predicted labels on benign and adversarial images after removing critical regions with varied sizes. The experiments consist of three steps, namely, finding critical regions, continuously removing the most important pixels from critical regions, and recording the corresponding probability ranking of original labels. We first develop an estimation algorithm to identify the critical regions with the importance score (see Section V-A). Then, we sort the positive importance of each pixel in a descending order and remove the 100 most important pixels each time and observe the ranking changes of the original label. Figure 1 depicts the trend averaged over 200 images generated using the method [13], where the ranking changes of benign images with gradual region removal grow slower than those of adversarial images. Also, experiments on other attacks show a similar trend. Thus, we can have the following key observation.

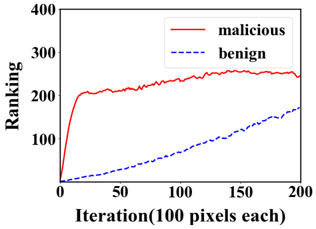


Fig. 1: Ranking variations between benign and adversarial images as the sizes of removed critical regions increase.

Key Observation. *The ranking changes of predicted labels on adversarial images are larger than those of benign images after removing the same small size of critical regions from both benign and adversarial input images.*

Figure 2 shows the high-level workflow of TaintRadar, which first estimates critical regions of input images and then analyzes the ranking difference before and after removing the regions for detection. TaintRadar is designed as a *generic* defense system that can detect both localized universal attacks and localized partial attacks in digital and physical worlds. It is *attack-agnostic*, since the defense does not require any prior knowledge on attacks including attack methods or the information of adversarial objects, e.g., their shapes, sizes, and locations. Also, TaintRadar is *agile* to be deployed in real world scenarios since it does not need to make any changes over the neural networks. The defense is *lightweight* so that it can be applied to detect attacks in real time without requiring extra training. Moreover, TaintRadar is *robust* to effectively detect various advanced attacks, e.g., adaptive attacks with white-box knowledge of both the neural network model and the detection method.

V. System Design

We first introduce the estimation approach to capture the critical region of an input image. Next, we analyze the influence of removing the estimated region and utilize the top increases of logits to enlarge the difference between benign and malicious images. Finally, we differentiate benign and malicious images by checking the ranking change of the predicted label.

A. Critical Region Estimation

We first locate the small and contiguous region that may be leveraged to launch an attack. As long as we can identify such region in an input image, we can check to see if they are being attacked. Inspired by [20], we utilize the gradient information of neurons in the last convolutional layer to generate a critical region of a given input image, where the high-level semantics are better than other convolutional layers and the preserved spatial information is also richer than the fully-connected layers. Since the adversarial objects usually do not block the original object, they not only need to increase the activation value of the target label, but also degrade the activation of the original object and some other related labels. We define a cross-entropy function to effectively capture the behaviors of both promoting and suppressing the activation.

$$l = -\texttt{CrossEntropy}(\bar{y}^c, p) = \sum_i \bar{y}_i{}^c log(p_i), \tag{1}$$

where $\bar{y}^c$ is the one-hot encoding of predicted label c, and p is the softened output of the model obtained by Eq. (2). Note that we add a negative sign to the cross-entropy function, indicating that we find a critical region that boosts the current identified label while suppressing the remaining labels. Intuitively, we want to find a region that offers the largest benefit to launch an untargeted attack on the current input so that the input will be misclassified. It is most likely that this region has been leveraged to construct a successful attack.

Considering that a malicious image might have high prediction confidence on a target label, p may be close to $\bar{y}^c$. Meanwhile, the gradient value of back-propagation may be 0 according to the rule for gradient calculation. To avoid the obtained gradient being 0, we divide logits Z by a parameter T before performing softmax, where T>1. We can get the softened output p_i as follows:

$$p_i = \texttt{Softmax}(\frac{Z}{T})_i = \frac{e^{\frac{Z^i}{T}}}{\sum_k e^{\frac{Z^k}{T}}}, \tag{2}$$

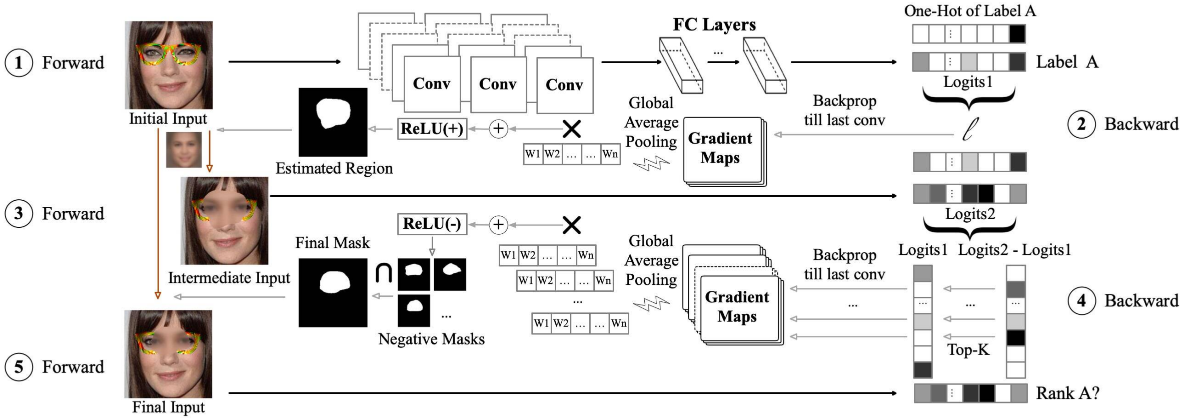


Fig. 2: Workflow of TaintRadar. Forward pass ① feeds the initial input to the model and gets the corresponding logits and label. We identify critical regions through backward pass ② using estimation function ℓ to the last convolutional layer (see Section V-A). Then, we replace the critical regions with a filling pattern to generate an intermediate input and compute the changes of logits with forward pass ③ . With the top-K logit changes, we obtain the negative masks by backward pass ④ (see Section V-B). We intersect these K negative masks to get the final mask and replace the region with the filling pattern. Finally, we use the ranking changes of the original label through forward pass ⑤ to detect attacks.

where p_i is the i-th node value in the output vector and Z^i is the corresponding i-th logit.

We then define the importance weight α_k of the kth feature map A^k as the global-average-pooling over gradients of each neuron A^k_{ij} with respect to the function l.

$$\begin{aligned}\alpha_k &= \frac{1}{N}\sum_i\sum_j \frac{\partial l}{\partial A^k_{ij}} = \frac{1}{N}\sum_i\sum_j\sum_s \frac{\partial l}{\partial Z^s}\cdot\frac{\partial Z^s}{\partial A^k_{ij}},\\ &= \frac{1-p_c}{T}\cdot\frac{1}{N}\sum_i\sum_j\frac{\partial Z^c}{\partial A^k_{ij}} - \frac{p_s}{NT}\sum_i\sum_j\sum_{s\neq c}\frac{\partial Z^s}{\partial A^k_{ij}},\end{aligned} \tag{3}$$

where N is the number of neurons in each feature map. The derivation of α_k in Eq. (3) shows that we can leverage all the information in the output vector to estimate the region. In particular, the first half of equation indicates the region that supports the predicted label, while the second half, with a negative sign, represents the regions suppressing the other labels. Then, we perform a weighted-sum over all the feature maps. By adding a ReLU function, we get a positive heatmap that represents the critical region. Ultimately, we normalize and binarize it to get L_{est} using the same threshold suggested in [20], i.e., 0.15, to obtain the final mask, where all the pixels inside and outside the estimated region are set as 1 and 0, respectively.

$$L_{est} = \texttt{Binarize}\left(\frac{\texttt{ReLU}(\sum_k \alpha_k A^k)}{\texttt{max}(\texttt{ReLU}(\sum_k \alpha_k A^k))}\right). \tag{4}$$

Note we can cancel T in Eq. (3) since it has influence only on p_c and p_s, but not the output. However, with the unchanged relative magnitude, the critical region is stable with different T values after the softening process. Our system sets T to 2.

B. Critical Region Based Detection

Now we aim to accurately identify adversarial perturbations by analyzing the estimated critical regions.

Strawman Approach. A straightforward approach is to simply remove the estimated critical regions and observe the ranking change of the original label. If the ranking of the original label changes drastically, the critical regions are most likely to be adversarial objects, meaning the image have been attacked. However, we find that the ranking changes may be large for both benign images and adversarial images, making it difficult to distinguish them. Particularly, when the estimated critical region covers most of the main object in a benign image, the rankings of the original label may change dramatically if we remove the critical region (see Figure 1).

Our Approach. We can tackle this problem in the above Strawman approach by reducing the impact on the benign images. We find that, in order to successfully construct an attack, adversarial objects are usually constrained in small contiguous regions where all pixels are utilized together. However, for benign inputs, different regions are relatively distributed as multiple parts and each region is with different semantics, e.g., the region may be a tail or an ear of a cat. Moreover, adversarial objects normally do not block the original object. Thus, to successfully launch an attack, adversarial objects will inevitably suppress logits of a large group of labels apart from increasing the logit of the target label. Although both benign and malicious objects suppress logits value of other labels to achieve the highest success probability, the regions suppressing each label are centralized and scattered for malicious and benign inputs, respectively.

Therefore, to enlarge the difference between the rankings (see Figure 1), we select the top-K labels that are suppressed most with the largest logit increases after overlaying the critical region with some filling patterns. Then, we leverage the counterfactual explanation [20] to find negative critical regions. First, we apply global-average-pooling on the gradients flowing from the logit Z^l to the last convolutional layer to obtain the weight of each feature map (see Eq. (5)).

$$\alpha^l_k = \frac{1}{N}\sum_i\sum_j\frac{\partial Z^l}{\partial A^k_{ij}}, \tag{5}$$

where l is one of these top-K labels, and α^l_k is the weight of the k-th feature map [20]. Second, we can use equation Eq.

(6) to find the negative critical regions of each label. Note that, before feeding the weighted map into the $ReLU$ function, we add a negative sign to obtain the region of a negative impact on the current logit, namely, suppressing the prediction of label l.

$$L^l_{negative} = \texttt{ReLU}(-\sum_k \alpha^l_k A^k). \quad (6)$$

After identifying the K negative critical regions, we can first binarize them using a threshold of 0.15 according to our empirical study and then intersect these K masks to generate a final mask (see Eq. (7)). For adversarial inputs, the intersection of these K suppression regions accounts for a significant ratio of the adversarial objects. For benign inputs, the suppression regions corresponding to these K labels are scattered, so the region after the intersection will be small enough, significantly reducing the impact on benign inputs.

$$L_{TaintRadar} = \bigcap_{l=1}^{K} \texttt{Binarize}\left(L^l_{negative}\right). \quad (7)$$

As analyzed before, the rankings of original labels are sensitive to the changes if we remove critical regions in malicious images. Small modifications in attacked regions will lead to obvious changes of the prediction outputs. Therefore, we inpaint the intersection region with a filling pattern and resend it to the classifier to get the final ranking of the original predicted label. Here, we define whether an input is malicious by a threshold ΔR, i.e., the ranking change of the original predicted label.

VI. Experiments

In this section, we evaluate TaintRadar in two typical scenarios with multiple attack variants.

A. Experiment Setup

To evaluate the effectiveness of TaintRadar, we use two typical scenarios, i.e., scene classification and face recognition. In particular, we construct universal and partial attacks by generating adversarial patch in the scene classification scenario, and construct partial attacks by generating accessory attack in the face recognition scenario.

Scene Classification. ImageNet dataset [22] is commonly-used in large-scale scene classification tasks [23]. It contains over 14 million images belonging to 1000 labels. Currently, there are a number of public models for scene classification that perform well on ImageNet dataset. Here, we use a pre-trained VGG16 model [24] that achieves over 92% top-5 test accuracy in ImageNet. To evaluate TaintRadar, we implement localized universal attacks [13] and develop partial attacks using the same methodology. We also conduct the corresponding physical attacks.

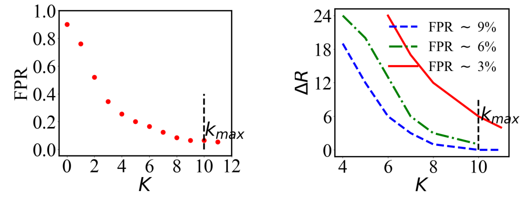
Face Recognition. To evaluate TaintRadar in the face recognition scenario in the digital world, we follow the experiment settings in [14] and use VGGFace dataset [1]. The VGGFace dataset is a large scale dataset, consisting of 2.6 million images of 2,622 celebrities. Besides, we use the pre-trained VGG-Face model proposed in [1], achieving over 98% test accuracy on the VGGFace dataset. We generate adversarial glasses to assess the performance of our approach. Moreover, to effectively evaluate TaintRadar in the real world, we retrain a new model that can recognize three authors of this paper and 140 celebrities in the PubFig dataset [25] using transfer learning [26].

We first show how we set the parameters in TaintRadar and then evaluate TaintRadar by measuring the true positive rate (TPR) and the false positive rate (FPR) under various settings in both digital and physical worlds.

B. Selection of Parameters in TaintRadar

To make TaintRadar attack-agnostic (i.e., requiring no prior knowledge of adversarial examples), we use a set of benign images to decide two parameters in TaintRadar, i.e., top-K and ΔR. Since our experiments show that the settings in different scenarios are similar, for simplicity, we take scene classification as an example to illustrate the parameter selection. First, we randomly select 200 images from the ImageNet Validation Dataset and run TaintRadar with different ΔR and K combinations. The results are shown in Figure 3a and Figure 3b. We find that when K is set too small, the intersection region found by TaintRadar may be too large for benign images. If removing the region, the ranking changes of the original labels would be large which result in a high FPR. As shown in Figure 3a, when K increases, the FPR will decrease. However, the intersection region of adversarial images gets smaller gradually, and the attacked region can not be found effectively as K increases to a certain level, leading to a low TPR. Thus, we set threshold k_{max} as the maximum value of K to trade off between FPR and TPR, where FPR is close to the minimum value and meanwhile the maximum value K ensures a highr TPR (see Section V-B).

Moreover, we analyze the ranking changes of the original label after removing the intersection region so that we can choose ΔR. In particular, when the ranking changes are larger than the threshold ΔR, we can safely detect them as adversarial images, which means TPR would be high if ΔR is small. For benign images, the correlations among ΔR, K and FPR is shown in Figure 3b. We can see that when FPR is given, the corresponding K decreases as ΔR increases. Therefore, we can select an acceptable FPR for benign images in advance, e.g., 6%. At the same time, ΔR and K are mutually constrained to maintain a FPR value for benign images. Considering that the change of K within a certain range has a small effect on the detection capability against adversarial images, we select ΔR as small as possible to improve the detection accuracy of TaintRadar.



(a) (b)

Fig. 3: (a) The changes of FPR as K increases; (b) The correlation among ΔR, K, and FPR.

C. Digital-World Experimental Results

We evaluate the effectiveness of TaintRadar in the digital-world where an attacker can manipulate each victim image on

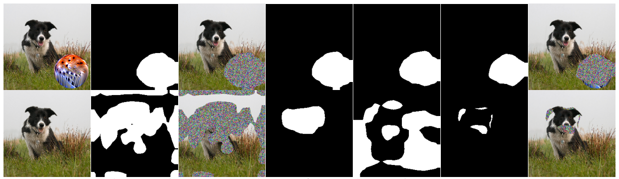


Fig. 4: The detection results of each TaintRadar step. Column 1 shows the benign and the victim input images, Columns 2 and 3 show the estimated critical regions and the intermediate inputs filled with filling patterns, Columns 4 and 5 illustrate two examples of sub-masks out of four, Columns 6 and 7 show the intersection regions of these sub-masks and final inputs.

the per-pixel level. We systematically measure the difficulty of generating an attack image under our defense. It is relatively easier for an attacker to deceive the classifiers without physical constraints. An attacker who cannot succeed in the digital world will mostly fail in the physical world.

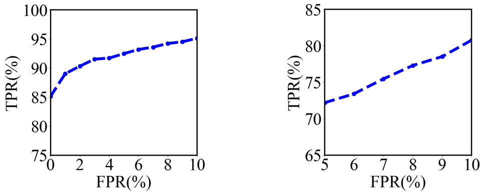
Scene Classification. To show that our defense is effective and robust, we construct both partial and universal attacks with varying attack settings when generating patches. The detailed attack has three settings. First, the batch size in patch generation is the number of images used to generate patches, showing the generalization capability of patches. For example, a patch created on 16 images has a more universal effect on held-out images than a generated patch for only 1 image. Second, for one size or multiple sizes in patch generation, an adversarial patch generated with multiple sizes is more robust against different shooting distance than those generated with one size. Third, we change the size of patches in victim images to estimate the effectiveness of TaintRadar against patches of different sizes. Lastly, the patch position in attacking is another factor to evaluate if the information loss of the original object will influence the detection performance of TaintRadar.

In this experiment, we randomly select 400 images from the ImageNet Validation Dataset as the benign image dataset. These 400 images are all correctly classified as ground truth labels. With the benign images dataset, similar to [13], we perform targeted attacks by using the cleverhans code [27] to construct patches. The attacks make the victims misclassified as a *toaster* (the corresponding label is 859). According to the method of setting parameters above, we use a randomly selected held-out benign dataset containing 200 images from validation set. We set the expected FPR as 6%, and the generated parameters are 9 for K and 2 for ΔR. The FPR evaluated on 400 held-out benign images is 6.25%.

A pair of examples is shown in Figure 4. The negative regions focus on the same area in the flow of malicious image detection, while for the benign example, they are scattered. From Table II, we can see that TaintRadar achieves a high TPR under different settings, which ensures the final success rate of patch attacks at a low level. The TPR decreases slightly when patches are placed at a random position in victim images. The reason is that they are likely to cover the main object in the original image, which incurs information loss of the original object and leads to a performance degradation of the detection.

Meanwhile, our approach is robust to different sizes of patches. There are a few abnormal results when the patch is generated with multiple sizes and the patch size in victim images is 0.2. The reason is that the number of adversarial images that are attacked successfully is really small. Therefore, there may exist deviations in the detection results with few samples. Also, we find that TaintRadar can detect both partial (batch size is 1) and universal attacks. In addition, the performances are stable in different settings when a smaller FPR is used. For example, when the FPR changes from 6% to 3%, the drops of performance range between 0% and 4% in Table II. Figure 5a shows the averaged trend under different FPR settings.

We reproduce SentiNet following the settings in [19]. It shows a similar 4% FPR and a 98.11% TPR on the adversarial patch task in [19]. As shown in Table II, using the same estimated curve and benign test set, SentiNet also achieves comparable results when the adversarial objects have strong generalization capabilities. However, when the batch size used in generation drops, the performances depicts a clear trend of degradation. Also, we note that the performance peaks at patch size 0.3. The reason behind is that patch of 0.2 has less chance of fooling the majority of benign sets, which is consistent with the successfully-generated rate shown in Table II. For patches with a size of 0.4, the average confidence drops compared to the size of 0.3, leading the output points cross the decision boundary. In total, TaintRadar achieves a better and more stable performance in different attack settings, including generalization capability, patch size, and patch position.



(a) Adversarial patches (b) Eye-glasses attacks

Fig. 5: TPR of TaintRadar under different FPR settings.

Face Recognition. In this scenario, we construct eye-glasses attacks, i.e., generating adversarial eys-glasses. All of them are partial attacks, since each pair of glasses target one person. To generate adversarial eye-glasses, we set victim image dataset by randomly sampling 20 from 2622 people and each with 50 aligned images (totally 1000 images). Then, we launch the attack targeting at *A.J Buckley* (the corresponding label is 0) with these images. We set the stop probability of 0.95 and with maximum 300 iterations following the settings in [14]. To determine parameters, we randomly select a held-out benign dataset with 20 people each with 10 images. Also, with the FPR set to 6%, we get parameters set as 13 and 13, for K and ΔR, respectively. The filling pattern here is the average face generated with all the VGGFace data, which preserves the facial information.

With 978 successfully attacked images out of 1000 attempted ones, the TPR of TaintRadar is 73.42%. The FPR evaluated on 200 held-out benign images is 5.50%. The TPR is mainly affected by two factors. First, the region found by TaintRadar is accumulated on the face area, which is relatively large for benign images. Second, the region covered by eye-glasses is critical for the VGG-Face model; however, the overlapping part of the original semantics will only impair the confidence of the original object's related labels. As our experiment shows, even when a large part of the critical regions is covered, it still has a minor impact on the final result. It conforms to the detection results of randomly placed patches

TABLE II: The first column of each setting is the successfully-generated rate (SR) out of 400 attempts. The other two show the TPRs of SentiNet and TaintRadar against adversarial patches. All numbers are shown in their percentages.

		Single Size									**Multiple Sizes**								
	Batch Size	1			4			16			1			4			16		
Position	**Patch Size**	SR	SentiNet	**Ours**	SR	SentiNet	**Ours**	SR	SentiNet	**Ours**	SR	SentiNet	**Ours**	SR	SentiNet	**Ours**	SR	SentiNet	**Ours**
Right-bottom	0.2	37.75	50.33	**97.96**	29.00	78.38	**99.08**	27.75	87.93	**98.25**	6.50	3.85	**80.00**	6.00	21.05	**100.00**	4.75	12.50	**95.83**
	0.3	97.50	81.28	**97.95**	97.25	99.48	**99.74**	96.00	**100.00**	97.94	78.75	58.41	**95.83**	83.25	98.76	**99.07**	80.75	**99.40**	97.29
	0.4	100.00	57.25	**91.50**	100.00	**97.49**	97.24	99.50	**99.75**	97.25	96.00	46.88	**86.98**	98.00	**94.39**	94.13	9800	**98.98**	88.27
Random	0.2	58.50	29.91	**95.26**	38.50	48.21	**96.39**	42.00	56.49	**95.45**	15.50	29.91	**79.03**	12.50	48.21	**91.30**	11.75	56.49	**81.63**
	0.3	98.75	58.99	**95.19**	97.00	87.47	**98.47**	97.75	88.92	**95.10**	86.00	58.99	**93.88**	88.00	87.47	**96.36**	89.75	88.92	**95.14**
	0.4	100.00	34.75	**88.25**	99.75	66.25	**95.75**	100.00	8.25	**71.18**	97.75	34.75	**82.10**	98.75	66.25	**88.92**	99.25	71.18	**80.76**

as discussed in Section VI-C, e.g., covering 20% of the critical region can still yield a 94% TPR. We discuss how to improve the detection performance in Section VIII. Also, SentiNet [19] achieves a 2% FPR, while only 0.72% of these 978 images are flagged as malicious with a nearly 0 fooled rate on the benign test set. Again, it proves our method generalizes better than SentiNet on localized partial attacks.

D. Physical-World Experimental Results

Scene Classification. We use the pre-trained VGG16 model in the physical world same as that in the digital world. Following the setting of [13], we captured 5 videos of a banana with and without a printed patch next to it, respectively. The relative position and the orientation of the banana and patch varies between each video. The VGG16 model without using TaintRadar classifies all 600 benign video frames (120 frames each) correctly and 598 out of 600 video frames with the printed patch as the target label *toaster*. By using the same parameters in the digital world, TaintRadar achieves 88.63% TPR on 598 victim image and 0.17% FPR on 600 benign inputs. In real deployment, the TPR can be further improved by verifying the prediction consistency of the contiguous frames, for example, using 5 contiguous frames to judge if a scene is under an attack during a short period of time.

Face Recognition. To construct partial attacks in the face recognition scenario, i.e., eye-glasses attack, we use the similar setting in the training of DNN_C in [14], which identifies 143 subjects composed by the first three authors of this paper and 140 celebrities from the PubFig evaluation dataset, each person with 40 training images. Based on the idea of transfer learning [26], we keep the first 37 layers of the VGG-Face Model [1], and append a fully connected layer with 143 output neurons followed by a softmax layer[1], obtaining a 94.93% accuracy on a held-out test set. Aiming at this model, we train a pair of eye-glasses against 30 benign images for each of the three authors. Videos containing 386, 309 and 410 frames for each author are collected, with 4.94%, 93.85% and 84.54% successful rates. Among these successful attacks, 68.42%, 79.31% and 80.83% are successfully detected by TaintRadar, with a 79.78% TPR in total. We achieve a better result than on the VGG-Face Model. The reason is that the model trained on a small dataset has less impact on benign images. However, the estimated critical region and the intersection region of the VGG-Face Model [1] are relatively large, leading to a higher FPR. This experiment result demonstrates that TaintRadar is also robust on the eye-glasses attack in the real world.

E. Run-time Computation Overhead

For deployment in a real-time neural network system, e.g., surveillance, the run-time overhead of TaintRadar should be small. Our experiments are executed on an RTX 2080Ti GPU with an i7-6850K CPU. We average the run-time computation delay over processing 100 images on VGG16 for ImageNet, which shares the same structure with VGG-Face. The overhead is about 79ms for each image when the K value set to 9. For each increase of K, TaintRadar incurs an extra delay of around 4ms. Actually, the delay can be further reduced by migrating the CPU computations to GPU or performing parallel computations. This result is around 100X faster than SentiNet [19] that requires around 7.73s for each image under the same environment. Therefore, TaintRadar is readily deployable in real time-bounded DNN systems.

VII. Robustness Evaluation

We evaluate the robustness of TaintRadar against two types of advanced attacks. First, the attackers use patches of different numbers or shapes to construct adversarial variants. Second, the attackers conduct adaptive white-box attacks using prior knowledge of TaintRadar. We study the attacks on scene classification tasks with the VGG16 model, including both universal and localized attacks. The effectiveness of TaintRadar against these attacks can be generalized to attacks over other tasks. In the experiments, we set FPR as 6% and use random noise as filling pattern the same as in Section VI.

A. Robustness against Adversarial Variants

Attacks Using Multiple Patches. To validate if TaintRadar can identify more than one critical region, we launch attacks using multiple patches. We construct 25 patches that target at mislabelling 400 benign images as *toaster*, i.e., one patch for 16 images. When the same patch, with a size of 0.4, is applied twice on the left and the right corners of victim images, all victim images are successfully misclassified as the target label. Our experiments show that TaintRadar can achieve 91.5% TPR on detecting adversarial examples with multiple patches.

Patches of Different Shapes. We also evaluate the effectiveness of TaintRadar on detecting adversarial patches of different shapes. We construct star-shaped and lightning-shaped patches as examples. We randomly select 200 images from ImageNet and fool the VGG16 model to misclassify them as *toaster*. To avoid covering main objects of original images, we put the patch on the bottom right of victim images. We successfully generate 156 adversarial images each with a star-shaped advesarial object and 158 adversarial images each with a lightning-shaped adversarial object. Among them, TaintRadar achieves 92.31% and 90.5% TPRs for star-shaped and lightning-shaped adversarial objects, respectively.

[1] [14] appended an extra sigmoid layer before the softmax. However, we think this was a typo. After normalized by a sigmoid, the final confidence of each label could not be greater than $\frac{e'}{\sum_{i \neq t} e^0 + e^1} \approx \frac{2.718}{142+2.718} \approx 0.0188$.

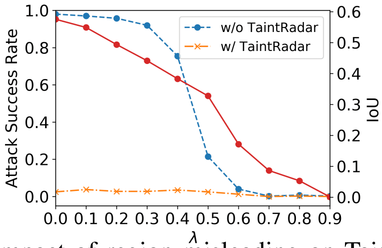


Fig. 6: The impact of region misleading on TaintRadar. Solid line shows the averaged IoU value of generated attacks; dashed lines show the attack success rates with or without TaintRadar.

B. Robustness against Adaptive Attacks

We show the robustness of TaintRadar against adaptive attacks that have white-box knowledge of TaintRadar. Specifically, an attacker can choose to mislead critical region estimation of TaintRadar or manipulate the rankings of target labels for fooling the critical region based detection of TaintRadar. Note that, partial attacks can be seen as an adaptive attack against defense methods that only consider universal attacks.

Misleading Critical Region Estimation. Inspired by a misleading attack that constructs imperceptible pixel-level perturbations [28], we construct localized attacks that can interfere with TaintRadar by adding pixel-level perturbations in a limited area[2]. The attack aims at deceiving CNN models and misleading estimated regions. Through our extensive experiments, we find that the attack does not have an impact on TaintRadar since it is hard for the attack to achieve both misclassification and region misleading. We use $1-\lambda$ and λ to balance the optimization goal of misclassification and region misleading, respectively, where a larger λ means the optimization focuses more on misleading the critical region estimation. We generate 25 patches with varid λ and apply them to 400 images for each λ. To show the accuracy of our critical region estimation under the misleading attacks, we use Intersection over Union (IoU), i.e., $\frac{|E \cap G|}{|E \cup G|}$, where E is the TaintRadar-estimated region and G is the actual adversarial region. Figure 6 shows that the value of IoU drops as λ increases, which means the attack misleads our estimated area away from the actually adversarial region. At the same time, the attack success rate (without TaintRadar) decreases, because the optimization process focuses less on fooling the classification task. In other words, we find these two optimization tasks are contradictory to each other. When using TaintRadar, the attack success rate remains below 10% all the time, illustrating that TaintRadar can effectively inhibit the adaptive attacks. Moreover, we build two other adaptive attack methods which aim to minimize TaintRadar-estimated regions and make TaintRadar identify a pre-defined area as the critical region, respectively. TaintRadar shows similar effectiveness against those attacks.

Deceiving Critical Region Based Detection. Attackers may attempt to evade the detection of TaintRadar by adaptively controlling the ranking changes of the target label to be lower than the threshold ΔR. More specifically, attackers can select a target label with the highest confidence in victim images. We implement both universal and partial attacks as follows.

To construct the universal attacks, we collect images of different source-labels, average the confidence of these images, and then choose the label with the highest ranking (instead of using the original label) as the target label. Here, we generate 50 patches and apply each to 16 randomly sampled images. Among all 800 generated images, 766 images can successfully fool the VGG16 model and 91.78% of them can be detected by TaintRadar. For the partial attacks, we collect images with the same source-label under different settings (e.g., light condition and background) and set the target label as the one with the highest ranking, achieving 789 successful attack attempts. TaintRadar still achieves a 68.44% detection rate. We see that the attacks do not have a significant impact on TaintRadar though some attacks can evade detection. The reason is that, in the presence of TaintRadar, the ability of the attacker is significantly weakened. First, the attack interfaces are greatly reduced under TaintRadar since an attacker can use a very limited number of labels to construct the attacks. For instance, an attacker can only use one label out of 1000 labels to construct the attack in our experiments. Second, in real world, it is difficult for attackers to accurately determine the target label since the rankings of the target label are not stable and will change under different environmental conditions.

End-to-end Attack. To evaluate the robustness of our entire detection process, we build an optimization-based end-to-end attack. Because of the non-differentiable step of binarization, TaintRadar cannot be directly attacked with a gradient-based approach. Backward Pass Differentiable Approximation (BPDA) [29] is an approach designed for this situation, which runs through the whole process directly to get the prediction and calculates gradients using a differentiable approximation with respect to the prediction. More specifically, we use $\hat{\beta}_{i,j} = \frac{1}{1+e^{-t(L_{i,j}-T)}}$ to approximate the binarization step, where t is the temperature increasing along iterations to generate result closer to binarization, $L_{i,j}$ denotes the value at the position (i,j) in the generated heatmap, and T is the binarization threshold. Then, we replace β with $\hat{\beta}$. When generating attacks, we initialize t as 1 and increase it by 1 after 100 iterations. The step size and iterations are set as 1 and 2000, respectively, using an Adam optimizer. We randomly select 200 images and run attacks targeting *toaster* individually against each image, with the knowledge of all the internal details of the defense. Note we ignore some physical constraints, such as location invariance and printability, to enhance the capability of attackers. Without TaintRadar, a total of 79 images are misclassified, but none of them are successfully attacked into the target label. With TaintRadar, the attack success rate is 25%. The overall robustness of TaintRadar against BPDA is 75% when 7% of the whole image is perturbed. This result largely surpasses DW [17] with only 5% robustness when 5% of the whole image is perturbed [15]. One main reason is that TaintRadar leverages regional information of deep representations at intermediate layers, which is more robust against pixel-level changes than methods using pixel-prediction correspondence. Thus, bypassing TaintRadar restricts the ability of achieving a misclassification attack in a small and contiguous area, and vice versa.

VIII. Discussion

GAN for Accurate Region Estimation and Image Recovery. In the current design of TaintRadar, we utilize filling patterns

[2] The original attack proposed in [28] needs to change pixels in the entire images, which cannot be directly applied in localized attacks.

to fill the critical regions so as to restore the original information of the images. Actually, we can leverage Generative Adversarial Networks (GAN) to inpaint this region instead of filling it with a filling pattern. Thus, we can restore the original information with a reduced FPR. We leave it as an interesting future work. Moreover, it is desirable to have more accurate image recovery in some scenarios, e.g., autopilots. Similarly, we can utilize GAN to recover the original label by leveraging image characteristics, such as image gradient, to eliminate the impact of residual perturbations.

Defenses against Backdoor Attacks. In this paper, we focus on localized adversarial examples without tampering with the model. Unlike post-training processes in the adversarial examples, the backdoor attack is a targeted attack launched during model training. Gu et al. [30] first demonstrate that CNN models can be easily backdoored by injecting poisoned data in the training dataset. Moreover, Liu et al. [31] show that they can launch trojan attacks by modifying some specific neurons without access to the training dataset. By using preliminary experiments, we find that the final outputs of TaintRadar on images with backdoor triggers are also centralized on the attacked area. Thus, TaintRadar can potentially be applied to defend against this type of attack.

Defenses against Attacks on Traffic Signs. The traffic sign attack [32] is another stealthy attack in the physical world, which uses stickers or camouflage graffiti to fool neural networks. For example, an attacker can put some stickers on a stop sign, which will not be perceived by human eyes but can cause misclassification to the classifier. However, The dimensions of the input features in road signs are very small in the commonly used models, e.g., road sign images are normally resized to 32×32 before the model is trained [33], [32]. Thus, the original attacked regions become a number of pixels in the resized image. Since our critical region estimation algorithm leverages deep representations at intermediate layers and neglects the input-prediction correspondence, it cannot locate pixels in images when they are scattered. In other words, as the granularity of the last convolutional layer is significantly larger than each scattered perturbed area, from the perspective of our approach, the attack stops to be localized. We consider addressing this problem using the traditional pixel-level threat model in our future work.

IX. Related Work

Adversarial Examples against Neutral Networks. Recent studies [3], [4], [5], [6], [7], [8], [34], [35] show that neural networks can be easily fooled by adversarial examples. They generate imperceptible perturbations bounded by a norm-ball constraint, e.g., l_0, l_2, or l_∞, in benign images so as to create traditional pixel-level adversarial examples. Normally, these attacks cannot be physically implemented in a real world environment. In this paper, we focus on localized attacks that can build small and physically visible adversarial objects. For example, adversarial patch can be printed and placed in an image to deceive the classifier [13], and traffic road signs with inconspicuous stickers might be misclassified by self-driving cars [32]. Moreover, state-of-the-art face recognition systems can be fooled by adversarial objects, e.g., crafted glasses [14] and graffiti stickers on hats [36].

Defenses against Traditional Adversarial Examples. Most existing defenses focus on imperceptible adversarial perturbations [12], [37], [38], [10], [39], [40], [41], [42]. Papernot et al. [12] proposed defensive distillation extracting the key information from a pretrained DNN to improve the resilience of a model to adversarial examples. Also, an autoencoder (AE) can be applied as a denoiser to purify adversarial effects of an input [39]. Similarly, PixelDefend [38] projects the adversarial input back to the training distribution. Moreover, adversarial training [10], [40], [37] improves the robustness of the DNN models by training against known attacks. Certified robustness approaches [41], [42] give a lower-bound on the adversarial accuracy. However, these defenses cannot be directly applied to throttle localized adversarial examples that are visible and constrained to small regions.

Defenses against Localized Adversarial Examples. Some defense approaches[15], [19], [17], [18], [16] have been proposed to resist localized adversarial examples. Input transformation approaches [18], [17] focus on mitigating the effect of adversarial perturbations and recovering to the original label of the benign input. However, none of these approaches show robustness under the BPDA attack [29], [15]. Chiang et al. [15] transferred certified robustness [41], [42], which gives a certificate when an output lies in the interval bound formed during the training process. Wu et al. [16] combined a new abstract attack model that represents physically realizable attacks with adversarial training [40], [37] to increase the robustness of a model. However, both robustness approaches require extra training with high training overhead and cannot work well at ImageNet scale [15], [10]. Also, robustness approaches do not block out adversarial examples completely and cannot be used to guard existing models. Thus, a robust detection approach is in urgent need. SentiNet [19] is a detection approach that sits the closest to our threat model. It leverages the property that adversarial objects can generalize to a large distribution of inputs; however, it is not held by localized partial attacks that only target a small subset of source labels, e.g., eyeglasses attack [14]. Moreover, the expensive selective search and testing algorithm make it hard to be deployed in time-bounded systems.

X. Conclusion

In this work, we propose TaintRadar to defend against localized adversarial examples. TaintRadar is the first generic defense system that can detect various types of localized adversarial examples, in particular the partial attacks such as the eye-glasses attack. TaintRadar uses feature maps in the last convolutional layers to identify critical regions in images, and accurately identifies the differences between benign and malicious images by evaluating the impacts of the regions. We evaluate the effectiveness and the robustness of TaintRadar using different settings in two typical scenarios, i.e., scene classification and face recognition. Experimental results show that TaintRadar can detect partial attacks, which cannot be captured by all existing defenses. Moreover, we show that TaintRadar can effectively throttle the localized attacks in the real world.

Acknowledgements

This work is supported in part by NSFC under Grant 61572278, BNRist under Grant BNR2020RC01013 and U.S. ARO under Grant W911NF-17-1-0447. Qi Li is the corresponding author of this paper.

References

[1] O. M. Parkhi, A. Vedaldi, A. Zisserman *et al.*, "Deep face recognition." in *bmvc*, vol. 1, no. 3, 2015, p. 6.

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[3] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[4] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2574–2582.

[5] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1765–1773.

[6] Y. Dong, T. Pang, H. Su, and J. Zhu, "Evading defenses to transferable adversarial examples by translation-invariant attacks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4312–4321.

[7] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 39–57.

[8] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*, 2019.

[9] G. K. Dziugaite, Z. Ghahramani, and D. M. Roy, "A study of the effect of jpg compression on adversarial images," *arXiv preprint arXiv:1608.00853*, 2016.

[10] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," *arXiv preprint arXiv:1611.01236*, 2016.

[11] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," *arXiv preprint arXiv:1704.01155*, 2017.

[12] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2016, pp. 582–597.

[13] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer, "Adversarial patch," *arXiv preprint arXiv:1712.09665*, 2017.

[14] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 1528–1540.

[15] P.-Y. Chiang, R. Ni, A. Abdelkader, C. Zhu, C. Studor, and T. Goldstein, "Certified defenses for adversarial patches," *arXiv preprint arXiv:2003.06693*, 2020.

[16] T. Wu, L. Tong, and Y. Vorobeychik, "Defending against physically realizable attacks on image classification," *arXiv preprint arXiv:1909.09552*, 2019.

[17] J. Hayes, "On visible adversarial perturbations & digital watermarking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1597–1604.

[18] M. Naseer, S. Khan, and F. Porikli, "Local gradients smoothing: Defense against localized adversarial attacks," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2019, pp. 1300–1307.

[19] E. Chou, F. Tramèr, G. Pellegrino, and D. Boneh, "Sentinet: Detecting physical attacks against deep learning systems," *arXiv preprint arXiv:1812.00292*, 2018.

[20] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 618–626.

[21] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14 410–14 430, 2018.

[22] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[25] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, "Attribute and simile classifiers for face verification," in *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 2009, pp. 365–372.

[26] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in neural information processing systems*, 2014, pp. 3320–3328.

[27] N. Papernot, F. Faghri, N. Carlini, I. Goodfellow, R. Feinman, A. Kurakin, C. Xie, Y. Sharma, T. Brown, A. Roy, A. Matyasko, V. Behzadan, K. Hambardzumyan, Z. Zhang, Y.-L. Juang, Z. Li, R. Sheatsley, A. Garg, J. Uesato, W. Gierke, Y. Dong, D. Berthelot, P. Hendricks, J. Rauber, and R. Long, "Technical report on the cleverhans v2.1.0 adversarial examples library," *arXiv preprint arXiv:1610.00768*, 2018.

[28] X. Zhang, N. Wang, H. Shen, S. Ji, X. Luo, and T. Wang, "Interpretable deep learning under fire," *arXiv preprint arXiv:1812.00891*, vol. 2, 2018.

[29] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," *arXiv preprint arXiv:1802.00420*, 2018.

[30] T. Gu, B. Dolan-Gavitt, and S. Garg, "Badnets: Identifying vulnerabilities in the machine learning model supply chain," *arXiv preprint arXiv:1708.06733*, 2017.

[31] Y. Liu, S. Ma, Y. Aafer, W. C. Lee, and X. Zhang, "Trojaning attack on neural networks," in *Network and Distributed System Security Symposium*, 2018.

[32] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning visual classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1625–1634.

[33] P. Sermanet and Y. LeCun, "Traffic sign recognition with multi-scale convolutional networks." in *IJCNN*, 2011, pp. 2809–2813.

[34] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[35] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2016, pp. 372–387.

[36] S. Komkov and A. Petiushko, "Advhat: Real-world adversarial attack on arcface face id system," *arXiv preprint arXiv:1908.08705*, 2019.

[37] Q.-Z. Cai, M. Du, C. Liu, and D. Song, "Curriculum adversarial training," *arXiv preprint arXiv:1805.04807*, 2018.

[38] Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman, "Pixeldefend: Leveraging generative models to understand and defend against adversarial examples," *arXiv preprint arXiv:1710.10766*, 2017.

[39] P. Ghosh, A. Losalka, and M. J. Black, "Resisting adversarial attacks using gaussian mixture variational autoencoders," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 541–548.

[40] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[41] S. Gowal, K. Dvijotham, R. Stanforth, R. Bunel, C. Qin, J. Uesato, R. Arandjelovic, T. Mann, and P. Kohli, "On the effectiveness of interval bound propagation for training verifiably robust models," *arXiv preprint arXiv:1810.12715*, 2018.

[42] M. Mirman, T. Gehr, and M. Vechev, “Differentiable abstract interpretation for provably robust neural networks,” in *International Conference on Machine Learning*, 2018, pp. 3578–3586.