Design and Interpretation of Universal Adversarial Patches in Face Detection

Xiao Yang[1*]   Fangyun Wei [2*]   Hongyang Zhang[3*]

Xiang Ming[4]   Jun Zhu[5]

Tsinghua University[1,5]   Microsoft Research Asia[2]   TTIC[3]   Xi'an Jiaotong University[4]

{yangxiao19[1],dcszj[5]}@mails.tsinghua.edu.cn   fawe@microsoft.com[2]

hongyanz@ttic.edu[3]   xjtustu.mx@stu.xjtu.edu.cn[4]

Abstract

We consider universal adversarial patches for faces — small visual elements whose addition to a face image reliably destroys the performance of face detectors. Unlike previous work that mostly focused on the algorithmic design of adversarial examples in terms of improving the success rate as an attacker, in this work we show an interpretation of such patches that can prevent the state-of-the-art face detectors from detecting the real faces. We investigate a phenomenon: patches designed to suppress real face detection appear face-like. This phenomenon holds generally across different initialization, locations, scales of patches, backbones, and state-of-the-art face detection frameworks. We propose new optimization-based approaches to automatic design of universal adversarial patches for varying goals of the attack, including scenarios in which true positives are suppressed without introducing false positives. Our proposed algorithms perform well on real-world datasets, deceiving state-of-the-art face detectors in terms of multiple precision/recall metrics and transferring between different detection frameworks.

1. Introduction

Adversarial examples are a central object of study in computer vision [33], machine learning [39, 23], security [26], and other domains [13]. In computer vision and machine learning, study of adversarial examples serves as evidences of substantial discrepancy between human vision system and machine perception mechanism [30, 25, 2, 9]. In security, adversarial examples have raised major concerns on the vulnerability of machine learning systems to malicious attacks. The problem can be stated as modifying an image, subject to some constraints, so that learning system's response is drastically altered, e.g., changing the classifier or detector output from correct to incorrect. The constraints either come in the *human-imperceptible* form

_____
[*]Equal contribution.

such as bounded ℓ_p perturbations [39], or in the *human-perceptible* form such as small stickers or patches [31, 7]. The focus of this work is the latter setting.

While image classification has been repeatedly shown to be broadly vulnerable to adversarial attacks [30], it is less clear whether object detection is similarly vulnerable [27, 21, 19, 20, 10]. State-of-the-art detectors propose thousands of candidate bounding boxes and the adversarial examples are required to fool all of them simultaneously. Nonetheless, for selected object categories the attacks and defenses have been studied extensively. They include objects like stop signs or pedestrians [22, 31, 7], but few attempts have been made on generating adversarial examples for faces. This is in spite of face detection as a task enjoying significant attention in recent years, due to its practical significance on its own and as a building block for applications such as face alignment, recognition, attribute analysis, and tracking. Publicly available face detectors [41, 24, 17, 40] can achieve performance on par with humans, e.g., on FDDB [12] and WIDER FACE dataset [35], and are insensitive to the variability in occlusions, scales, poses and lighting. However, much remains unknown concerning the behaviors of face detectors on adversarial patches. Our work sheds new light on this question and shows that a simple approach of pasting a single universal patch onto a face image can dramatically harm the accuracy of state-of-the-art face detectors. We propose multiple approaches for building adversarial patches, that address different desired precision/recall characteristics of the resulting performance. In addition to empirical performance, we are interested in understanding the nature of adversarial patch on face detection.

Significance. The study of adversarial patch in face detection is important in multiple aspects: a) In security, adversarial patch serves as one of the most common forms of physical attacks in the detection problems, among which face detection has received significant attention in recent years. b) The study of adversarial patch may help understand the discrepancy between state-of-the-art face detec-
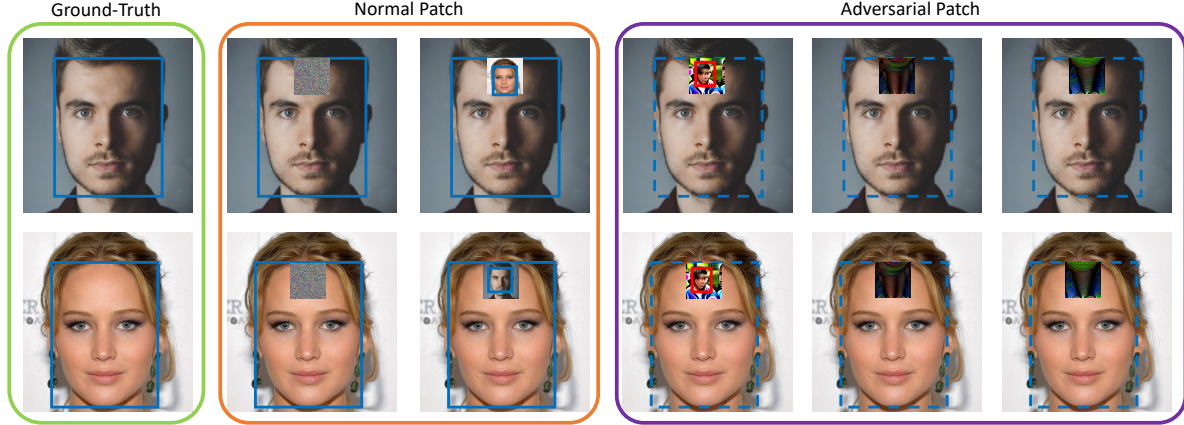
Figure 1. Properties and effect of different patches. In each image we show true positive (solid blue lines), false positive (red) and missed detection (dashed blue lines). **Left (green) box:** the clean input images. **Middle (orange) box:** pasting an un-optimized noise patch or a downsized face patch on the image does not affect the detectors. **Right (purple) box:** universal adversarial patches produced by our methods successfully suppress true positives, but their properties differ depending on the optimization technique. **From left to right:** *Patch-IoU* appears person-like and induces false positives; *Patch-Score-Focal* and *Patch-Combination* avoid the false positives. The patches are not necessarily pasted at forehead as demonstrated in Section 3.

tors and human visual system, towards algorithmic designs of detection mechanism as robust as humans. c) Adversarial patch to face detectors is human-perceptible and demonstrates significant interpretation as shown in this paper.

Challenges. In commonly studied classification problems, adversarial perturbations are inscrutable and appear to be unstructured, random noise-like. Even when structure is perceptible, it tends to bear no resemblance to the categories involved. Many observations and techniques for classification break down when we consider more sophisticated face detection tasks. Compared with other detection tasks, generating adversarial examples for face detection is more challenging, because the state-of-the-art face detectors are able to detect very small faces (e.g., 6×6 pixels [24]) by applying the multi-scale training and testing data augmentation. While there is a large literature on the algorithmic designs of adversarial examples in terms of improving the success rate as an attacker, in this work we focus on the *interpretation* of learning a small, universal adversarial patch which, once being attached to human faces, can prevent the state-of-the-art face detectors from detecting the real faces.

Our results. The gist of our findings is summarized in Figure 1. We consider state-of-the-art face detectors, that perform very accurately on natural face images. We optimize a universal adversarial patch, to be pasted at input face images, with the objective of suppressing scores of true positive detection on training data. This is in sharp contrast to most of existing works on adversarial examples for faces in the form of sample-specific, imperceptible perturbations, but a *universal* (independent of the input image) and *interpretable* (semantically meaningful) patch that reliably destroys the performance of face detectors is rarely studied in the literature. Our resulting patch yields the following observations.

- It succeeds in drastically suppressing true positives in test data. The attack also transfers between different face detection frameworks, that is, a patch which is trained on one detection framework deceives another detection framework with a high success rate.

- It looks face-like to humans, as well as to the detectors. Thus, in addition to reducing recall, it reduces precision by inducing false positives.

- Despite superficial face-likeness of the learned adversarial patch, it cannot be simply replaced by a real face patch, nor by a random noise pattern; affixing these to real faces does not fool the detectors.

- Surprisingly, these observations hold generally across different detection frameworks, patch initialization, locations and scales of pasted patch, etc. For example, even while initializing the patch with an image of a non-face object or a complex scene, after 100 epochs the resulting adversarial patch comes to resemble a face (see Figure 2).

In some scenarios the attacker may want to suppress correct detection without creating false positives (e.g., to hide the presence of any face). We propose modified approaches that produce patches with this property. Intuitively, the approaches minimize the confidence scores of bounding boxes as long as they are larger than a threshold. Experiments verify the effectiveness of the proposed approaches (see the last two columns in Figure 1).

Summary of contributions. Our work explores the choices in design of universal adversarial patches for face detection.

- We show how such patches can be optimized to harm performance of existing face detectors. We also show that when the objective is purely to suppress true de-
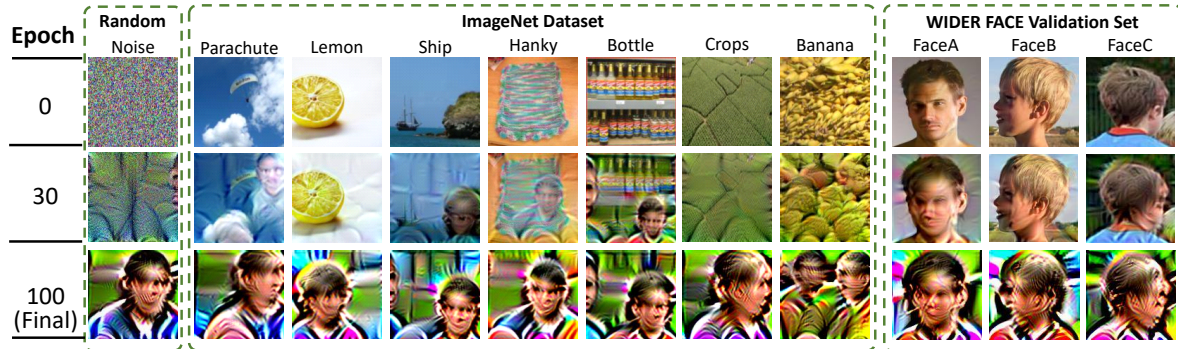
2

Figure 2. Adversarial patches from different *initialization* by *Patch-IoU*. The first row is the initial patches, and the second row and the third row represent the intermediate and final patches, respectively. All of the final patches here are detected as faces by face detectors.

tection, the resulting patches are interpretable as face-like and can be detected by baseline detectors, with this property holding true across different experimental settings.

- In response to some security-focused scenarios where the adversary may want to suppress correct detection without creating false positives, we describe methods to produce equally successful universal adversarial patches that do not look like faces to either humans nor the face detectors, thus reducing detection rate without increasing false positives. Our proposed algorithms deceive the state-of-the-art face detectors [24] on real-world datasets in terms of multiple precision/recall metrics, and transfers between detection frameworks.

2. Related Work

Adversarial examples on object detection. Adversarial examples on general object detection have been extensively studied in the recent years [37, 16]. A commonly explored domain for adversarial examples in detection is stop sign detection [6, 7, 8, 4]. Stop signs have many structural properties that one can exploit: standard red color, with fixed shape and background. By attaching a carefully-calculated sticker to the stop sign, [7] caused a mis-detection rate as high as 84.8% on the captured video frames obtained on a moving vehicle. Inspired by an observation that both segmentation and detection are based on classifying multiple targets on an image, [33] extended the methodology of generating adversarial examples to the general object detection tasks. More recently, [31] proposed a method to generate a universal adversarial patch to fool YOLO detectors on pedestrian data set. Another line of research related to our work is the perturbation-based adversarial examples for face detectors [18]. This line of works adds sample-specific, human-imperceptible perturbations to the images globally. In contrast, our adversarial patches are universal to all samples, and our patches are visible to humans and show strong interpretation. While optimizing a patch to fool detectors

has previously been used as a simulation of physical-world attacks, to our knowledge no properties of such patches to human visual system have been shown.

Adversarial examples in face recognition. To fool a face recognition system in the physical world, prior work has relied on active explorations via various forms of physical attacks [15]. For example, [29, 28, 34] designed a pair of eyeglass frames which allows a face to evade being recognized or to impersonate another individual. Other attacks in the physical world that may fool (face) classifiers include adversarial patches [3], hats [14], and 3D-printing toys [1]. However, these adversarial examples did not afford any semantic interpretation. Though scaled adversarial perturbations of robustly trained classifiers might have semantic meaning to humans [39], those adversarially trained classifiers are not widely used due to an intrinsic trade-off between robustness and accuracy [32, 39]. In contrast to this line of research, our goal is to demonstrate a reliable interpretation of adversarial patches for face detectors.

Face detection. Face detection is typically less sensitive to the variation of face scales, angles, and other external factors such as occlusions and image qualities. Modern face detection algorithms [41, 24, 17, 40] take advantage of anchor[1] based object detection methods, such as SSD [21] and RetinaNet [20], Faster R-CNN [27] and Mask R-CNN [10], and can achieve performance on par with humans on many public face detection benchmarks, such as FDDB and WIDER FACE dataset. They can detect faces as small as 6 pixels by applying multi-scale training and testing data augmentation, which serves as one of the primary differences with general object detection. Anchor based face detectors use IoU between anchors (or proposals) and ground truth to distinguish positive samples from negative ones in the sampling step during training. In the inference stage, average precision (*AP*) is a commonly used metric to evaluate the detection performance. However, as we il-

---

[1]Anchors are a set of predefined and well-designed initial rectangles with different scales and ratios. They are densely tiled on feature maps for object classification and bounding box regression.
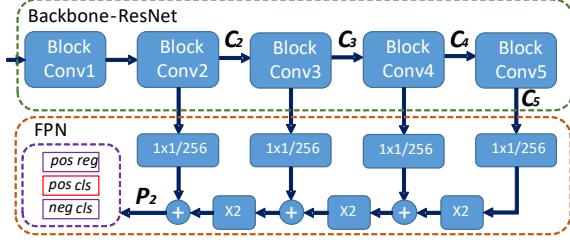
Figure 3. SLN framework in our face detection baseline model, where $\times 2$ represents the bilinear upsampling, $+$ represents the element-wise summation, and $1 \times 1/256$ represents the 1×1 convolution with 256 output channels. The feature map P_2 is used as the only detection layer where all anchors are tiled with stride 4 pixels. The positive classification loss is our main attacking target.

lustrate in the following sections, this criterion is not a fair metric to evaluate the impact of adversarial patches on a face detector.

3. Interpretation of Adversarial Patch as Face

In this section, we present our main experimental results on the interpretation of adversarial patch. We show that, on one hand, the adversarial patch optimized by the proposed *Patch-IoU* method looks like a face. The patch can be detected by the baseline face detection model, even in the absence of extra constraints to encourage the patch to be face-like. On the other hand, attaching a face picture to a real face does not fool the detector (see Figure 1). The phenomenon holds generally across different setups.

3.1. Preliminaries on face detection

Dataset. We use WIDER FACE [35] training dataset to learn both the face detector and the adversarial patch. The WIDER FACE dataset contains 32,203 images and 393,703 annotated face bounding boxes with high degree of variability in scales, poses, occlusions, expression, makeup, and illumination. According to the detection rate of Edge-Box [42], WIDER FACE dataset is split into 3 subsets: Easy, Medium and Hard. The face detector and adversarial patch are evaluated on the validation set. The set of ground-truth bounding boxes for an image is defined as $\{B_i\}$, where $B_i = (x^i, y^i, w^i, h^i)$, (x^i, y^i) is the center of the box, and w^i and h^i are the width and height of the bounding box, respectively.

Face detection framework. We use the state-of-the-art face detection framework [24] as the baseline model, and we name it as Single Level Network (SLN). Figure 3 illustrates the network structure. We use ResNet [11] as backbone, with a bottom-up feature refusion procedure in Feature Pyramid Network (FPN) [19]. We obtain a high-resolution and informative feature map P_2 (stride equals to 4 pixels). Anchors with scales $\{16, 32, 64, 128\}$ and aspect ratio 1 are tiled on P_2 uniformly. We denote by $\{A_i\}$ the

Table 1. Precision and recall of SLN baseline model and pasting various patches with (without) *Patch-IoU* algorithm on WIDER FACE validation set under $\delta = 0.99$ (see Figure 2 for visualized results).

Precision/ Recall	Easy	Medium	Hard	All
Baseline-SLN	99.0/ 73.4	99.4/ 62.4	99.4/ 27.9	99.4/ 22.5
Noise w/o *Patch-IoU*	99.1/ 54.9	99.4/ 41.5	99.4/ 17.6	99.4/ 14.2
Parachute w/o *Patch-IoU*	99.1/ 51.8	99.3/ 37.2	99.3/ 15.8	99.3/ 12.8
Lemon w/o *Patch-IoU*	98.9/ 53.4	99.2/ 39.4	99.2/ 16.7	99.2/ 13.4
Bottle w/o *Patch-IoU*	99.1/ 53.5	99.4/ 41.1	99.4/ 17.3	99.4/ 13.9
Banana w/o *Patch-IoU*	99.1/ 55.2	99.4/ 41.4	99.4/ 17.5	99.4/ 14.1
FaceA w/o *Patch-IoU*	51.8/ 30.2	61.4/ 24.2	61.8/ 10.3	61.8/ 8.3
FaceB w/o *Patch-IoU*	77.8/ 39.5	83.5/ 30.1	83.6/ 12.9	83.6/ 10.4
FaceC w/o *Patch-IoU*	98.4/ 38.3	98.9/ 29.8	98.9/ 12.7	98.9/ 10.2
Noise w/ *Patch-IoU*	2.7/2.7	6.5/3.7	7.3/1.8	7.3/1.4
Parachute w/ *Patch-IoU*	2.1/0.5	4.8/ 0.7	5.9/ 0.4	5.9/ 0.3
Lemon w/ *Patch-IoU*	0.2/0.1	0.9/0.3	1.0/0.2	1.0/0.1
Bottle w/ *Patch-IoU*	1.1/1.1	2.2/1.2	2.5/ 0.6	2.6/0.5
Banana w/ *Patch-IoU*	10.2/5.0	19.0/5.6	20.3/2.5	20.3/2.0
FaceA w/ *Patch-IoU*	0.0/0.0	0.0/0.0	0.0/0.0	0.0/0.0
FaceB w/ *Patch-IoU*	0.1/0.0	2.3/0.2	2.6/0.1	2.6/0.1
FaceC w/ *Patch-IoU*	0.1/0.1	0.2/0.1	0.3/0.0	0.3/0.0

set of all anchors. We apply IoU regression loss, anchor matching criterion, and group sampling strategy in [24] to train our baseline model. Formally, let $\text{IoU}(A_i, B_j)$ denote the IoU between the i-th anchor and the j-th ground-truth bounding box. Anchors with $\text{IoU}(A_i, B_j) > 0.6$ and $\text{IoU}(A_i, B_j) < 0.4$ will be set as positive and negative samples, respectively. Finally, we define a multi-task loss $\mathcal{L} = \mathcal{L}_{cls}^p + \mathcal{L}_{cls}^n + \mathcal{L}_{reg}$, where $\mathcal{L}_{cls}^p$ and $\mathcal{L}_{cls}^n$ denote the standard cross entropy loss for positive and negative samples, respectively, and $\mathcal{L}_{reg} = \frac{1}{N_{reg}} \sum_{(A_i, B_j)} \|1 - \text{IoU}(A_i, B_j)\|_2^2$ represents the IoU least square regression loss. If not specified, we use ResNet-18 as our defaulted backbone.

Training details of face detector. We use random horizontal flip and scale jittering as data augmentation during training. For scale jittering, each image is resized by a factor of $0.25 \times n$, where n is randomly chosen from $[1, 8]$. Then a random cropping procedure is used to crop a patch from the resized image to ensure that the longer side of the image does not exceed 1,200 pixels. We set the initial learning rate as 0.01 and decay the learning rate by a factor of 0.1 on the 60-th and the 80-th epochs. The model is trained for 100 epochs with synchronized stochastic gradient descent over 8 NVIDIA Tesla P100 GPUs and 8 images per mini-batch (1 image per GPU). The momentum is 0.9 and weight decay is set to be 5×10^{-5}. Backbone is initialized with ImageNet pre-trained weights. We fine-tune the model on the WIDER FACE training set and test on validation set with same image pyramid strategy as in the training procedure. We use Non-Maximum Suppression (NMS) as post-processing. The first line in Table 1 shows the precision and recall of the baseline model. *Easy*, *Medium*, *Hard*, and *All*[2] represent the results from easy subset, medium subset, hard

---

[2]Official WIDER FACE testing script (http://shuoyang1213.me/WIDERFACE/) only gives results of *Easy*, *Medium* and *Hard* subsets. We reimplement the test script to support testing on the whole validation set.
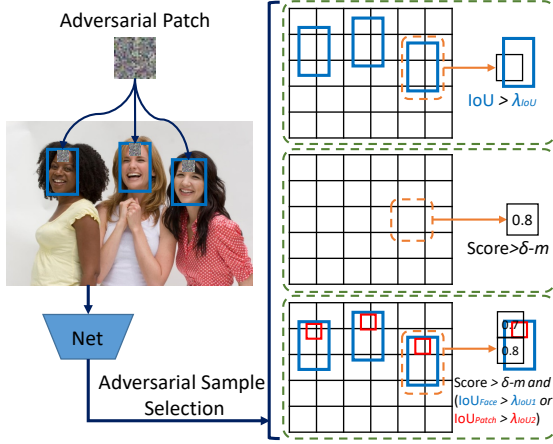
Figure 4. Different optimization methods for generating adversarial patches. The core difference is to involve different samples in the optimization of adversarial patch. From top to bottom: *Patch-IoU*, *Patch-Score-Focal* and *Patch-Combination*, respectively.

subset and the whole validation set, respectively.

3.2. Interpretation of adversarial patch

Training details of adversarial patch. For detection, as opposed to balanced classification problems, there are two types of errors: the false-positive error and the false-negative error. In response to an intrinsic trade-off between precision and recall in the face detection tasks, existing works set a score threshold δ to keep high precision: output proposals with confidence scores higher than δ are treated as faces. The goal of adversary is to decrease the confidence scores to be lower than δ by pasting a carefully-calculated, universal patch to human faces. We show that the adversarial patch can make the real faces invisible to various detectors. Formally, we define the patch P as a rectangle which is denoted by (x^P, y^P, w^P, h^P), where (x^P, y^P) is the center of the patch relative to the ground-truth bounding box B_i, and w^P and h^P represent its width and height, respectively. In our experiments, we set both w^P and h^P as 128 since the largest anchor size is 128 in the SLN face detection framework. For each of the ground-truth bounding box $B_i = (x^i, y^i, w^i, h^i)$ in the given training image, the patch P is resized to $\alpha\sqrt{w^i h^i}$ ($0 < \alpha < 1$) and then placed on B_i with its center position (x^P, y^P). We randomly initialize the patch, and set $\alpha = 0.5$ and $(x^P, y^P) = (w^i/2, \alpha\sqrt{w^i h^i}/2)$, unless otherwise specified. All of the training settings, including the training dataset and the hyper-parameter tuning, are the same as the SLN (or other face detection framework) baseline model.

Optimization. We define *Adversarial Sample Set* $\{AS_i\}$ as the set of selected samples which are involved in the training optimization of the adversarial patch. Each *Adversarial Sample* $AS_i = (A_i, B_i, S_i, P)$ contains four elements: the anchor A_i, the ground-truth bounding box B_i, the face confidence score S_i which represents the output of the classification layer with the softmax operation, and the adversarial patch P^3, respectively. We freeze all weights of the face detector; the patch P is the only variable to be optimized by gradient ascent algorithm. Our goal is to maximize the following loss function:

$$\mathcal{L}_{Adv}(P) = -\frac{1}{N}\sum_{i=1}^{N}\log(S_i), \qquad (1)$$

where N is the size of $\{AS_i\}$. We use the IoU to select $\{AS_i\}$, that is, each sample in $\{AS_i\}$ should satisfy $\text{IoU}(A_i, B_i) > 0.6$, which is exactly the same as the selection of positive samples in the baseline model. We term our algorithm as *Patch-IoU* (see the first row of Figure 4).

Evaluation details. We follow the same testing settings as that of the SLN (or other face detection framework) baseline model. Similar to the existing works, we set a threshold $\delta = 0.99$ to keep high precision: decreasing the scores of the ground-truth faces to be smaller than δ represents a successful attack. We show our visualized results in Figure 1 and the first column of Figure 2, i.e., the evolution of the adversarial patch with random initialization. Table 1 (see Baseline, Noise w/o *Patch-IoU* and Noise w/ *Patch-IoU* three lines) presents the corresponding numerical results on precision and recall with (without) *Patch-IoU* optimization. We have three main observations:

- The drop of recall implies that the detector fails to detect the real faces in the presence of the adversarial patch, i.e., $\text{Score}(\text{RealFace}) < \delta$.
- The patch with 100-epoch training appears face-like. The drop of precision implies that the detector *falsely recognizes* the adversarial patch as a human face, i.e., $\text{Score}(\text{AdversarialPatch}) > \delta$.
- Attaching a face photo to the real faces with the same size and location as the adversarial patch does not fool the detector, that is, we do not observe significant drop of recall.

3.3. Generality

The interpretation of adversarial patch is not a unique property of the setup in Section 3.2. Instead, we show that it is a general phenomenon which holds across different initialization, patch locations and scales, backbones, and detection frameworks.

Initialization. We randomly select seven images from ImageNet [5], three faces from WIDER FACE validation set, and one random image as our initialization. Figure 2 shows the evolution of patches across different training epochs. We observe that the patches come to resemble human faces, even while initializing the patches with non-face objects or a complex scene.

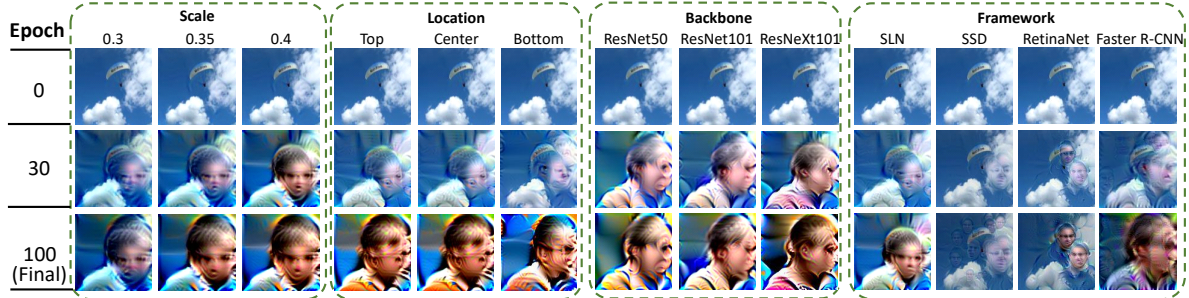[3] All AS_i's share an identical adversarial patch P.

Figure 5. Optimization results by *Patch-IoU* across different scales, locations, backbones and detection frameworks. *Patch-IoU* generates face-like adversarial patch which is falsely detected by various detectors.

Patch locations and scales. To examine whether the interpretation holds across different locations and scales of the patch, we run the algorithm with different patch *scales*: $\alpha \in \{0.3, 0.35, 0.4\}$ and *locations*: top $(x^P, y^P) = (w^i/2, \alpha\sqrt{w^i h^i}/2)$, center $(x^P, y^P) = (w^i/2, h^i/2)$, and bottom $(x^P, y^P) = (w^i/2, h^i - \alpha\sqrt{w^i h^i}/2)$. We observe a similar phenomenon for all these setups, as shown in Figure 5.

Backbones. We see in Figure 5 that the adversarial patches look like human faces for different backbones, including ResNet-50, ResNet-101, and ResNext-101.

Detection frameworks. Besides the face detection framework SLN, we also test three popular detection frameworks: SSD [21], RetinaNet [20] and Faster R-CNN [27]. For Faster R-CNN, we use the SLN as our region proposal network, and the RoIAlign [10] is applied on each proposal to refine face classification and bounding box regression. Except for the detection architecture, all of the experimental setups for the baseline model and the adversarial patch training are the same. Similarly, we observe that the adversarial patches come to resemble human faces (see Figure 5).

Generality across training datasets. We randomly split WIDER FACE training dataset into two subsets, namely, split 1 and split 2, to verify the generality of different training sources. Figure 6 shows the evolution of adversarial samples optimized by *Patch-IoU* method on each split. The final patches are face-like and can be falsely detected by baseline face detector. The two patches look slightly different, partially because of the non-convex properties of optimization [36, 38] and different initialization.

Numerical results. We also report the numerical results of the algorithm. We set $\delta = 0.99$ and show the precision and recall of using various patches to attack the SLN face detector on the WIDER FACE validation set. Table 1 illustrates the effect of eight representative kinds of initialization with (without) *Patch-IoU* optimization. We do not report the numerical results about different patch locations and scales, backbones and detection frameworks, since the results and phenomenon are identical as the initialization.
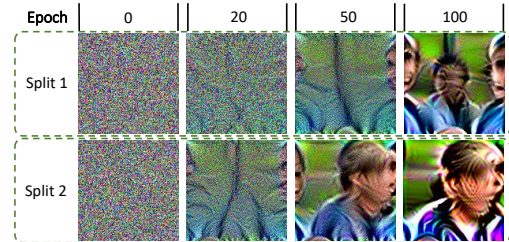

Figure 6. Optimization results by *Patch-IoU* on WIDER FACE training subset, split 1 and split 2. *Patch-IoU* generates face-like adversarial patch which is falsely detected by baseline face detector.

It can be seen that pasting a patch (even initialized as face) without any optimization will cause the recall to drop, but not so drastically. In contrast, the *Patch-IoU* can cause the recall to decrease dramatically across different initialization, leading to a successful attack. However, the adversarial patches also reduce the precision because the face-like patches are falsely detected and the scores of the patches are even higher than those of the true faces. We defer more discussions about evaluation metrics and the issue of precision drop in *Patch-IoU* method to Section 4.

Transferability between different frameworks. We also study the transferability of adversarial patch between different frameworks. Formally, we attach patches optimized from SSD, RetinaNet and Faster R-CNN, respectively, on each ground-truth bounding box in WIDER FACE validation set, and test their attacking performance on SLN baseline detector. Table 2 shows the numerical results. The patch trained on the Faster R-CNN framework enjoys higher success rate as an attacker on the SLN than the SSD and RetinaNet.

Attack by part of patch. To examine the attacking performance of only part of the adversarial patch that is optimized by *Patch-IoU*, we remove a half and one third area of the whole patch and test the performance of the remaining part of the patch on the WIDER FACE validation dataset. Table 3 shows the associated numerical results. We see that

6

Table 2. Precision and recall of different frameworks and transferability of adversarial patch attack from SSD, RetinaNet and Faster R-CNN to SLN under $\delta = 0.99$. A $\rightarrow$ B denotes that the adversarial patch is optimized by detector A and tested on detector B.

Precision/ Recall	Easy	Medium	Hard	All
Baseline-SLN	99.0/ 73.4	99.4/ 62.4	99.4/ 27.9	99.4/ 22.5
Patch-IoU-SLN	2.7/2.7	6.5/3.7	7.3/1.8	7.3/1.4
SSD $\rightarrow$ SLN	42.5/ 29.9	53.4/ 25.1	54.1/ 10.7	54.1/ 8.6
RetinaNet $\rightarrow$ SLN	37.4/ 28.7	48.5/ 24.5	49.2/ 10.5	49.2/ 8.5
Faster R-CNN $\rightarrow$ SLN	32.9/ 3.8	44.9/ 3.4	46.3/ 1.5	46.3/ 1.2

Table 3. Attacking performance of parts of adversarial patches. Half (One-third)-*Part* means removing one half (one-third) area of the whole patches.

Precision/ Recall	Easy	Medium	Hard	All
Baseline-SLN	99.0/ 73.4	99.4/ 62.4	99.4/ 27.9	99.4/ 22.5
Patch-IoU-SLN	2.7/2.7	6.5/3.7	7.3/1.8	7.3/1.4
Half-*Top*	93.3/ 38.8	95.9/ 35.6	96.1/ 15.6	96.1/ 12.5
Half-*Bottom*	99.2/ 47.4	99.5/ 40.1	99.5/17.2	99.5/13.8
Half-*Left*	98.5/ 41.2	99.1/ 36.3	99.1/15.8	99.1/ 12.7
Half-*Right*	82.0/ 30.6	88.8/28.7	89.2/12.5	89.2/ 10.1
One-third-*Top*	25.3/18.7	39.1/19.1	40.6/8.5	40.6/6.8
One-third-*Bottom*	74.6/23.0	83.2/12.0	83.7/9.1	83.7/7.3
One-third-*Left*	88.1/24.1	92.9/22.8	93.2/10.1	93.2/8.1
One-third-*Right*	12.8/11.9	22.44/12.7	23.6/5.7	23.6/4.6

removing a part of the patch hurts the performance of the adversarial patch as an attacker.

4. Improved Optimization of Adversarial Patch

In this section, we propose two improved optimization methods, *Patch-Score-Focal* and *Patch-Combination*, to prevent the adversarial patch from being face-like. We demonstrate the effectiveness of the proposed approaches by visualized and numerical results.

4.1. Evaluation metric

Attacking criteria. We set a confidence score threshold δ to keep high precision (> 0.99) in order to reduce the possibility of raising false positives. Note that the adversarial patch by *Patch-IoU* can be detected as faces (see Section 3.2). To successfully attack a face detection framework under the policy that none of the bounding boxes in the images should be detected as faces, we define our criteria as follows:

- *Criterion 1*: Reducing the confidence scores of true faces to be lower than δ;
- *Criterion 2*: Preventing the confidence score of adversarial patch from being higher than δ.

Shortcomings of Average Precision (AP) as an evaluation metric. AP[4] serves as one of the most common metrics to evaluate the performance of various face detection algorithms. When computing the AP on a face detection dataset,

---

[4]In general, a face proposal is treated as a True Positive if the IoU between the proposal and the ground-truth face is greater than 0.5 in face detection evaluation.

| 0.999 | 0.998 | 0.998 | 0.142 | 0.122 |

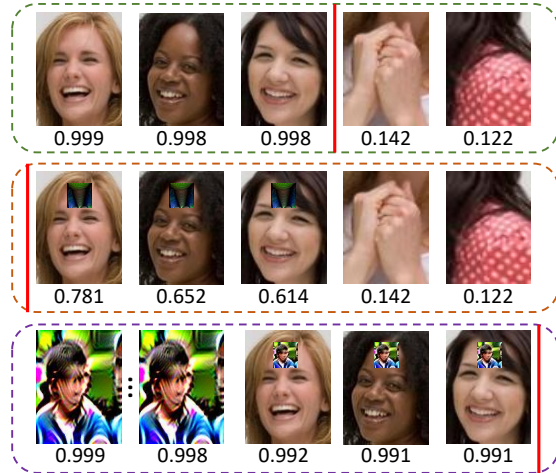| 0.781 | 0.652 | 0.614 | 0.142 | 0.122 |

| 0.999 | 0.998 | 0.992 | 0.991 | 0.991 |

Figure 7. The first row contains top-5 face proposals for the baseline model and the red line represents the threshold boundary. The second row represents a successful attack. However, the APs in the two rows are the same because the relative rank is the same. The third row illustrates a case where $\mathrm{Score}(\mathrm{AdversarialPatch}) > \mathrm{Score}(\mathrm{Face}) > \mathrm{Threshold}$. The AP becomes smaller because of the false positives, though the attack fails according to criterion 2. Thus we cannot use AP to evaluate the performances of adversarial patches.
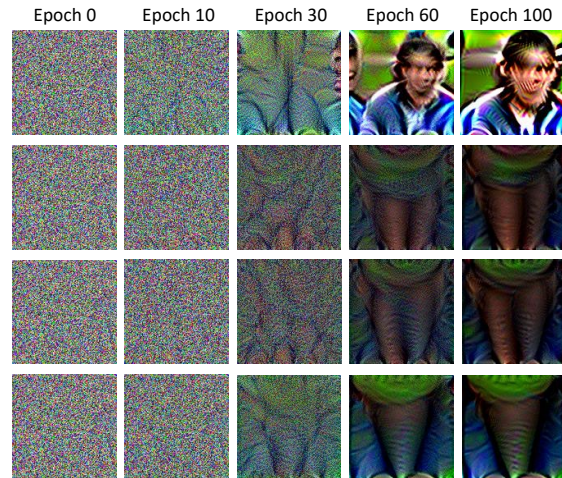

Figure 8. From top to bottom, the four rows represent the adversarial patches of *Patch-IoU*, *Patch-Score*, *Patch-Score-Focal* and *Patch-Combination*, respectively.

the proposed face candidates in the test images are sorted according to their normalized confidence scores. Then we can draw a precision-recall curve under different thresholds and the AP is defined as the area under the curve. Therefore, a straightforward way to evaluate the performance of an attacking algorithm on face detection might be to evaluate its AP (the lower, the better). However, there are two ways in which such an evaluation falls short:

- Reducing the confidence scores of true faces does not

change the relative rankings among positive (faces) and negative (backgrounds) proposals. As a result, AP remains unchanged even when the attack is successful.

- The fake faces appearing on the adversarial patch are treated as false positives. Thus the AP becomes small due to large amounts of false positives. However, this should be considered an unsuccessful attack when the goal is to prevent false positives while suppressing true predictions.

We illustrate the above arguments in Figure 7. We see that a successful attacking algorithm should reduce the recall of the test images while keeping the precision above the given threshold δ. The observation motivates us to use the recall conditioning on the high precision and the F_β score to evaluate the algorithms. F_β score is defined as:

$$F_\beta = \frac{1+\beta^2}{\beta^2 \text{Precision}^{-1} + \text{Recall}^{-1}},$$

where β is a hyper-parameter that trades precision off against recall; setting $\beta < 1$ sets more weights to recall and vice versa. We use Average F_β (AF_β), the area under the Threshold-F_β curve, to evaluate the attacking algorithms. A lower AF_β implies a better attacking algorithm.

4.2. Improved optimization

Given a threshold δ, we aim at decreasing the scores of true faces to be lower than δ under the constraints that the score of adversarial patch cannot be higher than δ. As described in Section 3.2, *Patch-IoU* method may violate *Criterion 2*. To resolve this issue, we first introduce a score-based optimization method named *Patch-Score*. Specifically, we set the adversarial sample set $\{AS_i = (A_i, B_i, S_i, P)\}$ as those samples with $S_i > \delta - m$, where m is a hyper-parameter on the relaxation of the constraint. This procedure for adversarial sample set selection forces the scores of both adversarial patch and true faces to be lower than predefined threshold δ. We set $\delta - m = 0.5$ as default.

Although *Patch-Score* satisfies *Criterion 1* and *Criterion 2* simultaneously, we show that some high-score negative samples may also be selected as adversarial samples AS_i, which may degrade the performance as an attacker. In response to this issue, we propose two solutions, namely, *Patch-Score-Focal* and *Patch-Combination*.

***Patch-Score-Focal* optimization.** Focal loss [20] aims at solving the extreme imbalance issue between foreground and background proposals in the object detection. The core idea is to assign small weights to the vast majority of easily-classified negatives and prevent them from dominating the classification loss. Our method is inspired from the Focal loss and adapts to the adversarial patch training. Formally, we replace the loss in *Patch-Score* by

$$\mathcal{L}_{Adv}(P) = -\frac{1}{N}\sum_{i=1}^{N} S_i^\gamma \log(S_i), \qquad (2)$$

where γ is a hyper-parameter and S_i^γ represents the modulating factor which sets different weights for different samples. In contrast to the Focal loss which assigns smaller weights to the easily-classified samples, our goal is to filter out negative proposals whose score are higher than $\delta - m$ and set smaller weights to these negative samples. We name this optimization method as *Patch-Score-Focal* (see the second row in Figure 4). We set $\delta - m = 0.5$ and $\gamma = 2$ as suggested in [20].

***Patch-Combination* optimization.** On one hand, *Patch-IoU* aims to select adversarial samples according to the higher IoUs of the ground-truth faces, without any score-related constraints in the adversarial patch optimization. On the other hand, *Patch-Score* is to select those samples with confidence scores higher than $\delta - m$, and thus the selected samples may include many negative proposals in the absence of information from the ground-truth faces. We combine the advantages of both methods and propose a new optimization method named *Patch-Combination*. Formally, we restrict each adversarial sample $AS_i = (A_i, B_i, S_i, P)$ to satisfy the following conditions: 1) $S_i > \delta - m$; 2) $\text{IoU}(A_i, B_i) > \lambda_1$ or $\text{IoU}(A_i, P) > \lambda_2$. The third row of Figure 4 illustrates the methodology. We set $\delta - m = 0.5$, $\lambda_1 = 0.3$ and $\lambda_2 = 0.3$ as default.

4.3. Experimental results

We use WIDER FACE training set and SLN baseline model for adversarial patch training; except for the adversarial sample set selection procedure, the same optimization setups for *Patch-IoU* training are applied to the *Patch-Score*, *Patch-Score-Focal* and *Patch-Combination* methods. We use WIDER FACE validation set to evaluate the algorithms (see the second to the fourth rows of Figure 8 for visualized results). In contrast to *Patch-IoU* method, no faces can be detected by our improved optimization algorithms since they all satisfy *Criterion 1* and *Criterion 2*.

Besides the visualized results, we also show numerical results. Figure 9 shows four Threshold-F_β[5] curves and AF_β (lower AF_β means a better attack) under different $\beta = [0.001, 0.01, 0.1, 0.5]$. Table 4 also shows the comparisons of precision and recall with $\delta = 0.99$. With better adversarial sample set design, *Patch-Combination* and *Patch-Score-Focal* achieve better performance than *Patch-Score*. This is because *Patch-Combination* benefits from the confidence scores and location information of both the ground-truth and the patch, and correctly filter out the negative samples in the optimization.

---

[5]Define s_p and s_n as positive and negative logits. We compute $s = s_p - s_n$ as the confidence score when plotting Threshold-F_β curve for better visualization.
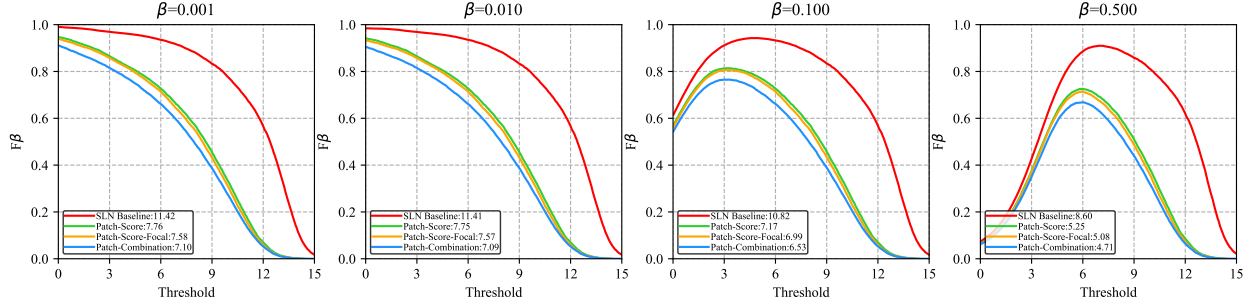
Figure 9. Threshold-F_β curve and AF_β under $\beta = [0.001, 0.01, 0.1, 0.5]$. *Patch-Combination* benefits from the confidence scores and location information of ground-truth and the patch and outperforms other methods.

Table 4. Precision and recall comparisons of Baseline-SLN, *Patch-Score*, *Patch-Score-Focal* and *Patch-Combination* with $\delta = 0.99$.

Precision/ Recall	Easy	Medium	Hard	All
Baseline-SLN	99.0 / 73.4	99.4 / 62.4	99.4 / 27.9	99.4 / 22.5
Patch-Score	98.5 / 25.5	98.9 / 19.7	99.0 / 8.3	99.0 / 6.7
Patch-Score-Focal	98.4 / 23.1	98.9 / 17.9	98.9 / 7.6	98.9 / 6.1
Patch-Combination	98.2 / **20.6**	98.7 / **15.6**	98.7 / **6.6**	98.7 / **5.4**

5. Conclusions

In this paper, we show a property of adversarial patches to state-of-the-art anchor based face detectors: the generated adversarial patches by *Patch-IoU* appear face-like and the detectors falsely recognize the patches as human faces. We show that this phenomenon holds true across different initialization, patch locations and scales, backbones, etc. The attack also transfers between different detection frameworks. In response to the scenario where the attacker wants to suppress true positives but not to introduce false positives, we propose *Patch-Score-Focal* and *Patch-Combination* methods to resolve this issue. Experiments verify the effectiveness of the proposed methods.

Acknowledgement. We thank Gregory Shakhnarovich for helping to improve the writing of this paper and valuable suggestions on the experimental designs.

References

[1] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok. Synthesizing robust adversarial examples. *arXiv preprint arXiv:1707.07397*, 2017.

[2] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 387–402, 2013.

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

[4] S.-T. Chen, C. Cornelius, J. Martin, and D. H. P. Chau. Shapeshifter: Robust physical adversarial attack on faster r-cnn object detector. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 52–68, 2018.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[6] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, F. Tramer, A. Prakash, T. Kohno, and D. Song. Physical adversarial examples for object detectors. *arXiv preprint arXiv:1807.07769*, 2018.

[7] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1625–1634, 2018.

[8] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, D. Song, T. Kohno, A. Rahmati, A. Prakash, and F. Tramer. Note on attacking object detectors with adversarial stickers. *arXiv preprint arXiv:1712.08062*, 2017.

[9] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

[10] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[12] V. Jain and E. Learned-Miller. Fddb: A benchmark for face detection in unconstrained settings. 2010.

[13] R. Jia and P. Liang. Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328*, 2017.

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

[15] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial examples in the physical world. In *International Conference on Learning Representations (ICLR) Workshops*, 2017.

[16] M. Lee and Z. Kolter. On physical adversarial patches for object detection. *arXiv preprint arXiv:1906.11897*, 2019.

[17] J. Li, Y. Wang, C. Wang, Y. Tai, J. Qian, J. Yang, C. Wang, J. Li, and F. Huang. Dsfd: dual shot face detector. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5060–5069, 2019.

[18] Y. Li, X. Yang, B. Wu, and S. Lyu. Hiding faces in plain sight: Disrupting ai face synthesis with adversarial perturbations. *arXiv preprint arXiv:1906.09288*, 2019.

[19] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

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

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

[22] X. Liu, H. Yang, Z. Liu, L. Song, H. Li, and Y. Chen. Dpatch: An adversarial patch attack on object detectors. *arXiv preprint arXiv:1806.02299*, 2018.

[23] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

[24] X. Ming, F. Wei, T. Zhang, D. Chen, and F. Wen. Group sampling for scale invariant face detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3446–3456, 2019.

[25] A. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 427–436, 2015.

[26] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *IEEE Symposium on Security and Privacy*, 2016.

[27] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[28] 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*, pages 1528–1540, 2016.

[29] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter. A general framework for adversarial examples with objectives. *ACM Transactions on Privacy and Security (TOPS)*, 22(3):16, 2019.

[30] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

[31] S. Thys, W. Van Ranst, and T. Goedemé. Fooling automated surveillance cameras: adversarial patches to attack person detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[32] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.

[33] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille. Adversarial examples for semantic segmentation and object detection. In *IEEE International Conference on Computer Vision*, pages 1369–1378, 2017.

[34] T. Yamada, S. Gohshi, and I. Echizen. Privacy visor: Method for preventing face image detection by using differences in human and device sensitivity. In *IFIP International Conference on Communications and Multimedia Security*, pages 152–161, 2013.

[35] S. Yang, P. Luo, C. C. Loy, and X. Tang. Wider face: A face detection benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[36] H. Zhang, J. Shao, and R. Salakhutdinov. Deep neural networks with multi-branch architectures are intrinsically less non-convex. In *International Conference on Artificial Intelligence and Statistics*, pages 1099–1109, 2019.

[37] H. Zhang and J. Wang. Towards adversarially robust object detection. In *IEEE International Conference on Computer Vision*, pages 421–430, 2019.

[38] H. Zhang, S. Xu, J. Jiao, P. Xie, R. Salakhutdinov, and E. P. Xing. Stackelberg gan: Towards provable minimax equilibrium via multi-generator architectures. *arXiv preprint arXiv:1811.08010*, 2018.

[39] H. Zhang, Y. Yu, J. Jiao, E. P. Xing, L. E. Ghaoui, and M. I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning (ICML)*, 2019.

[40] S. Zhang, L. Wen, H. Shi, Z. Lei, S. Lyu, and S. Z. Li. Single-shot scale-aware network for real-time face detection. *International Journal of Computer Vision*, 127(6-7):537–559, 2019.

[41] C. Zhu, R. Tao, K. Luu, and M. Savvides. Seeing small faces from robust anchor's perspective. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5127–5136, 2018.

[42] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *European conference on computer vision*, pages 391–405. Springer, 2014.