Adversarial Pixel Masking: Supplementary Materials

Ping-Han Chiang
bchiang@datalab.cs.nthu.edu.tw
National Tsing Hua University
Hsinchu, Taiwan R.O.C.

Chi-Shen Chan
csch@datalab.cs.nthu.edu.tw
National Tsing Hua University
Hsinchu, Taiwan R.O.C.

Shan-Hung Wu
shwu@cs.nthu.edu.tw
National Tsing Hua University
Hsinchu, Taiwan R.O.C.

ACM Reference Format:

In this document, we provide more details about the settings of our experiments. We also conduct more experiments to verify the effectiveness of APM.

1 ATTACK MODEL

APM can be used to defend against most existing physical attacks provided that the attacks have been considered by the threat model $T$ in Algorithm 1 in the main paper. However, generating patch pixels by replaying all known attacks could slow down adversarial training significantly. Since most existing physical attacks aim to manipulate the objectiveness scores and/or classification scores of candidate objects, we can define an unified objective for generating patch content. For ease of presentation, we consider YOLO [1, 8, 9] as the pre-trained object detector here. The objective can be easily adapted to other types of detectors. A YOLO detector divides the input space into grids $G$, and in every grid $g \in G$ there is a set $\mathcal{A}$ of pre-defined anchors. Given a masked input image $x \otimes m$, the detector (prepdensed by MaskNet) outputs three elements for every anchor $a \in \mathcal{A}$: the coordinates of a bounding box (relative to $a$), the corresponding objectiveness scores $o(x; \xi, \theta)_g, a \in \mathbb{R}$, and the corresponding classification scores $c(x; \xi, \theta)_g, a \in \mathbb{R}_{|C|}$ for all candidate classes $C$. The unified objective is then defined as

$$\arg \min_{p \in T} \sum_{g \in G} \sum_{a \in \mathcal{A}} o(x' \xi, \theta)_g, a \cdot c(x' \xi, \theta)_g, a + \lambda \Omega(p), \quad (1)$$

where $x' = x + p$ is a perturbed image and $\Omega(p)$ is a regularization term that encourages, for example, pixel smoothness. Eq. (1) is a realization of the inner max problem of Eq. (2) in the main paper. By considering both the objectiveness and classification scores, the generated adversarial examples can guide the MaskNet to defend against the ignorance attacks [4, 14, 16, 19, 20] which aim to make some important objects disappear, false-positive attacks [2, 12] which aim to create non-existing objects, and classification attacks [2, 12] which aim to mislead object labels.

2 DETAILED SETTINGS OF EXPERIMENTS

Local Gradients Smoothing (LGS). The performance of LGS is largely influenced by two hyper-parameters: threshold and grid size. As mentioned in Section 4 of the main paper, we followed the original paper [6] to set the threshold. On the other hand, we use more fine-grained grids by setting the grid size to $40 \times 40$ instead of the original $15 \times 15$. As Table 1 shows, this leads to better robustness.

Role of Spatial Concept (ROC). During training, the ROC [11] adds an additional regularization term that encourages a model to focus only on the features within the bounding box of each candidate object. This is done by maximizing the saliency maps within the bounding box, where each saliency map is the gradients of the object’s confidence score with regard to a feature map at a deep layer. We use the last layer of the feature extractor of the object detector to calculate the saliency maps.

Table 1: Performance of LGS with different grid sizes on IN-RIA dataset.

<table>
<thead>
<tr>
<th>Grid Size</th>
<th>High</th>
<th>Low</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean</td>
<td>87.2</td>
<td>86.7</td>
</tr>
<tr>
<td>Noise</td>
<td>86.4</td>
<td>86.0</td>
</tr>
<tr>
<td>ATK(10)</td>
<td>76.9</td>
<td>76.7</td>
</tr>
<tr>
<td>ATK(30)</td>
<td>72.8</td>
<td>64.8</td>
</tr>
<tr>
<td>ATK(50)</td>
<td>72.5</td>
<td>58.7</td>
</tr>
<tr>
<td>ATK(100)</td>
<td>69.4</td>
<td>49.7</td>
</tr>
</tbody>
</table>

Figure 1: Example masks given by the MaskNet after the (a) first and (b) second stages of adversarial training.
Table 2: Performance of different defenses with pre-trained RetinaNet on INRIA dataset. The RetinaNet weights are not fixed in APM.

<table>
<thead>
<tr>
<th></th>
<th>RetinaNet</th>
<th>APM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean</td>
<td>71.1</td>
<td>66.4</td>
</tr>
<tr>
<td>Noise</td>
<td>57.9</td>
<td>80.8</td>
</tr>
<tr>
<td>ATK(10)</td>
<td>6.5</td>
<td>63.0</td>
</tr>
<tr>
<td>ATK(30)</td>
<td>1.5</td>
<td>41.7</td>
</tr>
<tr>
<td>ATK(30)</td>
<td>0.7</td>
<td>32.1</td>
</tr>
<tr>
<td>ATK(100)</td>
<td>0.7</td>
<td>19.7</td>
</tr>
</tbody>
</table>

MaskNet architecture. We experimented with several architectures for the MaskNet in APM, and found that the U-Net architect [10] works the best. The U-Net consists of fully-convolutional layers, each is structured by a contracting path with an expansive path. So, it maintains the spatial information in feature space and increase the resolution of output. We also found that APM with U-Net converges faster than with other CNNs during adversarial training.

Transfer learning. As discussed in the main paper, we let the weights of the pre-trained object detector fine-tunable for a transfer learning task. To implement this, we employ a two-stage training process. In the first stage, we fix the weights of the pre-trained object detector and only train the MaskNet. Once the weights of MaskNet converge, we then fine-tune the weights of MaskNet and object detector jointly. These two-stage training stages both follow Algorithm 1 described in the main paper. Empirically, it prevents the random initial weights of MaskNet to ruin the object detector and results in more clear masks, as shown in Figure 1.

3 ADVERSARIAL ROBUSTNESS OF RETINANET

Here, we show that APM can also improve the robustness of a pre-trained object detector based on RetinaNet [5]. We consider the ignorance attack following the main paper. Here, we show that APM can also improve the robustness of a pre-trained object detector based on RetinaNet [5]. The RetinaNet weights are not fixed in APM, which is directly downloadable from the TensorFlow repository, only give an mAP of 71.1 for clean images. This is lower than the 87.4 given by the pre-trained YOLOv3 used in the main paper. Consider Eq. (2) in the main paper, the object detector needs to be able to correctly identify objects when the MaskNet ($\xi$) successfully masks an adversarial patch. Without a properly pre-trained object detector, the MaskNet cannot learn to “fix” the vulnerability of the detector at pixel space because the loss $L$ is largely resulted from $\theta$ rather than $p$. Empirically, we found that the MaskNet tends to output all-pass masks in failed cases, as shown in Figure 2.

4 APM IN THE REAL WORLD

Here, we show more results of the experiment described in Section 4.5 of the main paper. To see whether APM can defend physical attacks in the real world, we created an universal adversarial patch [14, 17, 18] by regularizing the perturbations $p$ (via the $O(p)$ term in Eq. (1)) such that $p$ is “universal” in the sense that it can be applied to different input images, and 2) takes into account some common post-print distortions such as white noise, rotation, brightness shift, etc. Subsequently, we print the universal patch out and see if it can mute objects when held by real people. Figures 5 in the main paper and 3 here show that APM can successfully defend against the universal patch in the real world. Interestingly, Figure 3 also shows that APM can defend against the universal patch even in the indoor scenes, which is very different from those in the INRIA training dataset. We also find that, APM can successfully defend against the attack by only masking some critical portions of the patch. An universal real-world patch seems to work only when all its pixels (or critical portions) are visible to the object detector. As our future work, we will conduct larger-scale experiments to further verify the effectiveness of APM in different real-world applications.

5 UNSEEN ATTACKS

To study the generalizability of APM, we at test time modify the threat model $T$ such that it generates different adversarial patches than the ones used during adversarial training. Recall from Section 4 of the main paper that, on the INRIA dataset, APM was trained by the adversarial patches of 1) square shape, 2) size equal to 0.8 of the width of the corresponding objects, and 3) horizontal placement at the middle of the corresponding objects. Table 3(a) and Figure 4 shows the performance of APM when the size of adversarial patches varies. We can see that APM performs well even if the size of adversarial patches varies at test time. This is because the “human” objects in the INRIA dataset are of different sizes, so APM was trained to generalize. We further modify the attack by changing the horizontal placement of adversarial patches to the bottom of the corresponding objects. Table 3(b) and Figure 5 shows the results. We also test APM using the adversarial patches of different aspect ratios, and the results are shown in Table 3(c) and Figure 6. APM can generalize to rectangular adversarial patches, despite they have never be seen by APM. We leave the study of more test-time variety as our future work.

1 See https://github.com/tensorflow/models/tree/master/research/object_detection.
Figure 2: When paired up with a pre-trained object detector having a high error rate, the MaskNet tends to output all-pass masks to help the detector see as much information as possible to make correct predictions.

Table 3: Generalizability of APM on INRIA dataset under the attacks with adversarial patches having (a) different sizes, (b) different locations, and (c) different aspect ratios ($M = 50$).

<table>
<thead>
<tr>
<th>Size (Ratio to Object Width)</th>
<th>Clean</th>
<th>Size (at Bottom)</th>
<th>Aspect Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9</td>
<td>0.7</td>
<td>0.6</td>
<td>0.5</td>
</tr>
<tr>
<td>82.3</td>
<td>85.4</td>
<td>84.9</td>
<td>84.7</td>
</tr>
</tbody>
</table>

(a) (b) (c)

Figure 3: APM against an universal physical attack in the real world.

Table 4: Robustness of APM under a black-box attack on INRIA dataset, where adversarial patches are generated using RetinaNet and then applied to YOLO.

<table>
<thead>
<tr>
<th>ATK(10)</th>
<th>ATK(10)</th>
<th>ATK(30)</th>
<th>ATK(50)</th>
</tr>
</thead>
<tbody>
<tr>
<td>YOLO</td>
<td>81.6</td>
<td>82.8</td>
<td>83.0</td>
</tr>
<tr>
<td>APM</td>
<td>91.3</td>
<td>91.0</td>
<td>91.3</td>
</tr>
</tbody>
</table>

However, the black-box physical attack does not seem to transfer well across object detectors as the vanilla YOLO already has high robustness against the attack. This contovers the seemingly-universal transferability of digital attacks in classification tasks [3, 7, 13, 15] and motivates further investigation, which we will leave as our future work.

REFERENCES


6 BLACK-BOX ATTACKS

We also consider black-box attacks, where the weights $\delta$ and $\xi$ in Eq. 2 in the main paper are not accessible to an adversary. The black-box attacks have shown to be possible in digital domains and/or for classification tasks [3, 7, 13, 15]. Nevertheless, to the best of our knowledge, there is no existing study that reports the existence of black-box attacks for object detection tasks in physical domains. To implement an black-box attack, we generate a physical attack using RetinaNet and then apply it to YOLO. Table 4 shows the results. As we can see, APM consistently improves the robustness.
Figure 4: Generalizability of APM in defending unseen adversarial patches of different sizes.

Figure 5: Generalizability of APM in defending unseen adversarial patches at different locations.

Figure 6: Generalizability of APM in defending unseen adversarial patches of aspect ratios.