Feasibility of Inconspicuous GAN-generated Adversarial Patches against Object Detection

Svetlana Pavlitskaya¹, Bianca-Marina Codău², J. Marius Zöllner¹,²

¹FZI Research Center for Information Technology
²Karlsruhe Institute of Technology (KIT)
Motivation: Adversarial Patch Attacks

Train an adversarial patch to suppress all detections
Motivation: Conventional vs. Inconspicuous Patches

Figure 2: Examples of state-of-the-art conspicuous adversarial patches against object detection: (a-b) applied in a digital setting, (d-e) printed on a t-shirt

Figure 3: Examples of state-of-the-art inconspicuous adversarial patches against object detection

SoTA: Combined Patch-GAN Training

- Examples: PhysGAN [6], PSGAN [9], IAP [20]
SoTA: Using a Pretrained GAN

- Examples: *Naturalistic Patch Attack* [5], *TnT attack* [7]
Research Gap & Research Questions

- So far only naturalistic patches by Hu et al. [5] evaluated on object detection
- However: suppressing selected objects by direct overlapping with a patch

Research questions:
- Which GAN-based approaches to generate inconspicuous patches are applicable to the object detection use case?
- What is the trade-off between performance and naturalistic patch appearance?
Experiments

**Model under attack** – YOLOv3

**Attack** – object vanishing attack with PGD

**Data**
- to attack: COCO dataset
- to train a GAN: clean Flower Recognition dataset (1,385 images)

**GAN Architectures**
- DCGAN
- BigGAN (only pretrained)
Baseline (Conventional Patches)

- Evaluated different patch sizes —> 100x100 px for further experiments
- Completely suppress all detections

(a) Patch size 100x100 pixels, 7K epochs, lr=0.01
(b) Patch size 80x80 pixels, 5K epochs, lr=0.5
Approach 1: Combined PGD-DCGAN Training

- 1 discriminator + 1 generator update (as in SOTA) —> highly distorted images
- 1 generator update, with YOLO gradients —> better-performing patches, but training data distribution is not learned

Additional generator update with YOLO gradients

(a) Cropping and horizontal flipping, 2K epochs
(b) Cropping and horizontal flipping, 4K epochs
(c) 2.5K epochs
(d) 5K epochs

Figure 8: Attacks with the combined PGD-GAN training using two generator updates per epoch
Approach 2: Using a Pretrained Generator (DCGAN)

- Patches get dark during training
- Two countermeasures:
  - Latent shift interpolation —> noisy patches
  - Patch transformations (random contrast, brightness transformations, adding random noise) [3]
- No control over the generated flower class

Figure 10: Attacks with a pretrained DCGAN
Differenteffect of countermeasures:

- Latent shift interpolation $\rightarrow$ only background gets dark
- Patch transformations $\rightarrow$ realistic background

Overall higher visual fidelity

**Approach 2: Using a Pretrained Generator (BigGAN)**

![Image showing the effect of different countermeasures](image.png)

**Figure 11**: Attacks with a pretrained BigGAN
Universal Attacks

<table>
<thead>
<tr>
<th></th>
<th>No attack</th>
<th>Black</th>
<th>Conventional PGD</th>
<th>[5] with class-daisy</th>
<th>Pretrained BigGAN + latent shift</th>
<th>Pretrained BigGAN + transformations</th>
<th>Pretrained DCGAN + transformations</th>
<th>Combined PGD-GAN training</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>mAP</strong></td>
<td>43.8</td>
<td>4.2</td>
<td>3.4</td>
<td>3.8</td>
<td>4.4</td>
<td>4.1</td>
<td>3.8</td>
<td>3.9</td>
</tr>
<tr>
<td><strong>AP_{person}</strong></td>
<td>80.3</td>
<td>55.2</td>
<td>28.4</td>
<td>40.4</td>
<td>41.8</td>
<td>32.0</td>
<td>41.2</td>
<td>46.0</td>
</tr>
<tr>
<td><strong>AP_{bicycle}</strong></td>
<td>2.2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.2</td>
<td>0.2</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td><strong>AP_{car}</strong></td>
<td>7.9</td>
<td>3.1</td>
<td>0.0</td>
<td>1.2</td>
<td>0.4</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
</tr>
</tbody>
</table>

Table 1
Mean average precision (mAP) and average precision (AP) for certain classes in % for universal patches, generated for a subset of COCO.
Examples with Universal Patches

- Pretrained DCGAN + transformations
- Pretrained BigGAN + transformations
- Pretrained BigGAN + transformations
Conclusion

- Evaluation of the existing GAN-based method for inconspicuous patch generation on the object detection use case

- Goal – suppress objects near the patch, possibly all objects in the input image
  - Attack strength of the conventional patches could not be reached
  - Using a pretrained generator leads to adversarial patches of higher visual fidelity, better results with BigGAN compared to DCGAN