Robustness Metrics for Real-World Adversarial Examples

Brett Jefferson
brett.jefferson@pnnl.gov

Carlos Ortiz Marrero
carlos.ortizmarrero@pnnl.gov

Pacific Northwest National Laboratory

Abstract

We explore metrics to evaluate the robustness of real-world adversarial attacks, in particular adversarial patches, to changes in environmental conditions. We demonstrate how these metrics can be used to establish model baseline performance and uncover model biases to then compare against real-world adversarial attacks. We establish a custom score for an adversarial condition that is adjusted for different environmental conditions and explore how the score changes with respect to specific environmental factors. Lastly, we propose two use cases for confidence distributions in each environmental condition.

1. Introduction

Now that Deep Learning is an established success [13], there is a rapidly expanding body of work assessing its limitations [18, 10, 2]. In particular, there has been a large number of papers published in recent years, interested in finding new ways to hack deep learning systems with a focus on manipulating convolutional neural networks into false and missed classifications [16, 15, 3]. Much of the early work with so-called adversarial attacks were only successful in virtual environments, e.g., the adversarial inputs were produced and evaluated digitally and without consideration of physical limitations. In the past year, researchers have expanded adversarial attacks to include physically created objects that can impact classifiers and detector models in real-world systems [11, 1, 9, 20]. The range and ability of physical attacks are improving at an impressive rate, accounting for a variety of real-world considerations including static scenes, robust angle and distance changes, and video recorded from a moving vehicle. Although researchers have established some guidelines for evaluating the robustness of virtual adversarial attacks [4], the analysis recommendations do not map subjectively onto physical adversarial attacks where perturbations are difficult to measure and success varies on a frame by frame basis. The consistent computational measure of success for physical attacks is the percent of frames the attack accurately manipulated the classifier or detector [8, 1, 21]. This is separate from adversarial generation metrics that are often included in the optimization loss function to improve physical challenges like imperceptibility and printability. In our work we propose an evaluation experiment and post-generation, effectiveness metrics for testing the robustness of real-world adversarial examples in different environmental conditions. In particular, we tested our metrics on adversarial “patches”, an idea clearly outlined in work done by Thys et al. in [20], but incorporating ideas from Athalye et al. in [1], Chen et al. in [6], and Eykholt et al. in [8].

2. Experiment

One of the primary goals of this paper is to introduce a methodology for evaluating adversarial patches in physical, real-world environments. As such, it is important to account for many environmental factors to systematically study patch performance against a no-patch condition baseline. While many researchers focus on the performance of an adversarial attack in native environmental conditions
(e.g. a patch on the bumper of a vehicle in actual traffic, a patch attached to a stop sign, or a patch attached to a person’s clothing in an office space), we assess performance in a well-controlled environment with little frame-to-frame variability due to moving objects, novel items entering the scene, changing lighting, etc. This tight control is necessary to accurately compare patch performance to a baseline condition where the model is allowed to detect objects without adversarial interference. In other words, we needed an environment that was reproducible and where happenstance occurrences were not a factor. Below we outline our methodology.

2.1. Equipment

We had two laboratory spaces for testing models and patches. One space was used to pilot the experiment for consistently performing target items, explore lighting conditions, test target item distances, determine how to affix patches in a consistent way, and debugging scripts. The second space was a larger laboratory for running the experiment and collecting the data (see Figure 1). Piloting is often not reported with this line of experimentation, but we feel that including some information about this procedure is in line with other disciplines that often detail piloting procedures at length. Some adaptations were required to accommodate the larger lab’s environment nuisances such as a vent that would slightly move our background from time to time. Barring a discussion on such minutiae, the second space was designed to be completely static during experimentation and data collection. No trials were discarded in our analysis. Equipment for the experiment included:

- Custom constructed mounting rail measuring roughly 7 feet long with attached platform (4 feet high) with plate for attaching camera devices
- Jetson AGX Xavier Camera set to a resolution of 1920 × 1080
- Pre-printed adversarial patches
- Tree Floor Lamp
- One 40 watt, 390 lumens halogen bulb and one 40 watt, 450 lumens LED bulb

2.2. Patch Generation

We generate our patches using the training technique outlined by Thys et al. in [20]. The broad idea can be summarized as follows:

1. Curate a set of training images that your object detector can recognize.

2. For each image we superimposed a patch (300 × 300 pixels, then scaled accordingly to fit the size of the object bounding box) to the image. We use the Expectation over Transformation algorithm to produce our adversarial patch using the following transformations: change of location, rotation angle, scale, brightness, contrast, and noise level [1].

3. We extract a classification score from these altered images, back-propagate the gradients to the input layer, and only update the pixels inside the region of the patch.

We leverage and extend the code-base provided in [19] to generate adversarial patches for all classes contained in the output of our model. In our case, these are all classes in the COCO dataset [14].

We trained three patches for the vase class using images from ImageNet [7] and OpenImages [12]. Our patches were obtained by minimizing two different objective functions: objectness score (O) and the product of class probability score and objectness score (CxO). For our ImageNet patches we triage images from the WordnetID n04522168, corresponding to the ImageNet “vase” class. For our Composite patch, we combined our extracted images from ImageNet with images extracted from OpenImages corresponding to the “vase” class name.

2.3. Procedure

During piloting we found our patches should be 2 × 2 inches with the generation method we opted to use [20]. We manipulated three environmental conditions/dimensions: target item distance (1 inch, 5 inches, 10 inches, 15 inches and 20 inches from Plexiglas surface), patch placement location (center and slight right of center), and lighting (2 bulb types) [5 × 2 × 2 = 20 environmental conditions]. There were five patch conditions including no-patch (baseline), ImageNet (O), ImageNet (CxO), Composite (CxO), and a white patch, which was a simple white 2 inch square cut from 8.5 × 11 inches, 92 bright copy paper. All patches were printed on the aforementioned office paper at 1200 dpi. See Section 2.2 for more details. Each patch type, except for the no-patch condition, was used in all environmental conditions. For the no-patch condition, the target item was always center-placed, but used with all light source and distance combinations.

Our camera was placed 5 inches from a Plexiglas surface affixed to a table. The Plexiglas served as a mounting structure to ensure consistent placement for each patch. There was a single light source used at a time and the source was positioned behind and above the camera, pointed toward the target placement region. The camera was placed on the rail platform and fixed for the duration of the experiment. A
large black board was used as background for the experiment.

Testing proceeded as follows:

1. For a given distance, the target item (a green vase, see Figure 3) was placed in one of two positions depending on the patch placement condition (center positioned or left of center).

2. With the target fixed, the no-patch condition was recorded first. Then each patch was placed in sequence of white patch, ImageNet (O), ImageNet (C xo), and Composite (CxO). Each time a patch was placed, it remained untouched through both lighting conditions. This was to minimize object and patch shifts across the conditions.

3. Once a scene was established, we allowed 30 seconds for the bulb to warm-up.

4. We ran a script that captured 500 frames and used each frame as an independent input to YOLOv2 [17]. We wanted enough frames for a single scene for a robust evaluation of that condition in lieu of natural image variation produced by the camera or by nature.

5. We recorded bounding boxes, confidences, objectness scores for each frame.

3. Results and New Metric

3.1. Patch Effectiveness

![Figure 2: Patches generated for experiment](image)

(a) ImageNet (C xo)  (b) ImageNet (O)

![Composite (CxO)](image)

Each of the patches in Figure 2 was designed to hide a target item (e.g. a green vase) from detection for the YOLOv2 classifier. In addition to these patches, a simple white square patch was also used to compare performance with an occlusion case. A first evaluation of each patch’s ability to hide the target was to simply count the number of frames the classifier was able to detect the target class in each scene (See Table 1). This is a standard measure. Higher values in the table indicate the patch was not effective at disrupting detection of the target.

There are a few conditions that stand out when looking at only frequencies. When the target item was placed very close to the camera (1 inch condition) and no patch was present, the classifier had difficulty detecting it. In all frames with LED lighting, the target was missed, while in the halogen bulb lighting, the target was detected in less than 40% of frames. Another stand-out is the LED, 1-inch, ImageNet (C xo) condition where the number of detections is higher than 500. In this case, the target is detected twice, once as a lower identification of the target and an upper identification of the target (see Figure 3).

Judging from frequency alone, one might conclude that the Composite (C xo) patch and ImageNet (C xo) patch are better than the other two patches. This conclusion would match intuition since one patch had a larger training set and both of these patches were trained using an objective function accounting for both class and objectness score. But there is more to be discovered. For instance, suppose one desired an all-around effective patch for a variety of physical environments. Is the Composite (C xo) patch better than the ImageNet (C xo) patch? A problem with error frequency is that is does not account for how well the model performs without adversarial interference. In the LED, 1-inch case, not having a patch at all is better than adding anything to the scene. We seek to develop a score that not only accounts for a variety of environmental changes, but also accounts for baseline performance in one summary.

![Figure 3: In this condition the vase was detected twice in nearly all frames.](image)
Table 1: Number of vase detections for each condition.

*When no patch was present, targets were center located. Thus, only one run of the model at different distances and bulb types was completed without a patch.

| Location | Bulb | Distance | None | Composite (CxO) | ImageNet (CxO) | ImageNet (O) | White Patch |
|----------|------|----------|------|-----------------|----------------|--------------|-------------|
| center   | halogen | 1 inch | 182  | 7               | 0              | 500          | 500         |
|          |       | 5 inches | 404  | 20              | 0              | 500          | 500         |
|          |       | 10 inches | 500  | 0               | 0              | 30           | 0           |
|          |       | 15 inches | 298  | 0               | 0              | 1            | 0           |
|          |       | 20 inches | 499  | 0               | 0              | 32           | 0           |
| LED      |   | 1 inch | 0    | 419             | 955            | 500          | 500         |
|          |   | 5 inches | 500  | 500             | 391            | 500          | 500         |
|          |   | 10 inches | 500  | 27              | 0              | 500          | 456         |
|          |   | 15 inches | 500  | 0               | 0              | 500          | 0           |
|          |   | 20 inches | 500  | 0               | 0              | 28           | 0           |
| right    | halogen | 1 inch | 182* | 0               | 18             | 499          | 479         |
|          |       | 5 inches | 404* | 0               | 206            | 500          | 313         |
|          |       | 10 inches | 500* | 206             | 0              | 302          | 1           |
|          |       | 15 inches | 298* | 0               | 0              | 26           | 0           |
|          |       | 20 inches | 499* | 0               | 0              | 1            | 0           |
| LED      |   | 1 inch | 0*   | 2               | 457            | 500          | 500         |
|          |   | 5 inches | 500* | 247             | 500            | 500          | 500         |
|          |   | 10 inches | 500* | 500             | 500            | 500          | 500         |
|          |   | 15 inches | 500* | 479             | 500            | 500          | 490         |
|          |   | 20 inches | 500* | 0               | 0              | 0            | 120         |

Table 2: Patch scores.

| Patch Condition | Score |
|-----------------|-------|
| No-Patch        | 0     |
| Composite (CxO) | 0.536 |
| ImageNet (CxO)  | 0.424 |
| ImageNet (O)    | 0.135 |
| White Patch     | 0.241 |

Table 3: ANOVA on scores.

|          | df | sum.sq | mean.sq | F     | PR(>F) |
|----------|----|--------|---------|-------|--------|
| Lctn.    | 1  | 0.16   | 0.16    | 1.04  | 3.11e-01|
| Bulb     | 1  | 2.25   | 2.25    | 14.89 | 2.16e-04|
| Dist.    | 4  | 4.11   | 1.03    | 27.21 | 9.31e-15|
| Patch    | 4  | 3.73   | 0.93    | 6.18  | 1.97e-04|
| Resid.   | 89 | 13.43  | 0.15    | NaN   | NaN    |

patch $P$ and a single environmental condition $e$ (e.g. lighting, distance, patch location), we compute the frequency of detection. Let the $f_{P,e}$ be the frequency for a given patch and environmental condition. Then we define the score for a patch conditionally over the set of environments to be

$$S_E(P) = \frac{1}{|E|} \sum_{e \in E} \frac{(f_{\emptyset,e} - f_{P,e})}{500} \quad (1)$$

where $E$ is the set of all environmental conditions. Note that this is the average difference of detection probabilities between the no-patch and patch conditions. For this score function, when the target is detected in all 500 frames and the patch successfully hides the target in all 500 frames, the score will be 1. When the baseline model is unsuccessful at detecting the target, the score is lower. Lastly, the score is averaged across all conditions run. The scores for our patches are given in Table 2. It is interesting to note that by our metric, the white patch outscored the patch trained using only the ImageNet corpus with objective function only utilizing the objectness score from YOLOv2. Further investigation is needed to explore why this occurred.

We also computed scores for each condition independent of the other conditions. We ran an ANOVA on this data with factors Patch Location, Bulb Type, Target Distance, and Patch. Bulb type, target distance, and patch were all significant factors. The analysis in the next section dives into the pattern of results on these dimensions.

### 3.2. Dimension Impacts

Recall that for a 1 inch distance, baseline model performance was particularly poor. However in the ImageNet and
white patch conditions model performance increased significantly, regardless of lighting or center/right patch placement. This may indicate that pre-trained YOLOv2 is not robust to large-scaled objects. Somehow, more generic occlusions provides YOLOv2 with enough context to make an accurate identification.

Figure 4: Image capture of YOLOv2 detection at 5 distances with white patch under LED lighting.

In addition to poor model/patch performance for close distances, the scores dip when the target item is 15 inches from the Plexiglas. The dip could be driven by either low baseline model performance at 15 inches, or by poor patch performance. A quick look at Table 1 reveals that baseline model performance also decreases at 15 inches (number of correct detections without the patch is 298 out of the possible 500 for halogen bulbs). When this occurs, the model score decreases since the score is relative to baseline.

Figure 5: Mean confidence per distance.

We also recorded confidence scores for each condition to study if there was a relationship with patch performance. Confidence scores did not influence patch performance across the distances. Figure 5 displays the average confidence for each distance and each patch condition. Baseline and ImageNet are the only two patch conditions that decrease from 5 inches to 20 inches. The two other simulated patches are consistently high for 5 to 15 inches and then effectively hide the target at 20 inches, while the white patch telescopes in performance. One might expect lower confidence scores leading to fewer detection. However, confidence scores are constructed independently of probability of detection within YOLOv2. The model detects a target when the objectness score is above 0.5 and to prevent multiple detections of the same object, the NMS threshold is set to 0.4. The two patch conditions with consistently high confidences for the vase are also the two patches that performed the best at hiding the vase. At 15 inches (where all simulated patches have a sudden decrease in score), the highest scoring patch, Composite (CxO), has a higher confidence when the target is detected than the baseline model. This provides some experimental evidence that confidence score alone, without context to other class scores, are not a clear predictor of success for patches designed to hide a target.

Figure 6: Score by distance for each patch.

An alternative explanation for model performance beyond 10 inches is that the target itself is mostly occluded from camera view at further distances. More than 97% of all frames at this distance were successfully either misclassified or not identified considering all patches. The sub-condition with the highest detection frequency occurred with the white patch placed slightly right of the target under LED lighting. Further testing is required to confirm occlusion is the main reason for poor performance. A counter condition is the ImageNet patch. When that patch was used there were 60 correct detections across lighting at center location, but only a single correct detection when the target was slightly right of the patch. The white patch, LED, right-position condition led to 120 of the 500 frames having correct detections at 20 inches.

Lighting was also a significant factor for patch scores.
Figure 7 displays the difference in lighting used for this experiment. We predicted that for the LED condition, the patches would be more effective at hiding the target. However, the LED condition resulted in fewer disappearances (more correct detections) than the halogen condition. This is surprising given the LED is more luminescent.

Figure 7: Image capture of YOLOv2 detection at 2 light configurations (LED and halogen with white patch at 10 inches.

Considering only changes in this factor, we find that across the two lighting conditions, the Composite (CxO) and ImageNet (CxO) patches outperform the other two patches. In addition, the white patch has higher scores than the ImageNet (O) patch in both lighting conditions. Performance in the halogen bulb condition is limited to 0.7 by the fact that we are averaging scores across the other dimensions (location and distance) and there are cases in which the baseline model had missed detection occurrences in these conditions.

Figure 8: Score by light for each patch.

The same trend, although on a smaller scale, occurs when marginalizing over location values. Figure 10 depicts mean values at each location. We computed 68% (roughly two standard deviations) confidence intervals by a bootstrapping procedure. Across the two target locations, the confidence intervals have high overlap which is an indicator that performance is likely equal regardless of target location. The Composite (CxO) and ImageNet (CxO) patches are the only two that had confidence intervals that did not overlap with a score of 0, indicating that for both location conditions, these patches had some effect on YOLOv2. However, after running one-way ANOVA’s, even these patches were not significantly different from 0 (p = [0.323, 0.219, 0.969, 0.595] for Composite (CxO), ImageNet (CxO), ImageNet, and white patch, respectively).

Figure 9: Image capture of YOLOv2 detection at 2 location configurations (center and right) with white patch at 10 inches under LED lighting.

Figure 10: Score by location for each patch.

3.3. Effectiveness Score

For our experiment, three of the four patches were virtually ineffective for the right-sided target placement under LED lighting. The remaining patch admitted many correct identifications of the vase under this condition. While we may ponder this case and speculate at the causes, this case highlights the need for a measure of patch effectiveness when detections are not eliminated. We propose a straightforward calculation and approach.

For two conditions under which a single detection model is being assessed (say condition 1 and condition 2), consider a single target class. For each condition, collect all model confidences corresponding to each camera frame in which the model detected the target class in roughly the same bounding box. Thus, we have a distribution of confidences for condition 1 and a distribution for condition 2 for a single target. We compare the two conditions by considering either the Kolmogorov - Smirnov statistic or the Wasserstein distance between the two distributions. This provides a relatively straightforward comparative computation in which researchers can apply several follow-on analyses. We highlight a few of these cases here:
Case 1. (Using Full Distribution of Confidences). As seen in Table 1, the ImageNet (O) patch performs particularly poorly in LED lighting. When baseline performs perfectly (that is, the target is identified in each frame), the global patch score derived above cannot distinguish the conditions. By considering the KS metric with baseline, there are three cases that are most similar to baseline, 1. right, 5 inch, 2. center, 5 inch, and 3. right, 10 inch (ks = 0.498, 0.664, and 0.948 respectively).

Case 2. (Exploring Data Patterns). Using the Wasserstein Distance we consider all pairwise distances between conditions for the ImageNet (O) patch. For the condition where there were no correct detections (20 inches, right-placed patch, and LED lighting) we impute the maximum relative distance. From this distance matrix on conditions, we embed them in a 2-dimensional representation using multidimensional scaling. The circled point is the condition in which ImageNet effectively hid the target. Clusters in this space reflect like-conditions (e.g. same lighting or patch placement). By color coding with respect to the levels of each dimension in Figure 11, we can see that the distance dimension provides a clear correlation with global score results. In fact, the distance dimension strongly dominates the linear shape of the points. We conjecture that the order (20 inches, 15 inches, 1 inch and 10 inches, and 5 inches) is an indicator of least to most ‘optimal’ viewing conditions for the baseline model.

4. Conclusion and Future Work

This paper makes two contributions. First, we propose a metric for adversarial attacks in the physical world. The metric compares attack performance to a baseline. We compute the metric in a controlled environment with reproducible environmental conditions. In addition, the metric reveals bias in the baseline models. We include two potential use cases for the proposed metric. The second contribution is that this is the most controlled systematic study of baseline model performance and its relationship to adversarial attacks that we are aware of. Chen et al. [6] had an indoor systematic assessment of sticker attacks but did not investigate varying controlled lighting and placement, rather distance and angle. While many of the current papers highlight that their patch generation methods work well in the real world, it is of importance to account for the weaknesses in any method to not only prevent other researchers from making the same mistakes, but to advance scientific understanding of deep learning models in general. It is our aim that the approach taken in this paper strives towards a full-report model and has added some understanding to real-world challenges facing adversarial attacks and understanding deep-learning models alike.

One may notice that our investigation, while exposing weaknesses in the YOLOv2 model with respect to trained physical adversarial attacks, does not provide an answer to why the models are vulnerable to certain real-world factors. This is a difficult problem that many recent advances have attempted to solve. A good summary on this work is [5]. In our own investigation, we suspect camera aspects are driving frame to frame variation and model training is interacting with environmental conditions to produce odd model results (such as the baseline model not detecting a vase six inches in front of the camera). In future work, we will extend the score derived here to study whether there are any systematic patterns to model and adversarial attack weaknesses.

References

[1] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. arXiv preprint arXiv:1707.07397, 2017.
[2] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Matusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro,
Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. arXiv preprint arXiv:1806.01261, 2018. 1

[3] Nicholas Carlini. A complete list of all (arxiv) adversarial example papers. https://nicholas.carlini.com/writing/2019/all-adversarial-example-papers.html. 1

[4] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, and Aleksander Madry. On evaluating adversarial robustness. arXiv preprint arXiv:1902.06705, 2019. 1

[5] Supriyo Chakraborty, Richard Tomsett, Ramya Raghavendra, Daniel Harborne, Moustafa Alzantot, Federico Cerutti, Mani Srivastava, Alun Preece, Simon Julier, Raghuveer M Rao, et al. Interpretability of deep learning models: a survey of results. In 2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), pages 1–6. IEEE, 2017. 7

[6] Shang-Tse Chen, Cory Cornelius, Jason Martin, and Duen Horng Polo 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. Springer, 2018. 1, 7

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee, 2009. 2

[8] Kevin Eykholt, Ivan Evtimov, Earlene Fernandes, Bo Li, Amir Rahmati, Florian Tramér, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Physical adversarial examples for object detectors. arXiv preprint arXiv:1807.07769v2, 2018. 1

[9] Kevin Eykholt, Ivan Evtimov, Earlene Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning models. arXiv preprint arXiv:1707.08945, 2017. 1

[10] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572, 2014. 1

[11] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. arXiv preprint arXiv:1607.02533, 2016. 1

[12] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. arXiv preprint arXiv:1811.00982, 2018. 2

[13] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. Nature, 521(7553):436, 2015. 1

[14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In European conference on computer vision, pages 740–755. Springer, 2014. 2

[15] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2574–2582, 2016. 1

[16] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In 2016 IEEE European Symposium on Security and Privacy (EuroS&P), pages 372–387. IEEE, 2016. 1

[17] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. arXiv preprint arXiv:1612.08242, 2016. 3

[18] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199, 2013. 1

[19] Simen Thys, Wiebe Van Ranst, and Toon Goedeme. Adversarial yolo. https://gitlab.com/EAVISE/adversarial-yolo. 2

[20] Simen Thys, Wiebe Van Ranst, and Toon Goedeme. Fooling automated surveillance cameras: Adversarial patches to attack person detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2019. 1, 2

[21] Yue Zhao, Qintao Shen, Ruigang Liang, Kai Chen, and Shengzhi Zhang. Practical adversarial attack against object detector. arXiv preprint arXiv:1812.10217, 2018. 1