# DetectorDetective: Investigating the Effects of Adversarial Examples on Object Detectors

Sivapriya Vellaichamy, Matthew Hull, Zijie J. Wang, Nilaksh Das, ShengYun Peng,
Haekyu Park, Duen Horng (Polo) Chau
Georgia Institute of Technology
Atlanta, Georgia, USA
[svellaichamy3, matthewhull, jayw, nilakshdas, speng65, haekyu, polo]@gatech.edu

## Abstract

*With deep learning based systems performing exceedingly well in many vision-related tasks, a major concern with their widespread deployment especially in safety-critical applications is their susceptibility to adversarial attacks. We propose DetectorDetective, an interactive visual tool that aims to help users better understand the behaviors of a model as adversarial images journey through an object detector. DetectorDetective enables users to easily learn about how the three key modules of the Faster R-CNN object detector — Feature Pyramidal Network, Region Proposal Network, and Region Of Interest Head — respond to a user-selected benign image and its adversarial version. Visualizations about the progressive changes in the intermediate features among such modules help users gain insights into the impact of adversarial attacks, and perform side-by-side comparisons between the benign and adversarial responses. Furthermore, DetectorDetective displays saliency maps for the input images to comparatively highlight image regions that contribute to attack success. DetectorDetective complements adversarial machine learning research on object detection by providing a user-friendly interactive tool for inspecting and understanding model responses. DetectorDetective is available at the following public demo link:* `https://poloclub.github.io/detector-detective`. *A video demo is available at* `https://youtu.be/5C3Klh87CZI`.

## 1. Introduction

Deep neural networks are now used in many computer vision tasks and have reached human-level accuracy in image classification [4, 9], object detection [19, 20], and semantic segmentation [8, 11]. With its phenomenal accuracy and applicability in various domains, they form a class of models that deserve greater understanding. Accuracy and robustness of models play a key role in their widespread deployment. Several works show that deep neural networks are vulnerable to digital perturbation [7, 13, 16] and physical realizable attacks [3, 6, 18]. Szegedy et al. [16] showed that adding visually imperceptible but structured noises are capable of fooling the deep learning system into making wrongful predictions. This can be especially disastrous in scenarios like self-driving cars with failed detection of a pedestrian or wrongful detection of stop sign as any other object. With the existence of such vulnerability, it is important to be able to visually understand and explain the process by which a system arrives at the predictions for the users to have trust in the system.

We propose DetectorDetective (Figure 1), an interactive tool that helps users understand how an original benign image and its adversarially-attacked version is processed as it goes from a collection of pixels through different modules of the object detector and finally to a collection of bounding boxes and classification labels. We generate the adversarial image from a user-chosen input image using *Projected Gradient Descent* [12] to enable this comparative visualization. The user then engages with the tool to interactively explore the key modules within the object detector, and those modules' components, with a side-by-side comparative visualization for the original and adversarial example. DetectorDetective also provides overall saliency maps based on Gradients [5] and GradCAM [15] to help users better understand the important pixels for both the images on the whole. GradCAM++ [2], a generalization of GradCAM is also found to work well for multiple detections. In addition, both techniques have been able to pass the basic sanity checks for saliency maps [1]. DetectorDetective is available at the following public demo link: `https://poloclub.github.io/detector-detective`. A video demo is available at `https://youtu.be/5C3Klh87CZI`. DetectorDetective's primary contributions include:

- **Visual interpretation of adversarial examples on object detector.** DetectorDetective helps users interpret

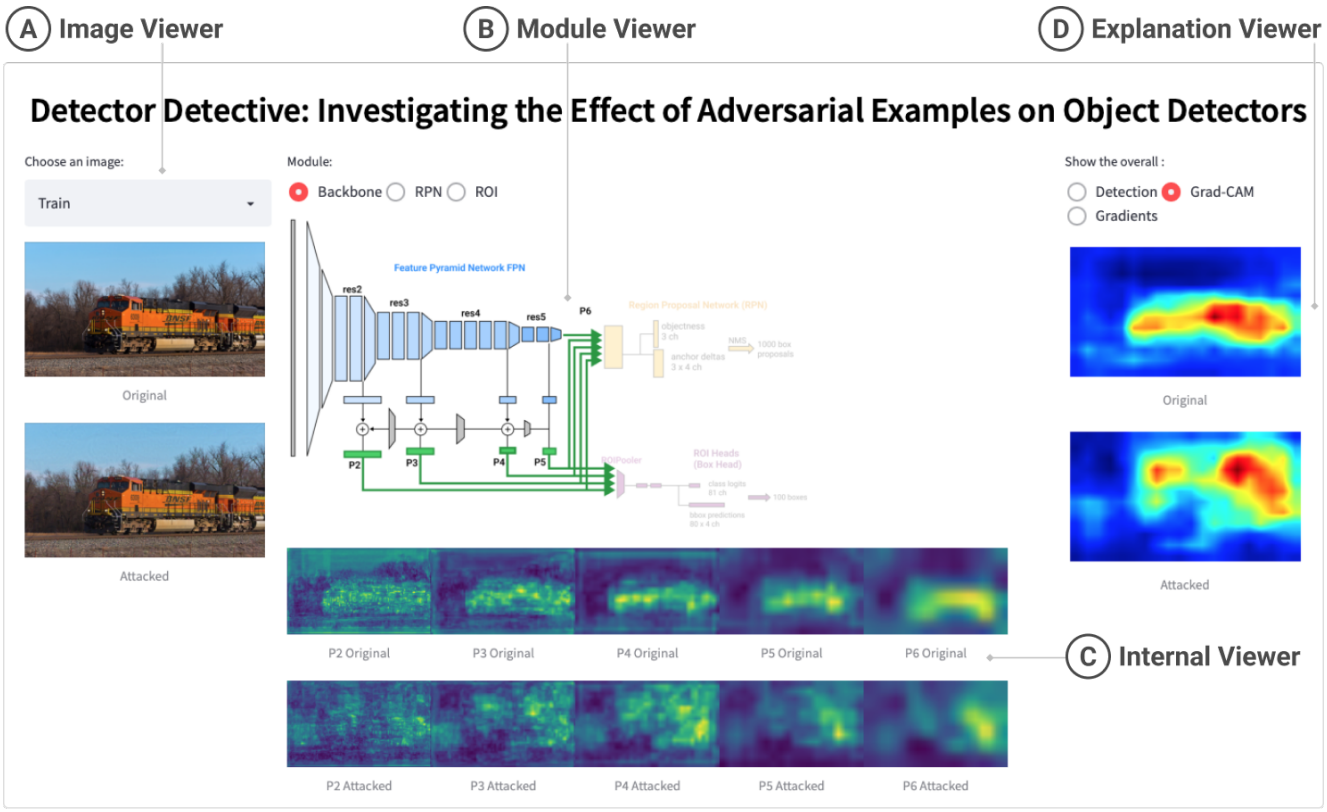



Figure 1. The DetectorDetective interface. (A) Image Viewer allows users to select an input image and presents the selected image with its attacked version. (B) Users can select an internal module to investigate in the object detector in Module Viewer. Module Viewer highlights the selected module in the architecture diagram. (C) Internal Viewer visualizes feature maps extracted by the selected module for benign and adversarial images, enabling side-by-side comparisons of how the module responds differently to the benign and adversarial images. (D) Explanation Viewer provides visual explanations of which part of the benign and adversarial images are used to make a model's outcome, also enabling side-by-side comparisons of model prediction on benign and adversarial attacks.

how an adversarial attack applied on input images fools an object detector into misclassification and mislocation, by (1) visualizing features extracted by the internal modules of the model, (2) explaining the model prediction with saliency maps (i.e., the regions of the images that most contribute to the prediction), and (3) enabling side-by-side visual comparisons of such feature visualization and prediction explanation between the benign and adversarial cases.

- **Open-sourced, web-based implementation.** DetectorDetective runs in modern browsers and is open-sourced[1], helping support reproducible research and broaden the public's access to investigating the effects of adversarial examples on object detectors.

- **Usage scenarios.** We provide three usage scenarios to showcase how DetectorDetective can help people understand the effects of adversarial examples on an object detector, such as why the object detector incorrectly detects multiple objects from the background of an adversarial image.

[1] https://github.com/poloclub/detector-detective

## 2. System Design and Implementation

In this section, we describe DetectorDetective's interface design and implementation (Figure 1). The *Image Viewer* (Figure 1A) presents a user-selected image and its adversarially-attacked image. The *Module Viewer* (Figure 1B) allows users to choose the module in the Object Detector to focus on. The *Internal Viewer* (Figure 1C) displays the visualizations of the key feature representation of the chosen module. Lastly, the *Explanation Viewer* (Figure 1D) explains the overall predictions using saliency maps like GradCAM and Gradients. We have chosen Faster R-CNN [14], a widely used object detector for this demo.

### 2.1. Image Viewer: Input Selection and Attack

Image Viewer (Figure 2) allows users to select an image and presents its adversarially attacked image. The drop down

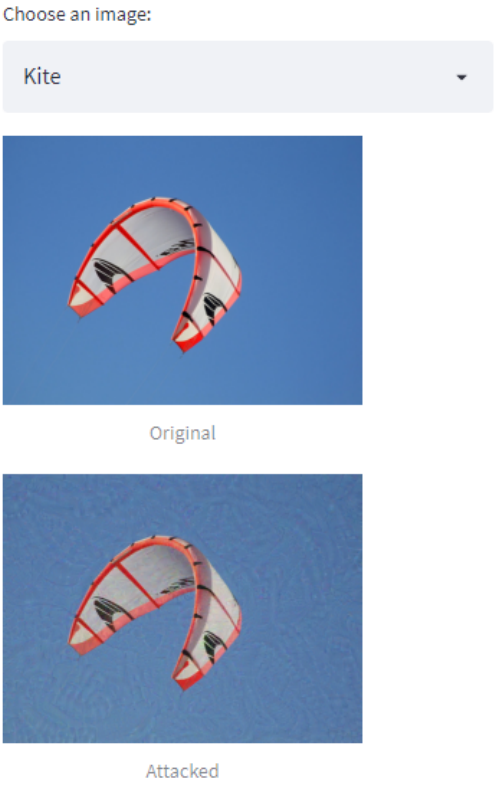



Figure 2. Image Viewer allows people to select an image to investigate, and it shows the selected image and its attacked version.

menu provides images from the COCO validation dataset 2017 [10] in this demo. The chosen image and its adversarial image perturbed by Projected Gradient Descent attack [12] on the Detectron2 model [17] are displayed in this viewer.

## 2.2. Module and Internal Viewer: Inside the Model

The Module Viewer (Figure 1B) allows users to choose an internal module in the object detector to investigate, and Internal Viewer (Figure 1C) visualizes internal features extracted by the selected module. On the top of Module Viewer, users can select one module by clicking a radio button. Module Viewer then helps the users ensure if they have chosen the right module, by visualizing the overall model architecture and highlighting the selected module. There are three modules in Faster R-CNN: *Backbone*, *Region Proposal Network (RPN)*, and *Region of Interest (ROI)* through which the image passes through sequentially (Figure 5). Next, we explain the three modules: Backbone (Section 2.2.1), RPN (Section 2.2.2), and RoI (Section 2.2.3).

### 2.2.1 Feature Pyramid Network

When users select the first radio button in Module Viewer (i.e., 'Backbone'), Internal Viewer visualizes feature maps


Figure 3. Module Viewer (top) and Internal Viewer (bottom), to visualize feature maps extracted by internal components of the object detector

extracted by Feature Pyramid Network (FPN). FPN is the first module of Faster R-CNN to boost its ability to predict images of all size by extracting multi-scale features. The five feature maps extracted at different scales are P2-P6, and deeper components extract larger scale features (i.e., have a higher receptive field). For example, Figure 3 shows that P2-P6 extract feature maps of both benign and adversarial images in different scales at varying depths, where brighter colors mean stronger features are extracted for the particular scale for the corresponding receptive field. After the feature extraction, they are fed into the other two modules Region Proposal Network (Section 2.2.2) and Region of Interest (Section 2.2.3).

### 2.2.2 Region Proposal Network

When users select the second radio button in Module Viewer (Figure 1B), Internal Viewer (Figure 1C) visualizes feature maps extracted by the second module of the object detector: Region Proposal Network (RPN). RPN generates the bounding box to identify the location of detected objects, and it internally uses two kinds of feature maps: *objectness maps* and *anchor deltas maps*. An objectness map visualizes the probability of the presence of an object in the image. An anchor deltas contains information about the shape and orientation of features with respect to anchors.Anchors of

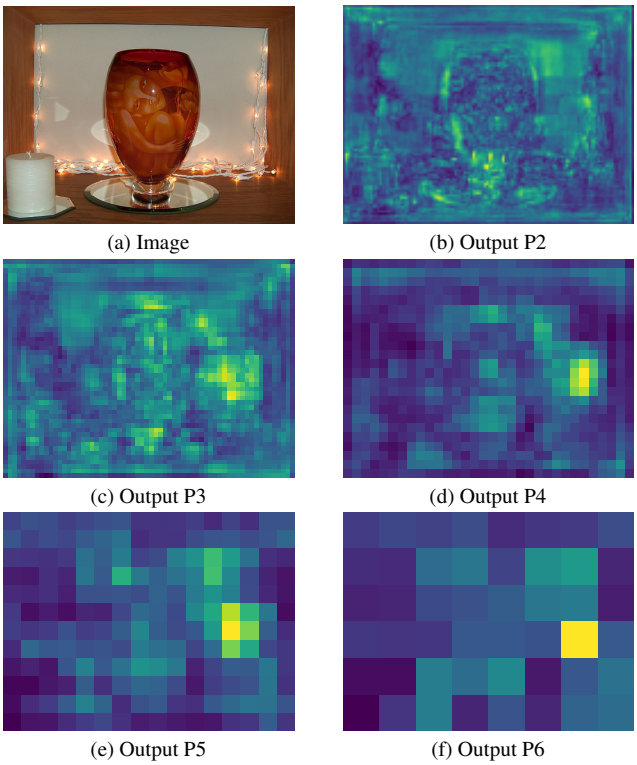
different sizes, aspect ratios are applied to the feature maps to classify the anchors in the foreground. A maximum of 1000 boxes are finally chosen.

#### 2.2.3 Region of Interest

When users select the third radio button in Module Viewer (Figure 1B), Internal Viewer (Figure 1C) visualizes the output of the last module: Region of Interest (RoI). RoI finally predicts bounding boxes and labels of the detected objects. RoI computes the score of each bounding box identified in RPN module, filters bounding boxes based upon a threshold, and predicts the label of the filtered bounding boxes. The final predictions are limited to 100 boxes.

### 2.3. Explanation Viewer: Display Prediction boxes

Explanation Viewer (Figure 1D) visualizes and explains the final prediction of the object detector, by highlighting the most influential pixels for the decision of the model (i.e., saliency map) in both benign and adversarial images, produced by GradCAM [15] and Gradients [5]. For example, Figure 6h shows that the object detector focuses its attention on the background sky of the adversarial image to predict cars in the sky.



(a) Image (b) Output P2 (c) Output P3 (d) Output P4 (e) Output P5 (f) Output P6

Figure 4. Feature maps extracted at varying depths in internal modules in the object detector

## 3. Usage Scenario

We provide three usage scenarios that showcase how DetectorDetective can help users investigate and interpret the effects of adversarial examples on the Faster R-CNN model.

### 3.1. Object Hallucinations

Object detectors may misidentify multiple objects with wrongly classified labels, when the models are optimized for both location and classification losses. For example, in Figure 6b, many objects are incorrectly detected on the background of the adversarial image, while a correct object (kite) is identified on the benign image as seen in Figure 6a.

Using DetectorDetective, we investigate such "object hallucinations", to understand why Faster R-CNN mis-perceives multiple objects that are not present in the adversarial image. We first use Explanation Viewer and observe that the saliency maps look very different between benign and adversarial cases, as seen in Figure 6g and Figure 6h. The saliency map for the benign image (Figure 6g) is faithful to the image, as the model focuses on the region where the kite exists. On the other hand, the saliency map for the adversarial image (Figure 6h) reveals that the portions of sky in the image strongly impact to the model's prediction, leading to multiple incorrect object detections finally.

To interpret the internal process of such hallucinated detections, we closely examine the internal features by using Module Viewer. On choosing the backbone module, we compare the internal features extracted by P2 component given the benign and adversarial images. As seen in Figure 6c, P2 well identifies features only for the kite in the benign image, as the highlighted region of the feature map are overlapped with where the kite locates. However, when the adversarial image is given, the activated parts in P2's feature map shown in Figure 6d seem to be much more scattered than that of the benign case, indicating that P2 component captures void features here and there in the adversarial image, which may cause confusion in subsequent modules. The objectness map for the adversarial image (Figure 6f) further reveals the scattered detections of void features in the adversarial image, compared to the objectness map for the benign image (Figure 6e).

### 3.2. Translated Bounding Box

We examine another incorrect object detection case, where an object is identified correctly but the bounding box coordinates are attacked. For example, as seen in Figure 7b, an adversarial train image is detected on the incorrect location in the image, compared to the correct detection of train in the benign image (Figure 7a). When inspecting Explanation Viewer, DetectorDetective reveals that the most salient region of the adversarial image for detecting the train seems to be out-focused as seen in Figure 7h; the region of

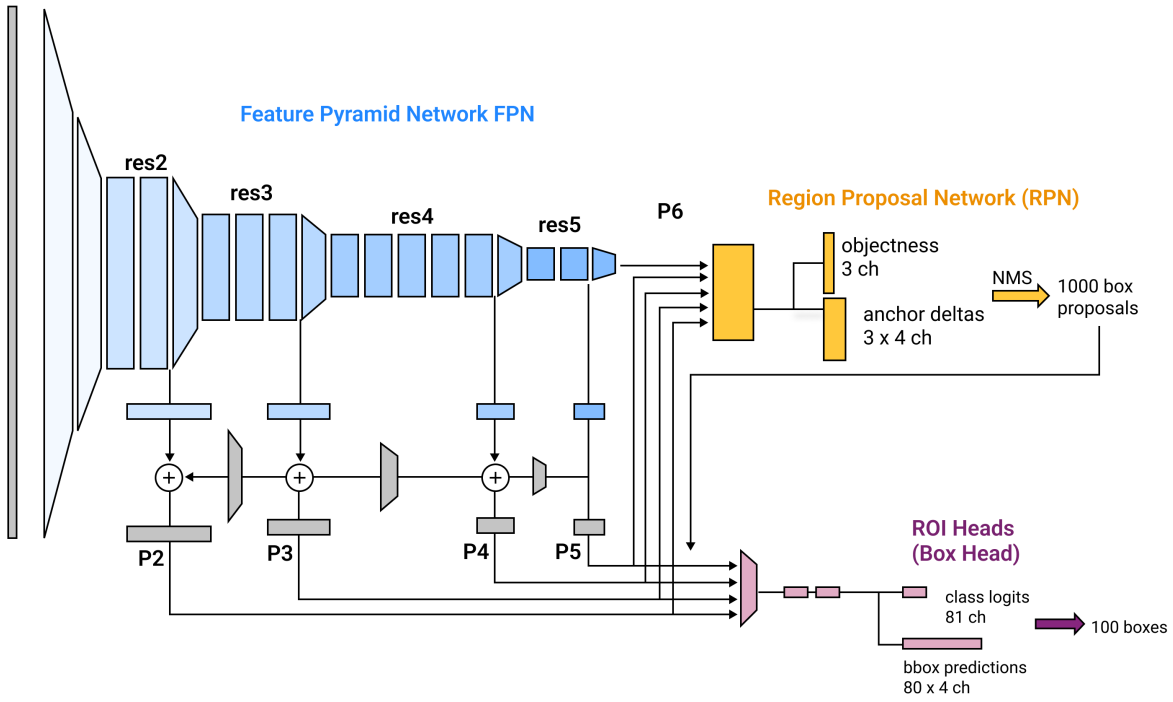



Figure 5. Model Architecture of Faster R-CNN, consisting of three modules: Feature Pyramid Network (FPN, also called as Backbone, on the left), Region Proposal Network (RPN, on the top right), and Region of Interest (RoI, on the bottom right).

attention has shifted up when we compare the saliency map with that of benign image seen in Figure 7g.

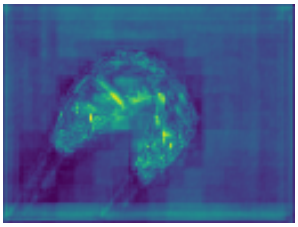
To examine the features extracted by internal modules, we then go to Module Viewer. We see that there are no stark differences between feature maps of earlier modules such as P2, as seen in Figure 7c and Figure 7d. We then move our focus on later modules such as P6 and start seeing more distinctive difference between the feature maps given the benign and adversarial images. For example, anchor delta maps extracted by P6 in RPN reveal difference between the original (Figure 7e) and adversarial (7f) cases. The highly activated parts in those feature maps are not aligned; P6 anchor delta map for benign image has horizontally activated regions as expected since the train is on the horizontal rail; however, the activated parts in P6 anchor delta map for adversarial image seem to be tilted, which may cause the final bounding box has shorter width than that of benign detection. We therefore see that an attack which results in the translation of the bounding box impacts parts of Region Proposal Network.

### 3.3. Incorrect Classification

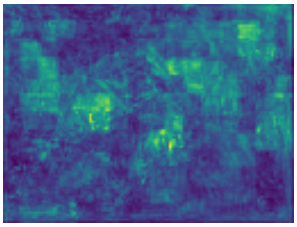
We found several incorrect detections given adversarial images, where the adversarial detections only attack the label and not the bounding box. While inspecting Explanation Viewer, we see subtle but noticeable differences in the salient pixels in benign and adversarial images, as shown in Figure 8g and Figure 8h. There is the emphasis on common features between dog and bear such as the lower half face on both benign and adversarial images. However, there is de-emphasis of unique features of a bear such as its eyes is reflected in the gradients of the adversarial image. To inspect the features extracted by different modules, we use Internal Viewer and compare the feature maps extracted by FPN and RPN. We then turn to the Inspector Viewer In the feature map by P6 in RPN, as shown in Figure 8c and Figure 8d, we see that the objectness is very similar for both benign and attacked images in that big objects at the center are detected. It may be because P6 generally captures features for large objects. On the other hand, when we inspect feature maps generated by P2 component in RPN module, we found bigger difference between the benign and adversarial images as shown in Figure 8e and Figure 8f. The RPN module does not seem to well detect the lower half of the face of the bear; compared to the benign feature map, the bear's nose is not highly activated in the adversarial feature map. This finding aligns well with what Explanation Viewer reveals about the de-emphasis on lower half of the bears' face in the adversarial image. In summary, DetectorDetective helps users more easily visualize and understand the impact of the feature extractor FPN over RPN module for this type of attacks.


(a) Original detection

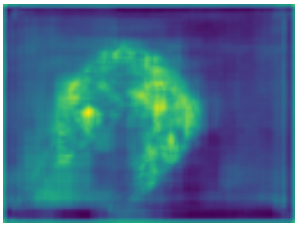

(b) Multiple Adversarial Detections



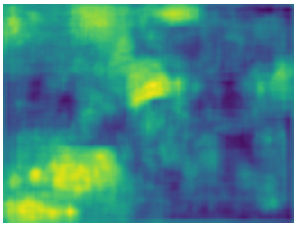
(c) Interpretable Features in P2



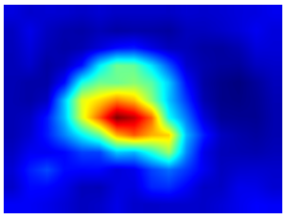
(d) Scattered Features in P2



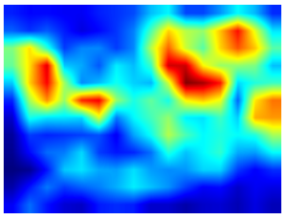
(e) P2-Objectness detects kite



(f) P2-Objectness detects sky



(g) Gradcam focuses on Kite

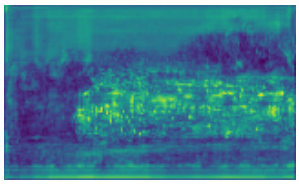


(h) Gradcam highlights sky

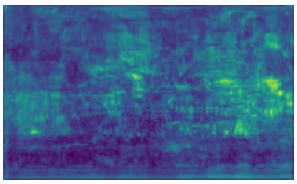
Figure 6. Hallucinated Detections: the object detector identifies multiple objects that are not present in the adversarial image. For example, in Figure 6b, multiple objects are detected on the background of the image, compared to the benign case where a kite is correctly identified in the benign image as seen in Figure 6a. In the adversarial image, DetectorDetective reveals that an internal component in the object detector (e.g., P2) extracts void features from where the sky is in the image, which may lead to the multiple incorrect detections (Figure 6d, 6f, 6h), compared to the benign cases (Figure 6c, 6e, 6g).


(a) Original detection


(b) Translated Detection

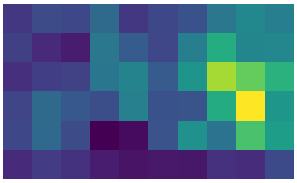


(c) Interpretable Features in P2

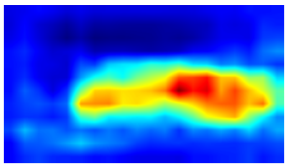


(d) Interpretable Features in P2

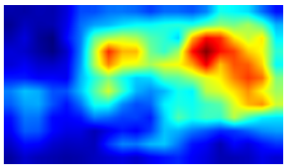
(e) P6-Anchor Delta Map highlighting correct location and orientation



(f) P6-Anchor Delta Map highlighting translated location and elongated orientation



(g) GradCAM highlights train



(h) GradCAM shifted above to accommodate box

Figure 7. Translated Object Detection: the object detector identifies the train object correctly, but the bounding box coordinates are not well aligned with the object in the adversarial image (Figure 7a and 7b). A potential reason why the bounding box is located in an incorrect position in the adversarial image is that the features detected by anchor delta map in P6 (Figure 7f) are not horizontally arranged, whereas the train object is on the horizontal rail.

## 4. Conclusion

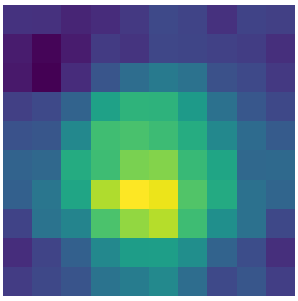
We present DetectorDetective, an interactive tool to help users understand the behaviors of an object detector, as adversarial images are processed inside through the model. DetectorDetective visually compares benign and adversarial features extracted by core components in the model to reveal the effect of adversarial attacks. Our tool is open-sourced, broadening the access to interpreting adversarial attacks on object detectors. We believe our visualization and comparison approaches help people gain insights into promoting defenses against attacks.

## 5. Acknowledgements

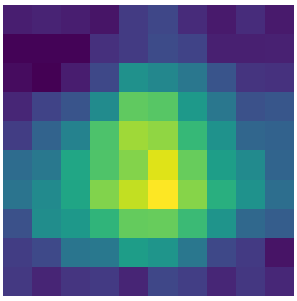
This work was supported in part by Defense Advanced Research Projects Agency (DARPA). Use, duplication, or disclosure is subject to the restrictions as stated in Agreement number HR00112030001 between the Government and the Performer. This work was also supported by gifts from Avast, Fiddder Labs, Bosch, Facebook, Intel, NVIDIA, Google, Symantec, Amazon.


(a) Original detection

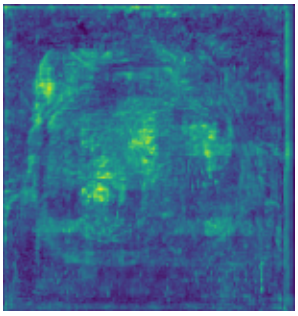

(b) Misclassified Detection



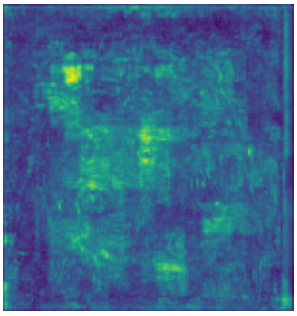
(c) Detection of central object in original image



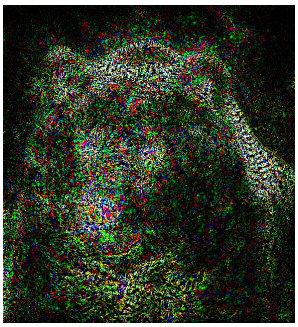
(d) Similar central object in attacked image



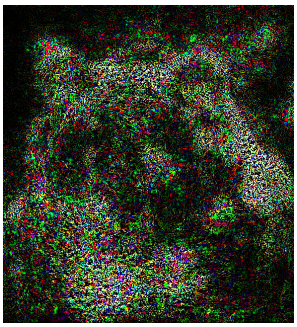
(e) Highlighted face at P2 feature extraction



(f) Highlighted chin and ears of bear at P2 Feature Extraction



(g) Gradients highlighting portions of original image



(h) Gradients highlighting portions of attacked image

Figure 8. Misclassified Detection: the object detector misclassifies the detected object (Figure 8a and 8b). Feature maps extracted by P2 (Figure 8f) and saliency map (Figure 8h) of the adversarial image reveal that the object detector may miss the feature of the bear's lower half of the face, resulting in the misclassification as a dog.

## References


[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31, 2018. 1

[2] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 839–847, 2018. 1

[3] 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

[4] 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. 1

[5] Christian Etmann, Sebastian Lunz, Peter Maass, and Carola-Bibiane Schönlieb. On the connection between adversarial robustness and saliency map interpretability. *arXiv preprint arXiv:1905.04172*, 2019. 1, 4

[6] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1625–1634, 2018. 1

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

[8] Yanming Guo, Yu Liu, Theodoros Georgiou, and Michael S Lew. A review of semantic segmentation using deep neural networks. *International journal of multimedia information retrieval*, 7(2):87–93, 2018. 1

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 1

[10] 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. 3

[11] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 1

[12] Bo Luo, Yannan Liu, Lingxiao Wei, and Qiang Xu. Towards imperceptible and robust adversarial example attacks against neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 1, 3

[13] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 1

[14] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 2

[15] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 1, 4

[16] 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

[17] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019. 3

[18] Kaidi Xu, Gaoyuan Zhang, Sijia Liu, Quanfu Fan, Mengshu Sun, Hongge Chen, Pin-Yu Chen, Yanzhi Wang, and Xue Lin. Adversarial t-shirt! evading person detectors in a physical world. In *European conference on computer vision*, pages 665–681. Springer, 2020. 1

[19] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 30(11):3212–3232, 2019. 1

[20] Zhengxia Zou, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *arXiv preprint arXiv:1905.05055*, 2019. 1