Role of Spatial Context in Adversarial Robustness for Object Detection

Aniruddha Saha*   Akshayvarun Subramanya*   Koninika Patil   Hamed Pirsiavash

University of Maryland, Baltimore County

{anisaha1, akshayv1, koni1, hpirsiav}@umbc.edu

Abstract

The benefits of utilizing spatial context in fast object detection algorithms have been studied extensively. Detectors increase inference speed by doing a single forward pass per image which means they implicitly use contextual reasoning for their predictions. However, one can show that an adversary can design adversarial patches which do not overlap with any objects of interest in the scene and exploit contextual reasoning to fool standard detectors. In this paper, we examine this problem and design category specific adversarial patches which make a widely used object detector like YOLO blind to an attacker chosen object category. We also show that limiting the use of spatial context during object detector training improves robustness to such adversaries. We believe the existence of context based adversarial attacks is concerning since the adversarial patch can affect predictions without being in vicinity of any objects of interest. Hence, defending against such attacks becomes challenging and we urge the research community to give attention to this vulnerability.

1. Introduction

The computer vision community has studied the role of spatial context in object detection for a long time. It is well-known that object detection performance is improved by considering information from the entire scene [6, 36, 5, 2]. Fast object detectors (YOLO [25], SSD [17], and Faster-RCNN [26]) process the entire image in one forward pass to reduce inference time and hence implicitly use contextual information for their predictions. Some previously used object detectors (e.g., RCNN [10]) do not use context since they do a forward pass for each region proposal separately. However, they are less accurate and have much slower inference compared to the fast detectors mentioned above.

In this paper, we study the scenario where an adversary can exploit the implicit contextual reasoning to fool object detectors in a practical setting. We show that category specific

---
*Equal contribution

cific contextual adversarial patches (e.g. all categories in PASCAL VOC) can be designed which when pasted on the corner of an image, having no overlap with any objects of interest can make the detector blind to a specific object category chosen by the adversary, while not influencing the detections of objects of other categories. For example, if the category chosen is "pedestrian" in self-driving car applications, the attack may have a high impact on safety. Since our adversarial patch does not overlap with the object being attacked, we believe the attack works because the detector considers the influence of not only the object but also the surrounding objects and the scene.

We believe this is of paramount importance because:

(1) This poses a question to the research community with regards to improving accuracy versus robustness when using detectors that utilize context. The problem is challenging since there is no straightforward way of limiting fast state-of-the-art object detectors to avoid using context even at the cost of accuracy. As mentioned above, these detectors process the image only once for fast inference. Also, in deep networks, it is difficult to limit the receptive field of the final layers so as to not cover the whole image.

(2) Adversarial patch attacks are easily reproducible in a practical setting as compared to standard adversarial attacks which add ϵ-bounded perturbation to the whole image. One can simply print an adversarial patch and expose it to a self-driving car by pasting it on a wall, on a bumper, or even by hacking a road-side billboard. In contrast, regular adversarial examples need to perturb all pixel values. Though the perturbations are small, this is not very practical.

(3) Defense algorithms developed for regular adversarial examples[19, 11] are not necessarily suitable for adversarial patches since the patches are not norm-constrained, i.e., can have large perturbations in a limited number of dimensions. Hence, in the space of pixel values, the adversarial image, i.e., image + patch, can be very far from the original image. There are two standard defense frameworks for regular adversarial examples: (a) Training with adversarial examples: This is not suitable for our case since learning the patch is expensive. (b) Training with regularizers in the input space: This is not suitable either since such defense algorithms as-

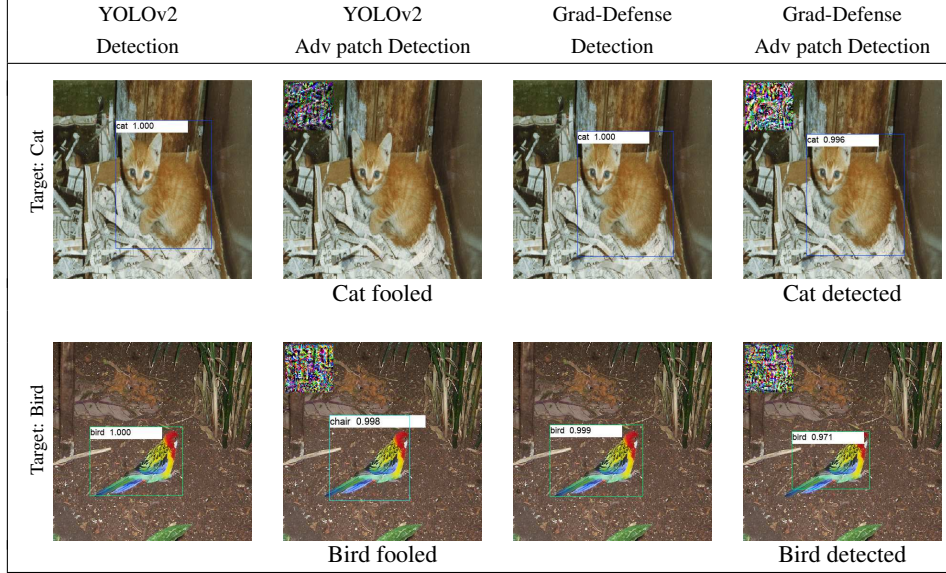| YOLOv2 Detection	YOLOv2 Adv patch Detection	Grad-Defense Detection	Grad-Defense Adv patch Detection

Figure 1: **Per-image blindness attack and Grad-Defense results-** We compare the detection results of YOLOv2 with our Grad-Defense model. We attack each model separately. We see that our Grad-Defense model is less susceptible to context-based attack. The target category is written below each example. The patch is always on the top-left corner.

sume the noisy image is close to the original one in the pixel space.

In this paper, we propose a defense which limits the use of context while training the object detector. We observe that it performs the best among all the other defense baselines. We show an example in Figure 1 where our defense model is more robust to adversarial patch attack. We believe that this is a much needed step towards developing defense algorithms which fix the vulnerabilities introduced by adversarial patches.

As an additional contribution, we investigate the efficacy of currently used explainability algorithms for object detectors. We observe that the well-known Grad-CAM algorithm [28] is not suitable for localization tasks. We modify this algorithm to localize the explanation of each detection and use it to visualize the use of context before and after training using our defense algorithm.

Our findings in this paper show that even though we believe in the richness of context in object detection, employing contextual reasoning can be a *double-edged sword* that makes the object detection algorithms vulnerable to adversarial attacks.

2. Related Work

Vulnerability to Adversarial Attacks: Convolutional neural networks have been shown to be vulnerable to adversarial examples. Szegedy *et al*. [34] first discovered the existence of adversarial images. Gradient based techniques such as Fast Gradient Sign Method (FGSM) [11] and Projected Gradient Descent (PGD) [19] were used to create these adversarial examples. [32] showed that perturbing

just one pixel is sufficient to fool classifiers. [20] presented an optimal way to create adversarial examples and later extended that to create an universal adversarial image [21]. However these attacks are not feasible for practical applications like IOT cameras and autonomous cars since we do not have access to all pixels and even perturbing single pixel that fools in multiple conditions becomes challenging. [38] showed that we can fool classifiers by adding an adversarial frame around the image that is trained to fool the classifier. Apart from classification, object detection and semantic segmentation networks have also shown to be vulnerable to adversarial examples. [37, 31] have shown that adversarial examples can be created for object detectors as well. Fischer et al. [8] showed the same for semantic segmentation.

Adversarial patches: Adversarial patches were introduced in [3, 14] which when pasted on an image can cause a classifier to output a target class. This poses a question as to whether object detection also has these vulnerabilities. Humans are rarely susceptible to mis-classifying objects in a scene if we introduce artifacts which do not overlap with the objects of interest. However, we show in this paper that object detection networks which use context can be fooled easily, taking advantage of this fact. In [31, 35], adversarial patches are pasted on top of the object (e.g. stop sign and person). This changes the appearance of the object and fools the detector. We consider a setting where the patch has no overlap with the object and thus highlight that contextual reasoning can lead to adversarial vulnerability making the threat model more challenging. [18] create a digital attack on object detectors where the pixel values are unconstrained. [15] designs a patch for YOLO which when

presented in a scene suppresses all detections. But we show that category specific contextual adversarial patches can be designed which can make the detector blind to a specific object category chosen by the adversary and not strongly influence detections of objects of other categories. [13] create adversarial signboards which when placed below each instance of a stop sign fools the detector. Our patch can be far from all objects and can produce blindness for all the attacker chosen category instances in the image. [24] shows the vulnerability of optical flow systems to adversarial patches. [33] shows that adversarial patches can be created which fool network interpretation algorithms as well.

Contextual reasoning in object detection: The relationship between objects and scene has been known for a long time to the computer vision community. Particularly, spatial context has been used in improving object detection. [6] empirically studies use of context in object detection. [36] shows that scene classification and object classification can help each other. [5] utilizes spatial context in a structured SVM framework to improve object detection by detecting all objects in the scene jointly. [12] learns to use stuff in the scene to find objects. [22, 2, 27] discuss the role of context in object detection while [29] for classification and segmentation. [40] shows that a network trained for scene classification has implicitly learned object detection which suggests the inherent connection between scene classification and object detection.

More recently, with the emergence of deep networks, utilizing context has become easier and even unavoidable in some cases. For instance, Faster-RCNN, SSD, and YOLO process the input image only once to extract the features and then detect objects in the feature space. Since the features are coming from all over the image, the model naturally uses global features of the scene.

3. Method

Background on YOLO: Given an image x, YOLO divides the image into a $S \times S$ grid where each grid cell predicts B bounding boxes, so there are BS^2 possible bounding boxes. Assuming C object classes, the model processes the image once and outputs, $P(object) \in R$: an objectness confidence score for each possible bounding box and $P(category|object) \in R^C$: scores of all categories conditioned on existence of an object at every grid cell, and a localization offset for each bounding box. During inference, the objectness score and class probabilities are multiplied to get the final score for each bounding box. In YOLO training, the objectness score is encouraged to be zero for background locations and closer to one for ground-truth object locations, and the class probabilities are encouraged to match the ground-truth only at the location of objects.

Adversarial patches: Assume an image x, a recognition function $f(.)$, e.g., object classification, and a constant bi-nary mask m that is 1 on the patch location and 0 everywhere else. The mask covers a small region of the image, e.g., a box in the corner. We want to learn a noise patch z that when pasted on the image fools the recognition function. Hence, in learning, we want to optimize:

$$z^* = \arg\min_z \mathcal{L}\big(f(x \odot (1 - m) + z \odot m; t)$$

where $\odot$ is the element-wise product, t is the desired adversarial target, and $\mathcal{L}$ is a loss function that encourages the fooling of the model towards predicting the target. Note that any value in z for which $m = \mathbf{0}$ will be ignored.

In standard adversarial examples, we learn an additive perturbation so that when added to the input image, it fools a recognition function, e.g., object classifier or detector. Such a perturbation is bounded usually by ℓ_∞ norm to be invisible perceptually. However, in adversarial patches z is not additive and is not bounded. It is only bounded to be in the range of allowed image pixel values in the mask location. This difference makes studying adversarial patches interesting since they are more practical (they can be used by printing and showing to the camera) and also are difficult to defend due to unconstrained perturbations.

3.1. Our adversarial attacks:

Per-image blindness attack: We are interested in studying if an adversary can exploit contextual reasoning in object detection. Hence, given an image x and an object category of interest c, we develop an adversarial patch that fools the object detector to be blind to category c while the patch does not overlap with instances of c on the image.

Since we are interested in blindness, we should develop attacks that reduce the number of true positives rather than increasing false positives. We believe increasing the number of false positives is not an effective attack in real applications. For instance, in self-driving car applications, not detecting pedestrians can be more harmful than detecting many wrong pedestrians. Therefore, in designing our blindness attack, we do not attempt to fool the objectness score, $P(object)$ of YOLO and fool only the probability of the object category conditioned on being an object, $P(category|object)$.

We initialize our patch to be a black patch (all zeros). Then, we tune the noise z to reduce the probability of the category c that we want to attack on all locations of the grid that match the ground-truth. We do this by simply maximizing the summation of the cross-entropy loss of category c at all those locations. For optimization, we adopt a method similar to projected gradient descend (PGD) [19] in which after each optimization step and updating z, we project z to be in the range of the acceptable pixel values [0-1] by clipping. Note that z will have no contribution at the locations off the patch where $m = \mathbf{0}$. We stop the optimization when

there is no detection for category c on the image or we reach the maximum number of iterations.

Universal blindness attack: Following [21], we extend our attack to learn universal adversarial patches. For category c, we learn a universal patch z on training data that makes the detector blind for category c across unseen test images. To do so, we adopt the above optimization by iterating through training images in the optimization while keeping z shared across all images.

Significance of blindness attack: Note that there are two ways of reducing mAP in object detection: (1) introducing lots of false positives, (2) reducing true positives. We believe (1) can be achieved without exploiting the contextual reasoning by generating lots of false positives at the patch location to dominate the mAP calculation, which may not necessarily affect the true positives. However, since we are interested in showing the exploitation of contextual reasoning, we focus on (2) where the adversary should make the detector blind by reducing true positives.

To see this effect, we modify the mAP calculation in our experiments by removing any false positives at the patch location. We believe considering regular mAP for evaluation does not necessarily show the contextual exploitation.

3.2. Defense for our adversarial attacks:

Defending against adversarial examples has been shown to be challenging [4, 1]. As discussed in the introduction, we believe defending against adversarial patches is even more challenging since the attack is expensive and the perturbation is not bounded to lie in the neighborhood of the original image.

Grad-defense: Since we believe the main reason for the success of contextual adversarial patches is the exploitation of contextual reasoning by the adversary, we design our defense algorithm to limit the usage of contextual reasoning during training the object detector.

In most fast object detectors including YOLO, each object location has a dedicated neuron in the final layer of the network, and since the network is deep, those neurons have very large receptive fields that span the whole image. To limit the usage of context by the detector, we are interested in limiting this receptive field only to the bounding box of the corresponding detection.

One way of doing this is to hard-code a smaller receptive field by reducing the spatial size of the filters in the intermediate layers. However, this is not a great defense since: (1) it will reduce the capacity and thus accuracy of the model, (2) it shrinks the receptive field independent of the size of the detected box, thereby hurting the detection of large objects. We conducted an experiment where we change the network architecture of YOLOv2 and set the filter sizes for all layers after Layer16 (just before the pass-through connection) to 1x1. We observe that this model gives poor mAP on clean images which is reported in Table 2.

We believe that a better way of limiting the receptive field would be use a data-driven approach. Network interpretation tools like Grad-CAM [28] that highlight the image regions which influence a particular network decision can be used. Grad-CAM works by visualizing the derivative of the output with respect to an intermediate convolutional layer (e.g., $conv5$ in AlexNet) that detects some high-level concepts. To limit the contextual reasoning in object detection, we should constrain such derivatives for a particular output to not span beyond the bounding box of the corresponding detected object. Hence, to defend against adversarial attacks, during YOLOv2 training, we calculate the derivative of each output with respect to an intermediate convolutional layer and penalize its nonzero values outside the detected bounding box.

More formally, assuming y^c is the confidence of an object belonging to category c detected at bounding box B and A_{ij}^k for the activation of a convolutional layer at location (i, j) and channel k, we calculate the derivative $\frac{\partial y^c}{\partial A_{ij}^k}$ and normalize it to sum to 1 over the whole feature map:

$$\hat{\beta}_{ij} = \frac{\beta_{ij}}{\sum_{i,j} \beta_{ij}} \quad \text{where} \quad \beta_{ij} = \sum_k \left| \frac{\partial y^c}{\partial A_{ij}^k} \right| \quad (1)$$

Then, we minimize the following loss to encourage $\hat{\beta}$ values to be completely inside the bounding box B.

$$\mathcal{L} = -\sum_{i,j \in B} \hat{\beta}_{ij} \quad (2)$$

Since $\hat{\beta}$ sums to 1, minimizing this loss will minimize the influence of image regions outside the detected bounding box on its corresponding object detection. We believe this is a regularizer that limits the receptive field of the final layer depending on the size of detected objects. So, it should limit the contextual reasoning of the object detector. Interestingly, this loss can be even minimized on unlabeled data in a semi-supervised setting.

Out-of-context(OOC) defense: Another way of limiting contextual reasoning is to remove the influence of context from the training data. We do so by simply overlaying an out-of-context foreground object on the training images. To create the dataset, we take two random images from the PASCAL VOC training data, crop one of the annotated objects from the first image and paste it at the same location on the second image. We blur the boundary of the pasted object to remove sharp edges and also remove the annotations of the second image corresponding to the objects occluded by the added foreground object. We keep non-overlapping annotations intact. We train YOLOv2 on the new dataset to get a model that is less dependent on context. Fig. 2 shows example out of context training images. A few other defense algorithms used as baselines are described in the experiments section.

Figure 2: **Out of context images-** This figure shows examples from the out of context dataset we curated to train our OOC defense model.

4. Experiments

We use PASCAL VOC dataset [7] for all our experiments [1]. For each chosen attack category, we create a subset of images which contain the object category and where none of the instances overlap with the patch area. This results in a filtered image set per category. We also evaluate the universal blindness attack on KITTI dataset [9] for "pedestrian" and "car".

4.1. Implementation details:

We use Pytorch [23] implementation of YOLOv2 and NVIDIA Titan X GPUs to run our experiments. The image size is fixed at 416×416 and our patch size is fixed at 100x100 which covers less than 6% of the image area. The top-left corner of the patch is at pixel $(5, 5)$. When optimizing, we initialize z from all zeros (black patch) and use the Adam optimizer. We also experimented with the standard l_∞ PGD attack but it did not provide any improvement in attack performance or convergence. For per-image experiments, we use the learning rate of 0.1 and run the attack for 250 iterations per image. For universal patch experiments, we use learning rate of 0.01, minibatch size of 1, 10 iterations per minibatch, and total of 100 epochs. The confidence threshold, NMS threshold, and IOU overlap threshold used for evaluations are 0.005, 0.45 and 0.5 respectively.

For Grad-defense experiments, we obtained best results considering the gradients at 16th layer of YOLOv2 (just before the pass-through connection), and $\lambda = 1.0$ which balances the YOLO standard loss and our loss in Eq. 2.

4.2. Evaluation:

Since we are interested in learning adversarial patches that do not overlap with the object, we fix the location of the patch, e.g., at top-left corner, and run evaluations only for images on which there is no object of interest overlapping with the patch location as described earlier.

Because our filtered image sets are independent, we run evaluations independently for each category and report the AP scores for the category in question. Hence, the mean average precision (mAP) we are reporting is a little different

---

[1]Code available at https://github.com/UMBCvision/Contextual-Adversarial-Patches

---

from common practice since it is mean over categories that use different image sets. This is due to the fact that there can be multiple objects in an image and such images can be part of different image sets. As mentioned earlier, we remove the false positives overlapping with the patch because we do not want our attack to work by making the patch as the most salient object.
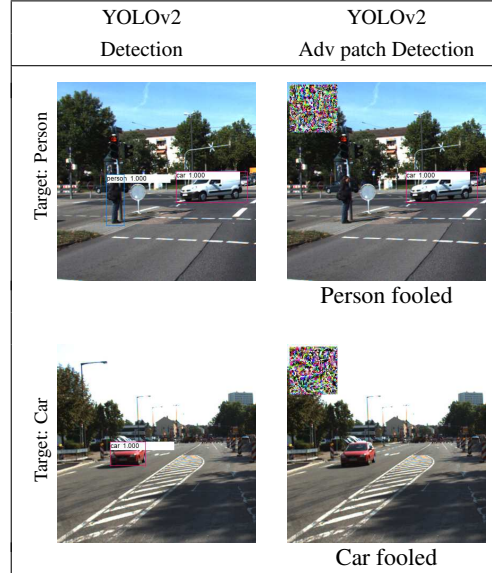

Figure 3: **Evaluation on KITTI dataset-** Examples of trained universal patch for car and person.

4.3. Results and discussion

In this section, we evaluate our attacks and defense algorithms in various settings.

Per-image blindness attack: We consider three baselines where we replace the adversarial patch with all white patch, an i.i.d. noise patch, or an out-of-context patch and show that our learned adversarial patch is better.

Table 1 shows the results of our per-image blindness attack experiments. We show the total number of images in PASCAL VOC 2007 and our filtered images. The "White patch" and "Random noise patch" baselines do not reduce the accuracy which means the attack needs to be tuned. Our attack reduces the mAP from 76.04% to 55.42% which is 20.61 points reduction. The highest drop of almost 35 points is for "train" and "aeroplane".

Universal blindness attack: Table 2 shows the results of our universal blindness attack experiments. We divide the filtered images to two halves for training and testing and then learn a universal adversarial patch per category on the training set and test it on the testing set. We see that our attack reduces the mAP for YOLOv2 from 76.85% to 56.24% which is almost 20.5 points reduction in mAP.

Defense evaluations: The results of defense algorithms are reported in Table 2. Our Grad-defense algorithm outper-

	Mean	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	dtable	dog	horse	mbike	person	pplant	sheep	sofa	train	tv
Total No. images	-	205	250	289	176	240	183	775	332	545	127	247	433	279	233	2097	254	98	355	259	255
No. filtered images	-	136	190	182	114	205	102	510	160	454	87	212	244	173	142	1286	174	73	273	148	187
YOLOv2 (clean)	**76.04**	75.05	81.02	75.22	66.58	50.59	81.08	79.86	80.96	64.40	85.19	76.32	85.35	85.91	80.08	75.62	57.28	79.90	79.83	83.30	77.18
White patch	76.33	75.30	80.59	76.01	67.00	50.69	81.20	79.85	80.89	64.18	84.83	76.99	85.93	86.33	80.23	75.69	57.75	79.86	81.02	84.82	77.42
Random noise patch	76.20	75.00	80.63	75.87	66.29	50.24	81.07	79.60	81.07	64.23	85.37	76.96	86.34	86.28	79.81	75.60	57.17	80.03	80.71	84.78	76.86
OOC attack	75.93	74.46	80.40	75.56	65.02	50.83	81.61	79.58	81.01	63.61	84.74	76.92	86.25	86.42	79.39	75.68	56.32	79.82	80.83	82.89	77.29
Adv patch attack (Ours)	**55.42**	40.89	71.51	44.11	38.46	39.90	60.25	62.28	57.25	54.33	54.03	71.27	62.90	67.98	66.77	59.87	38.48	55.53	64.14	47.96	50.56
Grad-Defense (clean)	**75.83**	73.41	80.71	75.33	61.32	47.28	81.33	81.05	84.19	63.11	86.02	77.01	86.14	85.91	82.84	74.39	53.34	81.01	81.12	84.73	76.40
Grad-Defense (attacked)	**61.31**	51.71	75.53	53.17	42.88	40.20	66.96	71.43	62.13	56.29	59.59	73.46	69.91	75.58	74.98	63.32	40.23	59.47	70.86	61.95	56.45

Table 1: **Per-image blindness attacks, and defense evaluation-** "YOLOv2 (clean)" shows the results of YOLOv2 on our clean per-image attack test sets. AP is calculated only on the filtered subset of images. "Adv patch attack" is our adversarial patch algorithm. We also evaluate three baseline attacks. Qualitative examples are shown in Fig. 1.

	Mean	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	dtable	dog	horse	mbike	person	pplant	sheep	sofa	train	tv
Total No. training imgs	-	68	95	91	57	102	51	255	80	227	43	106	122	86	71	643	87	36	136	74	93
Total No. testing imgs	-	68	95	91	57	102	51	255	80	227	43	106	122	86	71	643	87	36	136	74	93
YOLOv2 (clean)	**76.85**	79.25	83.17	77.19	63.88	49.70	80.61	79.47	80.59	64.92	85.76	77.39	86.65	81.32	84.78	75.41	56.82	89.05	76.96	87.59	76.56
YOLOv2 (attacked) (Ours)	56.24	29.66	71.51	39.7	34.14	44.67	65.21	60.26	44.41	58.28	61.94	77.12	67.52	67.82	59.16	65.2	46.17	69.87	72.04	42.07	47.96
AT-2000 (clean)	64.01	53.17	69.84	58.82	41.98	39.34	67.42	71.15	70.74	50.27	69.97	71.35	77.09	77.06	75.72	69.16	44.89	76.06	63.26	62.05	70.95
AT-2000 (attacked)	41.55	17.36	55.57	27.36	17.55	36.41	31.67	47.02	39.34	42.76	38.98	71.93	64.01	38.12	51.38	57.16	28.82	42.55	59.88	22.70	40.43
AT-30 (clean)	70.47	73.81	75.60	65.49	54.73	43.28	76.81	75.87	71.15	57.14	81.41	74.04	72.66	77.81	79.42	68.89	47.10	85.65	74.79	78.93	74.80
AT-30 (attacked)	50.47	29.23	75.83	27.16	42.23	38.86	46.15	65.27	41.22	51.20	61.10	75.73	46.88	40.40	76.21	53.05	37.21	60.19	56.96	31.00	53.42
OOC Defense (clean)	65.67	79.93	75.63	59.53	48.54	34.68	78.39	74.23	71.35	44.55	68.22	76.31	69.73	72.50	68.18	66.02	38.90	74.34	64.55	76.84	70.93
OOC Defense (attacked)	60.35	70.81	70.03	45.84	44.47	33.06	76.87	70.98	60.26	36.89	67.55	76.01	64.47	71.40	61.63	64.26	37.42	68.68	63.72	63.98	58.75
YOLOv2 1x1 (clean)	59.55	69.34	68.79	51.15	41.41	43.48	61.86	73.98	51.71	48.28	72.96	55.08	66.58	65.96	70.29	66.01	41.00	77.89	43.04	54.52	67.59
YOLOv2 1x1 (attacked)	59.57	69.25	68.84	51.07	41.13	43.67	61.66	74.12	52.09	48.46	72.89	55.11	66.34	65.96	71.02	65.85	41.36	76.67	42.95	54.54	68.42
Gradient w.r.t. input (clean)	65.80	62.18	75.85	58.91	46.55	31.78	76.12	62.96	72.72	53.83	68.74	73.16	69.46	76.41	75.47	58.32	62.30	74.45	73.06	72.20	71.51
Gradient w.r.t. input (attacked)	48.97	22.39	67.68	29.68	22.02	29.30	67.70	45.03	39.96	49.19	42.80	74.71	61.23	55.89	65.10	46.45	41.66	51.78	69.53	42.39	54.82
Grad-Defense (clean)	**76.09**	76.31	83.08	75.49	65.15	46.41	83.11	80.53	81.99	63.73	87.22	77.63	86.4	81.26	84.57	74.12	51.69	83.88	78.26	85.95	74.94
Grad-Defense (attacked)	**64.84**	50.9	80.47	50.11	48.00	46.23	78.93	68.83	63.11	60.13	67.96	77.62	77.26	77.67	68.92	67.05	48.50	70.99	73.97	60.3	59.79

Table 2: **Universal blindness attacks, and defense evaluation-** "YOLOv2 (clean)" shows the results of YOLOv2 on our clean universal attack test sets. Note that the image sets are different from Table 1. "YOLOv2 (attacked)" shows our universal patch blindness attack. We report results for two adversarially trained models (2000 and 30 iterations). "OOC Defense" refers to YOLOv2 trained on our out-of-context dataset. For YOLOv2 1x1, we limit the size of final conv layers to 1x1. "Gradient w.r.t. input" penalizes the gradient of the output w.r.t the input in pixel space. "Grad-Defense" is our main defense algorithm that outperforms other algorithms on both clean and attacked images. Qualitative examples are shown in Fig. 4.

forms other methods in terms of both accuracy on clean images (76.09%) and attacked images (64.84%). The original YOLOv2 model has an accuracy of 76.85% on clean images and 56.24% on attacked images. This shows the effectiveness of our defense method on limiting the contextual reasoning. In Table 2, we also evaluate a few other defense algorithms including training with out of context data. Our Grad-defense is also effective for per-image attacks. The results are reported in Table 1.

Adversarial training: Adversarial training is a common defense method for regular adversarial examples. Unfortunately, this is not suitable for adversarial patches since calculating each adversarial patch example is computationally expensive. For completeness, we use this as a baseline. We use 30 and 2000 iterations to generate adversarial patch examples and use them in training the model. The results are reported in Table 2. Even though we run the 2000 iteration attack for almost 6 days, its accuracy on clean images was still very low. The adversarially trained model has a clean accuracy of 70.47% but it has poor performance after attack and gives 50.47% accuracy.

Patch detection and Gaussian blurring: One might wonder that because the adversarial patches are not norm-constrained and look noisy, it should be easy to detect their presence on the image using a simple network or blur them using a Gaussian filter. In our experiments, we show that by incorporating these techniques into our patch training loop, we can still fool the patch detection or blurring system.

For the patch detection network, we create a dataset of clean images and attacked images and finetune a ResNet-18 network for a binary classification task - whether a patch is present in an image or not. We evaluate the performance of the patch detector on our attacked images. Then, we incorporate the patch detector into our adversarial patch optimization loop by adding another loss which when minimized fools the patch detector. We see that using the same attack setting, we are able to achieve comparable attack efficiencies and are now able to fool the patch detector as well. The results are shown in Table 3. We believe one can still train another detector, so a more advanced defense would be to train the attack and detector together in an adversarial game setting. However, we believe it is beyond the scope of this paper. For the Gaussian blurring case, we create universal adversarial patches using our method and add that to the image. The resultant image is blurred using a 7x7 Gaussian kernel to limit the effectiveness of the patch. We observed that after blurring, we see an increase of mAP from 56.24% to 71.53%. This means that the blur-

	Mean	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	dtable	dog	horse	mbike	person	pplant	sheep	sofa	train	tv
YOLOv2 (attacked)	56.24	29.66	71.51	39.7	34.14	44.67	65.21	60.26	44.41	58.28	61.94	77.12	67.52	67.82	59.16	65.2	46.17	69.87	72.04	42.07	47.96
Patch detector accuracy	**98.27**	100.00	95.79	100.00	92.98	100.00	100.00	98.43	98.75	99.12	100.00	91.51	96.72	100.00	95.77	98.44	100.00	100.00	97.81	100.00	100.00
YOLOv2 (fool detector)	57.66	33.17	67.18	39.30	34.27	44.47	77.57	60.49	51.15	57.58	61.91	77.40	76.05	68.39	60.18	66.97	46.41	63.29	67.97	42.56	56.91
Patch detector accuracy	**5.54**	2.94	4.21	7.69	1.75	0.97	25.49	1.57	18.75	1.76	11.36	0.00	4.10	2.30	2.82	4.98	6.90	2.70	0.00	4.05	6.38

Table 3: **Fooling patch detector-** We show that even though a norm-unconstrained adversarial patch can be detected because of its high frequency components, such a detector can be incorporated into the training loop and be fooled by our universal blindness attack with similar efficiency.

ring reduced the effectiveness of our patch. A similar idea was discussed as a form of defense for ℓ_∞ adversarial examples in [16]. However, it was shown in [4] that such a defense can be overcome by creating the adversary with the knowledge of blurring. We perform a similar experiment to show that adversarial patches can be created using the knowledge of blurring filter. We include the Gaussian filter as an additional layer at the input and train our universal patch iteratively by applying the blurring filter at randomly selected iterations. After attack, this gives us an mAP of 53.25% which is comparable to the results that we had obtained without incorporating blurring during training.

Per-image targeted attack: Though our main objective is to show the presence of blindness attack, we also evaluate a targeted attack. For each target object category, we choose 500 random images that do not contain the object. Then, we "artificially" change all ground-truth object labels to the target category in evaluation. We expect our targeted attack to increase the mAP on this artificial ground-truth from zero to a higher number. We use learning rate of 0.1 and run the attack for 2,000 iterations per image. On evaluation after attack, we see that the mAP increases from 0 to 18.61%. The detailed results are reported in the supplementary material.

Per-image objectness attack: Because YOLO makes its decision based on objectness as well as class scores, it is interesting to see whether we can fool objectness. This can be seen as a different form of attack. It is important to note that all mAP evaluations are done keeping the objectness threshold for acceptance at the original value of 0.005. To perform objectness attack, we run the patch optimization for 2,000 iterations per image and encourage the objectness scores to be below 0.005. We use learning rate of 0.05 for this experiment. We see a 17.5 point drop in mAP from 76.04% to 58.49%. The detailed results are provided in the supplementary material.

Universal blindness attack for KITTI: We evaluate our universal blindness attack on person (pedestrian) and car detection on KITTI dataset which is closer to self-driving car applications. Similar to PASCAL VOC experiments, we filter out the images for which the objects of interest overlap with the patch, but since the patch is on the top-left corner and people and cars are usually on the ground, we see that it mostly never overlaps with the objects of interest. Then we use the YOLOv2 model trained on PASCAL VOC dataset and learn universal adversarial patches for person and car categories. Since YOLOv2 needs to resize the image to be

square and KITTI frames have a different aspect ratio, we crop a random square box from the middle of KITTI frames so that we do not degrade the image quality by stretching it. Qualitative results are shown in Fig. 3. Our attack reduces car detection AP from 59.0% to 31.8% and person detection from 51.7% to 34.2% which are significant reductions.

4.4. DetGrad-CAM for visualization of context:

To analyze the usage of contextual information in object detection, we may use interpretation algorithms similar to those in object classification literature [28, 30, 39]. We consider a popular algorithm, Grad-CAM [28] which is for classification tasks. Specifically, we want to highlight the regions in the image responsible for the detection of each object. We can then analyze the amount of contextual information used by the model. To summarize the Grad-CAM algorithm, we use notations defined for Equation 1. The algorithm measures the importance score of each intermediate feature on the confidence y^c of an object by calculating:

$$\alpha_k^c = \frac{1}{Z} \sum_{i,j} \frac{\partial y^c}{\partial A_{ij}^k} \tag{3}$$

where Z is a normalizer. Then we calculate the weighted summation of activations and consider only the positive values, i.e, $G_{ij}^c = max(0, \sum_k \alpha_k^c A_{ij}^k)$. The final heatmap G_{ij}^c is resized to be in the size of input image.

However, the summation in Equation 3 has the effect of removing the spatial information when calculating the importance score. If an image has multiple objects of the same category, this can lead to highlighting all object instances for each detected instance. Since object detectors are trained to classify as well as localize the objects, the gradient values of the interpretation algorithm contain information regarding the location of the object. As a result, summation over all locations would result in loss of the spatial information. Hence, we modify the algorithm for detection task, namely **DetGrad-CAM**, by doing element-wise multiplication of gradients with the activations, and then summing over all features, i.e we calculate the heatmap as follows:

$$\tilde{G}_{ij}^c = max(0, \sum_k \frac{\partial y^c}{\partial A_{ij}^k} \odot A_{ij}^k) \tag{4}$$

We believe this simple modified method results in better explanation for object detection models as shown in Fig. 5. We also see that the Grad-defense model is using lesser contextual information compared to original model which leads to better defense against context-based adversaries.
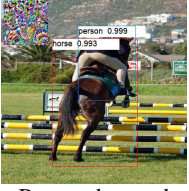
Figure 4: **Universal patch blindness attack and Grad-Defense results-** We compare the detection results of YOLOv2 with our Grad-Defense model . We attack each model separately. We see that our Grad-Defense model is less susceptible to context-based attack. The target category is written below each example. The universal patch per category is always on the top-left corner.
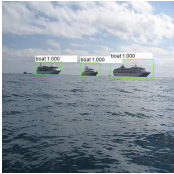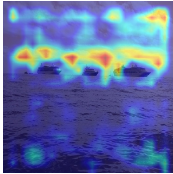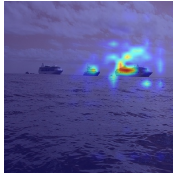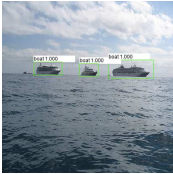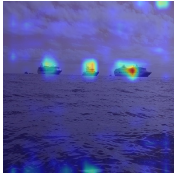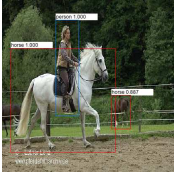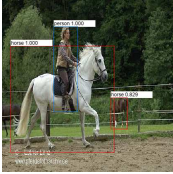

Figure 5: **DetGrad-CAM-** We observe that modifying Grad-CAM gives us better visual interpretations for the object detector. For each image we show the interpretation results of a detected object using Grad-CAM[28] and DetGrad-CAM from YOLOv2 and also our Grad-defense model. The category for which interpretation is calculated is written on the left. Note that for the first row, we are interested in finding the interpretation for the right-most boat and interestingly Grad-CAM highlights all three boats as important for detection.

5. Conclusion

Object detectors like YOLO, SSD and Faster-RCNN have become popular due to their fast inference time. They process the image only once rather than running the model on every proposal e.g., in RCNN. Hence, such models naturally learn to employ contextual reasoning which results in better accuracy. In this paper we show that reliance on context makes the detector vulnerable to category specific adversarial patches in which the adversary can make the detector blind for a category chosen by the attacker without occluding those objects in the scene or affecting the detection of other categories. This is also a practical attack that can cause major issues when deep models are deployed in real world applications like self-driving cars. We propose a defense algorithm by regularizing the model to limit the influence of image regions outside the bounding boxes of the detected objects. We show that the model trained this way improves robustness to the contextual attack. We believe this highlights the need for studying defense algorithms which are robust to contextual adversarial patch attacks but also accurate.

References

[1] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning*, pages 274–283, 2018. 4

[2] Ehud Barnea and Ohad Ben-Shahar. Exploring the bounds of the utility of context for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7412–7420, 2019. 1, 3

[3] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. In *Machine learning and Computer Security Workshop - NeurIPS*, 2017. 2

[4] Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 3–14. ACM, 2017. 4, 7

[5] Chaitanya Desai, Deva Ramanan, and Charless C Fowlkes. Discriminative models for multi-class object layout. *International journal of computer vision*, 95(1):1–12, 2011. 1, 3

[6] Santosh K Divvala, Derek Hoiem, James H Hays, Alexei A Efros, and Martial Hebert. An empirical study of context in object detection. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1271–1278. IEEE, 2009. 1, 3

[7] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, Jan. 2015. 5

[8] Volker Fischer, Mummadi Chaithanya Kumar, Jan Hendrik Metzen, and Thomas Brox. Adversarial examples for semantic image segmentation. *CoRR*, abs/1703.01101, 2017. 2

[9] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 5

[10] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 1

[11] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. 1, 2

[12] Geremy Heitz and Daphne Koller. Learning spatial context: Using stuff to find things. In *European conference on computer vision*, pages 30–43. Springer, 2008. 3

[13] Yi Huang, Adams Wai Kin Kong, and Kwok-Yan Lam. Adversarial signboard against object detector. 3

[14] Danny Karmon, Daniel Zoran, and Yoav Goldberg. Lavan: Localized and visible adversarial noise. In *International Conference on Machine Learning*, pages 2512–2520, 2018. 2

[15] Mark Lee and Zico Kolter. On physical adversarial patches for object detection. *arXiv preprint arXiv:1906.11897*, 2019. 2

[16] Xin Li and Fuxin Li. Adversarial examples detection in deep networks with convolutional filter statistics. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5764–5772, 2017. 7

[17] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 1

[18] Xin Liu, Huanrui Yang, Linghao Song, Hai Li, and Yiran Chen. Dpatch: Attacking object detectors with adversarial patches. *CoRR*, abs/1806.02299, 2018. 2

[19] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. 1, 2, 3

[20] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. *in CVPR*, 2016. 2

[21] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1765–1773, 2017. 2, 4

[22] Aude Oliva and Antonio Torralba. The role of context in object recognition. *Trends in cognitive sciences*, 11(12):520–527, 2007. 3

[23] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019. 5

[24] Anurag Ranjan, Joel Janai, Andreas Geiger, and Michael J. Black. Attacking optical flow. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 3

[25] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 1

[26] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1

[27] Amir Rosenfeld, Richard Zemel, and John K Tsotsos. The elephant in the room. *arXiv preprint arXiv:1808.03305*, 2018. 3

[28] 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 *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2, 4, 7, 8

[29] Rakshith Shetty, Bernt Schiele, and Mario Fritz. Not using the car to see the sidewalk–quantifying and controlling the effects of context in classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8218–8226, 2019. 3

[30] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image

classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. 7

[31] Dawn Song, Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramèr, Atul Prakash, and Tadayoshi Kohno. Physical adversarial examples for object detectors. In *12th USENIX Workshop on Offensive Technologies (WOOT 18)*, Baltimore, MD, 2018. USENIX Association. 2

[32] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 2019. 2

[33] Akshayvarun Subramanya, Vipin Pillai, and Hamed Pirsiavash. Fooling network interpretation in image classification. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 3

[34] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *CoRR*, abs/1312.6199, 2013. 2

[35] Simen Thys, Wiebe Van Ranst, and Toon Goedemé. Fooling automated surveillance cameras: adversarial patches to attack person detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 2

[36] Antonio Torralba, Kevin P Murphy, William T Freeman, Mark A Rubin, et al. Context-based vision system for place and object recognition. In *ICCV*, volume 3, pages 273–280, 2003. 1, 3

[37] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan L. Yuille. Adversarial examples for semantic segmentation and object detection. *CoRR*, abs/1703.08603, 2017. 2

[38] Michal Zajkac, Konrad Zolna, Negar Rostamzadeh, and Pedro O. Pinheiro. Adversarial framing for image and video classification. *CoRR*, abs/1812.04599, 2018. 2

[39] B. Zhou, A. Khosla, Lapedriza. A., A. Oliva, and A. Torralba. Learning Deep Features for Discriminative Localization. *CVPR*, 2016. 7

[40] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. *CoRR*, abs/1412.6856, 2015. 3