Defending Object Detection Networks Against Adversarial Patch Attacks

Thomas Gittings[1], Steve Schneider[2], John Collomosse[1,3]
[1]CVSSP, University of Surrey – Guildford, UK
[2]SCCS, University of Surrey – Guildford, UK
[3]Adobe Research, Creative Intelligence Lab – San Jose, CA.
{t.gittings,s.schneider,j.collomosse}@surrey.ac.uk

Abstract

We present a technique for defending object detection networks against adversarial patch attacks (APAs). APAs introduce carefully crafted overt regions into an image in order to fool the network to create false detections. We leverage adversarial training via a conditional Generative Adversarial Network (GAN) that seeks to produce effective attacks on the object detector whilst simultaneously training the detector to resist those attacks. We report experiments with several common detection networks (Faster/Mask R-CNN and RetinaNet). We show our training-time defence offers resilience against our GAN generated APAs that also translates to other unseen APAs targeting object detectors.

1. Introduction

Object detection is a fundamental computer vision capability underpinning applications including robotics, security, and content analytics. Contemporary object detection methods are enabled via convolutional neural networks (CNNs) and so are prone to adversarial attacks; minor modifications to images at inference time that induce a significant change in the network prediction [26, 9]. Adversarial attacks may be covert, via imperceptible changes distributed across an image [9, 20], or overt via *adversarial patch attacks* (APAs) that introduce visible regions or 'stickers' [3, 6, 7]. APAs have recently been developed for object detection, and thus there is an emerging threat to autonomous systems relying upon visual sensing.

This paper contributes the first training-time defence against APAs that target object detection networks. APAs targetting object detection have been sparsely researched and, consequently, few defences exist. Our core technical contribution is to harness adversarial training to improve the resilience of object detection models at training time. Such training need not be applied from scratch, enabling pre-trained models to be fine-tuned via our method in order to confer protection against APAs.

Our training time defence utilises a conditional Generative Adversarial Network (GAN) that synthesizes patches to attack the detector, whilst simultaneously improving the resilience of that network against those patches. The GAN synthesizes patches capable of hallucinating non-existent objects. We show our framework to confer protection against APAs to the network that generalizes beyond our own APAs. We apply our defence to three common object detection networks (Faster/Mask R-CNN and RetinaNet).

Figure 1. Defending a segmetation network (Mask R-CNN) against an adversarial patch attack introducing a spurious person detection. An undefended Mask R-CNN (top) detects the patch as 'person', whereas our defended Mask R-CNN (bottom) ignores the patch and predicts the same as the ground truth (inset).

2. Related Work

Adversarial examples were introduced by Szegedy *et al*., who used box-constrained L-BFGS optimisation to find the smallest possible perturbation that could induce misclassi-

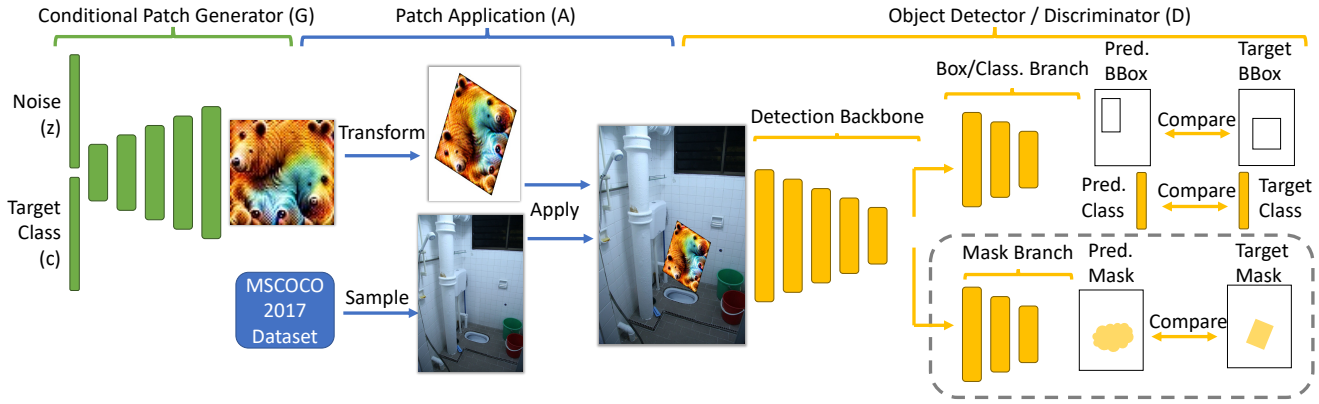

Figure 2. Proposed architecture for defending Faster/Mask R-CNN models against adversarial patches. The conditional patch generator G synthesises patches targeting all 80 MSCOCO classes. We alternate training of D and G to encourage the model to learn resilience to unseen patch attacks.

fication of an image [26]. The approach was later refined [4, 9] using SGD instead of L-BFGS to reduce computational overhead without compromising the efficacy of the adversarial examples produced.

Adversarial Patch Attacks (APAs) were introduced by Brown *et al*., who restricted the perturbation to a small region of the image but allowed it to be arbitrary in magnitude [3]. This allowed for an attack that would be robust to affine transformations and could be printed and used in the real world. Gittings *et al*. added a Deep Image Prior (DIP) as a form of regularisation, constraining the generated images to appear closer to natural images [7].

Adversarial Patches for Object Detectors were first explored by Liu *et al*., who created a patch that could cause the detector to ignore all objects in the image, limited by needing to be inserted digitally into the image in a specific place [17]. Chen *et al*. and Eykholt *et al*. independently created adversarial patches for object detection using stop signs [5, 25]. Thys *et al*. created a physical attack causing the disappearance of people when a patch was applied on them [27]. Braunegg *et al*. introduced an attack which attempts to create a new object at the patch location [2].

Defences Against Adversarial Images. The first defence against adversarial attacks was proposed by Szegedy *et al*. alongside the attacks themselves [26]. This was *adversarial training*, where adversarial examples are created during the training of the model and then included as part of the training data. When this method was originally proposed it was impractical, since there was no fast way to produce adversarial examples. This problem was resolved by the introduction of FGSM by Goodfellow *et al*. [9], and the method developed further by others [18, 24, 14]. Other defences use GANs or autoencoders to remove adversarial perturbations at inference time, by projecting the input onto a manifold of natural images [19, 23, 13].

Defences Against Adversarial Patches (APAs). Early APA defences made use of inference-time filtering to try and remove patches or otherwise modify them to make them less effective [10, 21, 1]. All these inference-time methods significantly degrade the classification performance on clean images in exchange for providing protection against adversarial patches. Braunegg *et al*. trained a detector to recognise adversarial patches as a specific class, along with MSCOCO object classes, by creating a dataset of patches in advance [2]. This is not true adversarial training, since the attack is not updated during the training process. Gittings *et al*. [8] applied adversarial training to patches via Vax-A-Net; using a conditional patch generator to allow the patches to be updated during training.

3. Method

In this work we study APAs on detection networks Faster-RCNN and RetinaNet, as well as the Mask-RCNN segmentation network. The attacks we consider are based on the APRICOT methodology of Braunegg *et al*. [2]. The goal of these attacks is to cause the detector to identify an object of some target class c in the location of the patch, when no such object is present in the image. Our aim is to defend the networks against these attacks at training time. Our training architecture (Figure 2) is inspired by Vax-A-Net [8]; an adversarial training method used to mitigate APAs on image classification networks. The architecture synthesises the patches using a conditional generative network (G), and applies an adversarial training process to update the generator while simultaneously training the detector (D) to build resistance to the patches.

3.1. Adversarial Patches for Object Detection

An adversarial patch for an object detection or semantic segmentation network D is an image p which is designed to cause D to return spurious detections around p when it is applied on top of any legitimate image x. Formally we define $A(p, x, l, t)$ to be to image x on which the patch p has been applied at location l with affine transformation t.

During training and testing we sample l randomly from

all possible patch locations and for t we randomly scale the patch to take up between 5% and 25% of the area of the image and rotate it randomly up to 20 degrees in either direction. During the training process the image x is sampled from the full MSCOCO training set, in order to produce general patches which are effective on any image.

Loss Function. The loss function for the patches is

$$L_G = L(A(p, x, l, t), C(l, t)) + \text{Print}(p) + \text{TV}(p), \quad (1)$$

where L is the standard loss function for training the detector in question (*i.e*. Faster/Mask R-CNN or RetinaNet), Print gives a non-printability score [27], which favours colours in the patch which can be easily represented on a CMYK printer, and TV is the total variation, also included to encourage better printability. $C(l, t, c)$ provides the 'ground truth' for the purpose of patch training, *i.e*. it returns a bounding box and mask for the adversarial patch applied to the image.

Patch Training. To provide a fair test of our methodology against the broadest range of APAs we implement two versions of the attack. The first (A-ADS) is the method of Braunegg *et al*. [2], which optimises the pixels of the adversarial patch directly. The other (A-DIP) is inspired by Gittings *et al*. [7], which uses a deep image prior to regularise the generation of the patch and provide a different appearance that will more thoroughly test our defences. Both methods produce adversarial patches of size 300×300.

3.2. A Training Time Defence for Object Detection

Patch Generator. The conditional patch generator G takes as its inputs an $N \times 100$ noise vector z and an $N \times 80$ class vector c, where N is the batch size. It uses five up-convolutional layers to produce patches of size 64×64 targeting the classes in c.

Discriminator/Detector. The loss function for D is defined as follows:

$$L_D = L(x, y) + L(A(G(z, c), x, l, t), y), \quad (2)$$

where y is the ground truth. The first term tries to ensure that clean images are correctly classified and the second is aiming for images with adversarial patches to be correctly classified.

Training Methodology Our network is trained in a similar manner to a Generative Adversarial Network (GAN). The object detection or segmentation network that we are defending takes the place of the discriminator in the GAN architecture. Instead of the discriminator acting as a tool to make the generator better, we are using the generator in order to produce a more effective discriminator. When training the networks we train alternately the discriminator and generator at each iteration, the same as for a normal GAN.

The generator is pre-trained for 270,000 epochs prior to training the discriminator. After this the generator and discriminator are trained together for another 270,000 epochs to provide protection against adversarial patches. Both the generator and discriminator use Adam optimisers, with a learning rate of 0.0001.

Method	Undefended	D-ROD (ours)
FRCNN R-50	41.79	41.38
FRCNN X-101	45.26	44.44
RN R-50	40.64	40.77
MRCNN R-50 BB	42.58	42.24
MRCNN R-50 Seg	41.70	41.30
MRCNN X-101 BB	45.70	45.54
MRCNN X-101 Seg	44.00	43.77

Table 1. Control: Mean Average Precision (mAP) of models on clean MSCOCO test images (*i.e*. without any adversarial patch).

4. Experiments and Discussion

We evaluate the efficacy of our training-time defence for Faster R-CNN (FRCNN) [22], RetinaNet (RN) [15] and Mask R-CNN (MRCNN) [11]. For Faster R-CNN and Mask R-CNN we test both ResNet-50 (R-50) [12] and ResNeXt-101 (X-101) [28] backbones, and for RetinaNet we test only the ResNet-50 backbone.

4.1. Datasets and Metrics

We evaluate using the MSCOCO 2017 dataset [16]. All our defended networks begin with networks pre-trained on MSCOCO, and are finetuned on the full MSCOCO 2017 training set, with patches generated to attack all 80 COCO classes. To evaluate our defences, we generate patches to attack the subset of 10 COCO classes picked by Braunegg *et al*. [2]. We report the Mean Average Precision (mAP) of the detectors, averaged across the set of 10 test classes. For Mask R-CNN networks we report the mAP for both detection (BB) and segmentation (Seg) tasks.

4.2. Our Defence vs Attacks

We evaluate the efficacy of our defence by subjecting it to two adversarial patch attacks, described in Section 3.1. We include both of these attacks because the DIP regularisation introduces substantially different textures in the patch when compared to the unregularised attack. We refer to our defended network as D-ROD, for 'Robust Object Detector'.

We test our defended network with attacks trained in two different ways. The first is a white box (WB) attack, in which the patches are trained on the final defended network. The second is a form of grey box (GB) attack, in which we use the same patches that are used for the undefended network, *i.e*. they are trained on the network with the publicly available weights before any adversarial training is applied. We also include a 'noise' patch for comparison; this is a 300×300 patch filled with uniform random noise on the interval $[0, 1)$, which provides a baseline for an optimal defence, since it is occluding the image in the same way as the adversarial patches but without any adversarial component.

Architecture	Defence	A-ADS	A-DIP	Noise
FRCNN R-50	Undefended	3.86	4.75	35.11
	D-ROD	33.06	9.71	36.39
FRCNN X-101	Undefended	4.49	12.74	37.60
	D-ROD	36.63	23.32	38.86
RN R-50	Undefended	3.52	3.81	33.48
	D-ROD	30.94	14.98	33.93
MRCNN R-50 (BB)	Undefended	4.00	5.39	35.64
	D-ROD	28.27	14.34	37.14
MRCNN R-50 (Seg)	Undefended	3.44	4.75	33.48
	D-ROD	27.11	13.26	34.99
MRCNN X-101 (BB)	Undefended	4.36	7.72	38.76
	D-ROD	40.55	20.39	40.38
MRCNN X-101 (Seg)	Undefended	3.75	6.93	35.88
	D-ROD	38.66	19.10	37.97

Table 2. mAP of models with grey box (GB) patches applied, *i.e.* patches trained on the original undefended network.

Architecture	Defence	A-ADS	A-DIP	Noise
FRCNN R-50	Undefended	3.86	4.75	35.11
	D-ROD	5.63	5.11	36.39
FRCNN X-101	Undefended	4.49	12.74	37.60
	D-ROD	7.34	19.64	38.86
RN R-50	Undefended	3.52	3.81	33.48
	D-ROD	4.43	4.86	33.93
MRCNN R-50 (BB)	Undefended	4.00	5.39	35.64
	D-ROD	7.91	9.19	37.14
MRCNN R-50 (Seg)	Undefended	3.44	4.75	33.48
	D-ROD	7.11	8.40	34.99
MRCNN X-101 (BB)	Undefended	4.36	7.72	38.76
	D-ROD	14.89	17.55	40.38
MRCNN X-101 (Seg)	Undefended	3.75	6.93	35.88
	D-ROD	13.70	16.11	37.97

Table 3. mAP of models with white box (WB) attacks, *i.e.* adversarial patches trained directly on the network they are applied to.

Control experiment. Table 1 shows the performance of the defended networks compared to the undefended network on clean images, *i.e.* MSCOCO images without any adversarial patch applied. In all cases the mAP for the defended network is within 1% of the undefended network, which demonstrates that the defence does not impact the performance on clean images.

Grey box (GB) attacks. In Table 2 we examine how the networks perform against attacks generated only on the undefedned network. The undefended network performs very poorly on these attacks, with the mAP in most cases being reduced to less than 5%. The noise patches reduce mAP for all the networks by 5-9% suggesting that this is the impact of occlusion, which is much less significant than the adversarial component of the patch. The defended networks do marginally improve the performance on these noise patches, suggesting that the network can learn to do a slightly better job of ignoring these occlusions if trained correctly. In the case of all the networks, apart from MRCNN R-50, the performance of against A-ADS is within 4% of the performance on noise patches, demonstrating good protection against this type of adversarial attack, despite it not being explicitly included during training. In the case of MRCNN R-50, the mAP on A-ADS is still within 8% of the mAP for noise patches, indicating quite good but imperfect protection. For A-DIP the mAP ranges from 9% to 24%. This indicates that the network is not able to completely defend against this style of attack, likely because the patches that the generator is able to produce are not sufficiently similar to the A-DIP patches, but even reasonable protection is afforded by our defence.

White box (WB) attacks. Results for fully white box attacks appear in Table 3. In the case of A-DIP on FRCNN X-101 and both attacks on MRCNN X-101, the mAP on the defended network is over 13%, which suggests that the defence is having a significant impact on the effectiveness of the attacks and providing substantial but not complete protection in these cases. For the remaining networks and attacks the mAPs are all improved over the undefended networks, but below 10%. This indicates that these networks are not defended very well against fully white box attacks. The reason for this poor performance on white box attacks compared to grey box attacks is probably that the generator was not able to synthesise enough attacks for the network to defend against to ensure complete protection against any possible attack, which could be caused by insufficient training time, incorrect choice of optimiser, sub-optimal training schedule, or others.

5. Conclusion and Future Work

In this paper we proposed the first training-time defence against APAs applied to object detection networks. Inspired by the Vax-A-Net architecture [8] for adversarial defence of image classification networks, we explored use of a conditional patch generator to synthesise patches, which the networks learn to defend against with an adversarial training methodology. Our experiments showed that the method produced good protection against unseen attacks created using the original network, and partial protection against those produced directly on the defended network. These results show good promise for this form of adversarial training against APAs in the novel domain of object detection networks.

The primary direction for future work is to modify the training process to improve the performance against white box APAs, to a level comparable with that demonstrated on grey box attacks. In scenarios an attacker may have access to a deployed network with weights for training purposes, motivating resilience to such attacks.

References

[1] Alaa E Abdel-Hakim. Ally patches for spoliation of adversarial patches. *Journal of Big Data*, 6(1):1–14, 2019. 2

[2] Anneliese Braunegg, Amartya Chakraborty, Michael Krumdick, Nicole Lape, Sara Leary, Keith Manville, Elizabeth Merkhofer, Laura Strickhart, and Matthew Walmer. Apricot: A dataset of physical adversarial attacks on object detection. In *Proc. ECCV*, 2020. 2, 3

[3] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017. 1, 2

[4] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *Proc. IEEE Symposium on Security and Privacy*, 2017. 2

[5] Shang-Tse Chen, Cory Cornelius, Jason Martin, and Duen Horng Chau. Shapeshifter: Robust physical adversarial attack on faster r-cnn object detector. *Proc. Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2018. 2

[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 *Proc. CVPR*, 2018. 1

[7] Thomas Gittings, Steve Schneider, and John Collomosse. Robust synthesis of adversarial visual examples using a deep image prior. In *Proc. BMVC*, 2019. 1, 2, 3

[8] Thomas Gittings, Steve Schneider, and John Collomosse. Vax-a-net: Training-time defence against adversarial patch attacks. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 2, 4

[9] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 1, 2

[10] Jamie Hayes. On visible adversarial perturbations & digital watermarking. In *Proc. CVPR Workshops*, 2018. 2

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proc. ICCV*, 2017. 3

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016. 3

[13] Andrew Ilyas, Ajil Jalal, Eirini Asteri, Constantinos Daskalakis, and Alexandros G Dimakis. The robust manifold defense: Adversarial training using generative models. *arXiv preprint arXiv:1712.09196*, 2017. 2

[14] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016. 2

[15] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proc. ICCV*, 2017. 3

[16] 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 *Proc. ECCV*, 2014. 3

[17] Xin Liu, Huanrui Yang, Ziwei Liu, Linghao Song, Hai Li, and Yiran Chen. Dpatch: An adversarial patch attack on object detectors. *arXiv preprint arXiv:1806.02299*, 2018. 2

[18] Chunchuan Lyu, Kaizhu Huang, and Hai-Ning Liang. A unified gradient regularization family for adversarial examples. In *Proc. International Conference on Data Mining*, 2015. 2

[19] Dongyu Meng and Hao Chen. MagNet. In *Proc. Conference on Computer and Communications Security*, 2017. 2

[20] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proc. CVPR*, 2016. 1

[21] Muzammal Naseer, Salman Khan, and Fatih Porikli. Local gradients smoothing: Defense against localized adversarial attacks. In *Proc. WACV*, 2019. 2

[22] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proc. NIPS*, 2015. 3

[23] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-GAN: Protecting classifiers against adversarial attacks using generative models. In *Proc. ICLR*, 2018. 2

[24] Uri Shaham, Yutaro Yamada, and Sahand Negahban. Understanding adversarial training: Increasing local stability of supervised models through robust optimization. *Neurocomputing*, 307:195–204, 2018. 2

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

[26] 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, 2

[27] Simen Thys, Wiebe Van Ranst, and Toon Goedemé. Fooling automated surveillance cameras: adversarial patches to attack person detection. In *Proc. CVPR Workshops*, 2019. 2, 3

[28] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proc. CVPR*, 2017. 3