Evaluating the Use of Fast Adversarial Training in Defending Against Adversarial Patch Attacks

Pedro Maia de Sampaio Ferraz - 21022845

School of Computer Science

University of Waterloo

pmaiades@uwaterloo.ca

Abstract

Adversarial patch attacks are a type of adversarial attack on deep neural networks (DNNs) that involve creating a small patch that can be physically applied to an object in the real world, causing the DNN to misclassify the object when it is subsequently captured in an image. These attacks can be particularly effective because they do not require any modification of the DNN itself, and the patch can be easily created and applied by an attacker with minimal resources.

In this study, we investigated the use of Fast Adversarial Training (FAST-AT) [1] and Fast Bi-level AT (FAST-BAT) [2] to defend against adversarial patch attacks. Our results showed that models trained using these methods significantly increased their accuracy when subjected to adversarial patch attacks, indicating that they had gained robustness against these attacks despite not being explicitly trained to defend against them. These findings demonstrate the potential of FAST-AT and FAST-BAT to enhance the robustness of DNNs against a variety of adversarial attacks.

1 Introduction

Deep neural networks (DNN) are being used in an ever increasing amount of applications and are becoming ubiquitous in our lives. However, these models present a vulnerability to maliciously crafted inputs that may cause DNNs to make incorrect predictions - these inputs are called *adversarial examples*. By making specific subtle changes to the input, it is possible to fool the DNN into making a completely different (and wrong) prediction.

Therefore, in order to deploy a deep learning model in a real-world application such as an autonomous driving car, it is crucial to ensure that the model is robust to adversarial inputs. The process of training the model with the goal of being robust to adversarial attacks is called *adversarial training* (AT).

Many AT techniques have been developed to train models to be robust to maliciously crafted inputs. However, those techniques usually use gradient methods to construct imperceptible adversarial examples, and their effect to defend against adversarial patches is, to the best of our knowledge, poorly understood. In this study, we aim to investigate the efficacy of novel AT techniques in defending against adversarial patch attacks on DNNs.

2 Background study

Adversarial training (AT) is a method for training neural networks to be robust against adversarial attacks. One traditional approach to AT is the min-max robust optimization approach, which was

formalized by Madry et al. [3] using Projected Gradient Descent (PGD) to create the strongest adversarial attack utilizing first order information about the network. However, this approach is computationally expensive and difficult to scale due to the many steps needed to generate each adversarial example.

To address this issue, Wong et al.[1] and Andriushchenko and Flammarion[4] introduced FAST-AT and FAST-AT-GA, respectively. These algorithms use the Fast Gradient Sign Method (FGSM) to efficiently generate adversarial examples for training the model. This allows for the training of robust models at a faster rate while maintaining good accuracy and robustness.

However, Zhang et al. [2] showed that FAST-AT and FAST-AT-GA can have a lack of stability or a poor accuracy-robustness trade-off. To address these issues, they introduced Fast Bi-level AT (FAST-BAT), which uses a bi-level optimization approach to improve the stability and accuracy-robustness trade-off of the model.

Regarding adversarial patch attacks, Eykholt et al. [5] have shown that it is possible to perform adversarial attacks with physical objects to fool image classifiers and object detection models under a variety of real-world conditions, with problematic consequences such as creating a patch to make a stop sign undetectable for the model.

In addition, Thys et al. [6] have shown that these types of adversarial attacks can be made even to target classes with lots of intra-variety, such as persons, to hide people from surveillance cameras.

To defend against adversarial patch attacks, Rao et al. [7] proposed an adversarial patch training approach to achieve robustness against Location-Optimized Adversarial Patches. This approach involves training the model to be robust against patches that are optimized for a specific location on the input image. However, it is unknown whether this approach is effective in defending against a wider range of adversarial patch attacks.

The goal of this study is to assess the effectiveness of state-of-the-art adversarial training techniques in defending against adversarial patch attacks. These techniques have previously demonstrated success in resisting regular adversarial attacks, but it is important to investigate their ability to protect against other types of attacks as well. By examining the performance of these techniques against adversarial patch attacks, we aim to gain a deeper understanding of their capabilities and limitations in safeguarding deep neural networks against a wider range of adversarial attacks.

3 Generating adversarial patch attacks

This study focuses on universal and untargeted adversarial patches. Universal adversarial patches are fixed patches that can be applied to any image in a dataset and are not specific to a particular label or class. In addition, untargeted adversarial patches do not target a specific label or class but rather aim to cause the model to make any incorrect prediction.

In order to optimize such patches, we maximize the cross-entropy loss between adversarial patch output and the specific target label.

More formally, consider a classification task with C classes. Let $\{(x_i, y_i)\}_{i=1}^{N}$ be the training set, where $x_i \in [0,1]^{W \cdot H \cdot D}$ is the input image and $y_i \in \{0, 1, \ldots, C-1\}$ is the associated label, with W, H and D representing the image width, height and number of channels, respectively. Let $f(x; w)$ be the trained classifier parameterized by w that output the logits for any input image x_i.

Consider an adversarial patch p with dimensions $W_p \times H_p \times D$, where $W_p < W$ and $H_p < H$. This patch can be represented as a tensor in the interval $(0,1)^{W_p \cdot H_p \cdot D}$. A padding function g is used to add zeros to the patch such that it has dimensions $W \times H \times D$ and is represented as a vector in the interval $[0,1]^{W \cdot H \cdot D}$. To apply the adversarial patch p to an input image x and obtain the patched input $\tilde{x}$, we first compute the padded patch $p' = g(p)$ and then calculate the values for each entry $\tilde{x}_{ijk}$ using the following equation:

$$\tilde{x}_{ijk}(p) = \begin{cases} x_{ijk}, & \text{if } p'_{ijk} = 0 \\ p'_{ijk}, & \text{otherwise} \end{cases}$$

This equation substitutes the zero-values of the padded patch p' with the corresponding values in the input image x to produce the patched input $\tilde{x}$. In addition, to ensure that the padded values on the

adversarial patch are properly represented as padding during the training process, the patch values are clamped to the $(0.000001, 0.999999)$ interval. This helps maintain the integrity of the padding and ensures that it is not misinterpreted as part of the patch during the training process.

Finally, in order to train a universal and untargeted adversarial patch, we can solve the following problem through stochastic gradient ascent, where L is the cross-entropy loss:

$$\max_{p} \sum_{i=1}^{N} L\big(f(\tilde{x}_i(p); w), y_i\big)$$

4 Experiments and results

To evaluate the effectiveness of modern adversarial training techniques in resisting patch attacks, we trained the Pre-activation ResNet (PreAct-ResNet) model using three different training procedures: standard Stochastic Gradient Descent, FAST-AT, and FAST-BAT. These models were trained on the CIFAR-10 dataset using open source code provided by Zhang et al. [2] available at `https://github.com/NormalUhr/FastBAT.git`. The adversarial patch training and evaluation code was implemented in Python using a Jupyter Notebook. The code was run on Google Colab with GPU runtime support.

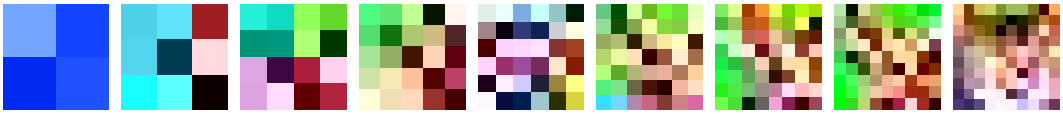
We then trained adversarial patches for each of the models using the method described in section 3. For patch training, we initialized the patches with fixed, constant values, resulting in a grey initial patch. We also tested two different approaches for placing the patches on the input image: one using fixed padding values to position the patch in the upper-left corner, and another using random padding values to position the patch anywhere on the input.

Placing the patch in the upper-left corner ensures that the patch does not obscure important features of the image. On the other hand, using random padding values allows the patch to be placed anywhere on the input, ensuring that the patch does not affect only a subset of the network's weights. This allows us to explore the potential impacts of different initialization and positioning methods on the effectiveness of the adversarial patches.

4.1 Fixed location patch results

By training square patches of size $W \times W \times D$ in the fixed upper-left position for various values of W, we obtain different patches. The patches obtained for the models trained with regular SGD, FAST-AT, and FAST-BAT can be seen in Figure 1, Figure 2, and Figure 3, respectively.

It is worth noting that the patches trained for the model trained with regular SGD appear to have more detail and information than those trained for the robust models. This suggests that the patch optimization routine was not able to identify a pattern that successfully exploits the weights of the robust models, in contrast to what happened with the SGD model. This visual difference may indicate that the adversarial training techniques used for the robust models were more effective at defending against patch attacks.



(a) W=2 (b) W=3 (c) W=4 (d) W=5 (e) W=6 (f) W=7 (g) W=8 (h) W=9 (i) W=10

Figure 1: Fixed location patches generated for the model trained by regular SGD.

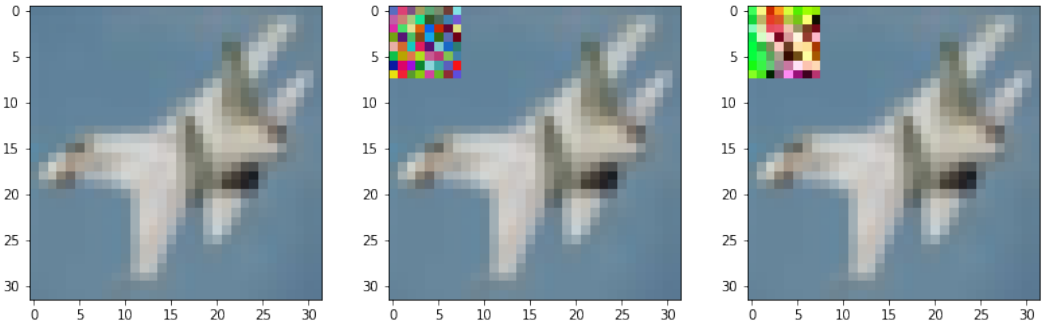
(a) W=2 (b) W=3 (c) W=4 (d) W=5 (e) W=6 (f) W=7 (g) W=8 (h) W=9 (i) W=10

Figure 2: Fixed location patches generated for the model trained by FAST-AT.

(a) W=2 (b) W=3 (c) W=4 (d) W=5 (e) W=6 (f) W=7 (g) W=8 (h) W=9 (i) W=10

Figure 3: Fixed location patches generated for the model trained by FAST-BAT.

In Figure 4, the regular SGD-trained model correctly classifies the original image and the image with the randomly applied patch, but incorrectly classifies the image with the adversarial patch as a deer. In contrast, when the same experiment is conducted using the robust models and their corresponding patches, the models are able to correctly classify the images even when an adversarial patch is present in the majority of cases.

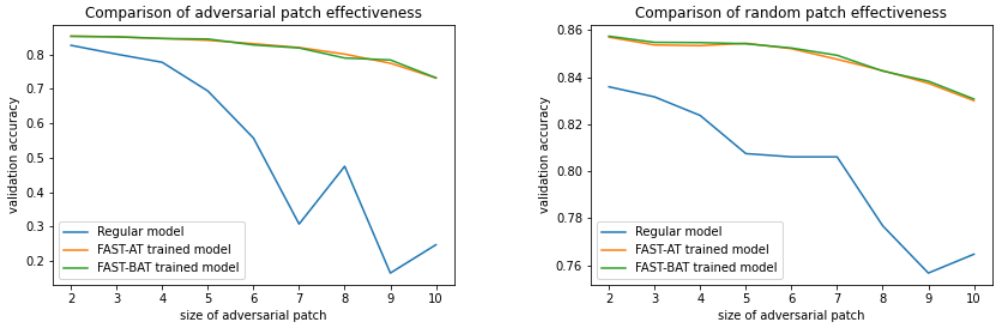


(a) Original image (airplane) (b) Random patch (airplane) (c) Adversarial patch (deer)

Figure 4: Adversarial patch attack (W=8) on regular SGD-trained model.

In Figure 5a, we can observe the effect of patch size on the model's ability to defend against fixed location adversarial patch attacks. As the size of the adversarial patch increases, the validation accuracy of the regular model decreases steadily. On the other hand, the robust models appear to be less affected by larger adversarial patches and maintain a relatively stable level of accuracy. This demonstrates the superior resilience of the robust models to patch attacks compared to the regular model. The numerical values for this figure can be found in Table 1.

In addition, Figure 5b suggests that the drop in accuracy seen in Figure 5a is primarily due to the adversarial nature of the attacks, rather than a loss of information caused by the patches. However, it is worth noting that the regular model's accuracy is still significantly more affected by the patch application, indicating that the robust model has a better understanding of the features associated with each class.



(a) Fixed location adversarial patches (b) Fixed location random patches

Figure 5: Effect of fixed location adversarial patch size on validation accuracy

Fixed location adversarial attack validation accuracy									
Patch size	2	3	4	5	6	7	8	9	10
Regular SGD model	0.826	0.800	0.777	0.693	0.556	0.3071	0.474	0.164	0.246
FAST-AT model	0.850	0.851	0.846	0.841	0.831	0.820	0.800	0.774	0.731
FAST-BAT model	0.850	0.851	0.846	0.844	0.827	0.818	0.789	0.783	0.732

Table 1: Values for Figure 5a.

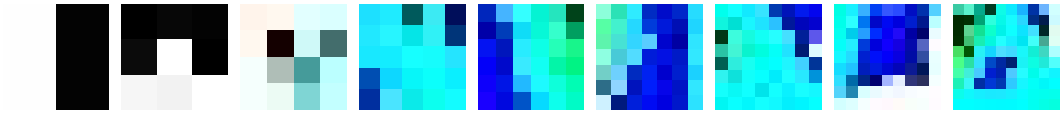
4.2 Random location patch results

Similarly to subsection 4.1, by training square patches of size $W \times W \times D$ in random positions for various values of W, we obtain different patches. The patches obtained for the models trained with SGD, FAST-AT, and FAST-BAT can be seen in Figure 6, Figure 7, and Figure 8, respectively.

Compared to the fixed-location patches, the randomly positioned patches of the robust models seem to contain more variety and information. However, it is still clear that the patches trained for the model trained with regular SGD contain more information overall, indicating that they were more successful at exploiting the model's vulnerabilities.

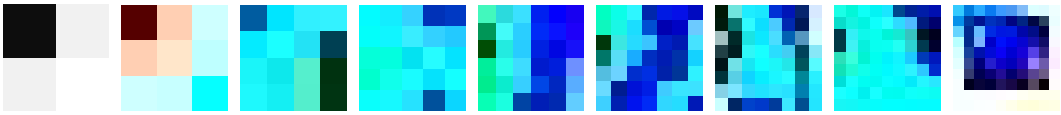
(a) W=2 (b) W=3 (c) W=4 (d) W=5 (e) W=6 (f) W=7 (g) W=8 (h) W=9 (i) W=10

Figure 6: Random location patches generated for the model trained by regular SGD.



(a) W=2 (b) W=3 (c) W=4 (d) W=5 (e) W=6 (f) W=7 (g) W=8 (h) W=9 (i) W=10

Figure 7: Random location patches generated for the model trained by FAST-AT.

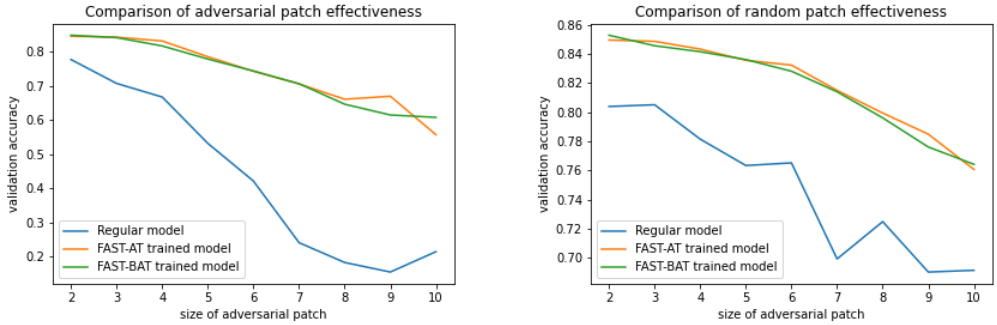


(a) W=2 (b) W=3 (c) W=4 (d) W=5 (e) W=6 (f) W=7 (g) W=8 (h) W=9 (i) W=10

Figure 8: Random location patches generated for the model trained by FAST-BAT.

In Figure 9a, we can see the impact of patch size on the model's resistance to random location adversarial patch attacks. Similarly to what was observed with the fixed location patches, the regular model is significantly more susceptible to adversarial attacks compared to the robust models. It is worth noting that the validation accuracy for random location patches is lower than that of fixed location patches, which is expected since randomly placed patches are more likely to obscure important image features. The numerical values for this figure can be found in Table 2.

Figure 9b shows that the randomly generated random location patches have a minimal impact on validation accuracy when compared to their adversarial counterparts. Additionally, just like with the fixed location patches, we see that when presented with randomly generated patches that occlude part of the image, the robust model's accuracy is also significantly higher than the regular model's accuracy. This further suggests that robust models are better able to understand the key features associated with each class, leading to improved performance in the face of disruptions or perturbations.



(a) Random location adversarial patches (b) Random location random patches

Figure 9: Effect of random location adversarial patch size on validation accuracy

Random location adversarial attack validation accuracy									
Patch size	2	3	4	5	6	7	8	9	10
Regular SGD model	0.777	0.707	0.667	0.532	0.422	0.241	0.183	0.155	0.215
FAST-AT model	0.845	0.842	0.831	0.785	0.742	0.705	0.660	0.670	0.557
FAST-BAT model	0.848	0.841	0.817	0.778	0.744	0.706	0.646	0.614	0.607

Table 2: Values for Figure 9a.

5 Conclusion

In this study, we explored the use of Fast Adversarial Training (FAST-AT) and Fast Bi-level AT (FAST-BAT) as a defense against adversarial patch attacks.

Our results showed that models trained using Fast Adversarial Training methods were significantly more accurate when subjected to adversarial patch attacks. This suggests that the model has gained robustness against patch attacks, despite not being explicitly trained to defend against them. These findings highlight the potential of Fast AT to enhance the robustness of deep neural networks against a variety of adversarial attacks.

In addition to their improved performance against adversarial patch attacks, this study also found that the robust models trained with FAST-AT and FAST-BAT demonstrated significantly higher accuracy than the regular models when presented with randomly generated patches that occluded part of an image. This suggests that the robust models had a better understanding of the key features associated with each class, enabling them to maintain a high level of accuracy even in the presence of disruptions or perturbations. These findings highlight the potential for FAST-AT and FAST-BAT to not only enhance the robustness of DNNs against adversarial attacks, but also improve their overall performance and resilience to various types of disruptions.

Future work could include evaluating the performance of FAST-AT and FAST-BAT against other types of adversarial attacks, examining the trade-offs involved in using these techniques in terms of accuracy on non-adversarial examples, and exploring the potential for combining them with other defenses against adversarial attacks. Research could also focus on the generalization properties of models trained with FAST-AT and FAST-BAT to unseen data and scenarios, as well as the effectiveness of these techniques in real-world applications.

Overall, this study has demonstrated the potential of FAST-AT and FAST-BAT to enhance the robustness of deep neural networks against adversarial patch attacks and other disruptions. Further research is needed to fully understand the capabilities and limitations of these techniques and to identify the best ways to apply them in practice.

References

[1] Eric Wong, Leslie Rice, and J. Zico Kolter. *Fast is better than free: Revisiting adversarial training*. 2020 (cit. on pp. 1, 2).

[2] Yihua Zhang, Guanhua Zhang, Prashant Khanduri, Mingyi Hong, Shiyu Chang, and Sijia Liu. *Revisiting and Advancing Fast Adversarial Training Through The Lens of Bi-Level Optimization*. 2021 (cit. on pp. 1–3).

[3] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. *Towards Deep Learning Models Resistant to Adversarial Attacks*. 2017 (cit. on p. 2).

[4] Maksym Andriushchenko and Nicolas Flammarion. *Understanding and Improving Fast Adversarial Training*. 2020 (cit. on p. 2).

[5] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramèr, Atul Prakash, Tadayoshi Kohno, and Dawn Song. "Physical Adversarial Examples for Object Detectors". *CoRR*, vol. abs/1807.07769 (2018). arXiv: 1807.07769 (cit. on p. 2).

[6] Simen Thys, Wiebe Van Ranst, and Toon Goedemé. "Fooling automated surveillance cameras: adversarial patches to attack person detection". *CoRR*, vol. abs/1904.08653 (2019). arXiv: 1904.08653 (cit. on p. 2).

[7] Sukrut Rao, David Stutz, and Bernt Schiele. "Adversarial Training Against Location-Optimized Adversarial Patches". In: *Computer Vision – ECCV 2020 Workshops*. Springer International Publishing, 2020, pp. 429–448 (cit. on p. 2).