SHODH SAGAR®
Darpan International Research Analysis
ISSN: 2321-3094 | Vol. 12 | Issue 3 | Jul-Sep 2024 | Peer Reviewed & Refereed

SURVEY OF ADVERSARIAL ATTACKS AND DEFENSE AGAINST ADVERSARIAL ATTACKS

Sanskar Agarwal*
Electronics and Communication Engineering, IIT Roorkee, Uttarakhand, India

Armaan Parreka
Electronics and Communication Engineering, IIT Roorkee, Uttarakhand, India

Vanshika Singh
Electronics and Communication Engineering, IIT Roorkee, Uttarakhand, India

Akshat Jain
Electronics and Communication Engineering, IIT Roorkee, Uttarakhand, India

DOI: http://doi.org/10.36676/dira.v12.i3.105

Accepted :12/09/2024 Published 18/09/2024 * Corresponding author

Abstract

In recent years, the fields of Artificial Intelligence (AI) and Deep learning (DL) techniques along with Neural Networks (NNs) have shown great progress and scope for future research. Along with all the developments comes the threats and security vulnerabilities to Neural Networks and AI models. A few fabricated inputs/samples can lead to deviations in the results of the models. Patch based Adversarial Attacks can change the output of a neural network to a completely different result just by making a few changes to the input of the neural network. These attacks employ a patch that is applied to the input image in order to cause the classifier to misclassify and make the incorrect prediction. The goal of this research is to develop effective defense strategies against these types of attacks and make the model/Neural Network more robust.

Keywords: Adversarial attacks, GAN(Generative adversarial networks), FGSM - FastGradient Sign method

Introduction

Deep learning is a branch of machine learning which uses multiple computational layers with high abstraction levels to learn from datasets to automate many real-life problems. It can have millions of parameters and has proved quite effective in solving the problems that prove hard for machine learning algorithms. As deep learning/machine learning algorithms are increasingly used in real-life applications, its flaws have become more and more evident. Szegedy et al. [1] discovered first that neural networks are susceptible to adversarial attacks. They work by changing a few pixel values in the input of the network which is imperceptible to the human eye but is able to trick the network into making the wrong decision/classification. Another type of attack works by applying a patch in the input image of the network and thus being able to trick the network into making the wrong prediction. There have been many types of research since which try to find and exploit such flaws in the neural networks and many countermeasures have also been presented regarding such issues. Here we will try to present some of such countermeasure techniques.

1. Adversarial Attacks

Adversarial attack can trick a model to give a false/wrong output. It can broadly be categorized into targeted and non-targeted attacks. In targeted attacks the aim is to change/modify the input orthe model in such a way that we can trick the model to give our intended classification instead of the correct one. While in untargeted attacks the aim is to simply make the model misclassify the input. They can of following types: Poisoning attacks- In this type of attack the training data set of the model is altered to influence the model, for example changing the tags of the dataset or so. Evasion attacks- These attacks are most frequently used in intrusion or malware scenarios which is the reason they are

the most researched and studied attacks. Here the attacker manipulates the data before it enters the network and fools the pre-trained classifiers to mislead the model. This is the most practical attack as it takes place during the deployment phase. Model extraction- In this the attacker aims to make a copy of a black box model by using its queries or to get the data set the model was trained on to manipulate the results. There are many adversarial examples (inputs that an attacker has designed to fool the model) that are popular these days. A few are listed below: FastGradient Sign method (FGSM)- This is a simple gradient based attack method that generates the adversarial examples with the minimum difference from the original image by slightly changing the pixels throughout the image. Deepfool attack- This attack aims to generate a sample which has the minimum euclidean distance from the original image and the perturbations are added to the image iteratively. Generative Adversarial Networks (GAN)- GANs have been used to generate adversarial attacks, where two neural networks compete with each other. Thereby one is acting as a generator, and the other behaves as the discriminator. The two networks play a zero-sum game, where the generator tries to produce samples that the discriminator will misclassify. Meanwhile, the. discriminator tries to distinguish real samples from ones created by the generator.

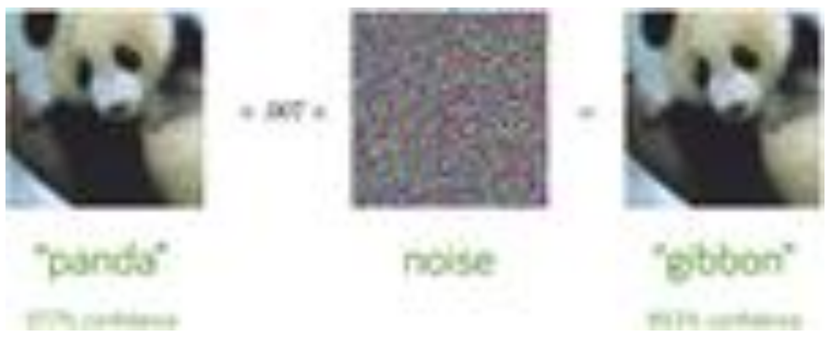

Figure 1: An example of FGSM attack

To understand how adversarial attacks work we need to see that in a basic ML/DL model the aim to minimize a loss function which is given by

$$\arg\min_{H} \sum_{x_i \in D} l(H(x_i), y_i) \tag{1}$$

While testing the error function is sum of all the loss functions which can given as,

$$\sum_{x_i \in T} l(H(x_i), y_i) \tag{2}$$

The aim of the adversarial attack is to change the input, x_i to $x_{i'}$ such that the network output H($x^{'}$) no longer belongs to prediction y_i but to new category $y_{i'}$ which may be targeted or non-targeted. As stated above it's clear that neural networks are susceptible to adversarial attacks. We are mainly interested in patch-based adversarial attacks on classification networks. We tried it on detection networks, eg - YOLO, as well but the intricacies of the detection network were too complex to try new techniques and computationally infeasible as well, so we switched to classification networks. Now the basic approach as to how to make a patch is to use a white image of the desired size (usually in pixels) and pass it through the neural network and change the pixel values of the patch till the desired misclassification is achieved. The image below shows the basic approach for patch formation. The best part about this technique is that you don't need to know the type of input to apply/generate the patch, you just need to know which network/architecture are we using and we can generate a patch from that. An example from [2] is given below.

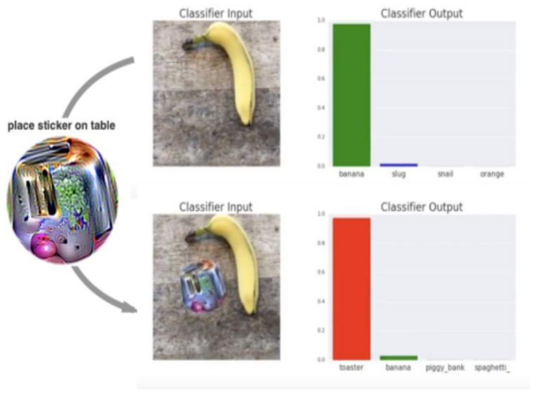

Figure 2: An adversarial attack on a neural network using a physical patch

The below picture shows the effect of the adversarial patch developed using the backpropagation method, the classification of banana changed to toaster just by applying the patch. Another example of the patch-based attack is present in [3] which is given below.

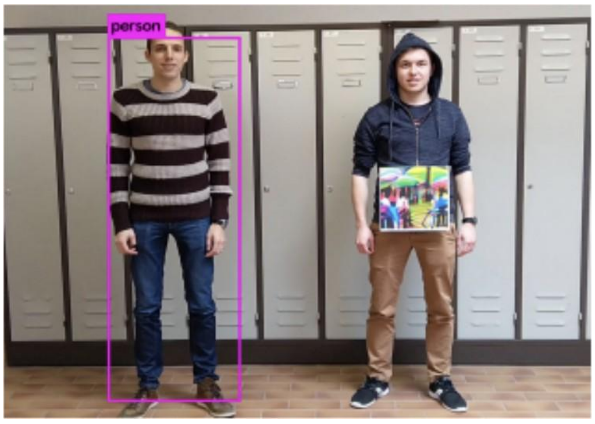

Figure 3: An illustration of the adversarial patch based attack

2. Defense Techniques and Implementation

First, we tried making GRAD-CAM to create class activation maps to gain insight on how the network works and progresses towards the result through various layers of the network. It also helps to gain insight into how the patch works and tricks the network into misclassifying the input image. An example is given below.

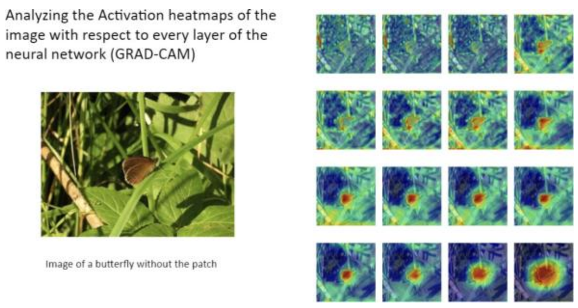
Analyzing the Activation heatmaps of the image with respect to every layer of the neural network (GRAD-CAM)
Image of a butterfly without the patch

Figure 4: Grad-CAM results with respect to various layers of Neural Network

Above figure 4 shows where the network focuses after each convolution layer in the InceptionV3 model. As we can see the model focuses on the butterfly as we proceed to the later layers and predicts based on those pixels. This gives us an intuition as to how the network focuses on the butterfly, to make more sense of this check Figure 5 where there is a comparison between the image with and without a patch in the last layer.

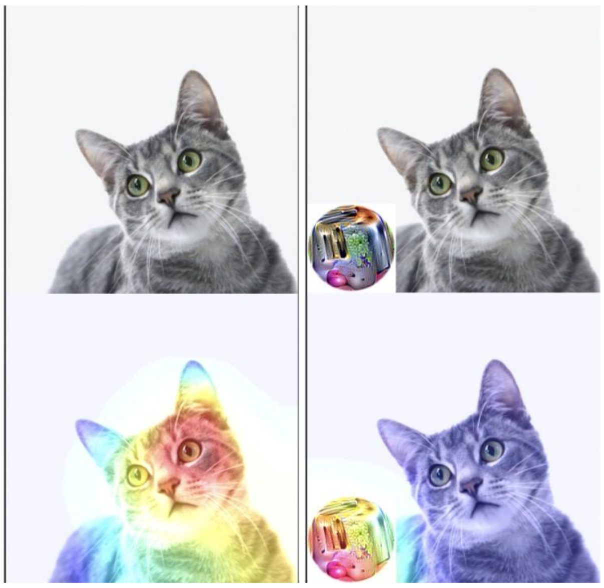

Figure 5: Grad-CAM heat map of images with and without the patch

The classification in the left image is 'tabby' (correct prediction) with a confidence score of 0.90 while in the image on the right(one with the patch) the image is classified as 'toaster' (incorrect) witha confidence score of 0.95. It can also be seen that in the image with patch the network focuses onthe patch instead of the cat. This was done to gain insight to the network and try a few basic ideas for defense, like freezing the weights of a particular layer, changing the activation from ReLu to ReLu6/tan h, quantization and binarization of weights, etc. These techniques were implemented and tested on a small scale but these showed low to none defensive effect against these kinds of attacks. The reason for failure in these techniques was that we were trying to change the network while the input was being attacked so the attack was not being properly mitigated.

Then, we carried out more experiments on single object single patch images, and various cases were examined. We performed image segmentation to create segmentation masks of images with and without the patch and GRAD-CAM to create class activation maps to gain insight on how the network works and progresses towards the result through various layers of the network. These techniques help us to gain insight into how the patch works and tricks the network into misclassifying the input image. The various cases that were examined are classified as follows: the classification model detects a patch or an object, and the image segmentation model detects both an object and a patch, or detects one of the two. So to summarize,

Classification (Gradcam)	**Segmentation**
Patch detected	Both patch, object detected
Object	Both
Patch	Object
Object	Object

Table 1: Comparison of Classification and Segmentation

The cases in which only the patch or none of the patch and object are detected in segmentation are extremely unlikely.

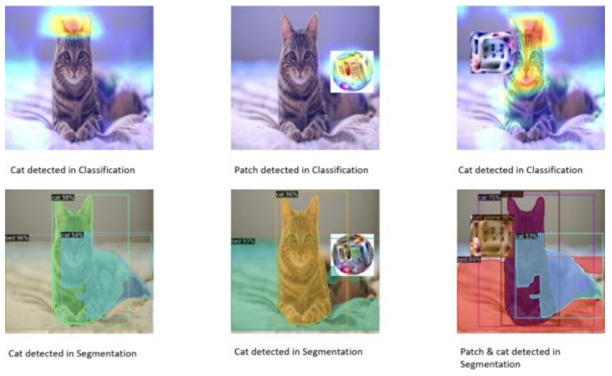
Cat detected in Classification
Patch detected in Classification
Cat detected in Classification
Cat detected in Segmentation
Cat detected in Segmentation
Patch & cat detected in Segmentation

Figure 6: Visualizations of Grad-cam and segmentation on patch and without patch images

Figure 6 depicts the Grad-Cam model and segmentation results on a cat image. The images in the first row are the results of the Grad-Cam model. The gradient is concentrated on the cat in the first image. The gradient is concentrated on the patch in the second image, and then on the cat again in the third image. The segmentation output of the preceding images is shown in the second row. In the second image, the patch is not detected in segmentation, whereas in the third image, both the object and the patch are detected in output. The second column in Figure 6 shows that the patch fools the model because the gradient in classification is concentrated on the patch but the segmentation is unableto detect it. As a result, we can't detect the patches using segmentation. To summarize, the results of these experiments were too varied and random for us to draw any conclusions, and we were unable to generalize these techniques as a proper defense technique. To get some solid results we moved onto two other techniques that gave satisfactory results. First is using an entropy filter and the other one is using an ensemble model to counter the effectiveness of the patch.

2.1. Entropy Filter

We attempted to segregate or filter out the patch from the image using an entropy filter because the patch usually has a very high entropy with respect to the image objects. We convert the image to grayscale first, then use the entropy filter to create masks, which we then apply to the original image to black out the patch. In a nutshell, the entropy function returns a value representing the level of complexity in a specific section of an image. The resulting values are, of course, constrained by the initial structuring element we selected. An example for various thresholds for the entropy filter applied on an image is given below. By choosing an appropriate threshold for the entropy we can filter out the patch and make the model resistant to adversarial attacks. Threshold: 0 (Since the threshold is zero all the information (entropy) is present in ¿ side)

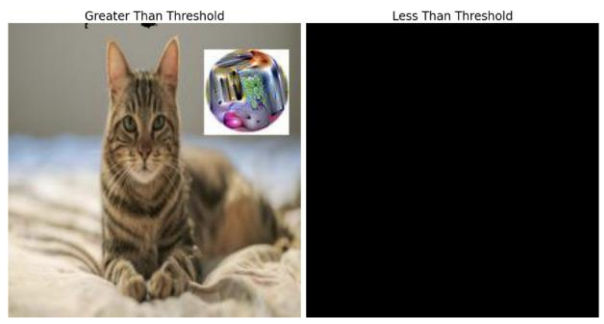

Figure 7: Coloured Image Segmented by entropy value (0)

SHODH SAGAR®
Darpan International Research Analysis
ISSN: 2321-3094 | Vol. 12 | Issue 3 | Jul-Sep 2024 | Peer Reviewed & Refereed

[h] Threshold: 0.7 (In this case the cat and the patch has high entropy so they are present in ¿ side)

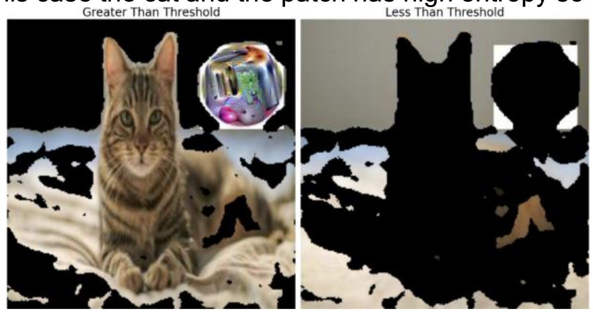

Figure 8: Coloured Image Segmented by entropy value (0.7)

Threshold: 0.92 (In this case only the patch is present in ¿ side and we are able to black out the patch in ¡ side)

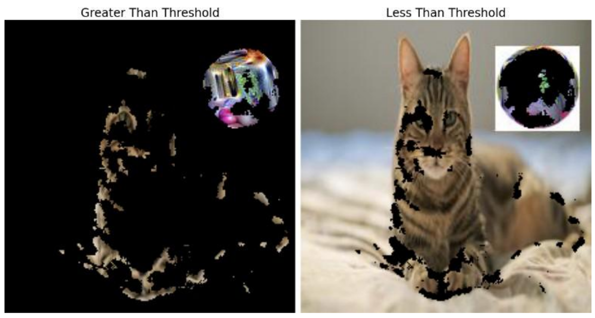

Figure 9: Coloured Image Segmented by entropy value (0.92)

Threshold: 1 (Since the threshold is one, nothing has greater entropy than 1 so everything is present in ¡ side)

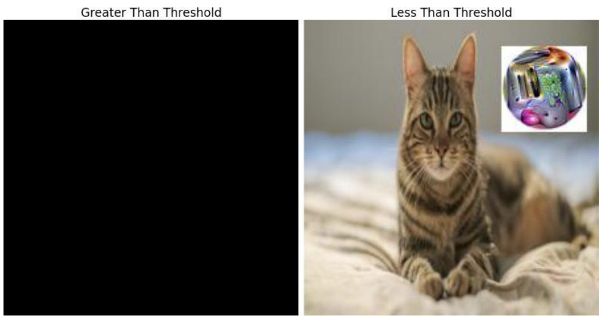

Figure 10: Coloured Image Segmented by entropy value (1)

2.2. Ensembling Technique

Ensemble learning refers to combining the predictions of multiple trained models instead of a single model. This technique reduces the variance of predictions and outperforms any single model used. The adversarial patch is created with the knowledge of the model weights and it will generally be only effective when the detecting/target model has similar weights as the one that was used to create the patch. This problem is resolved by adopting this ensembling technique. In this technique, multiple models vote on the same image. This is done in order to find the compromised model, for example: if there are 5 ensembling models and 4 of them detect an object with high confidence and one of them does not detect anything or detects the wrong class (patch), then we can conclude that model-5 is compromised and needs to either be replaced or removed.

3. Results and Conclusion

The first method, which employs an entropy filter, essentially filters or blacks out the patch from the image. Patches typically have a higher entropy (information) than other image objects. So, by determining an appropriate entropy threshold, we can prevent the model from being fooled by the patch.

We tested the entropy filter idea on 7000 images of imagenet validation set (7 images of each class). The patch we have taken is trained on VGG19. The patch size we have taken is 11 percent of the whole image. And the threshold for entropy is 0.965. The results are shown below: TABLE The first model (VGG19) shows the top-1 and top-5 accuracy of the images without the patch. We can see that this is the state of the art accuracy of classification. The second model shows the accuracies of the same model when tested on the images with the patch. We can see that by applying the patch the accuracy drops significantly. The third model uses an entropy filter before making the prediction and we can see the improvement in the accuracy from the second model.

The second technique based on ensembling uses multiple models and they vote on the same image for the purpose of majority voting. Five imagenet models (Inception, Resnet, vgg19, Xception, Mobilenet) are used to test on 7000 images of imagenet validation set (7 of each class). The patch we have taken is trained on VGG19. The patch size we have taken is 11 percent of the whole image. The results of ensembling technique are shown below:

Models	**Top-1 % Accuracy**	**Top-5 % Accuracy**
VGG19 Normal	0.7164286	0.9042857
VGG19 (Patch)	0.36957142	0.627
VGG19 (Entropy 0.955)	0.54742855	0.7714286

Table 2: Comparison of VGG19 Models

The above table shows the top-1 and top-5 accuracies of different models on 7000 images with the patch. We can see that VGG19 performs the worst, which is reasonable since the patch is trained on this network. ResNet and Inception also perform poorly which shows that they have similar weights to the VGG19. Xception and Mobilenet are the best models in this case. They must have weights that are very dissimilar than the weights used to train the patch.

4. References

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

[2] Adversarial Patch - Tom B. Brown, Dandelion Mané , Aurko Roy, Martín Abadi, Justin Gilmer

[3]Fooling automated surveillance cameras: adversarial patches to attack person detection -Simen Thys, Wiebe Van Ranst, Toon Goedeme

[4]I. Evtimov, K. Eykholt, E. Fernandes, T. Kohno, B. Li, A. Prakash, A. Rahmati, and D. Song. Robust physical-world attacks on deep learning models. arXiv preprint arXiv:1707.08945, 2017.

[5]A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial examples. arXiv preprint arXiv:1706.06083, 2017