Identifying FGSM Adversarial Attack Using Occlusion Heat Maps

Adithya Murali
Faculty of Mathematics
University of Waterloo
Waterloo, Canada
a9murali@uwaterloo.ca

Puneeth S.M. Saladi
Faculty of Mathematics
University of Waterloo
Waterloo, Canada
psmsalad@uwaterloo.ca

***Abstract*— Deep learning models are the state-of-the-art solution for many computer vision tasks such as image classification. Although these models deliver very high performance, they are still vulnerable to adversarial attacks which can have disastrous consequences in real-life scenarios. We propose a novel solution for identifying such attacks using a detector network trained on the occlusion heat maps of the original and adversarial images. Occlusion heat maps identify the most informative parts of an image according to the classifier confidence. We hypothesize that the heat map of an adversarial image will be noticeably different from that of the original class and can be identified by our detector network. Our solution achieves an overall accuracy of 75% on a 10 class subset of Imagenet dataset against the well known FGSM attack.**

***Index Terms*—Adversarial attack, Occlusion, FGSM**

I. Introduction

Convolutional Neural Networks (CNNs) are the state-of-the-art models when it comes to image classification. They have had a major impact in the field of computer vision since their inception in 2012. However, these powerful neural network models are still prone to adversarial attacks. Due to the pervasiveness of CNN's in diverse fields such as computer vision, health care, manufacturing, autonomous vehicles, and security, the vulnerability of these models to adversarial attack is a rapid growing threat. In many cases, we completely rely on the prediction provided by the neural models.

An adversarial attack is a slightly perturbed input to a neural network that makes the network output an incorrect prediction. In the case of CNNs, the adversarial inputs are slightly perturbed input images which are often indistinguishable from the original image by the human eye. Many such adversarial attacks have been developed by researchers. The Fast Gradient Sign Method (FGSM) is one such attack which was proposed by Goodfellow et al. [1] It relies on the fact that most neural networks have some linear component in them or designed to behave linearly and so perturbations of a linear model can make the network make wrong prediction. FGSM defines an adversarial example as follows -

$$\bar{x} = \boldsymbol{x} + \epsilon \operatorname{sign}\left(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y)\right) \tag{1}$$

Where $\bar{x}$ is the adversarial example, x is the original image, ϵ is a small constant, y is the target, θ is the hyperparameter associated with the model and $J(\boldsymbol{\theta}, \boldsymbol{x}, y)$ is the cost function given $\boldsymbol{\theta}$, $\boldsymbol{x}$ and y. Figure 1 shows a visualization of an FGSM attack.

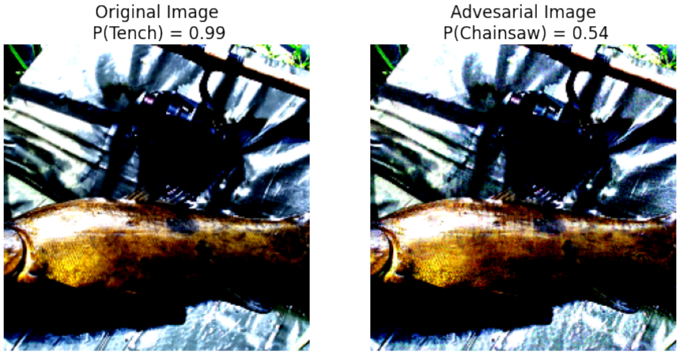

Fig. 1. The image on the left is correctly classified as a Trench. An FGSM attack was performed with an epsilon value of 0.1 and the image on the right was classfied as a chainsaw as a result of the FGSM attack.

Zeiler et al. [2] proposed an interesting concept of occlusion to find the parts of the image that contribute the most to a classification. We first find the classification of an image. Once we have a classification, we then block out a certain area of the image and find the probability that the image still belongs to the same class. This blocked out region can be moved around the original image to generate a heat map of the most important parts of the image. Figure 2 shows how occlusion can be used to identify the most important parts of an image for the purpose of classification.

In this paper, we attempt to use occlusion to generate heat maps of all correct classifications for each class. We then generate successful adversarial examples from our training image using the FGSM method and generate the heatmaps of the adversarial classification for each class. We now have heat maps of all correct classifications and adversarial classifications for each class. We train a detector neural network to classify the heatmaps generated to determine whether it belongs to a normal input image or an adversarial image. We hypothesize that this method can be an effective way to detect an adversarial attack on certain classes of images.

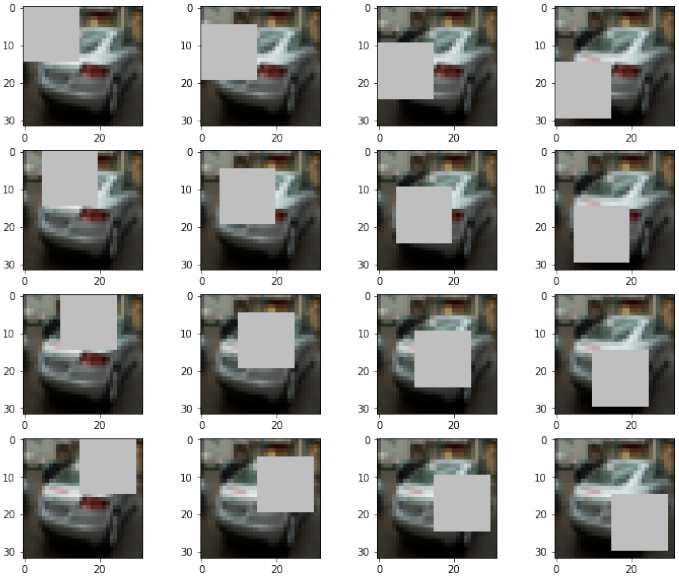

Fig. 2. We perform occlusion by blocking a part of the input image and moving that blocked area around the image. We calculate the probability of each image being a car to find the areas of importance.

The remainder of the paper is organized as follows: First we review some of the previous related works on adversarial defence in section II, then we present the details of our experimental setup in section III, report the performance of our proposed solution in section IV, discuss our results and share insights in section V, and finally conclude our paper outlining the possible future directions in section VI.

II. Related Works

Goodfellow et al. [1] conducted extensive experiments with adversarial examples and argued that the reason why adversarial examples break neural network models is due to the linearity, and used this hypothesis to propose the Fast Gradient Sign Method (FGSM) to generate adversarial examples. They also show that including these adversarial images during training has a regularization effect and helps in defending against such an attack.

Adversarial training is the most common approach to make neural networks robust to adversarial attacks in computer vision. This has also proven to be effective in other fields of machine learning such as Neural Machine Translation [3], Natural Language Understanding [4], and Automatic Speech Recognition [5].

An extension to the original adversarial training is ensemble adversarial training [6] , where multiple models are used to generate adversarial perturbations for input images. This decoupling of the models used for original classification and generating adversarial examples helps is generalization and results in better performance.

Label smoothing is another regularization technique which also helps in defending neural networks against adversarial attacks. The idea behind this technique is to reduce the overconfidence of the neural model by training it on soft targets rather than on one-hot labels. In these soft targets, the target probability of all wrong classes is set to a small pre-defined value rather than 0.

Meng et al. [7] proposes a novel solution: MagNet, which doesn't require retraining of an existing classfication model. They have 2 components - an autoencoder trained on the original images which acts as a detector and a reformer which tries to move the adversarial images towards the original image class. Our method is similar to theirs in the aspect that it can be applied to existing models without re-training them.

Another innovative solution proposed by Buckman et al. [8] changes the network architecture by discretizing the input images by replacing the pixel values with binary vectors using a thermometer encoding process. The idea behind this is that many adversarial attacks result in minor changes in the image pixel values which which can be offset by the process of discretization. This however is an expensive solution as it increases the model size significantly.

III. Experiments

A. Dataset

We use the CIFAR-10 dataset which contains 10 classes of images of size 32×32 pixels. The trainset contains $50,000$ images and the test set contains $10,000$ images. The 10 classes of images are:-

1) Airplane
2) Automobile
3) Bird
4) Cat
5) Deer
6) Dog
7) Frog
8) Horse
9) Ship
10) Truck

We also use a 10 class version of the Imagenet dataset called Imagenette[1] which contains images of size 224×224 pixels. We resampled this dataset so that the trainset is of size $12,394$ images and the testset is of size $1,000$ images. The 10 classes of images are:-

1) Tench
2) English Springer
3) Cassette Player
4) Chain Saw
5) Church
6) French Horn
7) Garbage Truck
8) Gas Pump
9) Golf Ball
10) Parachute

[1] https://github.com/fastai/imagenette

We test our hypothesis on these two datasets. It will be interesting to see how how our hypothesis holds against different image sizes and different image classes.

B. Training the first neural network

We build our first neural network to classify the input images which will later be subjected to a FGSM adversarial attack. For the Imagenette dataset we train a VGG11 network from scratch. The network used for CIFAR-10 and is very similar to the original VGG11 but differs in the linear layers towards the end to compensate for the different input image size.

TABLE I
THE NEURAL NETWORK ARCHITECTURE USED TO CLASSIFY IMAGENETTE DATASET

Layer	Output Shape	Parameters
Conv2d-1	[-1, 64, 224, 224]	1,792
BatchNorm2d-2	[-1, 64, 224, 224]	128
MaxPool2d-3	[-1, 64, 112, 112]	0
Conv2d-4	[-1, 128, 112, 112]	73,856
BatchNorm2d-5	[-1, 128, 112, 112]	256
MaxPool2d-6	[-1, 128, 56, 56]	0
Conv2d-7	[-1, 256, 56, 56]	295,168
BatchNorm2d-8	[-1, 256, 56, 56]	512
Conv2d-9	[-1, 256, 56, 56]	590,080
BatchNorm2d-10	[-1, 256, 56, 56]	512
MaxPool2d-11	[-1, 256, 28, 28]	0
Conv2d-12	[-1, 512, 28, 28]	1,180,160
BatchNorm2d-13	[-1, 512, 28, 28]	1,024
Conv2d-14	[-1, 512, 28, 28]	2,359,808
BatchNorm2d-15	[-1, 512, 28, 28]	1,024
MaxPool2d-16	[-1, 512, 14, 14]	0
Conv2d-17	[-1, 512, 14, 14]	2,359,808
BatchNorm2d-18	[-1, 512, 14, 14]	1,024
Conv2d-19	[-1, 512, 14, 14]	2,359,808
BatchNorm2d-20	[-1, 512, 14, 14]	1,024
MaxPool2d-21	[-1, 512, 7, 7]	0
Linear-22	[-1, 4096]	102,764,544
Dropout-23	[-1, 4096]	0
Linear-24	[-1, 4096]	16,781,312
Dropout-25	[-1, 4096]	0
Linear-26	[-1, 10]	40,970

The model was trained for 10 epochs on the CIFAR-10 dataset. Cross-Entropy was used to calculate the loss and Adam optimizer was used to optimize our CNN. We achieved a test accuracy of 78% on this dataset with our model.

The model was trained for 20 epochs on the Imagenette dataset. Cross-Entropy was used to calculate the loss and Adam optimizer was used to optimize our CNN. We achieved a test accuracy of 81% on this dataset with our model.

C. Generating the heatmaps

We obtain the heatmap as follows:-

1) Obtain the classification of an image by a forward pass through a neural network. Let this class be $\boldsymbol{y}$
2) Occlude the top left corner of the image with a square of side O_l.
3) Obtain the probability of the occluded image belonging to class $\boldsymbol{y}$. This is the value of one pixel in the heatmap.
4) Move by stride size O_s and perform the same until we reach the bottom right pixel. Direction of the movement of occlusion square is from top to bottom and left to right.

The width of the output heatmap is $\lceil \frac{(I_w - O_l)}{O_s} \rceil$ and the height of the output heatmap is $\lceil \frac{(I_h - O_l)}{O_s} \rceil$.

We obtain two sets of heatmaps for each dataset. The first set consists of heatmaps of correctly classified images from the training set for each class of image.

The second set consists of heatmaps for each successful adversarial image for each image class. It is important to note that the image class here refers to the output of the neural network after processing the adversarial image rather than the actual image class. A successful attack here refers to an image which is wrongly classified due to the FGSM attack which would otherwise have been correctly classified. The number of heatmaps obtained is shown in table II and III.

TABLE II
THE NUMBER OF CORRECT HEATMAPS AND WRONGLY CLASSIFIED HEATMAPS DUE TO FGSM ATTACK ON THE IMAGENETTE DATASET.

Class	Correct heatmaps	Adversarial heatmaps
Tench	1207	829
English Springer	1164	449
Cassette Player	1091	356
Chain Saw	919	1438
Church	1122	2260
French Horn	1063	309
Garbage Truck	1107	405
Gas Pump	1100	1143
Golf Ball	1154	1249
Parachute	1175	601

D. Training a detector neural network to classify heatmaps

We train a new neural network to process the heatmaps and detect a FGSM adversarial attack. This detector neural network consists of an input layer, 2 fully connected layers of 256 nodes each, 1 fully connected layer of 128 nodes, 1 fully connected layer of 64 nodes, 1 fully connected layer of 32 nodes and finally an output node of size 1.

For each class of image from each dataset, we have two classes of heatmaps which are the heatmaps of the images

TABLE III
THE NUMBER OF CORRECT HEATMAPS AND WRONGLY CLASSIFIED HEATMAPS DUE TO FGSM ATTACK ON THE CIFAR-10 DATASET.

Class	Correct heatmaps	Adversarial heatmaps
Airplane	4586	699
Automobile	4860	977
Bird	4430	6637
Cat	3669	2367
Deer	3963	5785
Dog	3630	337
Frog	4593	18211
Horse	4364	441
Ship	4562	1145
Truck	4641	1237

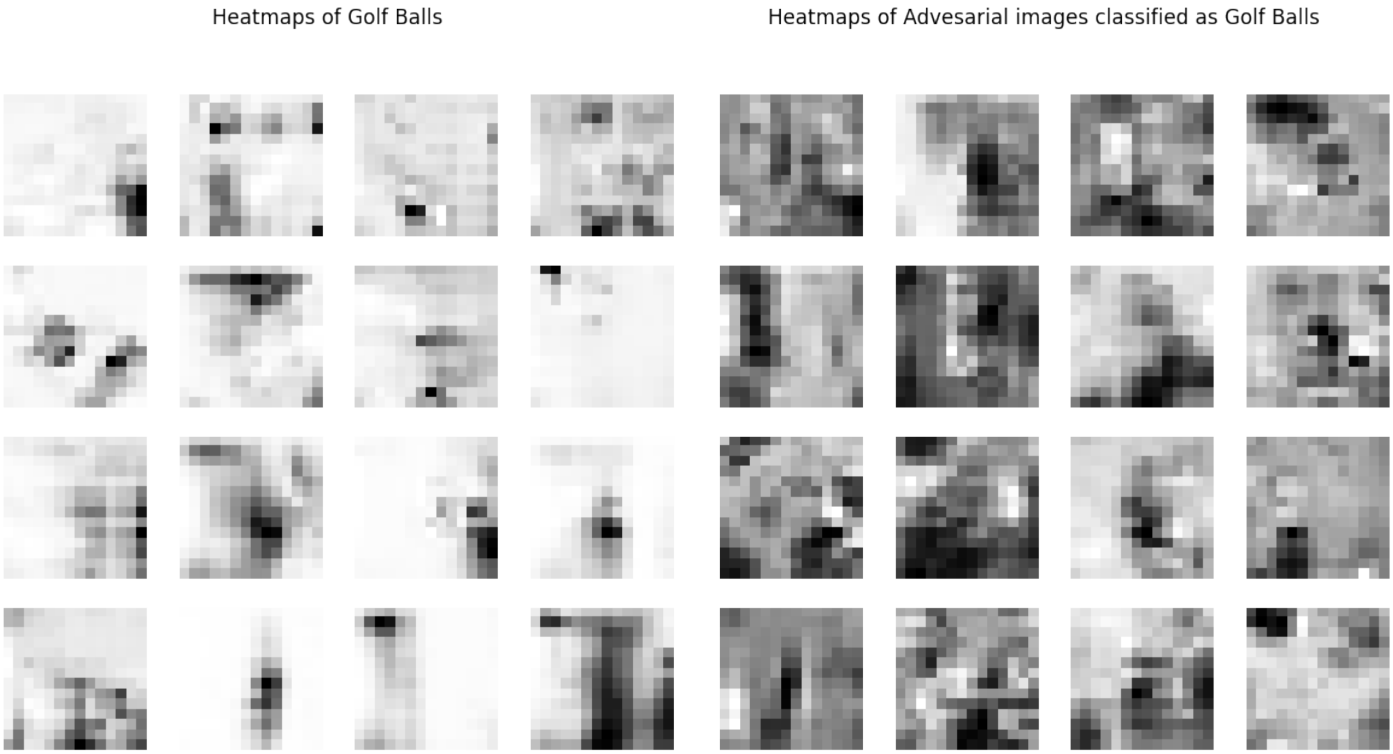

Fig. 3. The heatmaps on the left corresponds to golf balls from the Imagenette dataset correctly classified by our neural network. The heatmaps on the right correspond to images which were wrongly classified as golf balls after a FGSM attack. It is easy to see that the heatmaps on the right are of a darker shade.

which actually belong to the class and the heatmaps of the images which are adversarially classified to that particular class. This detector neural network is trained on this dataset to perform binary classification.

We have set $O_s = 14$ and $O_l = 32$ for the Imagenette dataset and $O_s = 5$ and $O_l = 15$ for the CIFAR-10 dataset. This gives a heatmap of size 4×4 for the CIFAR-10 images and a heatmap of size 14×14 for the Imagenette images.

IV. Evaluation

The performance of our detector neural network which detects a FGSM attack is given in IV and V. The neural network is trained on heatmaps from each image class on each dataset. It appears that it is easier to defend against a FGSM attack for certain classes of images than others. The Parachute class from the Imagenette dataset gives the best accuracy of 0.89. This intuitively means that we can detect whether the input image was an adversarial image or a normal image with an accuracy of 0.89 if the first neural network gives a classification of Parachute.

Table VI shows the overall performance of our detector neural network on both the datasets.

V. Discussion

The results of our experiment shows that this method works better on the Imagenette dataset than the CIFAR-10 dataset. This could be due to the larger image size of the Imagenette dataset. A larger image size can produce a larger heatmap which improves the performance of the detector neural network as it has more data to work with.

TABLE IV
Accuracy in detecting a FGSM adversarial attack on the CIFAR-10 dataset when an image is classified as one of the classes.

Class	Accuracy
Airplane	0.65
Automobile	0.63
Bird	0.67
Cat	0.69
Deer	0.70
Dog	0.50
Frog	0.56
Horse	0.56
Ship	0.73
Truck	0.68

Another interesting find is the varying performance between different image classes. The detector neural network which detects a FGSM attack is more effective on certain classes than others. As an example, The ship class in IV has a much higher accuracy than the other classes from the CIFAR-10 dataset. Similarly, Golf Ball and Parachute have a higher accuracy than the other classes from the Imagenette dataset as seen in V.

Figure 3 shows the reason behind why certain classes

TABLE V
ACCURACY IN DETECTING A FGSM ADVERSARIAL ATTACK ON THE 10 CLASS VERSION OF THE IMAGENETTE DATASET WHEN AN IMAGE IS CLASSIFIED AS ONE OF THE CLASSES.

Class	Accuracy
Tench	0.75
English Springer	0.73
Cassette Player	0.78
Chain Saw	0.73
Church	0.61
French Horn	0.70
Garbage Truck	0.75
Gas Pump	0.75
Golf Ball	0.81
Parachute	0.89

TABLE VI
OVERALL PERFORMANCE OF THE DETECTOR NEURAL NETWORK ON OUR DATASETS

Dataset	Accuracy
CIFAR-10	0.64
Imagenette (10 classes)	0.75

perform better. The heatmaps of actual golf balls usually have hot spots of important region which corresponds to the darker regions whereas the adversarial images generate heatmaps which shows that every part of the image contributes a significant amount to the classification. The reason behind what causes this phenomenon is worth further investigation.

One possible reason behind this phenomenon is that FGSM causes the texture of the image to change and the neural network gives more priority to the texture than the shape of the image which is indicative of an adversarial attack [9].

VI. CONCLUSION

Our experiments have shown that we can indeed use occlusion to detect a FGSM adversarial image. However, the accuracy in doing so is not high. We achieved an overall accuracy of 64% and 75% on the CIFAR-10 and Imagenette dataset respectively in detecting a adversarial image. There also seems to be a change in our success rate as we move across the different image classes in our dataset as seen in table IV and V. Our experimental result also show that our accuracy improves as the size of the image increase as seen in VI.

Future work can include experimenting with a wider range of adversarial attacks and datasets. We can maybe find that this method is more effective against certain adversarial attacks. It will be interesting to see if our performance can be further improved by making the image size and therefore the heatmap size much bigger. It is also worthwhile experimenting with the O_s and O_l hyperparameters and see how different occlusion size and stride makes a difference in the performance of the detector neural network.

REFERENCES

[1] I. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” in *International Conference on Learning Representations*, 2015. [Online]. Available: http://arxiv.org/abs/1412.6572

[2] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 818–833.

[3] Y. Cheng, L. Jiang, and W. Macherey, “Robust neural machine translation with doubly adversarial inputs,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019. [Online]. Available: https://www.aclweb.org/anthology/P19-1425

[4] C. Zhu, Y. Cheng, Z. Gan, S. Sun, T. Goldstein, and J. Liu, “Freelb: Enhanced adversarial training for natural language understanding,” 2020.

[5] A. H. Liu, H. Lee, and L. Lee, “Adversarial training of end-to-end speech recognition using a criticizing language model,” in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6176–6180.

[6] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, “Ensemble adversarial training: Attacks and defenses,” 2020.

[7] D. Meng and H. Chen, “Magnet: a two-pronged defense against adversarial examples,” 2017.

[8] J. Buckman, A. Roy, C. Raffel, and I. Goodfellow, “Thermometer encoding: One hot way to resist adversarial examples,” in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=S18Su–CW

[9] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, “Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness.” in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=Bygh9j09KX