Adversarial Attack Mechanisms on Convolutional Neural Networks for Traffic Signs [GitHub: Link]

Prasangsha Ganguly
50320914
Department of Industrial and Systems Engineering
prasangs@buffalo.edu

Anarghya Das
50213835
Department of Computer Science and Engineering
anarghya@buffalo.edu

Saiful Islam
50413531
Computational and Data Enabled Sciences and Engineering
saifulis@buffalo.edu

Abstract

Though in recent years neural networks have been widely used in several fields, they are often found to be vulnerable against adversarial examples, where the inputs are created by applying small but intentionally worst-case perturbations to the data samples, such that the perturbed input results in mis-classification. With an objective of exploring the robustness of different convolutional neural networks to classify traffic signals, we implemented different adversarial attack mechanisms. Using the German Traffic Sign dataset, and leveraging different neural networks, comparative analysis have been done to identify the effectiveness of the proposed attack mechanisms.

1 Introduction

GitHub: https://github.com/anarghya-das/DeepLearningProject

In recent times, adversarial attacks have demonstrated significant success in bringing down the performance of modern deep learning methods. An adversary can slightly perturb inputs to appear unchanged to humans but induce incorrect predictions by neural networks [1]. Studies have identified that, even the neural networks that perform at human level accuracy have an error rate of almost 100% on intelligently constructed examples which are very similar to the original examples. The lack of robustness exhibited by deep neural networks (DNNs) to adversarial examples has raised serious concerns for several applications like, *image identification and generation* [2], *text and language processing* [3], and *financial applications* amongst others. For example, image processing techniques applied to face recognition, automated driving by traffic signal identification among others. In this setting, often only a few pixels are altered in an image which results into a mis-classification. Often, these adversarial changes in the image are not even visually identified [2]. Furthermore, in this context, a neural network architecture is considered *robust* if the rate of incorrect prediction of the network is low under some adversarial attack scheme [4].

Self driving cars are one of the most interesting emerging technologies of the modern days. In self driving cars, vision systems of the vehicle with front facing camera modules used in variety of tasks such as traffic lane assist, traffic sign recognition (TSR), and object detection [5]. By creating intentional weak perturbations that get strong response in the output of the classifier, an attacker can produce malicious images or objects that can fool such systems. In this work, we consider adversarial attacks on the traffic signal images to induce incorrect predictions by the pre-trained convolutional neural networks. Using a comparative study of the performance of the neural networks and the percentage of successful attacks for different amount of perturbations for each type of attack, the best attack strategy and the most robust neural network architecture can be chosen. The remainder of the paper is organized as follows. In Section 2, the background of the convolutional neural network, and the adversarial attacks have been presented, the methodology employed in the paper has been depicted in section 3, the results have been discussed in section 4, and finally the paper is concluded in section 5.

2 Background and Problem Statement

Convolutional Neural Networks (CNNs) are widely studied in literature for image classification problems, which are simply neural networks that use convolution, instead of general matrix multiplication, in at least one of its layers. In our work, two different CNN models, viz., 1. `MicronNet`, 2. `AlexNet` are trained on the German Traffic Sign Recognition Benchmark data. The detailed architecture of the CNNs are described in section 3.

Often in literature, adversarial attacks and defences are formulated as a Stakelberg game between an attacker who wants to distort an input x to a new input $x + \delta$ via some distortion mechanism $\delta \in S$, where S is the feasible set of actions (distortions) of the attackers. On the other end, the defender tries to minimize the loss function. It can be expressed as follows [6],

$$\min_{\theta} \mathbb{E}_{(x,y)\in\chi}[\max_{\delta\in S} L(x+\delta; y; \theta)]$$

where χ is the set of training data consisting of inputs x and output labels y, L is the loss function, θ is the parameter of the network. For the attacker, there are different types of strategies which depend on the knowledge of the training mechanisms. For the *white-box* attack, the attacker has the knowledge of the training algorithm, on the other hand, the *black-box* attack doesn't have the knowledge of the training algorithm [7]. For defense strategies, there exists several methods like, adversarial training, defensive distillation or input transformations [8]. Several different attack mechanisms like NATTACK [9], DeepFool attack [10], universal perturbation attack [11] and BPDA attack [12] have been proposed in literature.

Though the adversarial attacks and defense mechanisms on the image data set have been well studied, adversarial attack mechanisms on the traffic data is not widely considered. Furthermore, the optimization algorithms used in the literature may not be guaranteed to converge to global optimality [13]. To address these gaps, the main objectives of the paper are,

- Train multiple CNNs which have been used in literature for the German Traffic Sign Recognition Benchmark data and analyze their performance on both the training and validation set data for different number of epochs and train-test split percentage.
- Proposed three types of adversarial attacks which shall perturb the input image such that the perturbed image is mis-classified by the neural network. Specifically, leveraging Simulated Annealing (SA) algorithm [14], an optimal attack has been realized. The details of the attacks have been depicted in the following section 3.
- Finally, we analyze the comparative performance of the different proposed attack mechanisms for the different neural networks used in our study.

3 Methodology

The overall research framework has been depicted in Fig. 1. In the framework, we first trained different CNN models. Then, we proposed three different attack mechanisms and using SA optimization algorithm, we obtained the optimal attacks on the input images.

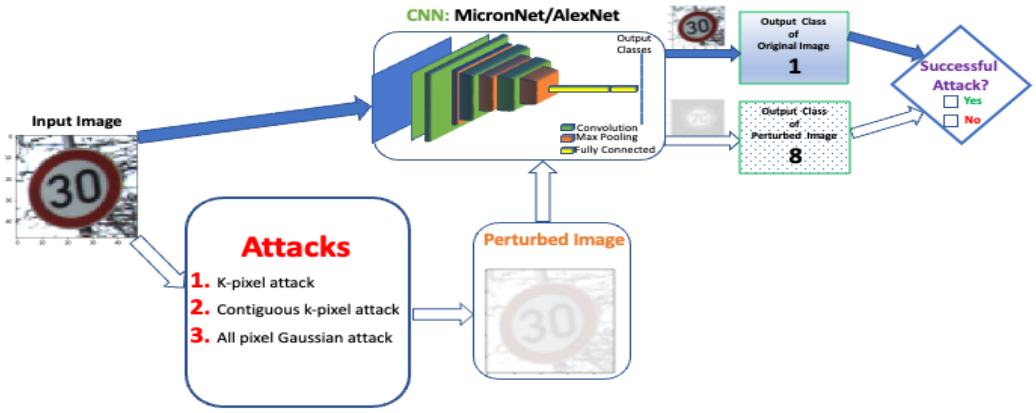

Figure 1: The overall research framework.

3.1 Data Description

The German Traffic Sign Recognition Benchmark [15] data has been used in our analysis. The dataset consists of 26,640 traffic sign images with 43 classes. For the test data, 12,630 images were used.

3.2 Training the CNNs

CNNs with different architecture have been trained in our model. The `MicronNet` [16] and `Alexnet` [17] uses a series of convolutional and max pooling and fully connected layers. The actual architecture have been depicted in Table. 1.

3.3 Background on Simulated Annealing

In this paper, we employed Simulated Annealing based optimization technique [14] to find the optimal attack which maximizes the difference between the actual label of the image and the predicted label of the perturbed image. Unlike population based techniques like genetic algorithm or differential evolution, the SA is guaranteed to converge to global optimal solution, given a suitable choice of the parameters.

Say, c is the current image (unperturbed) and n is the next image after perturbation with $Eval(c)$ and $Eval(n)$ denote the predicted classes of the images c and n by the neural network under consideration. The image n is a neighbor of the initial image c in the sense that the two images only differ by a set of pixels. In this case, we are not going to generate all the neighbors of c as there can be exponential number of neighbors. Instead, we are going to generate one random neighbor and decide whether to move to it

Table 1: Architectures of the CNNs used in the paper: (a) `MicronNet`, (b) `AlexNet`

(a)

Convolution $48X48X1$
Convolution $44X44X29$
Max Pool $22X22X29$
Convolution $20X20X59$
Max Pool $10X10X59$
Convolution $8X8X74$
Max Pool $4X4X74$
Fully Connected $1X1184$
Fully Connected $1X300$

(b)

Convolution $54X54X96$
Max Pool $26X26X96$
Convolution $26X26X256$
Max Pool $12X12X256$
Convolution $12X12X384$
Convolution $12X12X384$
Convolution $12X12X256$
Max Pool $5X5X256$
Fully Connected $1X4000$
Fully Connected $1X4000$
Fully Connected $1X300$

or not. Either we move to n from c or we remain at c. When we stay at c, we generate another random neighbor and again decide whether to move to it or not. We want decide based on the probability and the probability is governed by whether the move is good or bad. Initially we set T to very high. Then we lower the temperature gradually. Very often the algorithm has two loops. There is one inner loop where we generate a random neighbor, evaluate ΔE, and move to the neighbor with probability $P(c, n)$ to maximize ΔE. Then in the outer loop, we decrease T. The algorithm is depicted in Algo 1. As depicted in Fig.2, at high temperature, we will have a random walk like behavior (exploration) with probability 0.5. At low temperature, it will behave like hill climbing (exploitation of gradient).

Algorithm 1: Simulated Annealing for Adversarial Image Attack

begin

- $c \leftarrow$ Initial image ;
- $T \leftarrow$ Very high value ;
- **while** T *is not too low* **do**
  - **while** *Some number of iteration* **do**
    - $n \leftarrow$ Random Neighbor(c);
    - Evaluate $\Delta E = \text{Eval}(n) - \text{Eval}(c)$;
    - Move with probability $P(c, n) = \frac{1}{1+e^{-\Delta E/T}}$;
  - T decreases by some monotonically decreasing function ;

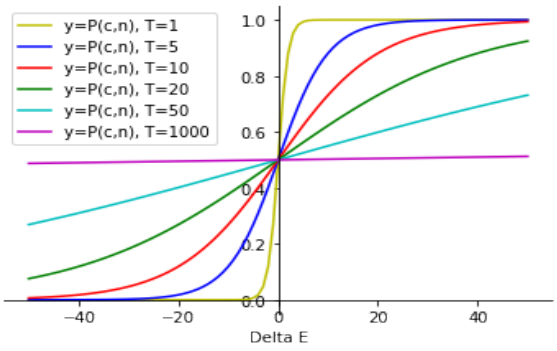

Figure 2: The transition probabilities versus the T for different ΔE for a maximization problem of SA.

To generate the attacks, we consider three types of perturbations to generate the neighbor image n as depicted below.

- **Attack 1: k-pixel attack**, inspired by the one pixel attack [13] where the image n and image c differs by exactly k pixel values. Essentially, we identify the minimum k, such that $Eval(n) \neq Eval(c)$. That is, in general, if δ is the perturbation on the input x and $f(x; y; \theta)$ is the predicted

Table 2: Performance of the CNNs for different epochs.

CNN	Epochs	Training Acc(%)	Test Acc(%)	Training time (sec)
MicronNet	5	51.12	95.91	158.67
	10	52.64	97.54	343.19
	15	52.97	98.43	507.63
Alexnet	5	77.0	89.43	241.57
	10	93.05	95.67	634.28
	15	96.12	96.92	914.6

output of the input x, it is given by,

$$Min \ \ ||\delta|| \tag{1}$$
$$S.t., \ \ f(x+\delta; y; \theta) \neq f(x; y; \theta) \tag{2}$$

- **Attack 2: Contiguous k-pixel attack**, where the perturbations made to the input image has to be confined within a neighborhood of the image ($\mathcal{N}_\delta$). This is particularly suitable for traffic signals, where physical traffic signs may be tampered.

$$Min \ \ ||\delta|| \tag{3}$$
$$S.t., \ \ f(x+\delta; y; \theta) \neq f(x; y; \theta) \tag{4}$$
$$\delta \in \mathcal{N}_\delta \tag{5}$$

- **Attack 3: All pixel Gaussian attack**, inspired by the FGSM attack [2], where an imperceptibly small vector whose elements are equal to the sign of the elements of the gradient of the cost function with respect to the input are added, we constructed a new attack algorithm. Here, all the pixels of the image are changed by a small Gaussian noise. That is $\delta \sim \mathbb{N}(\mu, \sigma^2)$ where, $\mu = 0.5$ and $\sigma = 0.1$.

4 Results

4.1 Performance of the CNNs

In Table 2, a comparative analysis of the neural networks for different number of epochs and different train train-test split have been presented. As it can be observed that the accuracy increases with the number of epochs. Furthermore, in between the two networks, the MicronNet runs faster compared to AlexNet while both of them have relatively similar performance for the test data set.

4.2 Performance of the attack methods

In Fig. 3, we depicted an example of the perturbation obtained by the different attack methods. As the performance of the CNNs have been very well on the validation data, the CNNs are robust enough and it requires many pixels (1617 pixels for Attack 1 and 753 pixels for Attack 2) to be perturbed for misclassification. Whereas, for the second example in Fig. 4, fewer pixels were sufficient to have a successful attack.

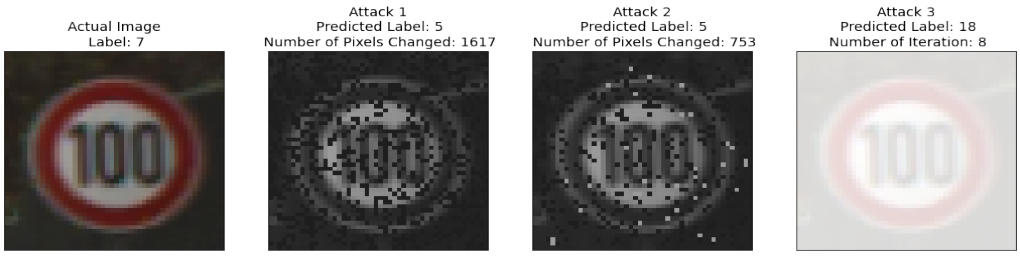

Figure 3: An example of the different types of attacks on an image of the traffic sign data set.

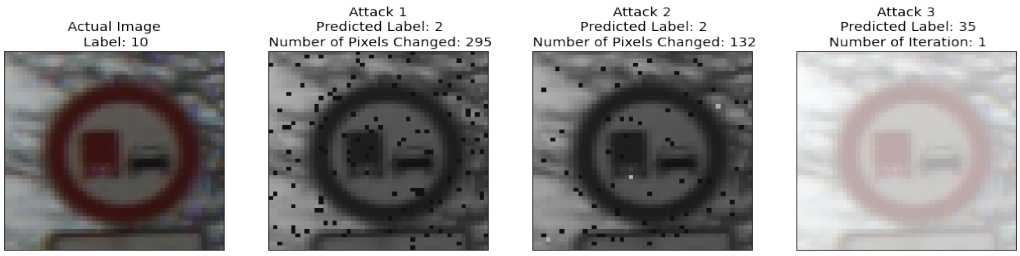

Figure 4: An example of the different types of attacks on an image of the traffic sign data set.

We evaluated the performance of the attack algorithms on the different neural networks. Say, we consider a sample of S images, then performance of the attack algorithm is given by the $\%$ of images in S which after perturbation yields a predicted output different than the original, i.e., $f(x+\delta; y; \theta) \neq f(x; y; \theta)$. As depicted in Fig. 5a, we plotted the $\%$ **of successful attack** versus the number of pixels perturbed for Attack 1 for a randomly sampled set of 30 images. Similarly, in Fig. 5b, we showed how the $\%$ of successful attack changes with the number of pixels perturbed. Finally, as in Attack 3, all the pixels are to be perturbed, we plotted the $\%$ of successful attack with the number of iterations taken to have a successful attack in Fig. 5c. Similarly, the performance of attack mechanisms on the `Micronet` model for a sample of 20 images have been depicted in Fig. 5d, Fig. 5e, and Fig. 5f respectively for Attack 1, Attack 2 and Attack 3.

For the `AlexNet`, we evaluate the percentage of successful attack for the different attack mechanisms as obtained in the Fig. 6.

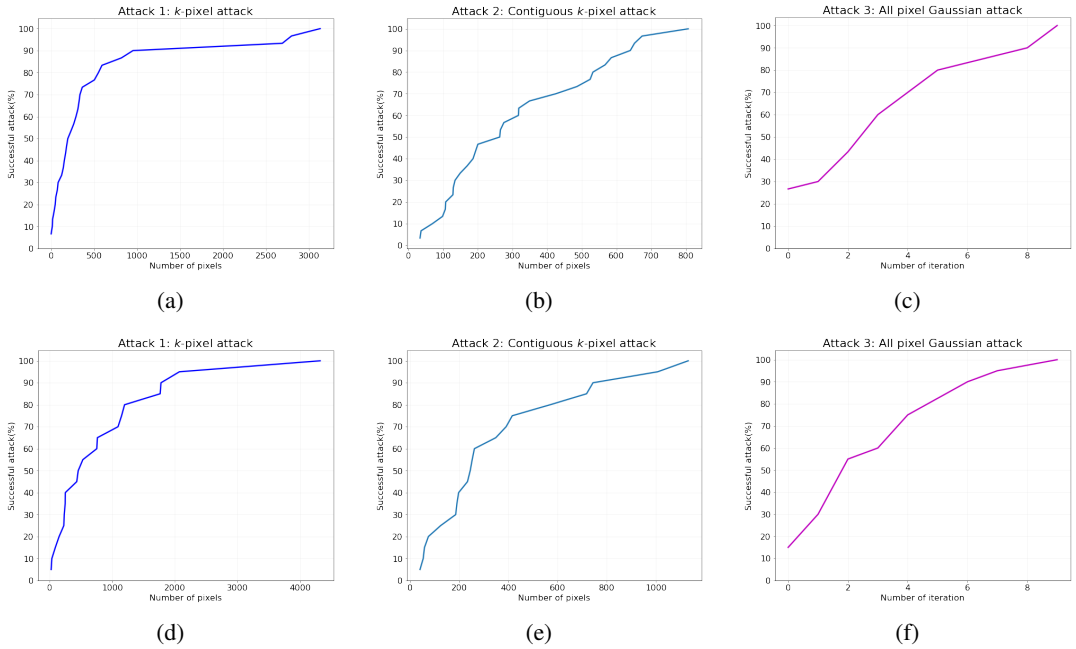

Figure 5: Performance of the attack algorithms for the `MicronNet` neural network for randomly sampled 30 images for (a) Attack 1, (b) Attack 2, (c) Attack 3 and sampled 20 images for (d) Attack 1, (e) Attack 2, (f) Attack 3

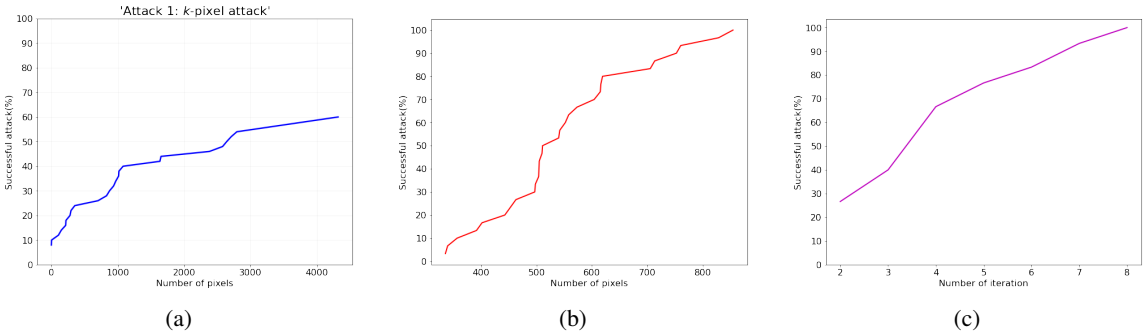

Figure 6: Performance of the attack algorithms for the `AlexNet` neural network for randomly sampled 30 images for (a) Attack 1, (b) Attack 2, and (c) Attack 3.

5 Conclusion

In this paper, we presented a framework to carryout adversarial attacks on traffic signal images on the CNN models. This framework can help in evaluating the robustness of neural networks to the adversarial attacks. In future, we would explore the defense mechanisms to make the neural networks more robust against the

adversarial attacks. Such defense mechanisms would further add a layer of robustness to the neural network architectures.

References

[1] Christian Szegedy et al. "Intriguing properties of neural networks". In: *arXiv preprint arXiv:1312.6199* (2013).

[2] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples". In: *arXiv preprint arXiv:1412.6572* (2014).

[3] Robin Jia and Percy Liang. "Adversarial examples for evaluating reading comprehension systems". In: *arXiv preprint arXiv:1707.07328* (2017).

[4] Nicholas Carlini and David Wagner. "Towards Evaluating the Robustness of Neural Networks". In: *2017 IEEE Symposium on Security and Privacy (SP)*. 2017, pp. 39–57. DOI: 10.1109/SP.2017.49.

[5] Nir Morgulis et al. "Fooling a Real Car with Adversarial Traffic Signs". In: (June 2019).

[6] Huan Zhang et al. "The Limitations of Adversarial Training and the Blind-Spot Attack". In: *ICLR* (Jan. 2019).

[7] Nicholas Carlini and Hany Farid. "Evading Deepfake-Image Detectors with White- and Black-Box Attacks". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2020, pp. 2804–2813. DOI: 10.1109/CVPRW50498.2020.00337.

[8] Aleksander Madry et al. "Towards Deep Learning Models Resistant to Adversarial Attacks". In: *ICLR* (June 2017).

[9] Yandong Li et al. "NATTACK: Learning the Distributions of Adversarial Examples for an Improved Black-Box Attack on Deep Neural Networks". In: (2019). DOI: 10.48550/ARXIV.1905.00441. URL: https://arxiv.org/abs/1905.00441.

[10] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. "DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 2574–2582. DOI: 10.1109/CVPR.2016.282.

[11] Seyed-Mohsen Moosavi-Dezfooli et al. "Universal Adversarial Perturbations". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 86–94. DOI: 10.1109/CVPR.2017.17.

[12] Anish Athalye, Nicholas Carlini, and David Wagner. "Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples". In: (2018).

[13] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. "One Pixel Attack for Fooling Deep Neural Networks". In: *IEEE Transactions on Evolutionary Computation* 23.5 (2019), pp. 828–841. DOI: 10.1109/TEVC.2019.2890858.

[14] Alexander G. Nikolaev and Sheldon H. Jacobson. "Simulated Annealing". In: *Handbook of Metaheuristics*. Ed. by Michel Gendreau and Jean-Yves Potvin. Boston, MA: Springer US, 2010, pp. 1–39. ISBN: 978-1-4419-1665-5. DOI: 10.1007/978-1-4419-1665-5_1. URL: https://doi.org/10.1007/978-1-4419-1665-5_1.

[15] J. Stallkamp et al. "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition". In: *Neural Networks* 0 (2012), pp. -.

[16] Alexander Wong, Mohammad Javad Shafiee, and Michael St. Jules. "MicronNet: A Highly Compact Deep Convolutional Neural Network Architecture for Real-Time Embedded Traffic Sign Classification". In: *IEEE Access* 6 (2018), pp. 59803–59810. DOI: 10.1109/ACCESS.2018.2873948.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Commun. ACM* 60.6 (May 2017), pp. 84–90. ISSN: 0001-0782. DOI: 10.1145/3065386. URL: https://doi.org/10.1145/3065386.