Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 4, Issue 4, 2026

ADVERSARIAL ROBUSTNESS EVALUATION OF CNN-BASED TRAFFIC SIGN RECOGNITION SYSTEMS

Raiyah Rub[*1], Shaheena Noor[2], Irfan Ahmed Usmani[3], Razia Maroof[4], Gul Munir[3]

[*1]*Electronic Engineering Department, Sir Syed University of Engineering and Technology, Karachi 75290, Pakistan*
[2]*Computer Engineering Department, Sir Syed University of Engineering and Technology, Karachi 75290, Pakistan*
[3]*Biomedical Engineering Department, Salim Habib University, Karachi,74900, Pakistan*
[4]*Computer Science Department, Iqra University, Karachi, 75500, Pakistan*

[*1]rrub@ssuet.edu.pk

DOI: https://doi.org/10.5281/zenodo.19658866

Keywords
Traffic Sign Recognition, Deep Neural Networks, Adversarial Robustness, FGSM, PGD, Benchmark Analysis

Article History
Received: 23 February 2026
Accepted: 02 April 2026
Published: 20 April 2026

Corresponding Author: *
Raiyah Rub

Abstract
Adversarial examples are a major challenge to deep neural networks used in safety-critical systems like intelligent transportation systems. Despite the outstanding performance of CNNs in traffic sign classification, their resilience to adversarial perturbations is not well-understood in various architectures. The paper is a benchmark study that assesses the adversarial robustness of VGG16, ResNet50, and EfficientNetB0 on the German Traffic Sign Recognition Benchmark (GTSRB) dataset. The models are evaluated to three gradient-based attacks FGSM, I-FGSM, and PGD with five $L\infty$ perturbation budgets ($\epsilon \in \{0.01, 0.02, 0.03, 0.05, 0.07$ to measure the accuracy loss and reveal architecture-specific patterns of vulnerability. To study failure modes qualitatively, performance is also studied using confusion matrices and feature space visualizations. Findings indicate that ResNet50 has the best adversarial robustness, retaining accuracy at over 94% in almost all settings, due to its residual connections that smooth the loss surface to gradient-based perturbations. EfficientNetB0 is the most susceptible to PGD, with a drop of 81.27% at $\epsilon = 0.07$, whereas VGG16 is the most susceptible to FGSM, decreasing to 80.95% at the same epsilon. These results emphasize the fact that adversarial robustness is inherently influenced by the architectural design decisions and provide a unified diagnostic benchmark to future studies of the secure deep learning models in autonomous driving systems.

1. Introduction

Deep neural networks have become the new paradigm of a broad range of computer vision tasks, and have demonstrated superhuman performance in fields like image classification, object detection, and semantic segmentation. The most promising use of this technology is in the creation of the autonomous vehicles and advanced driver-assistance systems (ADAS), in which the correct identification of the traffic signs is a pillar of safe navigation and decision making.

In the context of deep learning, many CNN designs have been suggested. This paper is devoted to the comparison of three influential models: VGG16 that introduced the idea of using deep stacks of small convolutional filters; ResNet50 that introduced the idea of residual connections to allow much deeper networks to be trained; and EfficientNetB0 that proposed a compound scaling approach to optimally trade-off network depth, width, and resolution.

Nevertheless, even with their impressive abilities, deep learning models have been demonstrated to be prone to adversarial attacks. These attacks entail the addition of well-designed, and usually unnoticeable, perturbations to the input data, with the aim of causing misclassification. Such vulnerabilities in the context of traffic sign recognition may prove to be disastrous, and a vehicle may fail to interpret a "Stop" sign as a "Speed Limit" sign.

This study presents a strict and methodical analysis of the adversarial resilience of various popular CNN designs to the task of traffic sign classification. We will test these models with a variety of adversarial attacks of different perturbation magnitudes to measure their resilience, determine their particular failure points, and gain a better insight into the obstacles that need to be overcome to make them safe in the real world. After winning the German Traffic Sign Recognition Benchmark competition [1], the first time CNN-based systems outperformed human-level accuracy, convolutional Neural Networks have become the new paradigm of TSR. Later architectures, such as VGG-Net [2], Res-Net [3] and Efficient-Net [4] have achieved over 99% accuracy under clean conditions on GTSRB.

The safety issue of these systems, however, is their adversarial robustness, which remains a safety challenge. The paper by [5] showed that CNNs are susceptible to imperceptibly small, intentionally designed input perturbations adversarial examples that are sure to cause a misclassification. Theoretical vulnerabilities were directly applied to the safety-critical domain in [6], which demonstrated the ability of physical-world adversarial perturbation of real STOP signs to defeat deployed classifiers in realistic drive-by scenarios.

Although there has been substantial research on adversarial attacks and defenses, comparative research on intrinsic (undefended) robustness of CNN architectures on various TSR datasets under a unified experimental protocol is limited. The majority of existing work assesses the robustness of a single architecture on a single dataset, and it is challenging to separate the effect of the architecture design, the properties of the dataset and the evaluation procedure on the achieved levels of robustness. The current paper will fill this gap by the following contributions:

1. Train VGG16, ResNet50 and EfficientNetB0 on GTSRB dataset.
2. Test all models with FGSM, I-FGSM and PGD attacks on both datasets using seven perturbation budgets with varying epsilon values.
3. Discuss architectural considerations, data properties, and implications to TSR system designers.

The paper is organized as follows. Section 2 is a review of related work. Section 3 explains the datasets and experimental procedure. Whereas, results and analysis discussed in section 4 and Section 5 defines conclusion.

2. Literature review

2.1 CNN Architectures for Traffic Sign Recognition

Initial CNN-based TSR methods used dedicated architectures that were optimized to the visual characteristics of traffic signs. Even before CNN domination other traditional computer vision systems also reported competitive performance; for instance, that text-based road signs could be detected with handcrafted feature pipelines [17] but these systems were not as scalable and generalizable as the current CNN-based systems. Furthermore, a multi-scale CNN that fused features of various convolutional layers with 98.97% on GTSRB [7] higher than human-level performance. Moreover, [8] Committee networks of deep CNNs were then used to reach 99.46% establishing CNNs as state of the art. In addition, VGG16 and ResNet50 fine-tuned on GTSRB attain 99.5% clean accuracy [3]. Similarly, Compound-scaled EfficientNetB0 has competitive accuracy with significantly fewer parameters [4] which makes it appealing to embedded automotive use. More recently, transformer-based architectures, including Vision Transformers (ViT) have also been considered in TSR, but CNN-based models still dominate embedded automotive applications because of their lower computational cost [19].

2.2 Adversarial Attacks

The Fast Gradient Sign Method [9] produces adversarial perturbations that give a single-step attack that is computationally efficient. This was extended to iterative FGSM (I-FGSM/BIM) using N gradient steps of size $\alpha = \epsilon/N$ with L^∞ projection which greatly boosts the success of attacks[10]. The Projected Gradient Descent (PGD) [11] introduces random initialization in the ϵ-ball prior to gradient steps and multiple restarts to estimate the worst-case adversarial example under the L^∞ constraint, the current standard of first-order L^∞ robustness evaluation. In addition to white-box environments also showed that adversarial examples are transferable across models [18], allowing practical black-box attacks on machine learning systems without model internals a threat especially in deployed automotive systems where model architectures are often confidential. The expectation over transformation (EoT) framework of physical attack robustness optimization was formalized [12]. It is also showed that adversarial examples generated in the frequency domain have a higher transferability across architectures than spatial-domain attacks, which presents a higher risk to black-box TSR systems in autonomous vehicles [20].

2.3 Adversarial Robustness of CNNs on GTSRB

The systematic study of adversarial attacks on GTSRB classifiers was carried out which showed that VGG-based TSR classifiers were susceptible to digital and physical adversarial perturbations [13]. The landmark physical-world attack study was demonstrated that sticker-modified STOP signs caused misclassification in CNN-based detectors [6]. A comparison of VGG-16, ResNet-50, and MobileNet-V2 to FGSM and BIM attacks revealed no consistent robustness benefits of deeper architectures but varying per-class vulnerability distributions [14]. AutoAttack was introduced to test GTSRB classifiers and it was shown that defenses tested using FGSM or single-restart PGD systematically overstated robustness [15]. This line of work is further developed in the present paper, which gives a three-model comparison under a single experimental protocol with fine-grained perturbation budget analysis. Most recently, it is shown that pre-training on large-scale datasets, followed by adversarial fine-tuning, can significantly enhance robustness-accuracy trade-offs, indicating that foundation model pretraining can be a viable route to robust TSR systems at a prohibitive cost of full adversarial training [21].

3. Methodology

This section presents the datasets, model architectures, and the end-to-end experimental protocol of this study. The implementation of the methodology was done on Python with TensorFlow v2.9.2.

3.1. Datasets

German Traffic Sign Recognition Benchmark (GTSRB): This data set consists of more than 50,000 images of 43 traffic signs classes. The pictures were taken in a very diverse range of real life conditions such as lighting, weather and the viewing angles. One of its key features is a harsh imbalance of classes, which is graphically verified in the Class Distribution chart of Figure1 GTSRB. An example is the Speed limit (50km/h) (class 2) and Priority-road (class 12) classes, which have more than 2,000 training samples, and the Go straight or left (class 36) and End of all speed and passing limits (class 32) classes, which have less than 250 samples.

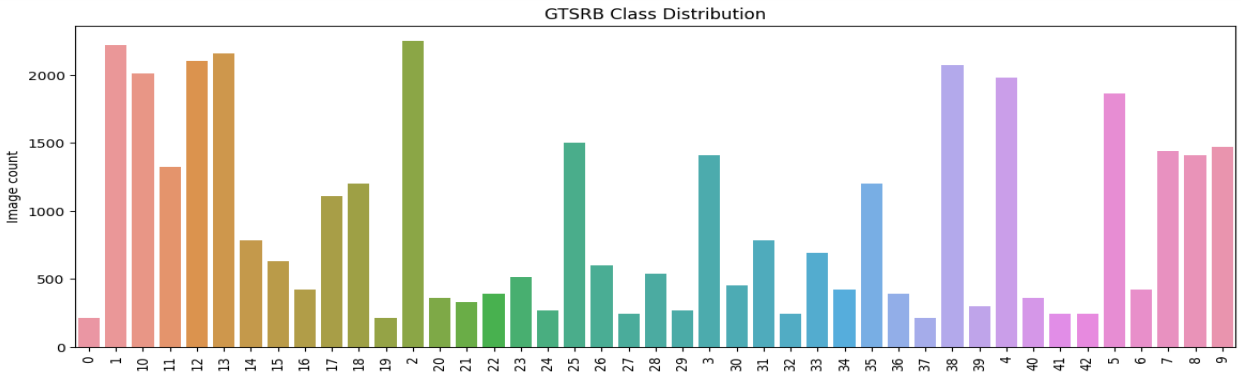


Figure 1 GTSRB Class Distribution.

3.2. Model Architectures

The study used a transfer learning model where VGG16, ResNet50, and EfficientNetB0 were pre-trained on ImageNet and used as feature extractors. Classification layers of each model were replaced with a custom head, whose construction is defined by the following:

Algorithm 1: Build Top Model Layers

Input: Base model M_base, Number of classes C, Dropout rate p **Begin**
1. x ←M_base.output 2. x ← GlobalAveragePooling2D(x) 3. x ← Dropout(p)(x) 4. x ← Dense(512, ReLU)(x) 5. x ← Dropout(p)(x) 6. out ← Dense(C, Softmax)(x) 7. M_final ← Model(M_base.input, out) 8. **Return** M_final **End**
Output: Final classification model M_final

The following three deep learning architectures were evaluated:

VGG16: [2] It has 13 convolutional layers with 3×3 kernels in five pooling blocks (channels: 64→128→256→512→512) followed by three fully connected layers (˜138M parameters). Its large parameter count and absence of residual connections produce sharp, high-curvature decision boundaries that we hypothesise are relatively susceptible to gradient-based adversarial attacks.

ResNet50: This architecture introduces the concept of residual connections, or "shortcuts," which allow the model to learn deeper representations without suffering from the vanishing gradient problem and improve robustness. [3] employs skip connections via bottleneck residual blocks across 50 layers (˜25M parameters). Residual connections have been shown to flatten loss landscapes [16], which we predict will produce relatively more consistent robustness under iterative gradient-based attacks.

EfficientNetB0: [4] Compound scaling over MBConv blocks with depthwise separable convolutions and squeeze-and-excitation modules (˜5.3M parameters) . Its parameter-efficiency and high clean accuracy make it attractive for deployment but its compact representational structure may concentrate vulnerability in fewer high-impact channels.

3.3. Adversarial Attacks

The models were subjected to three gradient-based white-box attacks, where the attacker has full knowledge of the model's architecture and parameters. The perturbation magnitude, denoted by ϵ, was varied across a range of values: [0.01, 0.02, 0.03, 0.05, 0.07].

3.3.1. Fast Gradient Sign Method (FGSM): FGSM [9] is a single-step white-box attack that computes the perturbation direction as the sign of the input gradient of the cross-entropy loss:

$$x^{adv} = x + \epsilon . sign(\nabla_x J(\theta , x, y))$$

(1)

where J is the cross-entropy loss, θ model parameters, y the true label, and ϵ the L∞ perturbation budget. FGSM requires one forward and one backward pass, making it the computationally cheapest attack. It establishes a lower bound on adversarial vulnerability and assesses sensitivity to single-step gradient-aligned perturbations.

Algorithm 2: Fast Gradient Sign Method (FGSM)

Input: Model M, Input image x, True label y, Perturbation strength ϵ
Begin
1. **Initialize** gradient tracking for input *x*
2. Compute model prediction: logits ← M(x)
3. Compute loss: L ← CrossEntropy(y, logits)
4. Compute gradient of loss w.r.t input: g ← ∇_x L
5. Generate adversarial perturbation: x^{adv} ← x + ϵ · sign(g)
6. Clip x^{adv} to valid range: x^{adv} ← Clip(x_adv, 0, 0.07)
7. **Return** x^{adv}

End
Output: Adversarial image x^{adv}

3.3.2. Iterative Fast Gradient Sign Method (I-FGSM): I-FGSM [10] also called Basic Iterative Method (BIM), applies N iterative FGSM steps with step size α and projects back onto the L∞ ϵ-ball after each step:

$$x_{t+1}^{adv} = Clip_{x,\epsilon}(x_t^{adv} + \alpha . sign(\nabla_x J(\theta, x_t^{adv}),y)))$$

(2)

We use N = 10 iterations and α = ϵ /N, initialised from the clean image and $Clip_{x,\epsilon}$ ensures perturbations remain within the ϵ-ball around the original input *x*.

Algorithm 3: Iterative Fast Gradient Sign Method (I-FGSM)

Input: Model M, Input image x, True label y, Perturbation limit ϵ, Iterations T
Begin
1. **Set** step size α ← ϵ / T
2. **Initialize** adversarial example: *adv* ← *x*
3. **For** t = 1 to T do
4. Initialize gradient tape for adv
5. Compute model prediction: logits ← M(*adv*)
6. Compute loss: L ← CrossEntropy(y, logits)
7. Compute gradient: g ← ∇_{adv} L
8. Update adversarial example: *adv* ← *adv* + α · sign(g)
9. Clip *adv* to valid range: adv ← Clip(*adv*, 0, 0.07)
10. **End** For
11. **Return** x^{adv} ← *adv*

End
Output: Adversarial image x^{adv}

3.3.3. Projected Gradient Descent (PGD): Widely considered one of the strongest first-order attacks, PGD [11] is an iterative attack that starts from a random point in the ϵ-ball around the clean image and takes multiple small steps in the gradient direction, projecting the result back onto the ϵ-ball after each step.

$$x_{t+1}^{adv}=\pi_{\beta_\epsilon(x)}(x_t^{adv}+\propto.sign(\nabla_x J(\theta,x_t^{adv},y))) \quad (3)$$

Where $\pi_{\beta_\epsilon(x)}$ projects the perturbed sample back into the allowed ϵ-ball around x.

Algorithm 4: Projected Gradient Descent (PGD)

Input: Model M, Input image x, True label y, Perturbation limit ϵ, Iterations T

Begin
1. Set step size $\alpha \leftarrow \epsilon / T$
2. **Initialize** adversarial example with random noise:
adv ← x + Uniform(-ϵ, ϵ)
3. Clip *adv* to valid range: *adv* ← Clip(adv, 0, 0.07)
4. **For** t = 1 to T do
5. **Initialize** gradient tape for *adv*
6. Compute model prediction: logits ← M(*adv*)
7. Compute loss: L ← CrossEntropy(y, logits)
8. Compute gradient: g ← ∇_adv L
9. Update adversarial example: *adv* ← *adv* + $\alpha \cdot$ sign(g)
10. Project adv into ϵ-ball around *x*:
adv ← min(max(*adv*, $x - \epsilon$), $x + \epsilon$)
11. Clip adv to valid range: *adv* ← Clip(*adv*, 0, 0.07)
12. **End** For
13. **Return** x^{adv} ← *adv*

End

Output: Adversarial image x^{adv}

3.3. Experimental setup

Data Preprocessing and Augmentation: All images were resized to 224x224 pixels. For training, extensive on-the-fly data augmentation was applied to enhance model generalization. The specific parameters are shown in the ImageDataGenerator instantiation was used to implement the training data augmentation, a rescaling factor of 1/255 was used to normalize the dataset and spatial transformations were performed: rotation (±15°)width and height shifts (up to 12%), zooming (up to 15%), and shearing (up to 8%). Moreover, the brightness was also varied within the range of 0.8-1.2 to replicate the different lighting conditions. A validation split of 20% was used to separate the training and validation sets.

Training Protocol: Models were trained for up to 20 epochs using the Adam optimizer and categorical cross-entropy loss. A suite of Keras callbacks, including ModelCheckpoint, EarlyStopping, and learning rate schedulers (ReduceLROnPlateau, LearningRateScheduler), was used to manage the training process and prevent overfitting.

Table 1. Model Hyperparameters for GTSRB

Parameter	GTSRB Dataset
Image Size	224 × 224
Batch Size	64
Epochs	20
Optimizer	Adam
Adam Parameters	ϵ = 1×10^{-8}, β_1 = 0.9, β_2 = 0.999
Learning Rate	0.0001
Loss Function	Categorical Cross-Entropy

4. Results and Analysis

In order to present a comprehensive evaluation of model robustness, quantitative measures and visual analytics are used**.** Clean ccuracy (Baseline Performance).

Indicates the accuracy of every CNN model (VGG16, ResNet50, EfficientNetB0) on unperturbed GTSRB datasets. This is used as the benchmark to measure the strength in adversarial situations.$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$

(4)

where *TP, TN, FP* and *FN* are the true positives, true negatives, false positives and false negatives respectively.

Table 2. Clean accuracy (%) of three CNN architectures tested on the GTSRB Dataset. ResNet50 and EfficientNetB0 show better performance.

Model	GTSRB Accuracy (%)
VGG16	97.7
ResNet50	98.1
EfficientNetB0	98.1

4.1. Adversarial Accuracy (Across Epsilon Values):

Classification accuracy is measured after applying adversarial perturbations generated using FGSM, I-FGSM, and PGD. The results are provided at various perturbation strengths (0.01, 0.02, 0.03, 0.05, 0.07) to give a degradation curve, which indicates sensitivity of each architecture. The overall accuracy equation is:

$$Accuracy_{adv}(\epsilon) = \frac{Correct\ Predictions\ under\ Attacks\ at\ \epsilon}{Total\ Samples}$$

(5)The general accuracy formula is:

$$Accuracy = \frac{1}{N}\sum_{i=1}^{N} 1(\hat{y}_i = y_i)$$

(6)

where N is the number of test samples in total, $\hat{y}_i$ is the predicted label, and y_i is the ground truth label. This is baseline accuracy.

When we introduce adversarial perturbations (x^{adv}):

$$Accuracy_{adv}(\epsilon) = \frac{1}{N}\sum_{i=1}^{N} 1(f_\theta(x_i^{\epsilon}) = y_i)$$

(7)where $f_\theta(\cdot)$ is the classifier.

- $x_i^{adv}$is the adversarially perturbed input, $x_i^{\epsilon} = \begin{cases} x_i, & if\ \epsilon = 0 (clean\ input) \\ x_i^{adv}, & if\ \epsilon > 0 (adversarial\ input) \end{cases}$

Table 3. Adversarial accuracy (%) comparison of baseline under FGSM, I-FGSM, and PGD attacks on the GTSRB dataset across different ϵ values.

Model	ϵ	VGG16	ResNet50	EfficientNetb0
FGSM	0.01	97.71	97.98	97.55
	0.02	97.21	97.88	96.58
	0.03	95.86	97.43	95.32
	0.05	90.35	92.85	92.23
	0.07	80.95	81.48	89.88
IFGSM	0.01	97.31	97.96	96.47
	0.02	96.84	97.88	96.48
	0.03	95.86	97.43	96.52
	0.05	95.72	97.98	96.48
	0.07	95.32	97.96	96.48
PGD	0.01	97.43	97.84	95.07
	0.02	96.69	97.83	94.15
	0.03	96.25	97.63	93.22
	0.05	94.54	96.51	88.25
	0.07	92.73	94.29	81.27

Table 3. indicates that the adversarial robustness of VGG16, ResNet50, and EfficientNetB0 on the GTSRB dataset when subjected to FGSM, I-FGSM, and PGD attacks have different vulnerability profiles across architectures. In FGSM, all models degrade monotonically as epsilon increases, with VGG16 decreasing the most by almost 17% between 97.71% and 80.95%, and EfficientNetB0 decreasing more slowly to 89.88%, indicating some resistance to single-step perturbation. Conversely, I-FGSM shows unexpectedly stable performance in all three models, with ResNet50 and EfficientNetB0 showing nearly constant accuracy over the entire epsilon range, presumably because of saturation of iterative steps in the perturbation constraint space. PGD is the most discriminative, with EfficientNetB0 being the most vulnerable architecture with a sharp loss drop between 95.07% and 81.27%, whereas ResNet50 is the most resilient overall with a loss drop of only 3.55% , from 97.84% to 94.29% which can be explained by its skip connections making the loss landscape less rough to gradient Altogether, these findings support the idea that architectural design, especially the existence of residual connections, is a determining factor in adversarial robustness, and that a single type of attack is not enough to perform a comprehensive vulnerability evaluation.

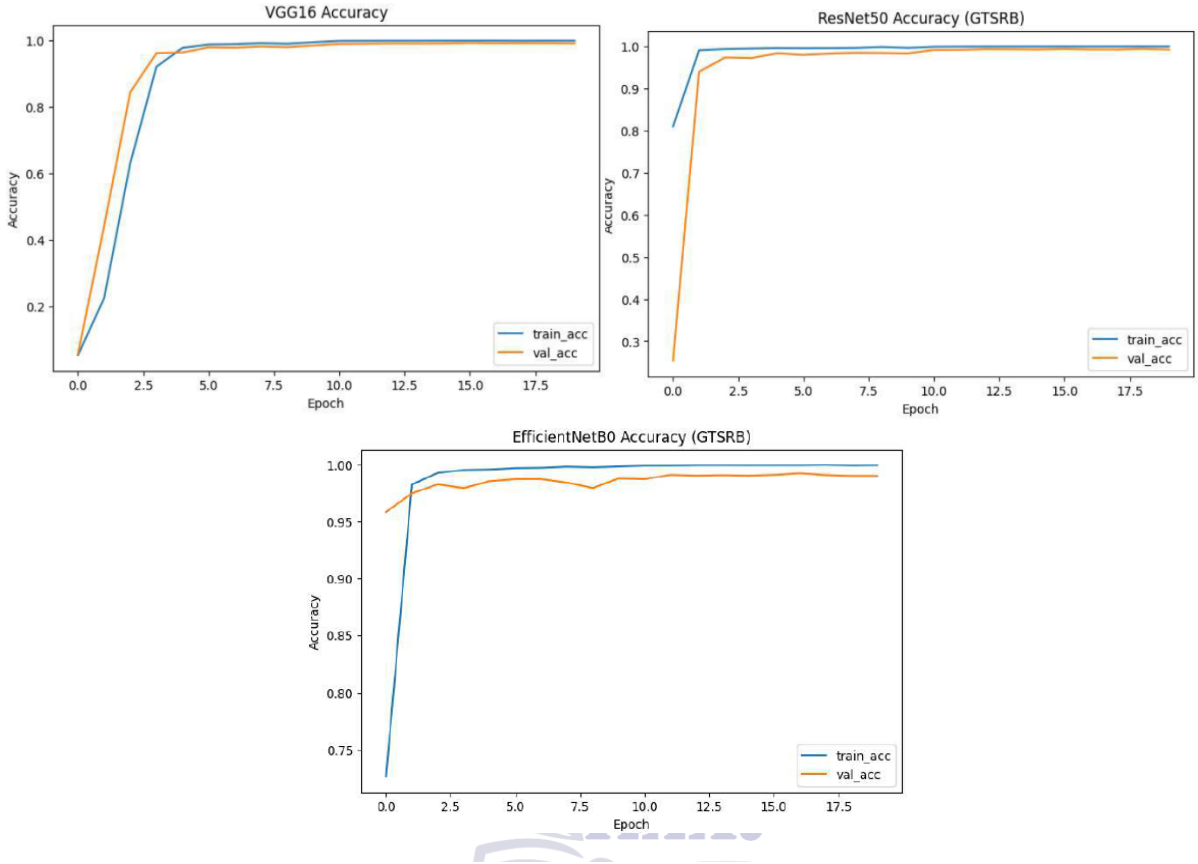
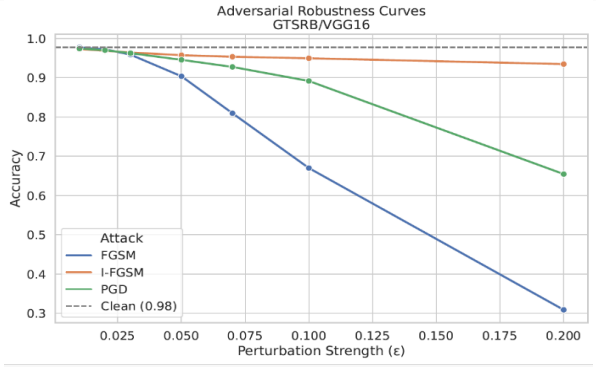
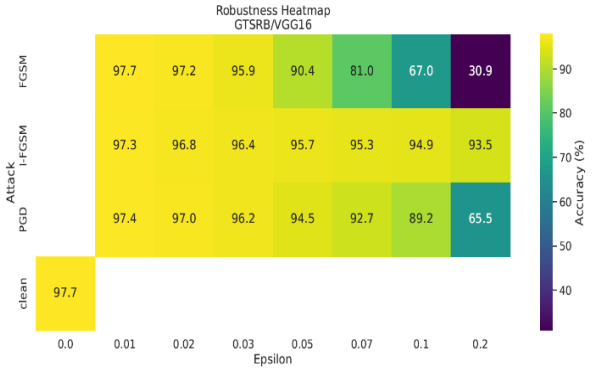


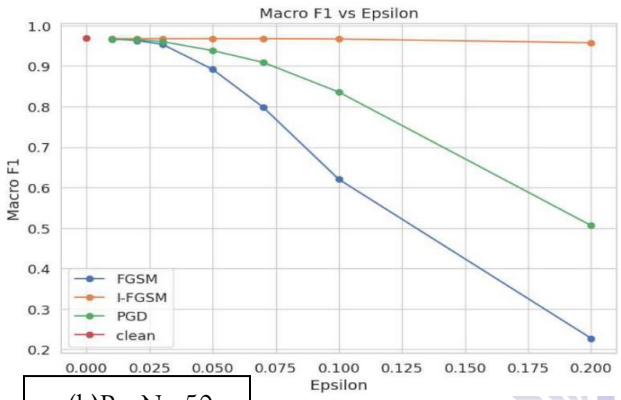
Figure 2. Comparison of training vs validation accuracy curve of GTSRB dataset on CNN models (VGG16, ResNet50, EfficientNetB0).

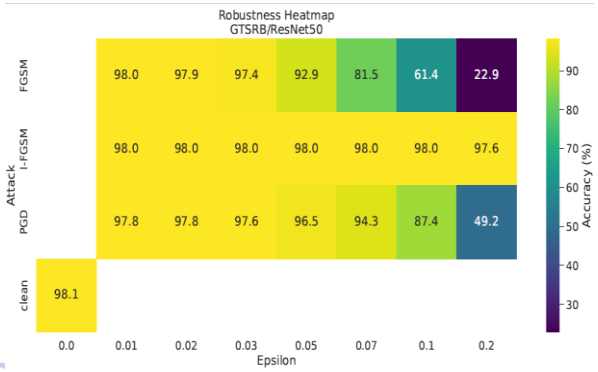
In figure 2 the VGG16 Accuracy graph with validation accuracy rapidly increasing to above 99% and following the training accuracy closely, which means that it is generalizing very well. Both EfficientNetB0 and ResNet50 were better convergent. Both the ResNet50 and EfficientNetB0 Accuracy graphs demonstrate validation accuracies of more than 98 per cent within the initial few epochs and finally more than 99 per cent. In order to intuitively describe the degradation patterns, heatmaps are produced of each CNN model with different adversarial intensities. These graphical depictions show the most drastic accuracy deteriorations due to which attacks and perturbation levels. Figure 3 demonstrates that the robustness Heatmap is a good visualization of the accuracy that decreases quickly with ϵ. In contrast to a similar deterioration in overall performance is seen in the Macro F1 vs Epsilon plot in Figure 3 Although the I-FGSM attack does not have such a strong effect on the F1 score, FGSM and PGD have a sharp drop, which means that it loses a significant amount of both precision accuracy and recall.



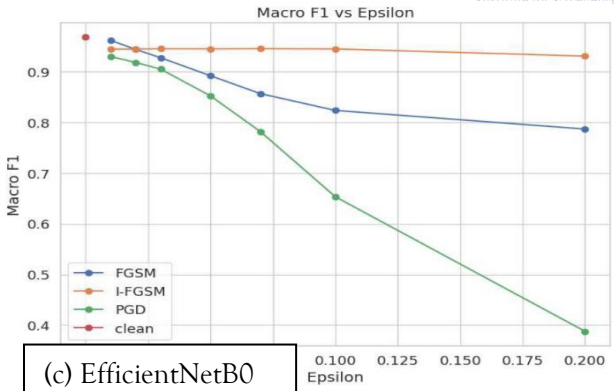


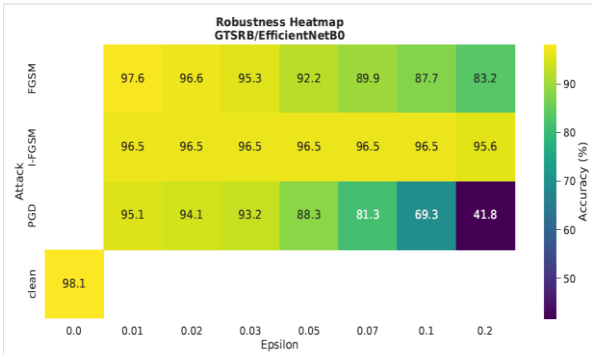
(a)VGG16





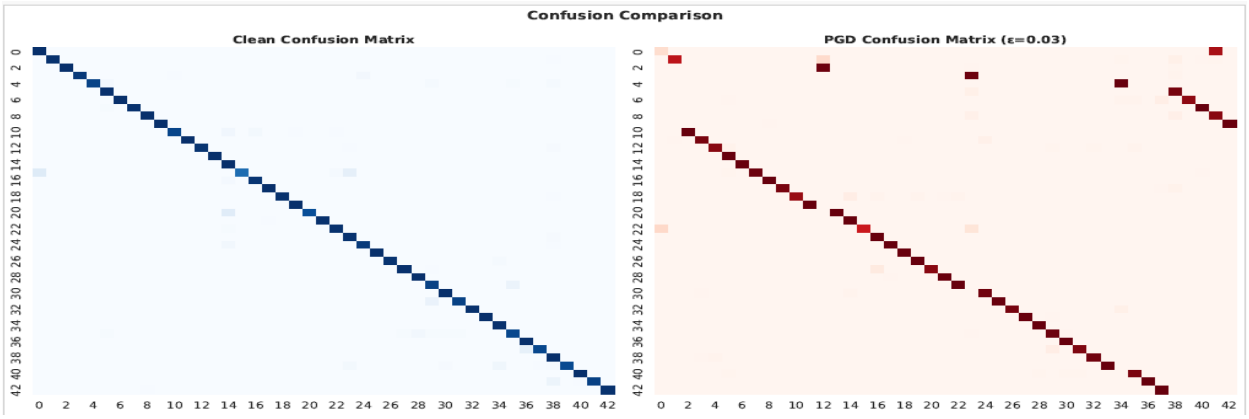
(b)ResNet50



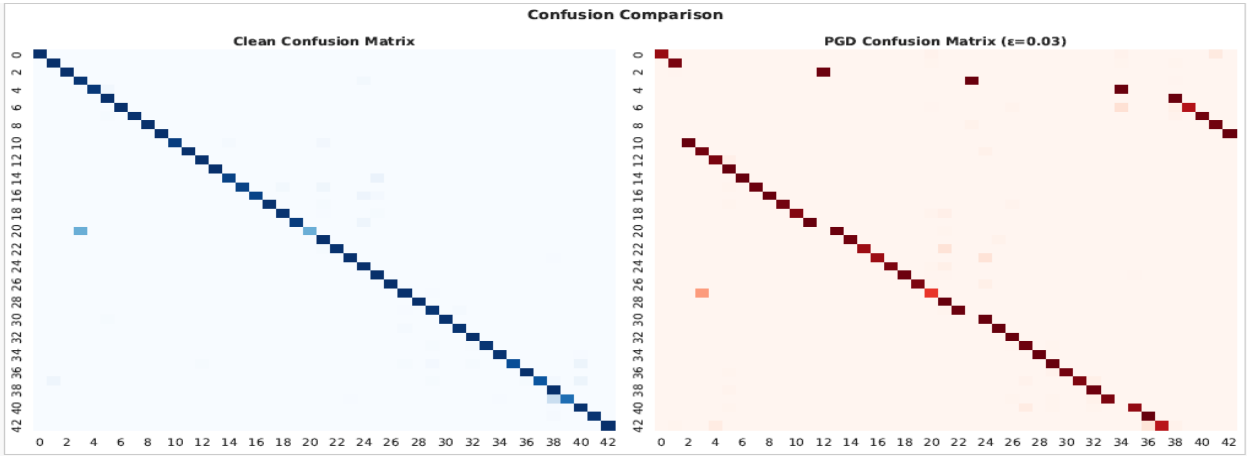


(c) EfficientNetB0

Figure 3 Heatmap and Adversarial accuracy versus perturbation strength (ϵ) for FGSM, I-FGSM, and PGD attacks for CNN models (a) VGG16 (b) ResNet50 (c) EfficientNetB0



(a)VGG16



(b) RESNET50
(c) EfficientNetB0

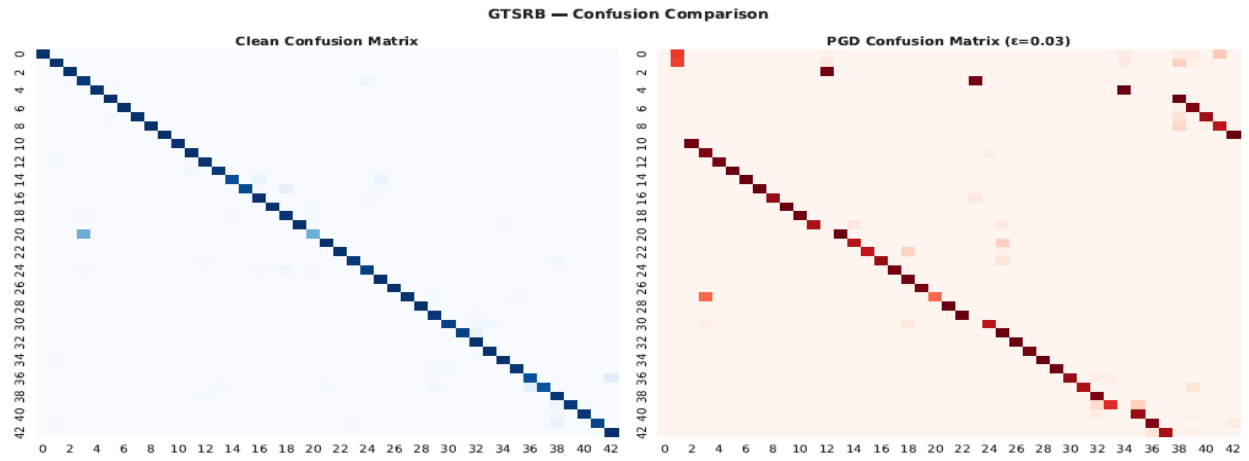
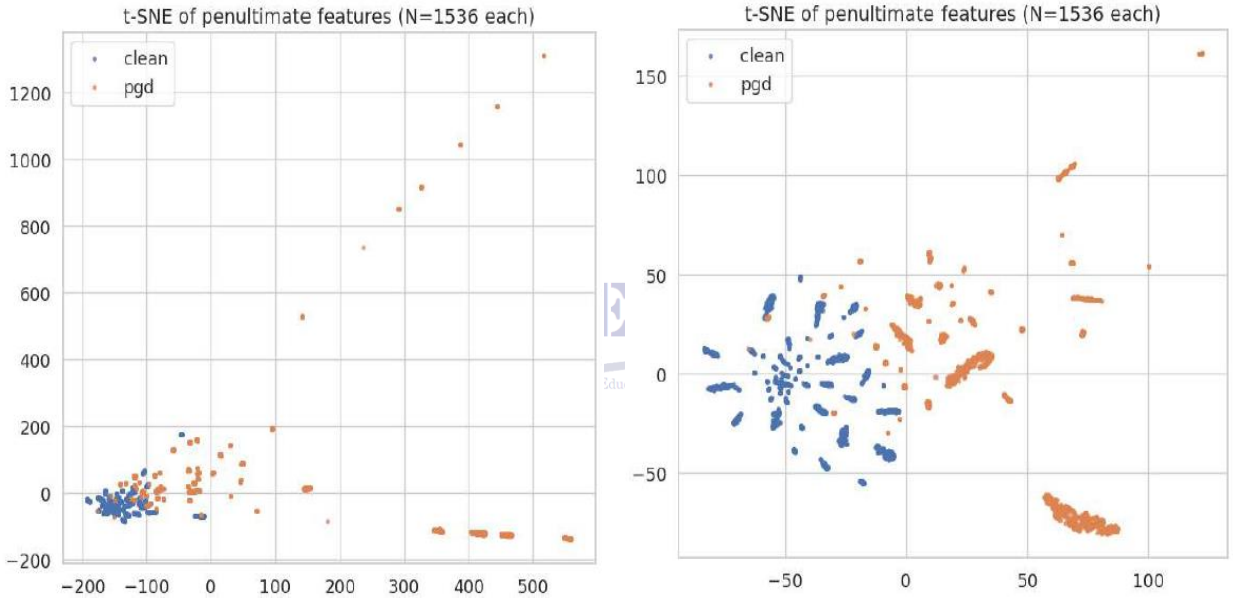


Figure 4. Comparison of confusion matrices between the performance of the model on clean and perturbed data (PGD at ϵ=0.03) of all models VGG16, ResNet50 and EfficientNetB0.

Figure 4. indicates the classification Performance between the model performance on clean and perturbed data of all three models in which VGG16 Confusion Matrix is virtually diagonal, which visually confirms the high accuracy, and indicates that the model is highly susceptible to adversarial attacks. The confusion matrices of both models ResNet50 and EfficientNetB0 are highly diagonal as well, which proves the almost perfect classification of the test set. On the other hand, the PGD Confusion Matrix (ϵ=0.03) is diffuse with many off-diagonal elements indicating a high level of misclassification in many of the classes of all three models.

In Figure 5. VGG16 the t-SNE of penultimate features plot is a powerful visualization of the effect of the attack. The characteristics of clean images are well-separated clusters. Nevertheless, the characteristics of the PGD-perturbed images are diffused and mixed, which means that the attack indeed disturbs the model-learned representations of features, shifting them through decision boundaries. The t-SNE of penultimate features in ResNet50 shows that the PGD attack is successful in perturbing the feature space, resulting in the overlap of the clusters of different classes, hence misclassification.



(a) VGG16 (b) RESNET50

Figure 5. t-SNE of penultimate features for (a) VGG16 (b) ResNet50

4.2. Macro Precision, Recall, and F1-Score:

Since the distribution of the categories of traffic signs is unequal, the macro-averaging is applied to give equal weight to all classes. Where N is the number of classes (43 in GTSRB).

$$Precicion_{macro} = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{TP_i+FP_i} \quad (8)$$

Macro Precision determines the proportion of the correctly predicted positive samples in all classes.

Macro Recall measures the ability of the classifier to recognize true positives in categories.

$$Recall_{macro} = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{TP_i+FN_i}$$

(9)

Macro F1-Score is a balance between precision and recall, providing one strong indicator of robustness.

$$F1_{macro} = \frac{1}{N}\sum_{i=1}^{N}\frac{2\times Precicion_i \times Recall_i}{Precicion_i+Recall_i}$$

(10)

Table 4. Macro-averaged precision, recall, and F1-score of VGG16 under FGSM, I-FGSM, and PGD attacks for varying perturbation strengths (ϵ).

Attack	Epsilon	precision_macro	recall_macro	f1_macro
Clean	0	0.9650	0.9719	0.9667
FGSM	0.01	0.9694	0.9730	0.9701
I-FGSM		0.9635	0.9612	0.9607
PGD		0.9640	0.9622	0.9615
FGSM	0.02	0.9632	0.9628	0.9616
I-FGSM		0.9597	0.9568	0.9562
PGD		0.9610	0.9593	0.9581
FGSM	0.03	0.9491	0.9394	0.9406
I-FGSM		0.9524	0.9512	0.9495
PGD		0.9519	0.9516	0.9494
FGSM	0.05	0.9151	0.8671	0.8724
I-FGSM		0.9442	0.9461	0.9433
PGD		0.9307	0.9353	0.9300
FGSM	0.07	0.8661	0.7868	0.7984
I-FGSM		0.9430	0.9442	0.9415
PGD		0.9204	0.9149	0.9125

Table 5. Macro-averaged precision, recall, and F1-score of ResNet50 under FGSM, I-FGSM, and PGD attacks for varying perturbation strengths (ϵ).

Attack	Epsilon	precision_macro	recall_macro	f1_macro
Clean	0	0.9787	0.9640	0.9686
FGSM	0.01	0.9747	0.9643	0.9673
I-FGSM		0.9768	0.9642	0.9677
PGD		0.9734	0.9624	0.9655
FGSM	0.02	0.9651	0.9633	0.9630
I-FGSM		0.9768	0.9640	0.9676
PGD		0.9709	0.9621	0.9650
FGSM	0.03	0.9539	0.9583	0.9540
I-FGSM		0.9768	0.9640	0.9676
PGD		0.9647	0.9593	0.9605
FGSM	0.05	0.9129	0.9021	0.8917
I-FGSM		0.9755	0.9642	0.9676
PGD		0.9458	0.9440	0.9381
FGSM	0.07	0.8856	0.7971	0.7981
I-FGSM		0.9755	0.9643	0.9676
PGD		0.9289	0.9168	0.9091

Table 6. Macro-averaged precision, recall, and F1-score of EfficientNetb0 under FGSM, I-FGSM, and PGD attacks for varying perturbation strengths (ϵ).

Attack	Epsilon	precision_macro	recall_macro	f1_macro
Clean	0	0.9749	0.9634	0.9678
FGSM	0.01	0.9754	0.9529	0.9614
I-FGSM		0.9560	0.9392	0.9441
PGD		0.9352	0.9293	0.9295
FGSM	0.02	0.9575	0.9373	0.9437
I-FGSM		0.9560	0.9397	0.9444
PGD		0.9261	0.9160	0.9179
FGSM	0.03	0.9498	0.9183	0.9272
I-FGSM		0.9567	0.9399	0.9449
PGD		0.9155	0.9017	0.9048
FGSM	0.05	0.9304	0.8794	0.8917
I-FGSM		0.9572	0.9395	0.9447
PGD		0.8726	0.8516	0.8519
FGSM	0.07	0.9086	0.8416	0.8564
I-FGSM		0.9586	0.9397	0.9450
PGD		0.8076	0.7892	0.7815

The comparative analysis of Tables 4-6 shows clearly that the performance of the models declines with the strength of perturbation (ϵ) though the degree of decline differs among architectures and attack types. It is noted that FGSM leads to a high decline in the precision, recall and F1-score at higher ϵ values especially in VGG16 and EfficientNetB0, which are vulnerable to single-step attacks. Conversely, I-FGSM shows comparatively consistent performance with all models, particularly with ResNet50, which achieves high scores even at ϵ = 0.07, which implies higher robustness. In addition, the fact that PGD is a more powerful iterative attack leads to moderate degradation, although not as severe as FGSM in certain instances. Overall, ResNet50 turns out to be the most resilient model, whereas EfficientNetB0 proves to be more vulnerable in more adversarial environments.

5. Conclusion

This paper has established that adversarial perturbations are a severe and real menace to CNN-based traffic sign recognition systems, and robustness to attack differs significantly among architectures and attack techniques. The major themes that come out of this exploration are that the effectiveness of the attack largely depends on the architecture design and that the I-FGSM stability plateau that exists in all models undermines the common belief that iterative attacks are more destructive than single-step approaches. Moreover, t-SNE feature space visualizations were a good indication that adversarial attacks essentially corrupt the internal representations that the models have learned, and that robustness cannot be measured by clean accuracy alone. Considering these findings, there is a strong sentiment that the deep learning community should focus on the creation and incorporation of strong defense systems such as adversarial training, certified defenses, and input sanitization that are specifically sensitive to the vulnerability characteristics of specific architectures. Expectantly, the results of this study will be valuable as a reference and diagnostic tool to researchers and practitioners that are on the path to designing secure and reliable deep learning systems and that future studies will elaborate on the trade-offs between robustness, accuracy and computational efficiency in the larger context of

autonomous driving and intelligent transportation systems.

REFERENCES

J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The German traffic sign recognition benchmark: a multi-class classification competition," in *The 2011 international joint conference on neural networks*, 2011: IEEE, pp. 1453–1460.

K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale Proc. Int. Conf. Learning Representations (ICLR), 2015.

K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," *Proc. Int. Conf. Machine Learning (ICML)*, 2019.

C. Szegedy et al., "Intriguing properties of neural networks," *Proc. Int. Conf. Learning Representations (ICLR)*, 2014.

Eykholt K, Evtimov I, Fernandes E, Li B, Rahmati A, Xiao C, Prakash A, Kohno T, Song D. Robust physical-world attacks on deep learning visual classification. InProceedings of the IEEE conference on computer vision and pattern recognition 2018 (pp. 1625-1634).

Sermanet P, LeCun Y. Traffic sign recognition with multi-scale convolutional networks. InThe 2011 international joint conference on neural networks 2011 Jul 31 (pp. 2809-2813). IEEE.

Dan C, Ueli M, Jonathan M, Jürgen SH. Multi-column deep neural network for traffic sign classification. Neural networks. 2012 Aug;32(1):333-8.

I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *Proc. Int. Conf. Learning Representations (ICLR)*, 2015.

A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," *Proc. Int. Conf. Learning Representations (ICLR)*, 2017.

A. Madry et al., "Towards deep learning models resistant to adversarial attacks," *Proc. Int. Conf. Learning Representations (ICLR)*, 2018.

A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," in *Proc. 35th Int. Conf. Mach. Learn. (ICML)*, Stockholm, Sweden, 2018, pp. 284–293.

Sitawarin C, Bhagoji AN, Mosenia A, Chiang M, Mittal P. Darts: Deceiving autonomous cars with toxic signs. arXiv preprint arXiv:1802.06430. 2018 Feb 18.

Zhang, J., Li, W., & Ogunbona, P. (2020). Adversarial robustness of deep learning models for traffic sign recognition. IEEE Transactions on Intelligent Transportation Systems, 22(10), 6525–6535.

Croce, F., & Hein, M. (2020). Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. Proceedings of the 37th International Conference on Machine Learning (ICML), 2206–2216.

Li, H., Xu, Z., Taylor, G., Studer, C., & Goldstein, T. (2018). Visualizing the loss landscape of neural nets. Advances in Neural Information Processing Systems (NeurIPS), 31.

J. Greenhalgh and M. Mirmehdi, "Recognizing text-based traffic signs," *IEEE Trans. Intelligent Transportation Systems*, vol. 16, no. 3, pp. 1360–1369, 2014

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z.B. and Swami, A., 2017, April. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security* (pp. 506-519).

A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2020.

M. Naseer, S. Khan, and F. Porikli, "Cross-domain transferability of adversarial perturbations," in *Advances in Neural Inf. Process. Syst. (NeurIPS)*, 2019.

S. Addepalli, S. Jain, and R. V. Babu, "Scaling adversarial training to large perturbation bounds," in *Proc. European Conf. Computer Vision (ECCV)*, Tel Aviv, Israel, 2022, pp. 301–316.