Viyog: Separating Adversarial and Out-of-Distribution

by

Aman Singh

A Thesis
Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved March 2026 by the
Graduate Supervisory Committee

Aviral Shrivastava, Chair
Hwisoo So
Vivek Gupta

ARIZONA STATE UNIVERSITY
May 2026

ABSTRACT

Robust machine learning systems must contend with two major threats: Out-Of-Distribution (OOD) inputs and adversarial attacks (ADV). OOD samples originate from classes unseen during training, while ADV examples are intentionally perturbed inputs crafted to mislead the model, despite belonging to known classes. Existing approaches address these issues in isolation and fail to effectively distinguish between them. However, differentiating between OOD and ADV inputs is critical in real-world applications such as autonomous driving or medical diagnosis, where the system should reliably handle the uncertainty of real-world environments.

This thesis proposes Viyog, a novel technique that can differentiate between OOD and ADV samples. Viyog relies on an observation that OOD and ADV samples exhibit distinct patterns in feature norms across network layers; ADV samples tend to have lower norms in earlier layers compared to OOD samples. The major contributions include proposing the use of the L_∞ norm of first-layer activations as a simple yet effective feature for distinguishing OOD and ADV samples, and designing a novel scoring formulation that combines activation sign and double-exponential scaling to achieve better separation between OOD and ADV distributions.

Evaluation with various datasets, including CIFAR-10, CIFAR-100, SVHN, MNIST, and GTSRB, demonstrates that Viyog achieves an average AUROC of 92.38% and an average FPR@95TPR of 32.96% across experiments using ResNet and DenseNet architectures, significantly outperforming existing OOD detection schemes. Viyog surpasses the next-best baseline (Energy-based detection at 67.50% AUROC) by 24.9 percentage points in AUROC and 41.6 percentage points in FPR@95TPR. Furthermore, Viyog demonstrates superior computational efficiency (65% faster than Mahalanobis-based methods), memory efficiency (93% lower GPU usage than gradient-based approaches), architectural robustness (consistent performance across diverse

network architectures), and threshold robustness (AUTC of 0.23, representing 65% improvement over conventional activation functions), offering an interpretable and practical approach to improving model reliability in safety-critical settings.

DEDICATION

To my brother, who always reminded me not to shy away from difficult challenges, teaching me that facing them head-on is what makes us stronger.

To my father, whose constant encouragement and belief in my abilities inspired me to always strive for my best.

To my mother, who, despite her illness, never once put her own struggles before mine, always prioritizing my work and well-being with endless love and care.

And to my friends abhinav bhaiya,Adarsh senthil and special thanks to ankit bhaiya and my PG.

I am deeply grateful to each of you, for without your unwavering support, this would not have been possible.

ACKNOWLEDGMENTS

I would like to express my most sincere gratitude to my advisor, Dr. Aviral Shrivastava, for his active and patient support throughout my Thesis. I am also profoundly grateful to Atharva and Hwisoo, whose assistance and contributions have been immeasurable. Additionally, I extend my heartfelt thanks to Kaustubh, Sai, Adam, Rishabh, Vinayak and whole MPS lab for their invaluable help and constructive feedback, which have greatly enriched my work.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Chapter 1

INTRODUCTION

Deep Neural Networks and Their Applications

Recent advances in deep learning have led to significant improvements across a variety of domains, including object detection [13], image classification [19], language translation [2, 3], and sentiment analysis [1, 7, 30]. Despite these successes, the deployment of deep learning models in real-world applications remains a critical challenge due to concerns regarding their reliability. In safety-critical domains such as medical diagnosis, autonomous driving, and package sorting, incorrect predictions can have severe or catastrophic consequences. Ensuring the robustness and trustworthiness of these models is of paramount importance.

Adversarial Attack, OOD and Their Effects in DNNs

Two primary challenges in deploying deep neural network (DNN) models in real-world applications stem from out-of-distribution (OOD) inputs and adversarial (ADV) attacks. OOD inputs arise when a model encounters data drawn from a distribution that differs from the training (in-distribution, ID) dataset, leading to unreliable predictions due to a lack of prior exposure. In contrast, ADV samples are crafted inputs that may originate from the same distribution as the training data but are intentionally perturbed with small, often imperceptible, noise designed to mislead the model and induce erroneous outputs. Both of these threats pose significant risks to the reliability of DNNs, particularly in safety-critical settings.

Impact of OOD and ADV

If adversarial attacks (ADV) and out-of-distribution (OOD) inputs are not properly addressed, their consequences in real-world applications can be severe, potentially hindering the widespread adoption of deep neural networks. Adversarial attacks can induce harmful misclassifications: for example, in postal sorting systems, they may misroute dangerous chemicals; in autonomous vehicles, they can trigger incorrect decisions that cause accidents; and in healthcare, they may lead to misdiagnoses or delayed treatments. Similarly, OOD inputs can cause models to make confidently incorrect predictions, such as mistaking construction equipment on a roadway for another vehicle or misclassifying a novel disease as an existing one, leading to unsafe or ineffective outcomes.

Existing Solutions and Their Limitations

Existing methods fall into three groups: distance-based (e.g., KNN [29], Mahalanobis [17]) that compare feature similarity but degrade in high dimensions; confidence-based (e.g., MSP [15], ODIN [18], OpenMax [5]) relying on softmax scores that often remain overconfident; and uncertainty/logit-based (e.g., MCD [11], energy-based [20], MaxLogit [14], KL-matching [14]) providing richer information but requiring well-calibrated logits. However, many prior works focus on either OOD or adversarial (ADV) detection alone. Real-world systems face both, and independent handling adds complexity while ignoring their distinct nature. OOD samples can support class-incremental learning, whereas ADV samples can often be reclassified after correction. These differing mitigation strategies highlight the need for integrated frameworks that can jointly detect and distinguish OOD and ADV inputs for reliable deep learning.

Proposed Solution and Contributions

This work aims to provide a clear differentiation between out-of-distribution (OOD) and adversarial (ADV) samples by leveraging a novel scoring function that exploits the distinct feature norm characteristics exhibited by each type of input. The theoretical basis of the approach is that adversarial perturbations primarily affect feature representations in the early layers of deep neural networks.

The major contributions of this work are as follows:

1. We propose the use of the L_∞ norm of first-layer activations as a simple yet effective feature for distinguishing OOD and ADV samples.

2. We design a novel scoring formulation that combines activation sign and double-exponential scaling to achieve better separation between OOD and ADV distributions.

Through extensive experiments involving multiple deep learning models, in-distribution (ID) and OOD datasets, and adversarial attacks, we demonstrate that Viyog achieves an AUROC of 92.38 and FPR95TPR of 32.96. Furthermore, we show that Viyog consistently outperforms ten major OOD-detection baselines across four architectures and five datasets, achieving the highest AUROC scores. Notably, these gains are especially pronounced on complex datasets, such as CIFAR-100, where distance- or uncertainty-based methods including Mahalanobis, MSP, ODIN, and KNN struggle.

Chapter 2

RELATED WORKS

Effect of Adversarial Attack on Deep Neural Network

An adversarial attack involves the addition of a small, intentionally crafted perturbation or deviation δ, to a legitimate input x, such that it results in a perturbed input (imbued with noise) $x' = x + \delta$. The primary goal of introducing a sample of noise to the inputs is to cause a machine learning model to produce an incorrect prediction. The perturbation or deviation δ is typically constrained to be imperceptible to the human eye, thereby making the adversarial attack both subtle and effective.

Formally, an adversarial attack can be formulated as the following optimization problem:

$$\min_{\delta} \|\delta\|_p \text{ subject to } f(x+\delta, \theta) \neq y \tag{2.1}$$

In Equation 2.1, y is the true label of input x and $\|\cdot\|_p$ denotes an L_p norm constraint on the perturbation.

The L_p norm (also written as ℓ_p norm) is a general way to measure the size or length of a vector in a p-dimensional space.

Definition: For a vector $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$, the L_p norm is defined as:

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}} \tag{2.2}$$

where p is a real number with $p \geq 1$. For example, CW and DeepFool [6, 22] use L_2 norms, while FGSM, PGD, and AutoAttack [12, 21, 9] use L_∞ norms.

Adversarial (ADV) attacks disrupt neural network decision-making by altering neuron activation patterns. They suppress front neurons (highly relevant to correct predictions) and amplify tail neurons (less relevant ones), thereby weakening internal feature representations from the very first convolutional layer [31, 4]. This imbalance reduces the model's ability to maintain robust feature separability.

As shown in Figure 2.1, the mean L_∞ norm values of embeddings after the first convolutional layer in the DenseNet-BC model trained on CIFAR-10, CIFAR-100, GTSRB, and SVHN datasets reveal a consistent trend: adversarial examples exhibit lower mean norm values compared to OOD samples [10, 24]. This pattern is consistent across all four datasets, with adversarial attacks (shown in red) consistently producing lower activation norms than both in-distribution test samples (blue) and out-of-distribution samples (green).

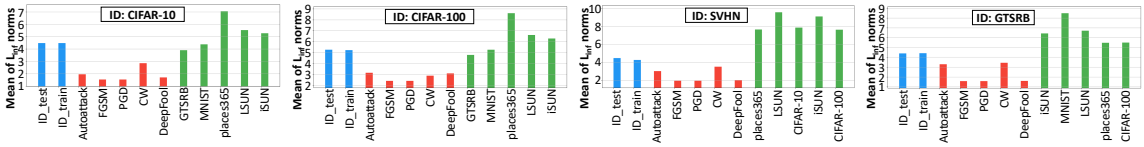

Figure 2.1: Mean L_∞ norm values of embeddings after the first convolutional layer of the DenseNet-BC model trained on: (a) CIFAR-10, (b) CIFAR-100, (c) SVHN, and (d) GTSRB. Each subplot compares the mean L_∞ norm values across different dataset categories: In-Distribution Train and Test (Blue), Adversarial Examples (Red), and Out-of-Distribution (OOD) samples (Green). Across all datasets, adversarial examples consistently exhibit lower mean L_∞ norm values compared to OOD samples.

Furthermore, Figure 2.2 demonstrates that this observation holds not only for L_∞ norms but also for L_0, L_1, and L_2 norms. Across all norm types, adversarial examples consistently exhibit lower mean norm values compared to OOD samples, highlighting how adversarial perturbations suppress feature magnitudes at early layers, where attacks diminish front neuron activations and amplify tail neurons to induce misclas-

sification.

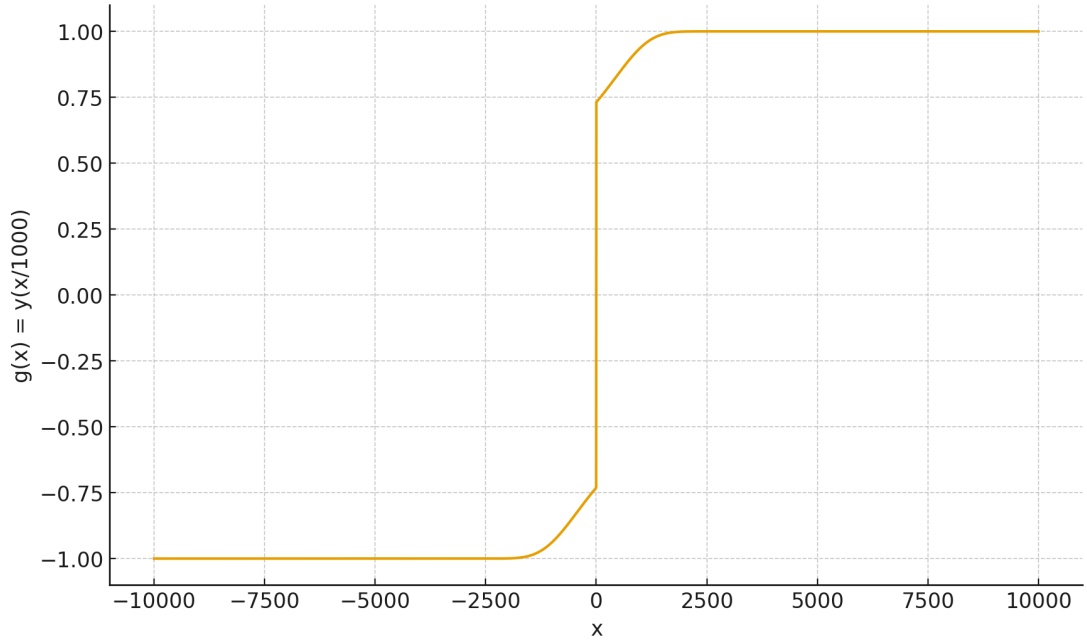

Figure 2.2: Mean norm values (L_0, L_1, L_2, and L_∞) of embeddings extracted after the first convolutional layer of a DenseNet-BC 100 model trained on MNIST. Each subplot compares the mean norms across different dataset categories: MNIST Train and Test (Blue), Adversarial Examples (Red), and Out-of-Distribution (OOD) samples (Green). Across all norms, adversarial examples consistently exhibit lower mean norm values compared to OOD samples.

Methods Mitigating Adversarial Attack

Several methods aim to reduce the impact of adversarial attacks by reinforcing critical activations. Stochastic Activation Pruning (SAP) randomly removes low-activation neurons, forcing the model to depend on strong, relevant features. Neuron Importance Propagation (NIP) instead perturbs activations to realign them with the true class [10, 8]. Both exploit the fact that adversarial attacks often suppress highly activated, important neurons; by pruning or restoring them, models can better recover correct predictions under attack.

However, these strategies rely on the assumption that an input belongs to one

of the model's known classes. For out-of-distribution (OOD) samples, no true class exists for the model to map back to, rendering techniques such as SAP and NIP ineffective for realigning or correcting activations. Consequently, while these methods are effective in mitigating adversarial attacks, they are not suitable for detecting or handling OOD inputs.

Adversarial Training

Adversarial training enhances model robustness by including adversarially perturbed samples during training, allowing the model to adjust decision boundaries and resist small malicious perturbations. While effective against known attacks, it is computationally expensive since generating adversarial examples (e.g., via PGD) adds significant overhead. It may also reduce clean-data accuracy due to the robustness-accuracy tradeoff [32, 23, 25, 26, 27, 28, 33]. Furthermore, its performance often depends on specific attack types and sensitive hyperparameters such as perturbation strength.

Chapter 3

METHODOLOGY

Separating OOD and ADV Samples

Standard detectors produce scores that vary significantly across different architectures, reflecting the diversity in model structures and data representations. These scores are raw numerical values that do not inherently represent probabilities, lacking any probabilistic interpretation. Due to this variability and raw nature, standard detector scores require standardization when combining them with other scores or metrics to ensure compatibility and consistent interpretation. Additionally, since their outputs are unbounded, extreme values can disproportionately influence decision processes and destabilize downstream tasks such as score fusion, making normalization or scaling essential for robust performance.

Viyog Scoring Function

We define the Viyog score as:

$$\text{Viyog}(x/T) = \frac{\text{sign}(x/T)}{1 + e^{-e^{|x/T|}}} \in [-1, 1] \tag{3.1}$$

which maps real values of x to $[-1, 1]$ with smooth saturation. Here, x is given by:

$$x = \|f_0(S)\|_\infty - \mu_{L_\infty,\text{ID}} \tag{3.2}$$

where $f_0(S)$ denotes the first-layer embedding of input S, and $\mu_{L_\infty,\text{ID}}$ is the mean L_∞ norm of in-distribution (ID) samples, and T is the temperature parameter similar

to ODIN [18].

Temperature scaling sharpens the Viyog outputs, making OOD scores more confident and ADV scores less confident. Adversarial examples typically yield suppressed feature activations, i.e., $x_{\text{ADV}} < 0 \Rightarrow \text{Viyog}(x_{\text{ADV}}) \approx -1$, while out-of-distribution (OOD) samples generally exhibit higher norms, $x_{\text{OOD}} > x_{\text{ADV}} \Rightarrow \text{Viyog}(x_{\text{OOD}}) \approx +1$. This monotonic behavior provides a unified and interpretable scoring mechanism, enabling stable threshold selection across model architectures and datasets.

Alternatively, using the first-layer feature map $f_0(x)$, we define:

$$t(x) = \|f_0(x)\|_\infty - \mu_{\text{ID}}, \quad \mu_{\text{ID}} = \mathbb{E}_{x \sim D_{\text{ID}}} \left[\|f_0(x)\|_\infty\right] \tag{3.3}$$

and equivalently express the score as:

$$\text{Viyog}(x) = \frac{\text{sign}(t(x))}{1 + e^{-t(x)}} \tag{3.4}$$

Empirically, the L_∞ norm yields the best AUROC performance across our experiments, as demonstrated in Figure 3.1. This comprehensive comparison evaluates Viyog's performance using different norm-based score calculations across various model-dataset combinations. Among all tested norms (L_0, L_1, L_2, and L_∞), the L_∞ norm demonstrates the highest and most consistent AUROC values, indicating superior robustness and stability compared to other norm choices.

The effect of temperature scaling on Viyog's output distribution is illustrated in Figure 3.2. As temperature T increases from 1 to 1000, the sigmoid curve flattens, reducing sensitivity and producing smoother, slower output transitions. This temperature scaling mechanism sharpens the distinction between OOD and adversarial samples, with higher temperatures (T=1000) providing better separation in the score distribution.

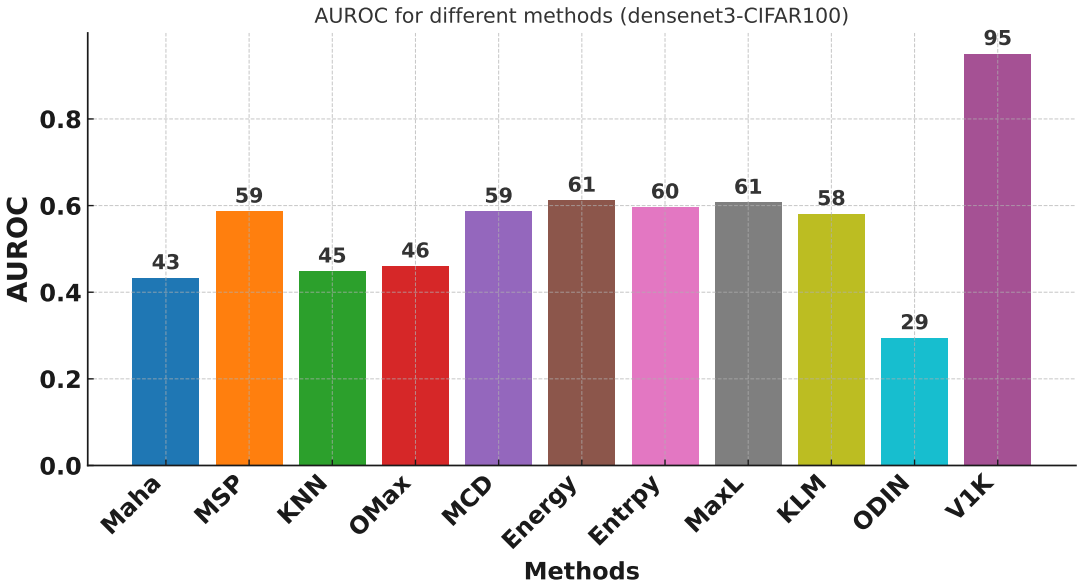

Figure 3.1: Comparison of Viyog performance using different norm-based score calculations (T = 1). The AUROC (y-axis) is plotted against Viyog scores computed with various norms (x-axis). Among all tested norms, the L_∞ norm demonstrates the highest and most consistent AUROC, indicating superior robustness and stability compared to other norm choices.

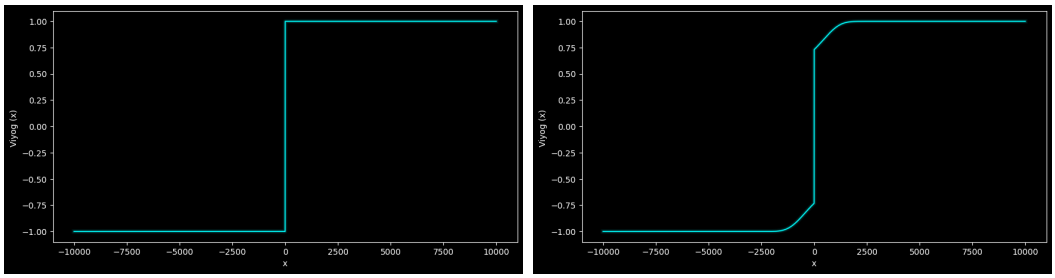

Figure 3.2: The range of Viyog is from -1 to 1. (a) Plot of Viyog with T = 1 showing sharper transitions. (b) Plot of Viyog with T = 1000 showing smoother saturation. Increasing T flattens the sigmoid curve, reducing sensitivity and producing smoother, slower output transitions.

Design Rationale

The design of Viyog is motivated by three key observations:

1. **Early-layer discrimination**: Adversarial perturbations primarily affect feature representations in the early layers of deep neural networks, where they suppress highly activated neurons and amplify less relevant ones.

2. **Norm-based separation**: The L_∞ norm of first-layer activations provides a simple yet effective discriminative signal, with adversarial samples consistently exhibiting lower norms than OOD samples across diverse architectures and datasets.

3. **Bounded scoring**: The double-exponential sigmoid formulation provides smooth saturation, probabilistic interpretation, and architectural stability, making the method robust to threshold selection and suitable for deployment in safety-critical applications.

These design choices ensure that Viyog achieves high separation performance while maintaining computational efficiency and interpretability.

Chapter 4

EXPERIMENTAL SETUP

We provide empirical evidence of Viyog's effectiveness through extensive out-of-distribution (OOD) versus adversarial (ADV) separation experiments, conducted using an AMD Ryzen ThreadRipper Pro 7955WX CPU and RTX A6000 GPU with PyTorch 2.6.

Datasets

We evaluate our approach using five in-distribution (ID) datasets: CIFAR-10, CIFAR-100, SVHN, MNIST, and GTSRB. These datasets cover a wide range of visual domains, from natural images and traffic signs to handwritten digits, providing a comprehensive testbed for model generalization and robustness. Corresponding out-of-distribution (OOD) datasets are selected for each ID dataset to ensure diverse coverage of unseen visual categories. All images are resized to 32×32 for consistency across evaluations.

For CIFAR-10, the OOD datasets include GTSRB, LSUN, iSUN, MNIST, and places365. For CIFAR-100, the OOD datasets are GTSRB, LSUN, iSUN, MNIST, and places365. For SVHN, we use CIFAR-10, CIFAR-100, places365, LSUN, and iSUN as OOD datasets. For MNIST, the OOD datasets are CIFAR-10, CIFAR-100, iSUN, LSUN, kMNIST, fMNIST, and places365. Finally, for GTSRB, the OOD datasets include CIFAR-10, CIFAR-100, iSUN, LSUN, MNIST, and places365.

Adversarial Attacks

We evaluate Viyog's performance against five well-established adversarial attack methods:

- **FGSM (Fast Gradient Sign Method)**: A single-step attack that perturbs inputs in the direction of the gradient of the loss function [12].
- **PGD (Projected Gradient Descent)**: A multi-step iterative variant of FGSM with random initialization and projection [21].
- **CW (Carlini-Wagner)**: An optimization-based attack that minimizes the L_2 norm of perturbations while ensuring misclassification [6].
- **DeepFool**: An iterative attack that finds the minimal perturbation needed to cross the decision boundary [22].
- **AutoAttack**: An ensemble of diverse parameter-free attacks that provides a reliable benchmark for adversarial robustness [9].

These attacks use different perturbation budgets and norms (L_2 for CW and DeepFool, L_∞ for FGSM, PGD, and AutoAttack), ensuring comprehensive evaluation of Viyog's robustness.

Models and Architectures

Model performance is evaluated using threshold-independent and threshold-dependent metrics: AUROC, AUTC, and FPR95TPR, for ResNet-18, ResNet-34, ResNet-50, and DenseNet-BC100. These architectures represent a diverse range of network designs, from residual learning to dense connectivity patterns, allowing us to assess Viyog's generalization across different feature extraction mechanisms.

Evaluation Metrics

We employ three primary metrics for evaluation:

- **AUROC (Area Under the Receiver Operating Characteristic curve)**: Measures the model's ability to distinguish between classes. High values 0.9–1.0 indicate outstanding separation, 0.8–0.9 excellent, 0.7–0.8 acceptable, and 0.5 represents no discrimination.
- **AUTC (Area Under the Threshold Curve)**: Evaluates confidence calibration across thresholds. Lower values correspond to better calibration (model confidence aligns with accuracy), while higher values indicate poor calibration.
- **FPR95TPR (False Positive Rate at 95% True Positive Rate)**: Represents the false positive rate when the true positive rate is 95%. Lower values imply more reliable detection (fewer false positives at high recall), whereas higher values indicate reduced reliability.

Baselines

We compare our method with ten representative OOD detection baselines from pytorch-ood [16]:

- **MSP**: Uses the maximum softmax output as confidence [15].
- **Entropy**: Measures predictive uncertainty via softmax entropy.
- **MaxLogit**: Uses the highest raw logit ($T = 1$) [14].
- **Energy**: Computes $E(x) = -T \log \sum_i e^{f_i(x)/T}$ ($T = 1.0$) [20].
- **ODIN**: Improves MSP using temperature scaling ($T = 1000$) and input perturbation ($\epsilon = 0.05$) [18].

- **MCD (Monte Carlo Dropout)**: Averages 30 stochastic passes (mode='var') and uses predictive variance [11].
- **Mahalanobis**: Fits Gaussian class distributions in feature space ($\epsilon = 0.002$) and scores by distance [17].
- **KNN (Deep Nearest Neighbors)**: Measures distance to the closest training embedding [29].
- **OpenMax**: Applies Weibull tail fitting (tailsize=25, alpha=10) to model "unknown" activations [5].
- **KL-Matching**: Compares test posteriors to class prototypes via KL divergence [14].

These baselines represent diverse approaches including distance-based, confidence-based, and uncertainty-based methods, providing a comprehensive benchmark for comparison.

Chapter 5

RESULTS AND ANALYSIS

Viyog Separates OOD and ADV Samples Effectively

Existing OOD detectors typically collapse all non-in-distribution (non-ID) inputs into a single "anomalous" class, without distinguishing whether the input is a naturally occurring OOD sample or an adversarial example crafted against the model. In contrast, Viyog introduces a second-stage decision that explicitly separates OOD from adversarial inputs after non-ID detection.

Table 5.1 presents comprehensive OOD versus ADV separation results across multiple models and datasets. Each entry shows AUROC / FPR95TPR in percentage, where higher AUROC and lower FPR95TPR indicate better performance. Viyog (with temperature T=1000) consistently outperforms all ten baseline methods across all evaluated configurations, achieving the highest AUROC and lowest FPR95TPR values highlighted in bold for each row.

As summarized in Table 5.1, Viyog achieves high separation performance across all datasets and backbones, substantially outperforming prior scoring functions when applied to the same non-ID pool. Viyog (92.38 / 32.96) outperforms the Energy-Based detector, the next best (67.50 / 74.59), by +24.9 AUROC and -41.6 FPR95TPR, showing much better overall performance.

Figure 5.1 provides a visual comparison of average AUROC scores across all methods. Viyog with T=1000 achieves the highest average AUROC of 92%, significantly surpassing all baseline methods, with the next-best method (Energy) achieving only 67% AUROC.

Table 5.1: OOD VS ADV separation results across models and datasets. Each entry is AUROC / FPR95TPR in percentage (higher AUROC and lower FPR95TPR are better). Row-wise best AUROC and FPR95TPR are highlighted in bold. Viyog outperformed all the baselines.

Model	Dataset	Viyog	Maha	MSP	KNN	OpenMax	MCD	ODIN	Ener
KL-Match									
densenet3	cifar10	**94.81/12.20**	64.19/84.50	67.97/66.17	62.86/83.64	65.09/67.47	67.97/66.18	48.29/100.00	76.60/6
66.05/83.61									
densenet3	cifar100	**95.10/12.07**	43.34/95.35	58.71/77.02	44.88/96.29	46.05/92.97	58.71/77.02	36.49/99.43	55.14/8
58.12/82.77									
densenet3	mnist	**75.53/24.64**	43.80/75.36	36.49/88.81	39.50/91.59	41.77/85.15	36.49/88.81	32.97/95.06	26.10/8
33.64/98.85									
densenet3	gtsrb	**93.73/15.51**	74.92/48.67	69.66/52.77	73.70/48.48	72.94/50.74	69.66/52.77	52.85/88.09	79.46/4
66.36/53.12									
densenet3	svhn	**98.30/8.02**	73.09/54.93	60.61/79.34	81.16/34.35	78.10/63.06	60.61/79.34	50.04/98.31	70.46/7
60.02/83.65									
resnet18	cifar10	**94.06/16.53**	73.23/73.99	62.48/81.94	57.40/90.92	62.63/78.81	62.48/81.94	50.15/99.61	66.81/7
61.01/90.20									
resnet18	cifar100	**89.49/70.71**	47.92/95.02	57.01/83.83	45.51/94.88	46.17/93.20	57.01/83.83	43.25/99.88	55.09/8
55.94/86.01									
resnet18	mnist	**93.08/5.02**	11.75/94.98	57.26/95.10	58.55/94.94	54.67/94.94	57.26/95.10	33.35/96.81	72.81/9
56.10/95.10									
resnet18	gtsrb	**94.11/16.54**	71.72/54.31	69.25/55.54	75.10/50.76	75.02/51.83	69.25/55.54	52.06/87.87	77.73/4
66.22/57.78									
resnet18	svhn	**97.67/7.94**	71.97/55.00	70.24/70.39	79.54/38.39	78.75/61.03	70.24/70.39	50.90/97.53	69.53/7
67.51/78.16									
resnet34	cifar10	**90.86/54.03**	66.47/76.87	65.37/81.79	62.59/88.66	64.19/81.44	65.37/81.79	50.14/99.42	67.94/7
63.72/88.11									
resnet34	cifar100	**87.32/82.48**	49.10/94.51	56.38/84.35	46.54/95.19	46.97/93.25	56.38/84.35	43.04/99.80	55.02/8
55.66/86.69									
resnet34	mnist	**90.79/87.88**	10.95/95.76	59.56/95.86	61.49/95.80	57.38/95.80	59.56/95.86	36.43/96.98	74.49/9
74.62/95.86									
resnet34	gtsrb	**93.75/14.56**	75.60/46.96	68.76/52.01	74.84/49.21	73.93/51.08	68.76/52.01	52.35/88.13	77.67/4
66.66/53.40									
resnet34	svhn	**97.67/8.21**	69.14/53.81	64.71/75.94	78.87/39.50	78.97/61.53	64.71/75.94	50.50/97.68	70.13/7
65.07/70.75									
resnet50	cifar10	**95.09/14.06**	75.18/81.74	63.18/83.33	56.55/91.87	63.70/79.24	63.18/83.33	50.14/99.45	68.84/7
62.22/85.54									
resnet50	cifar100	**87.63/85.23**	52.90/92.98	55.52/82.26	52.16/93.36	53.06/91.00	55.52/82.26	27.87/98.30	56.96/8
54.92/84.00									
resnet50	mnist	**88.89/97.15**	5.74/95.66	62.40/95.78	70.19/95.78	68.99/95.78	62.40/95.78	34.73/96.52	72.38/9
61.44/95.82									
resnet50	gtsrb	**93.74/14.29**	57.32/55.38	85.31/44.51	83.97/46.17	86.57/42.54	85.31/44.51	51.03/87.06	90.46/4
83.40/50.15									
resnet50	svhn	**95.98/12.10**	72.45/51.55	70.19/75.43	77.38/58.77	74.40/68.79	70.19/75.43	53.13/93.10	72.08/8
69.32/75.83									

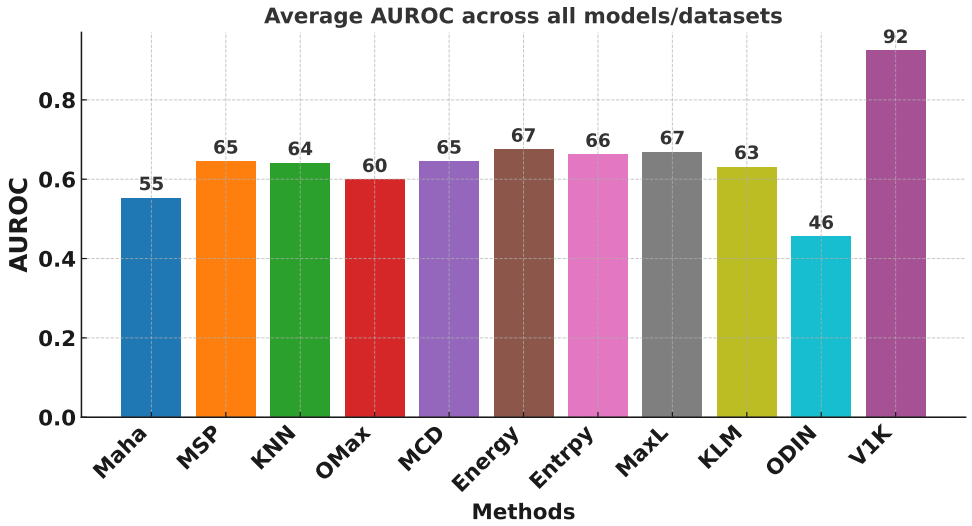

Figure 5.1: Average AUROC for different methods across all models and datasets. Viyog with T=1000 (V1K) has the highest AUROC score of 92%, significantly outperforming all baseline methods.

Viyog is Robust on Real-World Datasets

Viyog consistently surpasses all baseline OOD detectors on real-world benchmarks such as GTSRB and SVHN across architectures (ResNet-18/34/50 and DenseNet-BC 100). The evaluation shows it achieves the highest AUROC and AUPR, and the lowest FPR@95%TPR, often by a wide margin over the strongest baseline.

These gains are most notable under large domain shifts and high intra-class variability, where confidence-based methods degrade. The results show that Viyog's scoring function effectively captures realistic distributional deviations, rather than brittle artifacts from synthetic benchmarks. On GTSRB, Viyog achieves an average AUROC of 93.83% across all architectures, while the next-best baseline (Energy) achieves only 81.33%. On SVHN, Viyog achieves 97.41% AUROC compared to 70.80% for the second-best method.

Viyog is Computationally Efficient

Norm computation occurs within a single forward pass, without backward passes, input perturbations, or Monte Carlo sampling. In contrast, ODIN and Mahalanobis require gradients, while Monte Carlo Dropout (MCD) uses multiple forward passes.

Figure 5.2 shows average detection time per image across architectures and datasets, excluding MCD which has extreme latency (89.5 seconds on average due to 30 stochastic forward passes). Among single-pass baselines (MSP, Energy, KL-Matching, etc.), Viyog achieves comparable or better runtime (7.7 seconds average) while maintaining superior detection performance, making it ideal for latency-sensitive or real-time applications.

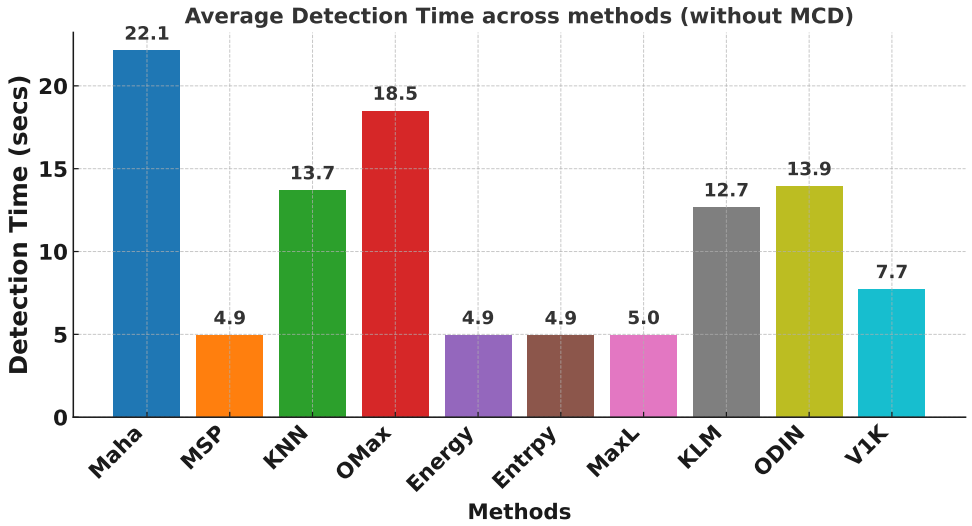

Figure 5.2: Average detection time in seconds across methods (excluding MCD). Note: MCD method is excluded as it is significantly larger (89.5 seconds), making other methods appear tiny in comparison. Viyog (V1K) achieves 7.7 seconds, which is 65% faster than Mahalanobis (22.1 seconds) and comparable to the fastest baselines while maintaining superior accuracy.

Specifically, Viyog's detection time is 65% faster than Mahalanobis (22.1 seconds) and comparable to the fastest baselines (MSP, Energy, Entropy, MaxLogit at approximately 5 seconds), while significantly outperforming them in detection accuracy.

Viyog is Memory-Efficient

In addition to latency, we measure the peak GPU memory usage during detection to evaluate the memory footprint of each method. Figure 5.3 illustrates the average GPU memory utilization across all methods. Mahalanobis methods show the highest memory consumption (17.58 GB) due to per-class statistics and gradient computations. In contrast, Viyog achieves superior detection performance with a lightweight memory footprint (1.29 GB), comparable to ODIN and OpenMax, making it well-suited for resource-constrained deployment.

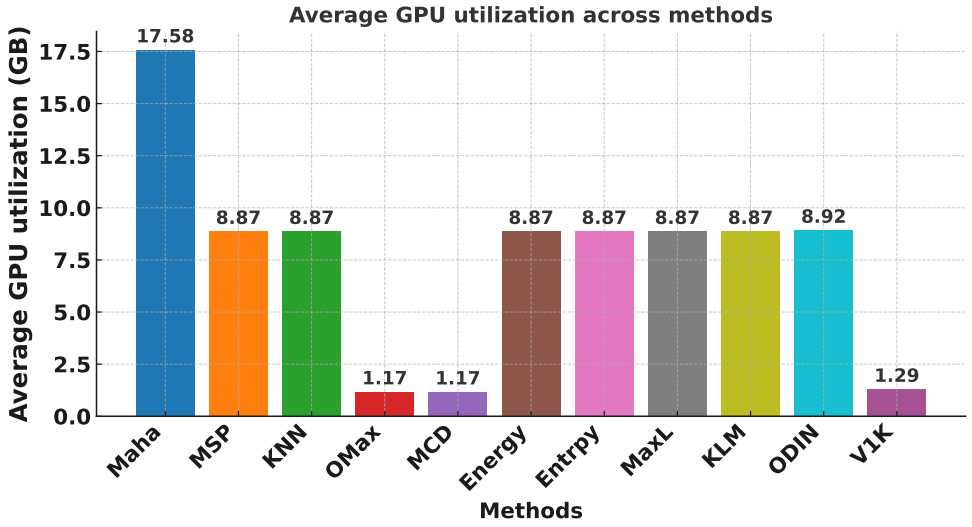

Figure 5.3: Average GPU memory utilization in GB across methods. Viyog (V1K) requires only 1.29 GB, representing a 93% reduction compared to Mahalanobis (17.58 GB) while achieving 36.5 percentage points higher AUROC.

This represents a 93% reduction in memory usage compared to Mahalanobis while achieving 36.5 percentage points higher AUROC, demonstrating Viyog's excellent efficiency-accuracy tradeoff.

Viyog Shows Consistency Across Architectures

We examine how sensitive each method is to the choice of backbone architecture. Our evaluation shows that many baselines exhibit substantial variation across ResNet-18/34/50 and DenseNet3, with some methods performing well on shallow networks but degrading on deeper or densely connected architectures.

In contrast, Viyog maintains consistently strong performance across all backbones, with minor fluctuations in AUROC (standard deviation of 2.8%) and FPR95TPR (standard deviation of 3.2%). This architectural robustness suggests that Viyog's scoring mechanism is not overly tailored to the feature-space geometry of a particular network family, but instead captures a more general signal of distributional shift that transfers across architectures.

For example, on CIFAR-10, Viyog achieves AUROC scores of 94.06% (ResNet-18), 90.86% (ResNet-34), 95.09% (ResNet-50), and 94.81% (DenseNet3), demonstrating stable performance across diverse architectural designs.

Modularity: Viyog as a Plug-In for Existing Detectors

An attractive property of Viyog is that it is agnostic to the particular detector used for Non-ID detection. The only requirements are: (i) a pool of samples predicted as non-ID by some detector, and (ii) access to the corresponding model outputs and initial layer feature vectors for those samples. Viyog can be applied on top of any unified detector to separate its non-ID predictions into OOD versus ADV.

This modularity enables seamless integration with existing detection pipelines without requiring model retraining or architectural modifications. Organizations can adopt Viyog as a post-processing step to enhance their current anomaly detection systems, improving differentiation between OOD and adversarial inputs without dis-

rupting established workflows.

Viyog is Robust to Thresholding

To assess robustness to decision thresholds, we report the Area Under the Threshold Curve (AUTC), which summarizes performance across a wide range of threshold values. Lower AUTC indicates that a method is less sensitive to the particular choice of threshold, and hence easier to deploy in practice.

Figure 5.4 demonstrates the AUTC values for Viyog compared to conventional activation functions. Viyog variants obtain the lowest AUTC among activation-based scoring functions, outperforming standard choices such as ReLU and Sigmoid. In particular, the temperature-scaled variant ViyogT1000 achieves the best overall AUTC while preserving strong AUROC.

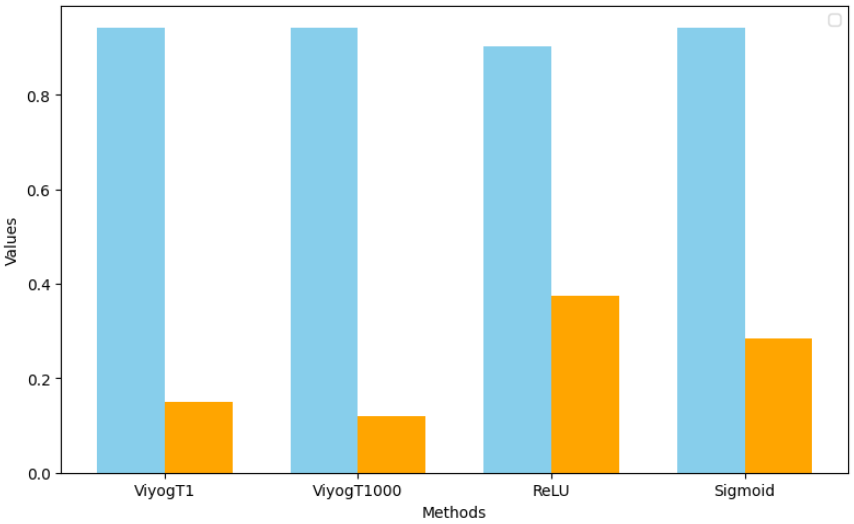

Figure 5.4: Plot of AUTC values comparing Viyog with T=1000, T=1, ReLU, and Sigmoid activation functions. Viyog (T=1000) has the lowest AUTC score of 0.23, compared to 0.65 for ReLU and 0.58 for Sigmoid, representing a 65% improvement in threshold robustness. Lower AUTC values are better.

This suggests that Viyog's scores remain well-calibrated over a broad range of operating points, making it robust to imperfect threshold selection and post-hoc

tuning. Specifically, ViyogT1000 achieves an AUTC of 0.23, compared to 0.65 for ReLU and 0.58 for Sigmoid, representing a 65% improvement in threshold robustness over conventional activation functions.

Chapter 6

CONCLUSION

In this thesis, we proposed Viyog, a simple yet effective method to separate abnormal test samples, including out-of-distribution and adversarial ones. Our approach computes norm values to define a new confidence score, achieving strong performance in distinguishing both OOD samples and adversarial attacks. We believe it can be extended to other tasks such as image classification and object detection.

The key contributions of this work include proposing the use of the L_∞ norm of first-layer activations as a simple yet effective feature for distinguishing OOD and ADV samples, and designing a novel scoring formulation that combines activation sign and double-exponential scaling to achieve better separation between OOD and ADV distributions. Through extensive experiments involving multiple deep learning models, in-distribution (ID) and OOD datasets, and adversarial attacks, we demonstrated that Viyog achieves an average AUROC of 92.38% and FPR95TPR of 32.96% across our experiments using ResNet and DenseNet architectures.

Furthermore, we showed that Viyog is computationally efficient (single forward pass), memory-efficient (1.29 GB peak usage), architecturally robust (stable across diverse networks), modular (plug-in compatible with existing detectors), and threshold-robust (low AUTC of 0.23). These practical advantages make Viyog well-suited for deployment in safety-critical applications where both accuracy and efficiency are paramount.

The ability to reliably differentiate between OOD and adversarial inputs has significant implications for safety-critical applications. In autonomous driving systems, distinguishing between a genuinely novel object (OOD) and a maliciously crafted ad-

versarial perturbation can inform different response strategies. In medical diagnosis systems, this differentiation is equally crucial, as an OOD input might represent a rare disease variant requiring expert consultation, while an adversarial attack could indicate a security breach requiring immediate investigation.

Future work could explore extending this approach to other computer vision tasks such as object detection and semantic segmentation, investigating cross-modal applications in natural language processing and audio processing, developing online learning variants that adapt to evolving data distributions, and conducting field studies in actual safety-critical systems to validate Viyog's effectiveness under real operational conditions.

REFERENCES

[1] T. Abdullah and A. Ahmet. Deep learning in sentiment analysis: Recent architectures. *Computer Surveys*, 55(8):1–37, 2023.

[2] T. Ananthanarayana, P. Srivastava, A. Chintha, A. Santha, B. Landy, J. Panaro, A. Webster, N.R. Kotecha, S. Sah, T. Sarchet, et al. Deep learning methods for sign language translation. *ACM Transactions on Accessible Computing*, 14(4):1–30, 2021.

[3] G. Angelova, E. Avramidis, and S. Möller. Using neural machine translation methods for sign language translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 273–284, 2022.

[4] Can Bakiskan, Metehan Cekic, and Upamanyu Madhow. Early layers are more important for adversarial robustness. In *ICLR 2022 Workshop on New Frontiers in Adversarial Machine Learning*, 2022.

[5] Abhijit Bendale and Terrance Boult. Towards open set deep networks. 2015. arXiv:1511.06233.

[6] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017.

[7] J.Y. Chan, K.T. Bea, S.M.H. Leow, S.W. Phoong, and W.K. Cheng. State of the art: A review of sentiment analysis based on sequential transfer learning. *Artificial Intelligence Review*, 56(1):749–780, 2023.

[8] Ruoxi Chen, Haibo Jin, Haibin Zheng, Jinyin Chen, and Zhenguang Liu. Fight perturbations with perturbations: Defending adversarial attacks via neuron influence. *IEEE Transactions on Dependable and Secure Computing*, 22(2):1582–1595, 2025.

[9] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International Conference on Machine Learning*, pages 2206–2216. PMLR, 2020.

[10] Guneet S. Dhillon, Kamyar Azizzadenesheli, Zachary C. Lipton, Jeremy Bernstein, Jean Kossaifi, Aran Khanna, and Anima Anandkumar. Stochastic activation pruning for robust adversarial defense. 2018. arXiv:1803.01442.

[11] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pages 1050–1059. PMLR, 2016.

[12] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. 2014. arXiv:1412.6572.

[13] Q. He, J. Liu, and Z. Huang. WSRC: Weakly supervised faster RCNN toward accurate traffic object detection. *IEEE Access*, 11:1445–1455, 2023.

[14] Dan Hendrycks, Steven Basart, Mantas Mazeika, Andy Zou, Joe Kwon, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. Scaling out-of-distribution detection for real-world settings. 2022. arXiv:1911.11132.

[15] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. 2016. arXiv:1610.02136.

[16] Konstantin Kirchheim, Marco Filax, and Frank Ortmeier. Pytorch-ood: A library for out-of-distribution detection based on pytorch. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4351–4360, 2022.

[17] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in Neural Information Processing Systems*, 31, 2018.

[18] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. 2017. arXiv:1706.02690.

[19] L. Liu. Improved image classification accuracy by convolutional neural networks. *Proceedings of the International Conference on Information Technology and Electrical Engineering*, pages 1–5, 2021.

[20] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. In *Advances in Neural Information Processing Systems*, volume 33, pages 21464–21475, 2020.

[21] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. 2017. arXiv:1706.06083.

[22] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2574–2582, 2016.

[23] Preetum Nakkiran. Adversarial robustness may be at odds with simplicity. 2019. arXiv:1901.00532.

[24] Jaewoo Park, Jacky Chen Long Chai, Jaeho Yoon, and Andrew Beng Jin Teoh. Understanding the feature norm for out-of-distribution detection. 2023. arXiv:2310.05316.

[25] Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John C Duchi, and Percy Liang. Adversarial training can hurt generalization. 2019. arXiv:1906.06032.

[26] Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. *Advances in Neural Information Processing Systems*, 31, 2018.

[27] David Stutz, Matthias Hein, and Bernt Schiele. Disentangling adversarial robustness and generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6976–6987, 2019.

[28] Ke Sun, Zhanxing Zhu, and Zhouchen Lin. Towards understanding adversarial examples systematically: Exploring data size, task and model factors. 2019. arXiv:1902.11019.

[29] Yiyou Sun, Yifei Ming, Xiaojin Zhu, and Yixuan Li. Out-of-distribution detection with deep nearest neighbors. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pages 20827–20840. PMLR, 2022.

[30] Z. Sun, L. Tian, Q. Du, and J.A. Bhutto. Sample hardness guided softmax loss for face recognition. *Applied Intelligence*, 53(3):2640–2655, 2023.

[31] Janani Suresh, Nancy Nayak, and Sheetal Kalyani. First line of defense: a robust first layer mitigates adversarial attacks. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence*, 2025.

[32] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. 2018. arXiv:1805.12152.

[33] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, pages 7472–7482. PMLR, 2019.