Machine Learning Classification Boundaries and Robustness for Adversarial Attacks

Jian Ren, Jaxon Hancock, Tongtong Li
Electrical and Computer Engineering
Michigan State University, East Lansing, MI, USA
{renjian, hancoc79, tongli}@msu.edu

***Abstract*—Deep learning models have achieved remarkable accuracy in classification tasks, yet their vulnerability to adversarial attacks remains a critical security concern. This paper presents a framework for quantitatively analyzing classification robustness and decision boundary characteristics in neural networks. Using MNIST digit classification as a case study, we introduce two key metrics: robustness, measuring the tolerable range of sample distortion before misclassification, and separability, quantifying the distance between correctly and incorrectly classified samples near decision boundaries. We propose a novel boundary exploration algorithm that efficiently identifies inside and outside borderline samples through iterative sample mixing, enabling generation of adversarial examples with minimal perceptual distortion. Our evaluation across three CNN architectures reveals that while these networks demonstrate good robustness, they consistently exhibit poor separability, meaning visually indistinguishable samples can be classified into different categories. We further investigate pixel pruning as a defense mechanism, demonstrating a fundamental trade-off between separability and robustness: improving one metric necessarily degrades the other. Our findings provide quantitative insights into the inherent vulnerabilities of classification systems and establish a foundation for developing more robust defenses against adversarial attacks.**

***Index Terms*—Adversarial examples, classification boundaries, robustness, separability, SSIM, MNIST, deep learning security**

I. Introduction

Machine learning has been widely deployed across critical domains including autonomous driving, medical diagnosis, fraud detection, and facial recognition, demonstrating prediction accuracy comparable to or exceeding human performance. Deep learning (DL) has emerged as the primary driver for recent artificial intelligence (AI) breakthroughs, enabling systems to learn complex patterns from massive datasets. However, the widespread adoption of these systems has exposed a fundamental vulnerability.

Adversarial examples—inputs deliberately manipulated to cause misclassification while remaining imperceptible to humans—pose severe security threats to AI systems [1], [2]. Despite extensive research on generating adversarial examples and developing defense mechanisms [3]–[5], fundamental questions about classification robustness remain inadequately addressed. The decision boundaries that separate different classes in feature space are often abstract and difficult to characterize quantitatively, particularly in high-dimensional spaces where deep neural networks operate, due to the lack of appropriate quantitative metrics [6]. The transition from one class in the feature space to another can be nearly seamless without clear separation. Moreover, due to the largely black-box nature of neural network training architectures [7], the underlying logic behind classification algorithms remains highly ambiguous, making it challenging to predict or prevent adversarial vulnerabilities.

Existing robustness metrics focus primarily on worst-case perturbation bounds [8], [9] or adversarial accuracy under specific attack methods [10], [11]. However, these approaches provide limited insight into the geometric and topological properties of decision boundaries that fundamentally determine classifier vulnerability. Understanding not only how far a sample can be perturbed before misclassification (robustness), but also how well different classes are separated near decision boundaries (separability), is crucial for comprehensive security assessment.

Furthermore, classification algorithms are inherently incapable of determining whether an input falls within their intended scope, regardless of whether the input itself is meaningful [12]. This limitation becomes particularly problematic at decision boundaries where the distinction between in-scope and out-of-scope becomes ambiguous.

The contributions can be summarized as follows:

1) We introduce two essential metrics for evaluating classification robustness (measuring tolerable distortion within a class) and separability (measuring distinction between classes at boundaries), providing a systematic framework for quantitative security assessment.
2) We propose an efficient iterative algorithm for identifying inside and outside borderline samples by systematically mixing samples from different classes, enabling precise characterization of decision boundaries.
3) We demonstrate how the boundary exploration framework naturally extends to generating adversarial examples with minimal perceptual distortion.
4) Our experiments on MNIST digit classification reveal a consistent pattern of strong robustness but poor separability across all tested models, indicating fundamental vulnerabilities in standard classification approaches.
5) We analyze pixel pruning, and uncovering a fundamental trade-off between robustness and separability, which has important implications for designing practical defense mechanisms.

The remainder of this paper is organized as follows. Section II provides an overview of the paper. Section III presents our quantitative framework for characterizing classification

robustness. Section IV details our simulation results. Section V investigates pixel pruning as a defense mechanism and analyzes the robustness-separability trade-off. We concludes the paper in Section VI.

II. Overview

A. Classification Decision Boundaries and Characterization

The conceptual decision boundary in machine learning that separates different classes, while in some cases can be easily described, is often abstract and difficult to clearly characterize in most practical applications due to the lack of appropriate quantitative metrics. The transition from one class in the feature space to another can be nearly seamless, without clear separation. Moreover, due to the largely black-box nature of the training architecture, the underlying logic behind the classification algorithm remains highly ambiguous.

It is generally technically challenging to ensure high classification accuracy without integrating additional modules. Using image classification as an example, the classification algorithms must be combined with filters to exclude clearly out-of-scope image subjects. This is because classification algorithms are inherently incapable of determining whether an image falls within the intended scope-regardless of whether the image itself is meaningful or not.

B. Adversarial Examples

Adversarial examples [1], [2] are inputs to machine learning models that have been deliberately manipulated in subtle ways to cause the model to make incorrect predictions, even though a human would likely make the correct classification. These intentionally crafted inputs are often imperceptible to humans—such as tiny perturbations to the pixels and changing luminosity of certain area of an image—yet they can drastically change the model's classification output with high confidence. Adversarial examples pose significant security risks by enabling attacker to fool AI systems, for instance by causing a stop sign to be misclassified, spam to go undetected, or malware to bypass security scanners.

C. Threat Models

Similar to many prior works [13], in this work, we assume that the adversary has full knowledge of the classification model, including its architecture, parameters, and defense strategies. With this assumption, adversaries can train their models independently and generate adversarial examples. It is worth noting that adversarial examples generated for one model may also successfully fool other models, including black-box models with different architectures and datasets [2], [14].

D. Image Perceptual Metrics

The Structural Similarity Index Measure (SSIM) [15] is a perceptual metric in computer vision that assesses image quality by evaluating pixel data, structural patterns, and perceived realism. It quantifies the degradation of a processed image relative to its original reference. Unlike the mean squared error (MSE), which measures the average pixel-by-pixel difference, SSIM better reflects how the human visual system perceives image quality. The SSIM index is a full-reference metric, meaning that image quality is evaluated with respect to an uncompressed or distortion-free reference image.

TABLE I: Three CNN architectures for MNIST classification, with training/testing accuracies: 99.42%/99.28%, 99.80%/99.41%, and 99.44%/99.12%, respectively.

Layer	Architecture A	Architecture B	Architecture C
1	Input (28×28)	Input (28×28)	Input (28×28)
	Conv2D 6 (3×3) ReLU	Conv2D 20 (3×3) ReLU	Conv2D 32 (3×3) ReLU
	Batch Normalization	Batch Normalization	
3	MaxPooling2D (2×2)	MaxPooling2D (2×2)	Conv2D 64 (3×3) ReLU
4	Conv2D 16 (3×3) ReLU	Conv2D 50 (3×3) ReLU	MaxPooling2D (2×2)
	Batch Normalization	Batch Normalization	Flatten Batch Normalization
5	MaxPooling2D (2×2)	MaxPooling2D (2×2)	Fully Connected 128 ReLU
6	Fully Connected 120 ReLU	Fully Connected 500 ReLU	Fully Connected 10 ReLU
7	Fully Connected 84 ReLU	Fully Connected 84 ReLU	Softmax
8	Fully Connected 10	Fully Connected 10	Output
9	Softmax	Softmax	
10	Output	Output	

The SSIM between samples x and y is defined as follows:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \quad (1)$$

where μ_x and μ_y are the pixel sample means and σ_x^2 and σ_y^2 are the sample variances of x and y, respectively; σ_{xy} is the the sample covariance of x and y; $c_1 = (k_1 L)^2$, $c_2 = (k_2 L)^2$ are variables to stabilize the division with weak denominator, where L is the dynamic range of the pixel values, and $k_1 = 0.01$ and $k_2 = 0.03$ by default.

E. A Generic Security Attack and Boundary Example Generation for MNIST-based Classifications

In this paper, we introduce a generic approach to reach the boundary of any digit in the MNIST dataset. We compare implementations of three CNN architectures (see Table I) for MNIST digit recognition using MATLAB. Here, pure white and pure black refer to 28×28 grayscale images with all pixel values equal to 255 and 0, respectively.

By repeatedly mixing two samples from different classes [6], we can generate a pair of samples such that one remains within the class while the other lies just outside, as close as possible to the classification boundary—forming adversarial examples. Using the same algorithm, we can derive adversarial examples with minimum computer vision distortion.

TABLE II: Classification accuracies of the aforementioned architectures and classification accuracies on MNIST digits.

	Architecture A	Architecture B	Architecture C
Training accuracy	99.417%	99.795%	99.437%
Testing accuracy	99.28%	99.41%	99.12%
Pure white	Classified: 0 Prob: 98.8%	Classified: 8 Prob: 55.8%	Classified: 2 Prob: 91.5%
Pure black	Classified: 1 Prob: 98.5%	Classified: 1 Prob: 90.6%	Classified: 1 Prob: 35.5%

III. Quantitative Characterization for Classification Robustness

There are two essential quantitative measurements for evaluating classification robustness. The first is the tolerable range of sample distortion within which a sample can still be accurately classified into its original category—essentially, the gap between a training sample and its corresponding inside borderline sample—referred to as *robustness*. The second is the distance between the inside borderline sample and the nearest outside borderline sample—referred to as *separability*. The former reflects the robustness of the classifier in maintaining accurate predictions under reasonable distortion, such as environmental noise and sensor limitations, while the latter is critical to ensuring the ability to distinguish among different classification categories.

For images-based classification, the tolerable range refers to the acceptable level of image degradation or adversarial manipulations within which the digit image remains correctly classified. When the manipulation exceeds this limit, misclassification occurs. The gap between the original image and the inside borderline sample reflects the classifier's robustness to image quality degradation. Ideally, the classification robustness should align with human visual recognition of the digits—meaning the SSIM value between the original and the inside borderline sample should be as small (i.e., different) as possible. However, it cannot be too small due to computer vision constraints. Regarding separability, we hope that the inside borderline and outside borderline samples are significantly different and clearly distinguishable, which corresponds to relatively small SSIM values.

For an MNIST digit sample x_s, we can select another sample x_b that belongs to a different category. It should be noted that x_b may or may not represent a valid digit in the usual sense. Given x_s and x_b and their corresponding categories, c_{x_s} and c_{x_b}, we generate a new sample $x_c = \text{uint8}(x_s/2 + x_b/2)$. Depending on how x_c is classified, either x_b or x_s will be updated. If x_c is classified the same as x_s, it lies closer to the decision boundary, and in this case, we replace x_s with x_c. Otherwise, we replace x_b with x_c. This update guarantees that x_s remains in the same classification category, while x_b remains in a different one. This process is repeated until either x_b and x_s are sufficiently close, measured using the SSIM value, or a predefined iteration limit is reached.

For grayscale images such as MNIST digit images, the maximum pixel value is 255. After only 8 repetitions, the maximum value becomes 1. In Algorithm 1, the goal is to minimize the gap—achieved through maximizing the SSIM value—between the inside and outside borderline samples. That is

$$\begin{aligned} &\text{maximize } \mathsf{SSIM}(x_b, x_s), \\ &\text{such that } C(x_s) = C(x_o) \\ &\qquad\quad C(x_b) \neq C(x_o), \end{aligned} \tag{2}$$

where C denotes the classification scheme applied to a sample. Fig. 1 shows the steps to reach the borderline samples. Alternatively, we can modify equation (2) to generate adversarial

Algorithm 1: Generate a pair of inside and outside borderline samples.

Data: A trained network; a source image x_o to be classified; a background image x_b; an ssim threshold δ_s and the iteration threshold it.

Result: A pair of digit images x_b and x_s from 2 different classes, cxb and cxs, respectively, that have very high ssim value.

```
os = xs ;                          % Source image
cxb = classify(net, xb);
cxs = classify(net, xs);
while isempty(xe) do
    xc = uint8(xb/2 + xs/2);
    cxc = classify(net, xc);
    if (cxc == cxs) then
        xs = xc ;          % xs shifts backward
        cxs= cxc;
    else
        xb = xc ;           % xb shifts forward
        cxb = cxc;
    end
    s = ssim(xb, xs);
    if (s > δs) || (i >= it) then
        xe = [xe; [xb, cxb, xs, cxs]];
        break
    end
    i = i + 1;
end
ssim_xb_xs = ssim(xb, xs);
ssim_xs_os = ssim(xs, os);
ssim_xb_os = ssim(xb, os);
```

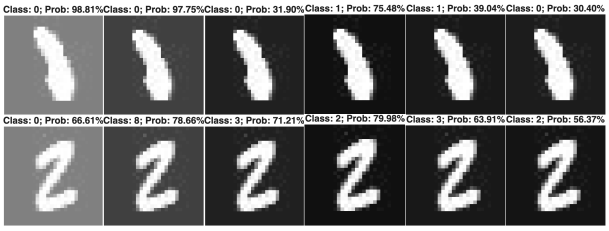

Fig. 1: Sample steps to reach the borderline pairs.

examples that approach the classification boundary of x_s using Algorithm 1 as follows:

$$\begin{aligned} &\text{maximize } \mathsf{SSIM}(x_b, x_o), \\ &\text{such that } C(x_s) = C(x_o) \\ &\qquad\quad C(x_b) \neq C(x_o). \end{aligned} \tag{3}$$

IV. Simulation Results

Our evaluation of various training architectures, including the three architectures listed in Table I, has demonstrated that these classification architectures can tolerant a reasonable range of image quality degradation. In other words, they are generally robust. However, all these schemes share one common property: they all have separability issues where the distinction between inside and outside borderline samples is quite ambiguous, in that two visually indistinguishable

Fig. 2: Selected MNIST digits and their inside and outside borderline samples.

samples with an SSIM value as close to 1 as possible can be classified into different categories. In this case, the visually indistinguishable sample can work as an adversarial example to fool the machine learning system.

Fig. 2 shows selected MNIST digits along with their inside and corresponding outside borderline samples generated using architecture A. The first row of each category displays the original digits from the MNIST test set, the second row shows their corresponding inside borderline samples, and the third row presents their corresponding outside borderline samples, which are the adversarial examples of the corresponding source digits.

In generating the borderline samples, x_b is either pure white in grayscale—analogous to increasing the image's luminosity, pure black in grayscale—analogous to reducing the image's luminosity, or a sample taken from the MNIST test database. In particular, for cases where x_b is pure white or pure black, the results are shown in Table III. In our simulations, the SSIM threshold was set to 0.99, and the maximum number of iterations to 9.

From this table, we can see that all three architectures demonstrate strong robustness, and the adversarial examples

TABLE III: Mean robustness and separability when mixing with x_b set to pure white or pure black using Algorithm 1.

Architecture	x_b	**Separability**		**Robustness**		**AE**	
		SSIM	STD	SSIM	STD	SSIM	STD
Architecture A	white	0.9950	0.0018	0.3231	0.0644	0.3076	0.0633
	black	0.9951	0.0066	0.5512	0.0819	0.5310	0.0811
Architecture B	white	0.9944	0.0066	0.4858	0.0505	0.4687	0.0448
	black	0.9942	0.0083	0.4610	0.0855	0.4487	0.0849
Architecture C	white	0.9944	0.0070	0.3499	0.0762	0.3348	0.0581
	black	0.9946	0.0067	0.4898	0.0787	0.4748	0.0784

(AEs) are generally quite different from the original samples. Moreover, the quantitative variations are stable and consistent with their differences in classification accuracy. However, they also consistently exhibit very high SSIM values in separability, meaning that the inside and outside borderline samples are nearly indistinguishable. This indicates that separability is an issue that cannot be resolved through classification schemes alone; instead, alternative techniques must be explored.

V. Pixel Pruning and Classification Robustness

With the intention of enlarging the gap between the inside and outside borderline samples, we need to devise technical approaches or filters that can limit the number of available samples around the borderlines for classification algorithms.

TABLE IV: Mean robustness and separability for architecture A when mixing with x_b set to pure white or pure black under pixel pruning using Algorithm 1.

Pixel Pruning	x_b	**Separability**		**Robustness**		**AE**	
		SSIM	STD	SSIM	STD	SSIM	STD
$t = 64$	white	0.6872	0.2413	0.5033	0.2328	0.2584	0.0606
$T = 255$	black	0.7128	0.1406	0.5910	0.1202	0.4050	0.0770
$t = 128$	white	0.3213	0.1209	0.7376	0.1006	0.1987	0.0678
$T = 255$	black	0.5550	0.1442	0.7847	0.1946	0.4260	0.0714
$t = 128$	white	0.1036	0.0007	0.9004	0.0001	0.0041	0.0008
$T = 128$	black	0.4791	0.0634	0.8253	0.1700	0.4234	0.0714

Pixel pruning—replacing less significant pixels with black or white—is one of the most natural color depth reduction approaches to consider. It is similar to feature squeezing analyzed in [16], [17].

We perform various pixel pruning operations when mixed with a white image (analogous to increasing luminosity) and a black (analogous to reducing luminosity) background, replacing pixels with values below t with 0 and those above T with 255. We first set $t = 64$ and $T = 255$ for both white and black x_b, meaning that, for all MNIST digits, pixel values below 64 are replaced with black. Our simulation results show that separability in these two cases increases—measured by SSIM, the values decrease from 0.9955 (white) and 0.9951 (black) to 0.6872 and 0.7128, respectively. Meanwhile, the robustness decreases from 0.3231 and 0.5512 to 0.5033 and 0.591, respectively.

We further set $t = 128$ and $T = 256$ for both white and black x_b. Our simulation results indicate that separability increases further while the robustness continues to decrease. Specifically, the SSIM values for separability in the two cases drop to 0.3213 (white) and 0.5550 (black), while robustness decreases to 0.7376 (white) and 0.7847 (black), compared with 0.3231 (white) and 0.5512 (black) in the previous setting.

We also conduct a more extreme experiment by setting $t = T = 128$. In this case, classification separability improves further, accompanied by an additional reduction in robustness. Detailed simulation results are provided in Table IV.

These analyses demonstrate a clear trade-off between separability and robustness: as one improves, the other deteriorates. Therefore, identifying an appropriate balance between these two conflicting performance metrics is an important research challenge in filter design.

We also found that, compared with Table III, while AE generation with increasing luminosity tends to produce adversarial examples less similar to the original samples (as shown in Fig. 3), the impact is relatively small when luminosity decreases. It is also interesting to note several inconsistencies. First, based on visual observation, the digits in the first row appear to have better fidelity than those in the second row; however, both the prediction probabilities and the SSIM values in the second row are higher than those in the first row. This discrepancy is somewhat difficult to fully explain.

VI. Conclusion

This paper introduces a quantitative framework for analyzing classification robustness using two metrics: robustness (tolerable distortion before misclassification) and separability (distance between decision boundaries). Our boundary exploration algorithm efficiently identifies borderline samples for security assessment and adversarial example generation. Experiments on MNIST across three CNN architectures reveal that all models exhibit poor separability, meaning visually indistinguishable samples are classified differently—a fundamental vulnerability. Pixel pruning analysis uncovers an inherent trade-off: improving separability degrades robustness and vice versa.

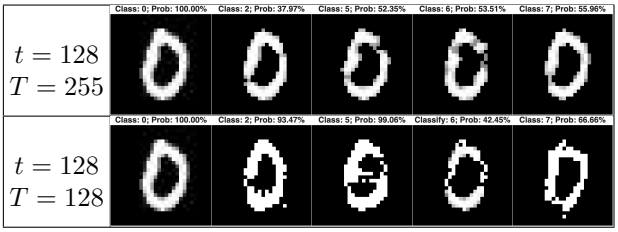

Fig. 3: Selected adversarial examples for architecture A, where the leftmost digit is the original sample, and the corresponding mean SSIM values between the AEs and the original samples are 0.6297 and 0.7468, respectively, .

References

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," 2014.

[2] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[3] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations (ICLR)*, 2018.

[4] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *IEEE Symposium on Security and Privacy (SP)*, p. 582–597, 2016.

[5] C. Guo, M. Rana, M. Cisse, and L. van der Maaten, "Countering adversarial images using input transformations," in *International Conference on Learning Representations (ICLR)*, 2018.

[6] H. Karimi, T. Derr, and J. Tang, "Characterizing the decision boundary of deep neural networks." https://arxiv.org/abs/1912.11460, 2020.

[7] Z. C. Lipton, "The mythos of model interpretability," *Queue*, vol. 16, no. 3, pp. 31–57, 2018.

[8] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *IEEE Symposium on Security and Privacy (SP)*, p. 39–57, 2017.

[9] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *International Conference on Machine Learning (ICML)*, p. 274–283, 2018.

[10] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, , and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," in *International Conference on Learning Representations (ICLR)*, 2018.

[11] W. He, J. Wei, X. Chen, N. Carlini, and D. Song, "Adversarial example defenses: Ensembles of weak defenses are not strong," in *11th USENIX Workshop on Offensive Technologies (WOOT 17)*, 2017.

[12] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in *International Conference on Learning Representations (ICLR)*, 2017.

[13] W. He, J. Wei, X. Chen, N. Carlini, and D. Song, "Adversarial example defenses: Ensembles of weak defenses are not strong." https://arxiv.org/abs/1706.04701, 2017.

[14] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," 2017.

[15] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[16] W. Xu, D. Evans, and Y. Qi, "Feature squeezing mitigates and detects carlini/wagner adversarial examples." https://arxiv.org/abs/1705.10686.

[17] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks." https://arxiv.org/abs/1704.01155.