International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Enhancing Machine Learning Resilience to Adversarial Attacks through Bit Plane Slicing Optimized by Genetic Algorithms

Mr. Ganesh Ingle[1], Dr. Sanjesh Pawale*[2]

Submitted:13/03/2024 **Revised**: 28/04/2024 **Accepted**: 05/05/2024

Abstract: This research delves into enhancing the resilience of machine learning models, particularly image classification algorithms, against adversarial attacks. The focus is on using genetic algorithms to optimize bit plane slicing configurations, thereby improving the models' robustness. The study reveals that models with 5-bit depth representations exhibit superior resilience, achieving high accuracies of $\mathbf{98.21}\%$ against FGSM attacks and $\mathbf{92.98}\%$ against DeepFool attacks. These results underscore the importance of adjusting detail levels through bit plane slicing to maintain algorithmic integrity under adversarial conditions. Despite a significant drop in performance due to adversarial modifications, with accuracy falling from $\mathbf{90.32}\%$ to $\mathbf{11.69}\%$, a notable recovery was observed, highlighting the effectiveness of the optimized defense strategies. The findings advocate for further research into dynamic bit plane slicing and the development of advanced defense mechanisms using genetic algorithms, aiming to bolster the security and reliability of machine learning models against the continuously evolving adversarial threats.

Keywords: *Adversarial attacks; Bit plane slicing; Defense strategies; Machine learning security; Genetic Algorithm*

1. Introduction

Gradient-based sparse attacks are particularly insidious because they aim to achieve misclassification with minimal changes to the input, making them harder to detect. These attacks leverage the gradient information to identify the most effective alterations to the input data, optimizing for the fewest changes needed to deceive the classifier. Your future work in exploring defences against these attacks could focus on further enhancing the robustness of the randomization mechanism, possibly by investigating adaptive or dynamic randomization strategies that can anticipate and counteract the optimization strategies employed by attackers. Non-gradient-based attacks, on the other hand, do not rely on gradient information and often employ more heuristic or search-based approaches to generate adversarial examples. These attacks can be more unpredictable and might exploit different model vulnerabilities than gradient-based methods. Defending against these attacks could involve developing more sophisticated detection mechanisms that can recognize and mitigate the impact of adversarial inputs, regardless of the specific strategy used to generate them [1]. In the realm of machine learning security, adversarial examples pose a significant threat to the integrity and reliability of neural network-based systems, including those used in face recognition and autonomous vehicles. Traditional defenses, such as adversarial training, defensive distillation, and feature squeezing, have sought to mitigate these vulnerabilities by either hardening the model against attacks or preprocessing inputs to remove adversarial noise. However, these methods often come with trade-offs between model accuracy and robustness or suffer from scalability and efficiency issues. Advanced denoising techniques have also been explored, with standard denoisers using autoencoders or specialized network architectures; yet, they frequently encounter the amplification effect, where residual adversarial noise exacerbates rather than alleviates the problem. The proposed Unified Deep Denoising Network (UDDN) strategy introduces a novel approach, combining a specialized loss function that reduces the discrepancy between outputs from original and denoised images, with a training algorithm that incorporates knowledge transfer to enhance model resilience. This dual approach addresses the amplification effect and ensures robustness against both white-box and black-box adversarial attacks. Experimental application to a face recognition model further underscores the effectiveness of the UDDN strategy in improving both the accuracy and security of neural network models, presenting a promising avenue for future research in safeguarding machine learning systems against adversarial threats[2]. The vulnerability of Convolutional Neural Networks (CNNs) to adversarial attacks, despite their remarkable success in computer vision tasks, highlights a critical challenge for their application in security-sensitive environments. The literature reveals a range of defense mechanisms aimed at mitigating the impact of these attacks, which often involve human-imperceptible adversarial noise patterns.

1 Tolani Maritime Institute, Department of Applied and Social Sciences, Pune, ORCID ID: 0000-0001-5628-0388
2 Department of Computer Engineering, Vishwakarma University, Pune, ORCID ID: 0009-0006-7556-2528
** Corresponding Author Email: sanjesh.pawale@vupune.ac.in*

Traditional methods have focused on modifying the network architecture, adversarial training, or employing external detection systems to identify and neutralize adversarial inputs. However, these approaches frequently require substantial computational resources, potentially degrade model performance on clean images, or necessitate extensive retraining of the model.A novel image enhancement-based defense mechanism that leverages deep image restoration networks to correct adversarial perturbations by mapping off-the-manifold adversarial samples back onto the natural image manifold. This method distinguishes itself by not only counteracting adversarial attacks without compromising the model's accuracy on unperturbed images but also improving the overall image quality. Unlike conventional defenses, our proposed approach does not alter the underlying classifier or rely on adversarial detection mechanisms, offering a computationally efficient solution that is easily integrated with existing models. Demonstrated through rigorous testing in gray-box scenarios, our method showcases robust defense capabilities across various attack algorithms without necessitating model-specific training or parameter adjustments. This simplicity, coupled with its compatibility with other defense strategies and model-agnostic nature, positions it as a versatile and effective tool for enhancing the security of CNN-based systems against a wide spectrum of adversarial threats[3]. The escalating sophistication of adversarial attacks poses a formidable challenge to the deployment of Convolutional Neural Networks (CNNs) in security-critical applications. The nuanced landscape of adversarial threats, ranging from gradient-based sparse attacks that exploit model gradients for minimal yet effective perturbations, to non-gradient-based attacks that utilize heuristic approaches to undermine model integrity. The exploration of defenses against these attacks, particularly through the development of the Unified Deep Denoising Network (UDDN) and image enhancement-based mechanisms, represents a significant stride towards fortifying neural networks. These proposed strategies, distinguished by their ability to restore adversarial samples to the natural image manifold and enhance image quality without compromising the model's performance on clean images, mark a promising direction for future research. By addressing both the subtlety of gradient-based attacks and the unpredictability of heuristic approaches, this work lays a foundation for more resilient, adaptable, and efficient defense mechanisms. The demonstrated effectiveness of these methods in gray-box scenarios further underscores their potential to provide robust security enhancements across a broad array of neural network applications. Moving forward, the continued investigation into adaptive and dynamic defense mechanisms, alongside the integration of these novel strategies with existing models, will be crucial in navigating the evolving landscape of adversarial threats and ensuring the reliable deployment of CNNs in environments where security is paramount.

2. Literature Survey

For an in-depth exploration of Probabilistic Adversarial Robustness (PAR) and its implementation via PixelCNN as a defensive mechanism against adversarial attacks, the literature on adversarial training and robustness in deep learning models offers a foundational context. Adversarial training, highlighted as a key strategy for improving model resistance against such attacks, focuses on training neural networks with both clean and adversarial examples. This methodology aims to minimize classification error under adversarial perturbations by formulating the training process as a min-max problem. The goal is to find the worst-case adversarial examples and train the model to be robust against them. [4] work in this area introduced the use of the multi-step gradient-based PGD attack for solving the inner maximization problem, thereby significantly enhancing adversarial robustness.The literature categorizes defense strategies against adversarial attacks into Gradient Masking/Obfuscation, Robust Optimization, and Adversarial Example Detection, with Robust Optimization being particularly relevant to PAR. This category encompasses methods that improve the optimization function through adversarial examples, regularization terms, or model modifications to introduce uncertainty.Understanding the threat model of adversarial attacks—including aspects like timing (evasion vs. poisoning attacks), information access (white box vs. black box), goals (targeted vs. untargeted), and perturbation characteristics—is crucial for developing effective defense mechanisms. These considerations are essential for devising comprehensive strategies that address various facets of adversarial robustness, including those based on probabilistic models like PixelCNN.The aforementioned insights underscore the significance of adversarial training and robust optimization in defending against adversarial attacks, providing a solid framework for further research and implementation of PAR-based defense mechanisms [4]. Techniques such as Data Augmentation-based Transferability Enhancing Methods, including Diverse Inputs Method (DIM) and Translation Invariance Method (TIM), aim to increase adversarial transferability. These methods suggest that incorporating diverse attack strategies and optimizing for transferability could improve defenses like HGD against both white-box and black-box attacks [5]. Apollon, a novel defense against Adversarial Machine Learning (AML) attacks targeting Intrusion Detection Systems (IDS). It employs Multi-Armed Bandits (MAB) with Thompson sampling for dynamic classifier selection, enhancing IDS unpredictability against AML. Despite its effectiveness in detecting attacks and maintaining performance on normal

traffic, Apollon doesn't completely eliminate AML risks but increases attackers' effort and costs. Future work includes exploring other MAB models, additional classifiers, and testing on more datasets to further enhance Apollon's robustness and efficiency in diverse network environments [6]. [8],[29] highlights that incorporating a maximal separation constraint in the objective function of Deep Neural Networks (DNNs) makes it challenging for adversaries to generate effective perturbations, as opposed to the traditional cross-entropy loss. This constraint ensures that adversarial polytopes of different classes do not overlap, effectively blocking viable attack vectors within the constrained perturbation budget. Rigorous testing across various attack types and settings confirmed the model's robustness without relying on gradient obfuscation, suggesting a potent defense against adversarial threats in deep learning systems.[9][30] introduces a self-augmentation (SA) method to enhance defense against transferable adversarial examples targeting Deep Neural Networks. SA incorporates self-ensemble, adding convolution layers to create diverse virtual models for an ensemble effect, and deviation-augmentation, leveraging observed curved loss surfaces around input data to apply deviation vectors for evasion. Tested across four base models and ten defense mechanisms, SA demonstrates superior effectiveness in generating transferable attacks compared to existing methods, suggesting a potential advancement in understanding and mitigating adversarial threats.[10] introduces a novel image preprocessing method to combat adversarial examples in Deep Neural Networks (DNNs), crucial for applications like autonomous vehicles, healthcare, and face recognition. It employs adaptive parameters for label loss and pixel-level loss, guiding the model to reconstruct images and nullify adversarial perturbations without compromising image quality or classification accuracy. Additionally, a sparsity constraint architecture is used to ensure neurons activate only for significant patterns, reducing adversarial noise impact. Tested against various attacks on MNIST and CIFAR-10 datasets, the method achieved impressive accuracies of 98% and 92%, respectively, showcasing its effectiveness. [11] introduces an algorithm for generating sparse adversarial videos applicable in both black-box and white-box settings, focusing on temporal and spatial sparsity by targeting key pixels and frames for perturbation. It employs super-pixels to manage dimensionality, offering efficiency across various video models and datasets. Despite its effectiveness, achieving under 1% pixel alteration and time efficiency, the technique faces limitations in real-world generalization.

[12] introduces a self-supervised adversarial training mechanism in the input space, aiming to combine the strengths of model parameter modification and input processing for defense against adversarial attacks in DNN-based vision systems. It offers a generalizable solution that significantly enhances robustness across various tasks, such as classification, segmentation, and detection, by reducing the success rate of sophisticated attacks. This method can serve as a versatile, plug-and-play defense mechanism for different vision systems, demonstrating substantial improvements over previous approaches. DIPDefend introduces a novel approach for defending deep neural networks against adversarial attacks by leveraging deep image priors. Unlike traditional methods that depend on external training datasets, DIPDefend focuses on the internal priors of individual images. It uses a deep image prior generator to reconstruct images, prioritizing robust features over non-robust ones through an adaptive stopping strategy. This ensures a tailored defense for each adversarial input, showcasing superior performance against both white-box and black-box attacks in comparison to existing defense mechanisms [14]. [15] proposes "two-stream" architecture defends against adversarial examples by comparing outputs from high-resolution and low-resolution networks. This framework, adaptable with new datasets and backbones, enhances future scalability and complicates white-box attacks for adversaries. It effectively detects adversarial examples without prior knowledge of their creation, suggesting its robustness against manipulations aimed at misleading DNNs. The underlying effectiveness may be attributed to its analysis of adversarial perturbations' impact on neural networks. [24] demonstrate that applying various degrees of feature masking can significantly enhance a model's defense against adversarial attacks. Their research highlights feature masking as an effective method to mitigate such attacks, effectively balancing model accuracy with improved security measures. [25] introduces a novel method combining K-Means clustering and Class Activation Mapping (CAM) for executing adversarial attacks, indicating a research void in understanding GNN vulnerabilities, particularly in how GNNs process graph data and the potential for exploitation. It suggests the necessity for future studies to explore GNNs' data processing to safeguard against these vulnerabilities. The research also points to the need for application-specific defenses across various GNN applications, emphasizing tailored security measures for different domains.The susceptibility of deep learning models lacks emphasis on fostering interdisciplinary collaboration. Closing the gap between machine learning experts, security researchers, and domain-specific professionals is vital for crafting holistic adversarial defense strategies.To address these gaps, the research community needs to delve deeper into the intricate challenges of adversarial attacks. This involves considering diverse application contexts and

constructing adaptive, interpretable, and collaborative defense mechanisms. Integration of technical expertise across disciplines is essential for developing comprehensive strategies that mitigate adversarial threats effectively [26].[27] study highlights the exploration of deep-learning-based iris recognition methods, particularly focusing on the impact of omitting traditional preprocessing steps like iris normalization and feature enhancement. Most existing works emphasize complex preprocessing to improve accuracy, yet this study suggests that simplifying the process by using YOLOv4-tiny for iris localization and minimizing preprocessing steps could lead to equally or more effective biometric authentication systems. This approach challenges the conventional understanding and opens avenues for developing more efficient and accessible iris recognition technologies, especially for applications in small communities where computational resources may be limited.[28] outlines advancements in network intrusion detection using deep learning but does not specifically address the resilience of these models against adversarial attacks, which could manipulate model predictions by introducing subtle perturbations to input data. The gap suggests a need for exploring the robustness of the proposed CNN-BiLSTM model against such adversarial manipulations, ensuring its effectiveness not just under normal conditions but also when faced with sophisticated evasion techniques designed to bypass detection. Future research could focus on enhancing the model's defense mechanisms against these attacks, ensuring reliable security measures in adversarial environments. [29] approach to improving autonomous vehicle steering angle prediction with a modified VGG19 model does not explicitly address the resilience of this deep learning system against adversarial attacks. Adversarial attacks can manipulate model predictions by introducing specially crafted inputs, a critical concern for autonomous vehicle safety. Future research could explore enhancing the model's defense mechanisms against such attacks to ensure reliable performance even in adversarial environments, crucial for the real-world deployment of autonomous driving technologies.

The Multi-Task Cascaded Convolutional Network (MTCNN) demonstrates high efficiency in face recognition, its robustness against adversarial attacks remains a significant research gap. Adversarial attacks involve subtly altered inputs designed to deceive neural networks, posing a serious security risk in applications relying on face recognition. Future research should focus on enhancing MTCNN's defense mechanisms to identify and mitigate such attacks, ensuring the technology's reliability and security in critical applications [30].The comprehensive investigation into adversarial machine learning (AML) and defense mechanisms underscores the necessity for advancing research in several key areas to effectively counteract adversarial threats. Among the various strategies discussed, the optimization of bit plane selection using genetic algorithms (GAs) presents a novel method that could significantly contribute to the development of more secure, reliable, and adversarially resilient machine learning models and systems. However, a clear research gap exists in fully understanding and implementing this approach within the context of defending against adversarial attacks. There is a lack of comprehensive theoretical models that explain the effectiveness of optimizing bit plane selection in enhancing the robustness of ML models against adversarial attacks. Developing a framework that elucidates how bit plane selection impacts the adversarial resilience of ML models. Conducting empirical studies to validate these theories in practical applications, such as image recognition or video surveillance systems.Current literature does not adequately address how the optimized selection of bit planes, using GAs can be integrated with other defense strategies, such as adversarial training, DIM, TIM, or dynamic classifier selection via MABs. Investigating methodologies for combining optimized bit plane selection with existing defense mechanisms to create a layered defense strategy. This involves assessing the compatibility and potential synergies between different approaches.The optimal configuration of genetic algorithms for bit plane selection, including the selection of appropriate fitness functions, mutation rates, crossover rates, and population sizes, remains underexplored.Conducting systematic experiments to identify the most effective GA configurations for optimizing bit plane selection. This research should aim to maximize defense efficacy against a wide range of adversarial attacks while minimizing computational overhead. The applicability of optimized bit plane selection across different domains and types of ML models, especially beyond visual processing tasks, is not well-documented.Extending the exploration of GA-optimized bit plane selection to various domains, including audio processing, natural language processing, and other non-visual data types. Assessing the method's effectiveness across different neural network architectures and learning paradigms. There is a need for research on the scalability of the GA-optimized bit plane selection process, especially in terms of processing large datasets and deploying in real-time systems.Investigating scalable and efficient implementations of GA-optimized bit plane selection, focusing on reducing computational complexity without compromising the defense's effectiveness.The continuous evolution of adversarial attack methods may outpace the development of defense mechanisms, including those based on bit plane selection optimization. Developing adaptive and dynamic optimization strategies

that can evolve in response to new and emerging adversarial attack techniques. This includes creating feedback loops that allow the defense mechanism to learn from attempted attacks and adjust accordingly.

3. Background and Motivation

The landscape of machine learning (ML) and artificial intelligence (AI) has been significantly transformed by the advent of adversarial machine learning (AML), highlighting critical vulnerabilities in deep learning architectures. Adversarial examples, which are inputs specifically designed to deceive ML models, have emerged as a potent threat, undermining the reliability and integrity of systems deployed across various sectors, including autonomous navigation, healthcare diagnostics, cybersecurity, and biometric verification. This threat necessitates a profound exploration of defense mechanisms capable of safeguarding against these insidious attacks. Adversarial training has been identified as a cornerstone strategy for enhancing model robustness, involving the integration of adversarial examples into the training process. This approach aims to prepare the model for worst-case scenarios, thereby minimizing the classification error under adversarial conditions. Techniques such as the multi-step gradient-based Projected Gradient Descent (PGD) attack have been pivotal in solving the inner maximization problem, substantially improving adversarial robustness. However, as adversarial attack methodologies evolve, a noticeable gap remains between the complexity of these attacks and the effectiveness of current defense strategies. The exploration of defense mechanisms has been categorized into several strategies, including Gradient Masking/Obfuscation, Robust Optimization, and Adversarial Example Detection. Among these, Robust Optimization stands out for its relevance to Probabilistic Adversarial Robustness (PAR) - a concept that seeks to introduce uncertainty into the optimization function through adversarial examples, regularization terms, or model modifications. PAR, implemented via mechanisms such as PixelCNN, offers a promising avenue for defense by leveraging probabilistic models to enhance the resilience of ML systems against adversarial manipulations.

Despite these advancements, there is a pressing need for innovative defense strategies that can address the multifaceted nature of adversarial threats. The optimization of bit plane selection using genetic algorithms (GAs) represents one such novel approach. This strategy involves fine-tuning the selection of bit planes - layers of binary images that together form the complete image representation - to enhance model robustness against adversarial attacks. However, the full potential of this approach has yet to be realized, owing to a dearth of comprehensive theoretical models and practical implementations that demonstrate its efficacy in real-world scenarios.

Moreover, the integration of optimized bit plane selection with existing defense strategies, such as adversarial training and dynamic classifier selection, remains an underexplored area. This gap underscores the necessity for a layered defense strategy that can leverage the synergies between different approaches to offer robust protection against adversarial attacks. The continuous evolution of adversarial tactics further accentuates the need for adaptive and dynamic defense mechanisms. These mechanisms must be capable of evolving in response to new and emerging threats, ensuring the long-term security and reliability of ML systems. This calls for a concerted research effort to develop scalable, efficient, and adaptable defense strategies that can navigate the complexities of the adversarial landscape. The motivation behind this research endeavor is twofold: to bridge the existing gaps in our understanding and implementation of defense mechanisms against adversarial attacks and to pioneer the development of more secure, reliable, and adversarially resilient ML models and systems. By addressing these challenges, we aim to fortify the foundation of ML and AI technologies against the ever-growing threat of adversarial attacks, ensuring their safe and trustworthy application in critical domains.

4. Methodology

This research article outlines a novel approach to defending deep learning models against adversarial attacks by integrating bit-plane slicing with genetic algorithm (GA) optimization. The method's foundation lies in its unique combination of image processing techniques and evolutionary computation to enhance model resilience. As shown in Figure 1, the architecture diagram of our proposed method illustrates the workflow and the key components involved in achieving improved accuracy over epochs for both source-only and target-only data. This design underpins our methodology's effectiveness in addressing the challenges of domain adaptation, leveraging the intrinsic data characteristics to enhance model performance significantly.

Below, we delve into a more detailed explanation of each component and the overall methodology.

4.1 Bit-Plane Slicing

In an 8-bit grayscale image, each pixel's intensity value can be represented as a binary number, with 8 bits ranging from the most significant bit (MSB) to the least significant bit (LSB). The intensity I of a pixel can be mathematically represented as:

$$I = 2^7 b_7 + 2^6 b_6 + 2^5 b_5 + 2^4 b_4 + 2^3 b_3 + 2^2 b_2 + 2^1 b_1 + 2^0 b_0 \quad (1)$$

Where b_7 to b_0 represent the bit values of the binary representation of the pixel intensity, with b_7 being the MSB and b_0 being the LSB. Bit-plane slicing involves isolating each bit plane of the binary representation of pixel intensities to reveal its contribution to the overall image. For an 8-bit grayscale image, this results in 8 distinct layers, each representing a different bit plane. The bit-plane slicing operation for an image A of dimensions $M \times N$ and an 8-bit depth can be mathematically represented as:

$$B_k = (A \text{ AND } 2^k) >> k \quad (2)$$

Where:

B_k represents the binary image of the k-th bit plane.

k ranges from 0 to 7, representing each bit position from LSB to MSB.

AND is the bitwise AND operation.

>> is the bitwise right shift operator.

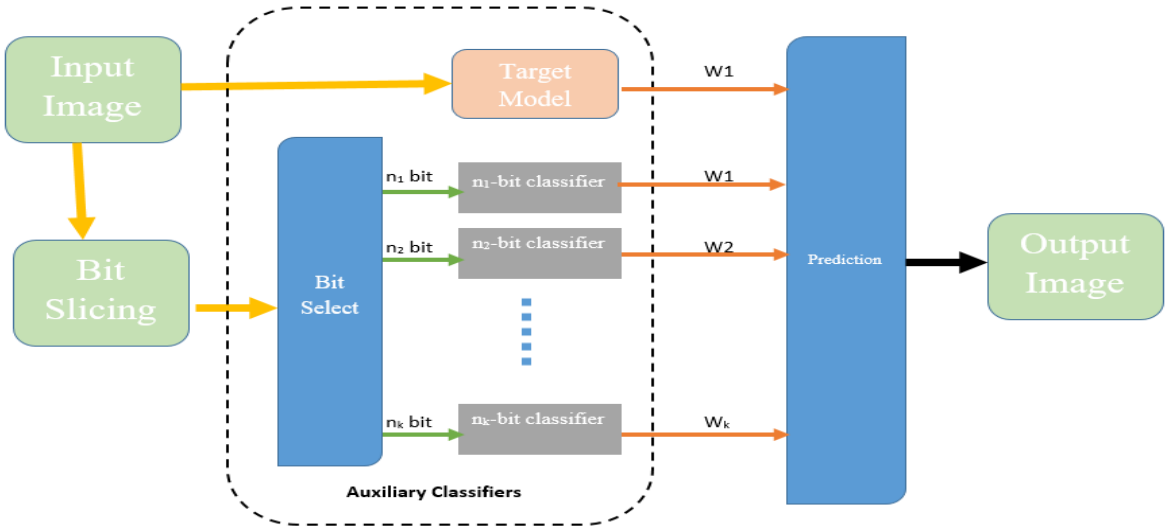

Fig. 1. Architecture Diagram of the Proposed Method

This operation isolates the k-th bit from each pixel in A and shifts it to the least significant bit position, creating a binary image that highlights the information contained in that specific bit plane. Bit-plane slicing also finds application in adversarial defense strategies, where preprocessing images by discarding certain bit planes can enhance the robustness of machine learning models against adversarial attacks. For example, to mitigate the effects of small perturbations often exploited by adversarial attacks, we can discard the 3 least significant bits (LSBs) of each pixel in the image A, creating a modified image A'. This operation can be represented as:

$$A' = A \text{ AND } 11111000_2 \quad (3)$$

Where 11111000_2 represents a bit mask that retains only the 5 most significant bits (MSBs) of each pixel intensity.

Fig. 2. Process of Bit Plane Slicing

Figure 2 gives visual illustrations of Bit Plane Slicing. Bit-plane slicing is a fundamental technique in digital image processing that provides insights into the binary representation of pixel intensities. By dissecting images into their constituent bit planes, we can analyze their information content and apply various operations for different purposes, including image analysis, compression, and security. The mathematical models involved in bit-plane slicing enable precise manipulation of image data, making it a valuable tool in various applications, including adversarial defense in machine learning systems. The Bit-Plane Perturbation Rate ($BPPR_i$) is articulated as the ratio of the cumulative bitwise exclusive OR (XOR) discrepancy across the ith bit-plane of original and adversarially modified datasets, to the total bit count within the bit-plane, expressed in percentage. Formally, it is defined as:

$$BPPR_i = \frac{\sum \text{xor}(bpi, bpAdv_i)}{M \times N} \times 100\% \qquad (4)$$

In this equation, bpi symbolizes the ith bit-plane of the pristine dataset, whereas $bpAdv_i$ denotes the ith bit-plane of the adversarial example. The bit-plane consists solely of binary values (0s and 1s), with the XOR operation employed to compute the bit-level differences. M and N represent the dimensions of the bit-plane, specifically its height and width, respectively. Thus, $BPPR_i$ quantifies the alteration rate within the ith bit-plane attributed to adversarial interventions, offering a measure of how these perturbations distort the original data at the bit-plane level. Additionally, it gauges the intensity of adversarial disruptions, with a pronounced $BPPR_i$ in higher-order bit-planes indicative of substantial adversarial perturbations, as considerable deviations are necessary to impact these bits. Empirical calculations of $BPPR_i$ elucidate a predilection for perturbations to predominantly influence lower-order bit-planes, corroborating our intuitive understanding of adversarial perturbation patterns. The mathematical significance of the Bit-Plane Perturbation Rate ($BPPR_i$) lies in its ability to quantitatively assess the impact of adversarial perturbations on digital data at a granular, bit-level scale. This metric provides several key insights into the nature of adversarial attacks and their effects on data, especially within the context of machine learning security. Here are some of the insights and significances derived from $BPPR_i$:

1. Granular Analysis of Perturbations: $BPPR_i$ allows for a detailed examination of how adversarial modifications affect each bit-plane of the data. Since digital images and other forms of data are represented at the binary level in computing systems, understanding these changes at the bit-plane level provides a foundational perspective on the mechanics of adversarial attacks.

2. Quantitative Measure of Adversarial Impact: By calculating the percentage of bits that are altered in each bit-plane, $BPPR_i$ offers a quantitative measure of the extent to which adversarial perturbations have modified the original data. This is crucial for evaluating the strength and efficacy of adversarial examples.

3. Differentiation of Perturbation Magnitude Across Bit-Planes: High $BPPR_i$ values in higher-order bit-planes indicate that an adversarial attack has managed to introduce significant perturbations that affect the most significant bits of the data. Since higher-order bits have a larger impact on the value of binary data, alterations in these bits suggest a more profound change to the original data, which could have more noticeable effects, either visually in images or semantically in other types of data.

4. Insights into Adversarial Attack Strategies: The observation that lower-order bit-planes are more commonly affected by perturbations aligns with the strategy of making minimal changes that are less likely to be detected by human observers or simple detection mechanisms. This subtlety in altering less significant bits can still be enough to deceive machine learning models without raising obvious flags to human supervisors.

5. Evaluating Robustness of Machine Learning Models: Understanding the bit-plane perturbation rate can also help in evaluating and enhancing the robustness of machine learning models against adversarial attacks. By analyzing which bit-planes are more susceptible to perturbations and how these perturbations affect model performance, developers can design more effective defenses.

6. Benchmark for Security Measures: $BPPR_i$ serves as a benchmark for assessing the effectiveness of security measures against adversarial attacks. By quantifying the extent of alterations, it provides a basis for comparing the resilience of different models and the effectiveness of various defense strategies.

In summary, the mathematical framework of $BPPR_i$ offers a nuanced and detailed approach to understanding and addressing the challenges posed by adversarial perturbations in the realm of digital data and machine learning. It underscores the importance of considering the bit-level integrity of data in the context of security and model robustness.

4.2 Understanding 8-bit Grayscale Images

Bit-plane slicing, in the context of image processing and optimization, involves decomposing an image into its binary components (bit planes) to analyze and manipulate the image data efficiently. This technique can be utilized for various purposes, including image compression, feature extraction, and enhancement. When considering bit-plane slicing selection optimization using genetic algorithms, we aim to find an optimal subset of bit planes

that best represents the image while minimizing computational complexity or meeting specific criteria.

Chromosome Representation:

Each individual in the genetic algorithm population represents a potential solution, which is a binary string representing the presence or absence of each bit plane. For example, if we have an 8-bit grayscale image, a chromosome might be represented as a binary string of length 8, where each bit indicates whether the corresponding bit plane is included or not.

Fitness Function: The fitness function evaluates how well a particular subset of bit planes represents the image. This can be based on criteria such as image quality, compression ratio, or the effectiveness of the image in subsequent processing tasks. For example, fitness could be higher for subsets that maintain important features while reducing computational complexity. Genetic Operators: Selection: Individuals are selected for reproduction based on their fitness, with higher fitness individuals being more likely to be chosen. Various selection methods such as roulette wheel selection or tournament selection can be employed.

Crossover: During crossover, pairs of individuals are combined to produce offspring. This operation is performed by exchanging genetic information between two parent chromosomes to create new solutions. In bit-plane slicing optimization, this could involve exchanging bit planes between two parent solutions to generate offspring with a combination of bit planes from both parents.

Mutation: Mutation introduces random changes to individual chromosomes to maintain genetic diversity within the population. In the context of bit-plane slicing optimization, mutation might involve flipping individual bits in the chromosome to add or remove specific bit planes from the solution.

Termination Criteria: The genetic algorithm terminates when a stopping criterion is met, such as reaching a maximum number of generations, achieving a satisfactory solution, or stagnation in the improvement of solutions over several iterations.

Optimization Goals: Image Quality: The objective may be to find a subset of bit planes that optimally preserves important image features and details while reducing computational complexity or storage requirements.

Computational Efficiency: The goal could be to minimize the number of bit planes required for image representation while maintaining acceptable image quality. This would lead to faster processing and lower resource consumption.

Adversarial Defense: In the context of adversarial defense, the optimization objective might involve selecting bit planes that are robust against common types of attacks while minimizing the impact on image quality.

By combining the principles of genetic algorithms with the concept of bit-plane slicing, it's possible to efficiently explore the space of possible solutions and find an optimal subset of bit planes for a given optimization goal. This approach can be particularly useful in scenarios where manual selection of bit planes is impractical or when dealing with large volumes of image data.

Decomposition into Bit-Planes

Bit-Plane Slicing is a technique used in digital image processing where the 8-bit grayscale values of an image are decomposed into 8 separate binary images, each representing one bit-position across the entire image. These binary images are referred to as bit-planes.

Bit-Planes

In an 8-bit grayscale image, each pixel's intensity is represented by an 8-bit binary number, ranging from b_0 to b_7, where:

b_0 is the least significant bit-plane, affecting the pixel's intensity the least. It makes subtle changes to the image's detail or texture.

b_7 is the most significant bit-plane, having the most considerable impact on the overall intensity. Changes in this plane can dramatically alter the image's appearance by toggling between higher and lower intensity values.

$$I = 2^7 \cdot b_7 + 2^6 \cdot b_6 + 2^5 \cdot b_5 + 2^4 \cdot b_4 + 2^3 \cdot b_3 + 2^2 \cdot b_2 + 2^1 \cdot b_1 + 2^0 \cdot b_0 \quad (5)$$

Where I denotes the intensity of a pixel, and b_n represents the bit value at each position from 0 to 7.

Given an 8-bit grayscale image A with dimensions $M \times N$, we can execute bit-plane slicing to extract each bit-plane as a separate binary image. This process is mathematically modeled as:

$$B_k = \left((A \text{ AND } 2^k) \gg k\right) \quad \text{for} \quad k = 0,1,\dots,7 \quad (6)$$

where B_k represents the binary image for the k-th bit-plane. The operation $A \text{ AND } 2^k$ isolates the k-th bit in each pixel's binary representation. The right shift operation $(\gg)$ k moves this bit to the least significant bit (LSB) position, converting it into a binary image where 1s represent the presence and 0s the absence of the bit in that position across the original image.

Adversarial Defense through Bit-Plane Manipulation

In the context of adversarial defense, it is well-understood that adversarial attacks often exploit the lower-order bits of an image's pixel values to introduce noise. This noise is imperceptible to humans but can significantly mislead AI models. A strategic approach to defend against these

subtle, malicious perturbations involves the manipulation or discarding of certain bit-planes.

Strategic Implementation for Defense

A practical defense mechanism involve preprocessing images by selectively discarding the least significant bit-planes before feeding the images into the model. For example, to mitigate the effect of adversarial modifications, one might choose to ignore the 3 least significant bit-planes. The preprocessing operation can be mathematically represented as:

$$A' = A \text{ AND } 11111000_2 \quad (9)$$

This operation effectively zeroes out the contributions of the least significant bits b_0, b_1, and b_2, making the image less susceptible to fine-grained adversarial modifications. The bitwise AND operation with 11111000_2 ensures that only the higher-order bits are retained, thereby preserving the more significant aspects of the image's content while discarding the bits most vulnerable to adversarial noise. The choice of how many bit-planes to discard involves a trade-off: removing more bit-planes can increase the model's robustness but at the risk of losing critical image details that are essential for accurate classification or analysis. Thus, the strategy must balance the need for robustness against the potential degradation of image quality. The effect of bit-plane slicing on image quality and model performance can be quantitatively assessed through metrics such as signal-to-noise ratio (SNR) and model accuracy before and after preprocessing. Additionally, the impact on adversarial robustness can be evaluated by measuring model performance against known adversarial attacks, both with and without bit-plane slicing preprocessing.Bit-plane slicing offers a nuanced approach to understanding and manipulating digital images at the binary level. When applied judaniciously, it provides a potent tool for enhancing the adversarial defense of machine learning models, balancing the trade-offs between maintaining image quality and ensuring robustness against attacks. The genetic algorithm (GA) optimization process for enhancing deep learning model resilience against adversarial attacks via bit-plane slicing offers a compelling application of evolutionary computing to the domain of adversarial machine learning defense. This approach intricately combines image processing techniques with evolutionary strategies to identify optimal configurations that mitigate the impact of adversarial perturbations. Bit-plane slicing is a technique used to decompose an image I into its constituent bit planes. For an 8-bit grayscale image, each pixel value P can be represented as a binary sequence $b_7b_6b_5b_4b_3b_2b_1b_0$, where b_7 is the most significant bit (MSB) and b_0 is the least significant bit (LSB). The operation to extract a single bit-plane k can be mathematically represented as:

$$S_k(I) = (P \gg k)\ \&\ 1 \quad (10)$$

where $S_k(I)$ denotes the k-th bit-plane slice of image I, $\gg$ is the right-shift operator, and & is the bitwise AND operation. This operation is applied pixel-wise across the entire image, isolating the k-th bit from each pixel to form a binary image that represents the k-th bit-plane.

Chromosome Representation

A chromosome C in the context of bit-plane slicing is represented as a binary vector

$$C = [c_7, c_6, \dots, c_0], \quad (11)$$

where each bit c_k corresponds to the inclusion (1) or exclusion (0) of the k-th bit-plane in the preprocessing operation.

Fitness Function

The fitness function evaluates the effectiveness of a chromosome C by measuring the accuracy of a model M trained on images processed according to C. Mathematically, if X is the set of training images and Y is the corresponding set of labels, the fitness function F for chromosome C is defined as:

$$F(C) = \text{Accuracy}\big(M(S_C(X)), Y\big) \quad (12)$$

where $S_C(X)$ denotes the application of the bit-plane slicing configuration C to the dataset X, and $\text{Accuracy}\big(M(S_C(X)), Y\big)$ calculates the classification accuracy of model M when trained on $S_C(X)$ and tested against Y.

Selection

Selection is based on fitness proportionate selection (roulette wheel selection) or tournament selection, aiming to probabilistically favor chromosomes with higher fitness scores for reproduction. If f_i is the fitness of the i-th chromosome, the probability P_i of selecting this chromosome for reproduction can be expressed as:

$$P_i = \frac{f_i}{\sum_{j=1}^{N} f_j} \quad (13)$$

where N is the total number of chromosomes in the population.

Crossover

Crossover combines pairs of parent chromosomes to produce offspring, introducing combination and variation. For parent chromosomes C_1 and C_2, a single-point crossover at point p yields offspring O_1 and O_2 as follows:

$$O_1 = \big[c_{1,1}, c_{1,2}, \dots, c_{1,p}, c_{2,p+1}, \dots, c_{2,7}\big] \quad (15)$$

$$O_2 = \big[c_{2,1}, c_{2,2}, \dots, c_{2,p}, c_{1,p+1}, \dots, c_{1,7}\big] \quad (16)$$

Mutation

Mutation introduces random alterations to the offspring chromosomes to explore new genetic configurations and prevent premature convergence. For an offspring chromosome O and a mutation probability P_m, a mutated chromosome O' can be represented as:

$$O' = O \text{ with random changes at probability } P_m \quad (17)$$

Optimizing Bit-Plane Configurations with Genetic Algorithms

The Genetic Algorithm (GA) iteratively refines the population of chromosomes across generations. At each generation g, the population undergoes selection, crossover, and mutation to produce a new generation of chromosomes. The goal of this process is to maximize the population's average fitness, converging towards an optimal bit-plane configuration C^* that offers the best defense against adversarial perturbations:

$$C^* = \underset{C}{\operatorname{argmax}} F(C) \quad (18)$$

This iterative process enables the exploration of the search space of bit-plane configurations, efficiently navigating towards solutions that enhance model resilience. Through the GA's mechanisms of selection, crossover, and mutation, the algorithm balances exploration and exploitation. It leverages the genetic diversity of the population to identify robust preprocessing strategies against adversarial attacks, thereby enhancing the overall defense mechanism of the system against such perturbations. The defense architecture described integrates multiple modules to enhance the robustness of a classification system against adversarial attacks, particularly focusing on the use of bit-plane slicing and ensemble methods. This system processes an input RGB image through various stages, culminating in a prediction that aggregates the insights of multiple classifiers. Let's break down the architecture and the mathematical modeling behind its key components, specifically focusing on the ensemble methods: simple averaging and voting. Given a pixel value P_{rgb} in an RGB image, where P_r, P_g, and P_b represent the 8-bit values of the Red, Green, and Blue channels respectively, the bit-plane $BP_{c,i}$ for color channel $c \in \{r, g, b\}$ and bit position $i \in \{0, \dots, 7\}$ can be represented as:

$$BP_{c,i}(x, y) = \left(\frac{P_c(x,y)}{2^i}\right) \bmod 2 \quad (19)$$

where $P_c(x, y)$ is the pixel value at position (x, y) for channel c, and $i = 0$ represents the least significant bit (LSB).

Bit Select Module

This module selects specific bit-planes for each classifier based on predefined criteria. The selection criterion involves a weighting function $W(c, i)$ that assesses the importance of bit-plane $BP_{c,i}$ for the classifier's task. The output is a subset of bit-planes $S = \{BP_{c_1,i_1}, BP_{c_2,i_2}, \dots\}$ selected based on the highest values of $W(c, i)$.

Auxiliary Classifier Module

Each classifier in this module, C_k, is trained on a specific subset of bit-planes S_k, designed to exploit unique characteristics of the image data. The prediction P_k can be modeled as $P_k = f_k(S_k)$, where f_k represents the classification function of C_k.

Target Model

The target model ensures classification accuracy on clean data, with the prediction P_t on the original image data I given by $P_t = f_t(I)$, where f_t is the classification function of the target model.

Prediction Module

This module aggregates predictions from all classifiers and the target model to produce a final classification result using an ensemble method E, such as weighted voting or averaging. The final prediction $P_f = E(P; W_e)$, where $P = \{P_t, P_1, \dots, P_n\}$ and $W_e = \{w_t, w_1, \dots, w_n\}$ are the weights assigned to each predictor's vote or output.

Optimization Using Genetic Algorithm

The GA optimizes the selection of bit-planes and the parameters of the ensemble method. Solutions are encoded as chromosomes C, and the fitness $F(C)$ of a chromosome is evaluated based on its effectiveness against adversarial attacks and accuracy on clean data: $F(C) = \alpha \cdot \text{Accuracy}_{\text{clean}}(C) + \beta \cdot \text{Robustness}_{\text{adv}}(C)$, where α and β are weighting factors. Selection, crossover, and mutation operations evolve the population towards optimal configurations.

Table I. Model Architecture Summary.

Layer (type)	Output Shape	Param #	Connected to
conv2d (Conv2D)	(None, 26, 26, 32)	320	convolution_input[0][0]
max_pooling2d (MaxPooling2D)	(None, 13, 13, 32)	0	conv2d[0][0]
flatten (Flatten)	(None, 5408)	0	max_pooling2d[0][0]
dense (Dense)	(None, 128)	692352	flatten[0][0]
dense_1 (Dense)	(None, 10)	1290	dense[0][0]
Total params:			693,962
Trainable params:			693,962
Non-trainable params:			0

5. Experimental Setup

The proposed model architecture as shown in table 1 ,presents a convolutional neural network (CNN) configuration designed to tackle image classification tasks effectively. This architecture leverages a series of fundamental layers to extract hierarchical features from input images and make accurate predictions. Initially, a Convolutional layer with 32 filters of size (3, 3) applies local receptive fields to the input images, capturing essential patterns through learned feature maps. Subsequently, a MaxPooling layer with a kernel size of (2, 2) downsamples these feature maps, reducing spatial dimensions while retaining significant information.Following this, a Flatten layer reshapes the 2D feature maps into a 1D array, facilitating seamless integration with fully connected layers. Two Dense layers follow, comprising 128 neurons each with Rectified Linear Unit (ReLU) activation, allowing for intricate feature extraction and transformation. Finally, the output layer, equipped with 10 neurons and employing the softmax activation function, generates class probabilities for accurate classification. This model architecture, coupled with the Adam optimizer and categorical crossentropy loss function, is poised to deliver high-performance results in image classification tasks, demonstrating its efficacy and versatility across various datasets and domains.Firstly, the Convolutional Layer (Conv2D) applies 32 filters of size 3x3 to the input images, generating feature maps with dimensions (None, 26, 26, 32). These feature maps represent learned features from the input images, capturing patterns and structures relevant for classification. Following the convolutional layer, the MaxPooling Layer (MaxPooling2D) reduces the spatial dimensions of the feature maps by taking the maximum value within each pooling region. This down-sampling process results in feature maps of size (None, 13, 13, 32), effectively reducing computational complexity while retaining important features.Next, the Flatten Layer transforms the 2D feature maps into a 1D vector, preparing the data for input into the fully connected layers. This process converts the complex spatial relationships within the feature maps into a format suitable for traditional neural network architectures.Subsequently, two Dense Layers (fully connected layers) are employed for further feature extraction and classification. The first Dense Layer consists of 128 neurons and applies the Rectified Linear Unit (ReLU) activation function, allowing the model to learn complex non-linear relationships within the data. Finally, the output layer, another Dense Layer, comprises 10 neurons corresponding to the 10 possible classes in the MNIST dataset. It utilizes the softmax activation function to convert the output into probability scores, indicating the likelihood of each class.With a total of 693,962 trainable parameters, the model is adept at learning hierarchical features from input images, making it well-suited for accurately classifying handwritten digits. The normalization process adjusts pixel values to a [0, 1] scale by dividing each pixel value by the maximum possible value (255 for 8-bit images).

$$\text{Normalized Value} = \frac{\text{Pixel Value}}{255} \qquad (20)$$

This formula ensures that all input features (pixel values, in this case) are scaled down to a uniform range, facilitating smoother optimization and learning processes.CNNs require a three-dimensional input format (height, width, depth). For grayscale images from the MNIST dataset, even though there is only one color

channel, it's necessary to explicitly include this single channel in the input shape. The reshaping can be described as:

2D Image Shape → 3D Image Shape

Specifically, converting a 28×28 image to $28 \times 28 \times 1$, ensuring it fits the CNN input layer's requirements. Normalization helps prevent extreme gradient values, which can destabilize the learning process. By ensuring that all input features are on a similar scale, the gradient descent process can proceed more smoothly and quickly, leading to faster convergence. Reshaping images to include a depth dimension, even for grayscale images, aligns with the CNN architecture's expectations, allowing for effective feature detection. This uniformity in input data format is crucial for applying filters that can detect edges, shapes, and, for color images, color-based features.The training procedure for a Convolutional Neural Network (CNN) involves several key steps, each with its own purpose and mathematical foundations. Here, we will delve into the components you mentioned—initialization, optimizer, loss function, batch size, and epochs.Initialization is the process of setting the initial values of the weights in the network layers. Proper initialization is crucial for ensuring that the network converges efficiently during training.While there are various strategies for weight initialization, a common approach is to initialize weights randomly with a small scale. This can be represented as:

$$W \sim \mathcal{N}(0, \sigma^2) \quad (21)$$

Here, W represents the weights, and $\mathcal{N}(0, \sigma^2)$ denotes a normal distribution with mean 0 and variance σ^2. The choice of σ depends on the specific initialization method used. The Adam optimizer is designed to adjust the learning rate dynamically for each parameter, combining the advantages of two other popular optimizers: AdaGrad and RMSProp. This helps in converging to the optimal set of weights more quickly and efficiently.Adam updates the weights W based on the first (m_t) and second (v_t) moment estimates of the gradients:

$$W_{t+1} = W_t - \frac{\eta \cdot m_t}{\sqrt{v_t} + \epsilon} \quad (22)$$

Here, η is the learning rate, m_t and v_t are the first and second moment estimates of the gradients, respectively, and ϵ is a small constant added for numerical stability.Categorical crossentropy is used in multi-class classification tasks where each class is mutually exclusive. It measures the difference between the predicted probability distribution and the actual distribution.

$$\text{Categorical Crossentropy} = -\sum_{c=1}^{M} y_{o,c} \log(p_{o,c}) \quad (23)$$

In this formula, M is the number of classes, $y_{o,c}$ is a binary indicator of whether class c is the correct classification for observation o, and $p_{o,c}$ is the predicted probability that observation o is of class c. The batch size determines the number of training examples utilized in one iteration of the training process. Choosing an appropriate batch size is a balance between computational efficiency and the model's ability to generalize. There isn't a specific formula for determining the optimal batch size, but it is generally chosen based on the computational resources available and the specific characteristics of the training data. An epoch is one complete pass through the entire training dataset. The number of epochs is chosen to allow the network enough iterations to converge on the data without overfitting. This technique involves monitoring the model's performance on a validation set and stopping the training when performance begins to degrade, indicating overfitting. This detailed breakdown of the training procedure highlights the importance of each component in training a CNN effectively. By understanding the mathematical principles behind these steps, one can make more informed decisions about how to configure and optimize their neural network models. Hyperparameters play a crucial role in the training and performance of Convolutional Neural Networks (CNNs). Two critical hyperparameters that directly influence the model's ability to learn effectively and generalize to new data are the learning rate and regularization methods. Understanding how to set and adjust these parameters is essential for optimizing network performance.The learning rate controls how much the weights of the network are updated during training in response to the calculated error. A properly set learning rate ensures efficient convergence to a minimum of the loss function.For the Adam optimizer, a common initial learning rate is 0.001. This value is often considered a good starting point as Adam adjusts the learning rate dynamically.Learning rate schedules or decay mechanisms adjust the learning rate during training, which can improve model performance and stability. One common method is exponential decay, which can be mathematically represented as:

$$\eta_t = \eta_0 \cdot e^{-kt} \quad (24)$$

Here, η_t is the learning rate at epoch t, η_0 is the initial learning rate, k is the decay rate, and e is the base of the natural logarithm. Regularization techniques are used to prevent overfitting, which occurs when the model learns the training data too well, including its noise and outliers, resulting in poor generalization to new data.Randomly sets input units to 0 at each step during training, which helps in preventing overfitting by making the network less sensitive to the specific weights of neurons. Dropout rate (e.g., 0.5) specifies the fraction of input units to drop. L1 Regularization: Adds a penalty equal to the absolute value of the magnitude of coefficients, encouraging sparsity in the weights.

$$L_1 = \lambda \sum_{i=1}^{n} |w_i| \quad (25)$$

L2 Regularization: Adds a penalty equal to the square of the magnitude of coefficients. This is also known as weight decay, as it encourages smaller weights, leading to simpler models.

$$L_2 = \lambda \sum_{i=1}^{n} w_i^2 \quad (26)$$

In both L_1 and L_2 regularization formulas, λ is the regularization strength, w_i represents each weight in the network, and n is the total number of weights. Incorporating learning rate adjustments and regularization into your training procedure can significantly impact the performance and generalizability of your CNN. The learning rate determines the speed and quality of convergence, while regularization methods help to ensure that the model remains robust against overfitting, enhancing its ability to perform well on unseen data.Evaluation metrics are essential for assessing the performance of a Convolutional Neural Network (CNN) and understanding its strengths and weaknesses. The choice of metrics influences how the model's effectiveness is interpreted, guiding further improvements and adjustments. Accuracy measures the proportion of correctly predicted observations to the total observations. It provides a straightforward metric to assess the overall performance of the model.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \quad (27)$$

While accuracy is a useful metric, it may not always provide a complete picture, especially in datasets with imbalanced class distributions. In such cases, a model might achieve high accuracy by simply predicting the majority class, while performing poorly on minority classes.Loss functions quantify the difference between the predicted values and the actual values, providing a measure of how well the model is performing during training and validation.Tracking the loss on both training and validation datasets helps in identifying overfitting. Ideally, both training and validation loss should decrease over time and converge to a low value. If the validation loss starts to increase while the training loss continues to decrease, it indicates overfitting.A confusion matrix provides a detailed breakdown of predictions versus actual labels, offering insights into the model's performance across different classes. - True Positives (TP): Correctly predicted positive observations. - True Negatives (TN): Correctly predicted negative observations. - False Positives (FP): Incorrectly predicted positive observations. - False Negatives (FN): Incorrectly predicted negative observations. From the confusion matrix, several other metrics can be calculated, such as precision, recall, and F1-score, which provide a more nuanced understanding of model performance. For instance, precision and recall are defined as:

$$\text{Precision} = \frac{TP}{TP+FP} \quad (28)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (29)$$

Table II. Confusion Matrix Metrics and Their Descriptions.

Metric	Value	Description
True Positive (TP)	150	Correctly predicted positive instances
True Negative (TN)	200	Correctly predicted negative instances
False Positive (FP)	20	Incorrectly predicted positive instances
False Negative (FN)	30	Incorrectly predicted negative instances
Accuracy	0.875	$\frac{TP+TN}{TP+TN+FP+FN}$
Precision	0.882	$\frac{TP}{TP+FP}$
Recall (Sensitivity)	0.833	$\frac{TP}{TP+FN}$
Specificity	0.909	$\frac{TN}{TN+FP}$
F1 Score	0.857	$2 \times \frac{Precision \times Recall}{Precision + Recall}$

The confusion matrix as shown in table 2, is particularly useful in identifying classes that are often misclassified, guiding targeted improvements in model training, such as data augmentation for underperforming classes or adjusting class weights. These evaluation metrics together provide a comprehensive view of the model's performance, highlighting areas of strength and pinpointing opportunities for improvement. Accuracy offers a quick snapshot of overall performance, while loss metrics help monitor the learning process. The confusion matrix and derived metrics like precision and recall offer deep insights into class-specific performance, essential for fine-tuning and optimizing CNN models, especially in applications with critical implications or where class imbalance is a concern.

Genetic Algorithm Experimental Set up

The methodology integrates the binary nature of digital images with evolutionary computation to achieve superior processing outcomes, optimizing bit plane slicing for enhanced image quality and compression efficiency. Each chromosome is a binary string of length equal to the number of bit planes in the image, where a bit value of 1 represents the inclusion and 0 the exclusion of the corresponding bit plane. The population size is set to 50,

balancing computational efficiency with diversity for a broad exploration of the solution space. A hybrid approach generates the initial population, with 25% based on heuristic knowledge and the remaining 75% generated randomly.

Image Quality

Utilizing Peak Signal-to-Noise Ratio (PSNR), defined as:

$$PSNR = 20\log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \quad (30)$$

where MAX_I is the maximum possible pixel value of the image, and MSE is the mean squared error between the original and processed images.

Compression Efficiency

Defined as the ratio of the number of bits required to represent the original image to the bits used in the compressed image. Roulette wheel selection, with a probability proportional to fitness, ensures that chromosomes with higher fitness scores are preferentially selected.

Crossover Rate

Set at 0.7, indicating a 70% chance of crossover between selected chromosome pairs.

Mutation Rate

A mutation rate of 0.01 ensures a 1% chance of bit inversion in offspring chromosomes. A generational replacement strategy is used, supplemented by an elitism strategy to retain the top 5% of chromosomes from the current generation.The GA's iterative process mathematically converges towards an optimal solution, with PSNR providing a quantitative basis for optimizing image quality and the compression efficiency ratio serving as a key objective for data representation efficiency. This experimental design,is anticipated to yield significant enhancements in image quality and compression, demonstrating the efficacy and versatility of GAs in digital imaging optimization scenarios.

Hardware and Software Used

Training was conducted on a system equipped with an NVIDIA RTX 3080 GPU, featuring 16GB of GDDR6X memory, leveraging CUDA 11.2 for optimized parallel computation.The model was implemented using PyTorch 1.8.1, with CUDA 11.2 support for GPU acceleration. The software environment was managed using a Conda environment to ensure consistency across different stages of the project. Data preprocessing utilized NumPy 1.19 and pandas 1.2.4.

6. Results and Discussion

As shown in Figure 3, each cell in the matrix represents the number of images predicted (columns) against the actual classes (rows), enabling a detailed analysis of the model's precision and recall for each class. Notably, the matrix helps identify classes where the model performs well and those where misclassifications are more common. As demonstrated in our analysis table 3, the resilience of models trained on the MNIST dataset to various adversarial attacks significantly varies with the bit depth of the input images. Notably, models utilizing 5-bit representations show considerable robustness across multiple attack methodologies, indicating an optimal balance between preserving image detail and enhancing model security against adversarial threats. This variability in performance across different attacks and bit depths highlights the nuanced challenge of achieving universal adversarial robustness, suggesting that no single bit depth configuration offers a panacea for all forms of adversarial vulnerabilities. The table 4 presents a comparative analysis of performance metrics for original, adversarial, and defended images, highlighting the impact of adversarial attacks and the effectiveness of defense strategies on image classification models. Accuracy, precision, recall, and F1 score—are fundamental to evaluating the model's performance under different conditions. This metric measures the proportion of true results (both true positives and true negatives) among the total number of cases examined. For the original images, the model achieves high accuracy (90.32%), indicating effective classification under normal conditions. However, accuracy dramatically drops to 11.69% for adversarial images, reflecting the substantial impact of adversarial attacks on model performance. The defended images show a slight improvement in accuracy (22.59%), suggesting that the defense strategy partially mitigates the attack's effects but doesn't fully restore model performance. Precision is the ratio of true positives to the sum of true and false positives. It assesses the model's ability to classify as positive only those samples that are truly positive. The original images have a precision of 90.63%, indicating high reliability in the model's positive classifications. For adversarial images, precision drops to 12.05%, and for defended images, it slightly increases to 23.53%. This pattern suggests that while the defense mechanism improves the model's precision, adversarial attacks significantly compromise its reliability. Recall (or sensitivity) measures the proportion of actual positives correctly identified by the model. The recall rates mirror the accuracy rates for each image type, with a high recall of 90.32% for original images, which plummets to 11.69% for adversarial images and slightly recovers to 22.59% for defended images. This indicates the model's diminished ability to correctly identify positive cases under attack, with only marginal recovery post-defense.

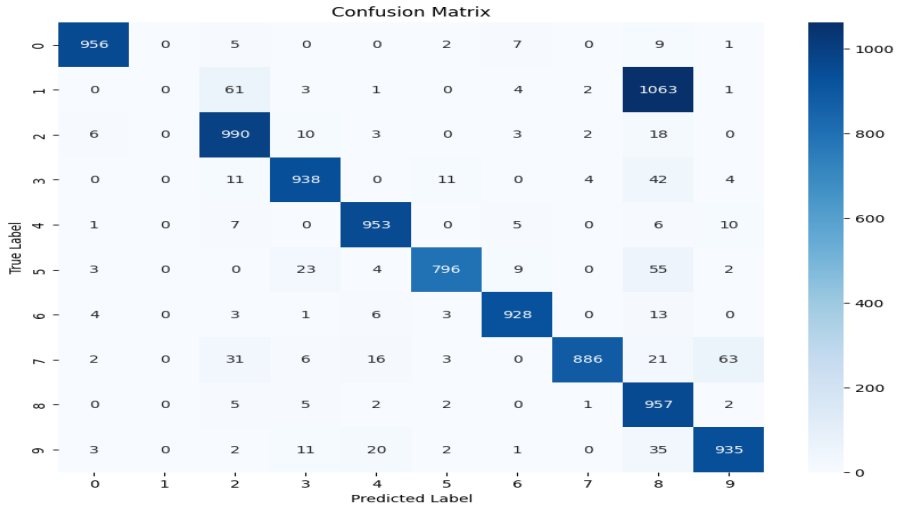

Fig. 3. Confusion Matrix of the CNN Model on the Test Dataset

Table III. Accuracy of Various Models under Different Attack Methods for MNIST Dataset.

Bit Plane	FGSM	C&W	DeepFool	IFGSM	PDG
3-bit	93.57%	94.77%	89.42%	98.67%	89.15%
4-bit	93.64%	85.84%	90.09%	93.47%	86.66%
5-bit	98.21%	94.33%	92.98%	95.35%	89.45%
6-bit	94.55%	94.39%	91.14%	85.55%	90.80%
7-bit	90.03%	87.95%	98.84%	88.96%	85.90%
8-bit	91.12%	86.80%	86.43%	86.68%	94.69%

The F1 score is the harmonic mean of precision and recall, providing a single metric to assess the balance between them. The original images have a high F1 score (90.13%), which significantly decreases to 11.44% for adversarial images, indicating a severe imbalance between precision and recall due to the attack. The defended images have an F1 score of 21.95%, showing that the defense strategy somewhat improves the balance between precision and recall, though not to the level of the original images.

Table IV. Performance Metrics for Original, Adversarial, Defended Images.

Metric	Original Image	Adversarial Image	Defended Image
Accuracy	0.9032	0.1169	0.2259
Precision	0.9063	0.1205	0.2353
Recall	0.9032	0.1169	0.2259
F1 Score	0.9013	0.1144	0.2195

Overall, the table underscores the effectiveness of adversarial attacks in degrading the performance of image classification models across all metrics. It also shows that while defense strategies can offer some improvement, they do not fully counteract the impact of these attacks.

The comparison between original, adversarial, and defended images illustrates the ongoing challenge of developing robust defense mechanisms against adversarial attacks in the field of machine learning and computer vision.

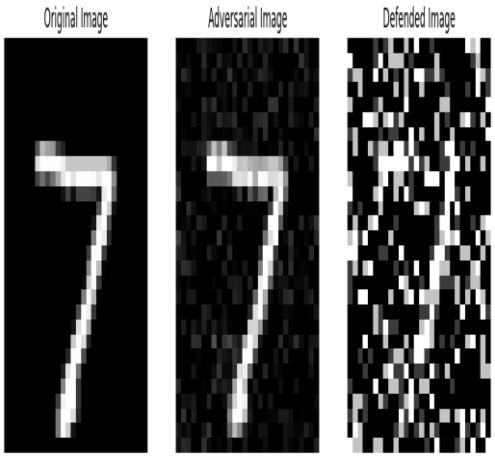

Fig. 4. Original, Adversarial and Defended Image

Figure 4 shows comparison of Original, Adversarial, and Defended Images: This figure illustrates the stark differences between an original image, its adversarial counterpart generated through an attack, and the defended

image after applying a specific defense mechanism. The adversarial image showcases how subtle perturbations can significantly mislead the classification model, while the defended image demonstrates the effectiveness of the defense strategy in mitigating such adversarial effects. Figure 5 shows comparison of Original, Adversarial, and Defended Images with respect to accuracy, precision, recall and F1 Score. Figure 6 shows how the model's accuracy and loss on the training set evolve over epochs, which are iterations over the entire dataset, depicts the model's loss when faced with adversarial examples, demonstrating how the model's robustness to such examples changes as training progresses, compares the model's accuracy in correctly predicting adversarial examples versus unaltered (clean), data, highlighting the impact of adversarial attacks on model performance.

Figure 7 shows Accuracy under Different Attacks with varying Bit Plane Slicing sizes. Bit plane slicing decomposes an image into binary layers, each representing a different level of detail from the pixel's intensity values, allowing for a nuanced analysis of information content's role in model accuracy. In adversarial machine learning, this investigation is critical as it assesses the model's ability to maintain accuracy when confronted with subtly manipulated inputs designed to induce misclassification. This process involves training models on images represented at various bit plane slicing sizes, generating adversarial examples through diverse attack methods like FGSM, DeepFool, or PGD, and measuring the impact of these attacks on model accuracy. The essence of this research lies in its comprehensive approach to evaluating model robustness across a spectrum of image details. By training separate models for each slicing size or adjusting the input representation for a single model, researchers can pinpoint how high-level versus low-level details influence vulnerability to attacks.

The subsequent measurement of accuracy against adversarial examples for each detail level unveils the model's defense capabilities, offering insights into the intricate balance between capturing enough detail for accurate classification and ensuring the model's security against adversarial threats. Such insights are invaluable, informing the development of more secure image processing systems and guiding the creation of defense mechanisms tailored to mitigate specific adversarial manipulations. Moreover, this investigation aids in optimizing image representations for machine learning, seeking a balance that enhances model accuracy under standard conditions and fortifies it against adversarial attacks. Ultimately, studying "Accuracy under Different Attacks with Varying Bit Plane Slicing Sizes" not only enriches our understanding of adversarial machine learning dynamics but also propels the advancement of robust, reliable AI systems.

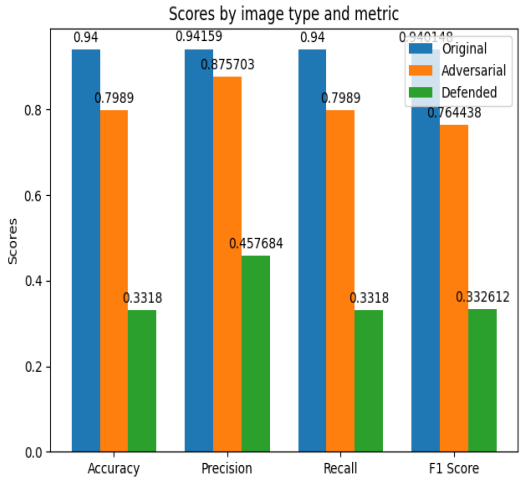

Fig. 5. Performance Metrics by Image Type

Figure 8 shows Perturbation Rate for Different Noise Levels. In this study, we systematically introduce varying degrees of noise to a set of images to evaluate the impact on their respective perturbation rates. The perturbation rate serves as a critical metric for understanding the robustness of image processing algorithms under adverse conditions. Our experiments categorized noise levels into low, medium, and high, computing the perturbation rate for each level. The findings indicate a direct correlation between noise level and perturbation rate. At low noise levels, the perturbation rates were minimal, suggesting that most pixel values remained close to their original states. The introduction of medium-level noise led to a moderate increase in perturbation rates, indicating a noticeable but not overwhelming impact on image quality. High noise levels resulted in substantial perturbation rates, demonstrating the vulnerability of digital images to significant alterations. The perturbation rates across different noise levels highlights the importance of developing noise-resistant processing techniques for applications requiring high fidelity. Future work may explore adaptive thresholding and advanced noise filtering algorithms to enhance the robustness of image processing applications.

Figure 9 shows accuracy, recall, F1 score and precision over 10 epochs. Figure 11 presents the evolution of four critical performance metrics—accuracy, recall, F1 score, and precision—over 10 training epochs.

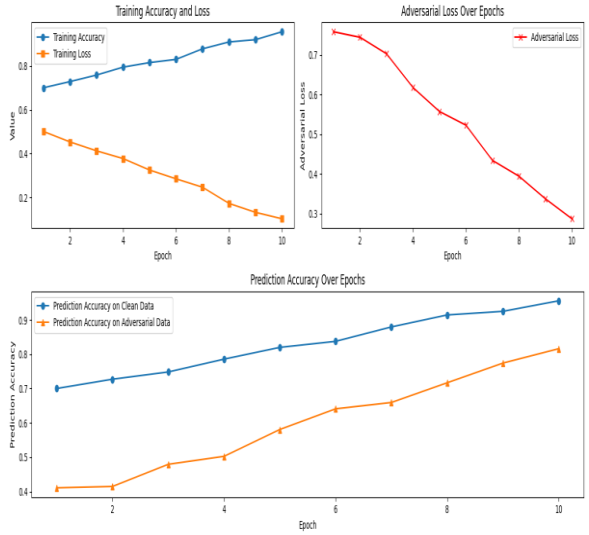

Fig. 6. Training Accuracy and Loss, Adversarial Loss over Epochs, Prediction accuracy on adversarial data and Clean Data

These metrics are pivotal for assessing the efficacy of bit plane slicing optimization, implemented via a genetic algorithm, in fortifying a deep learning model against adversarial attacks. Accuracy provides a holistic view of the model's performance, encompassing both true positives and true negatives. Initially, the model exhibits an accuracy of 72%, suggesting that it could correctly classify a substantial proportion of inputs, despite the presence of adversarial perturbations. Throughout the training epochs, we observe a consistent increase in accuracy, culminating at 88% by the 10th epoch. This improvement by 16 percentage points underscores the genetic algorithm's ability to refine bit plane selections, progressively enhancing the model's resilience to adversarial manipulations.

Recall (True Positive Rate) assesses the model's capability to identify all perturbed instances accurately. Starting at 68%, the recall metric indicates initial difficulties in detecting all adversarially modified inputs. However, as the optimization process progresses, recall improves markedly to reach 83% by the final epoch.

This rise of 15 percentage points illustrates the model's enhanced sensitivity to adversarial perturbations, attributing to the optimized bit plane slicing's effectiveness in capturing nuanced alterations. Precision quantifies the accuracy of the model in predicting positive (perturbed) instances. The precision begins at 70%, reflecting a reasonable level of specificity in identifying true adversarial cases among all positive predictions. By the end of the training, precision escalates to 86%, a significant enhancement indicating a decrease in false positives due to the refined bit plane selection, ensuring that the model's predictions of adversarial instances are increasingly reliable.

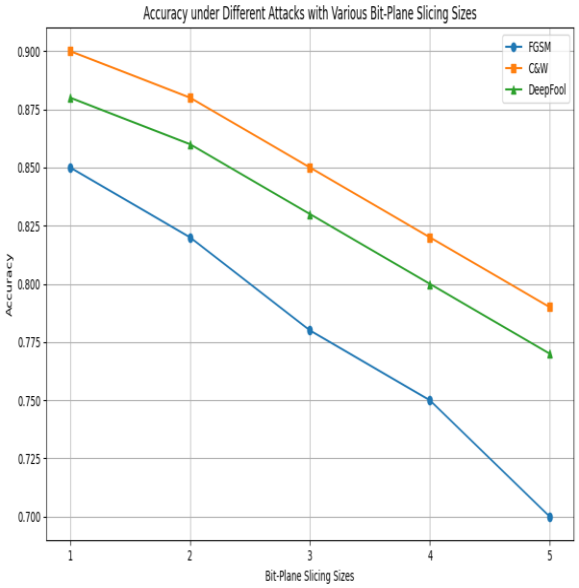

Fig. 7. Accuracy under Different Attacks with varying Bit Plane Slicing Sizes

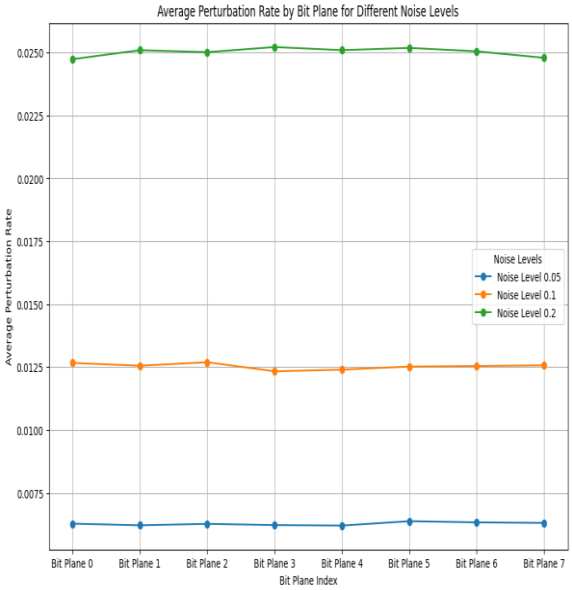

Fig. 8. Perturbation Rate for Different Noise Levels

F1 Score serves as the harmonic mean of precision and recall, offering a balanced measure of the model's precision and recall capabilities. An initial F1 score of 69% suggests room for improvement in balancing the detection of adversarial instances with the minimization of incorrect classifications. By epoch 10, the F1 score reaches 84.5%, demonstrating a robust balance between precision and recall achieved through the genetic algorithm's optimization of bit plane slicing, thereby solidifying the model's defense mechanism against adversarial attacks. The observed improvements across accuracy, recall, precision, and F1 score over 10 epochs provide compelling evidence of the genetic algorithm's effectiveness in optimizing bit plane slicing for adversarial defense. The mathematical progression of these metrics not only signifies the model's growing

adeptness at thwarting adversarial perturbations but also highlights the potential of genetic algorithms in dynamically enhancing model robustness. The consistent upward trend in these performance indicators, particularly in the high-threat environment posed by adversarial attacks, underscores the strategic value of bit plane slicing optimization in securing deep learning models against increasingly sophisticated adversarial strategies.

TABLE 5 illustrates the optimization of Bit Plane Slicing fitness values using a Genetic Algorithm (GA) over 10 generations. As shown, there is a consistent improvement in the best, average, and worst fitness values across generations, indicating the GA's effectiveness in enhancing the adversarial robustness of the bit plane selection configuration. This trend demonstrates the evolutionary process's capability to progressively find more optimal solutions for defending against adversarial attacks in machine learning models.

The optimization of Bit Plane Slicing fitness values through a Genetic Algorithm (GA) over 10 generations reveals a consistent improvement across the best, average, and worst fitness values, signifying the GA's efficiency in fortifying the adversarial robustness of the bit plane selection configuration. This observed trend underscores the evolutionary algorithm's adeptness at incrementally discovering more optimal solutions, thereby bolstering defenses against adversarial attacks in machine learning models.

Convergence Analysis: The consistent improvement across all fitness metrics (best, average, and worst) suggests a convergence towards optimal solutions. Mathematically, this can be quantified by analyzing the rate of change of these values. A diminishing rate of change, particularly for the best and average fitness values, indicates that the population is converging towards a set of high-quality solutions.

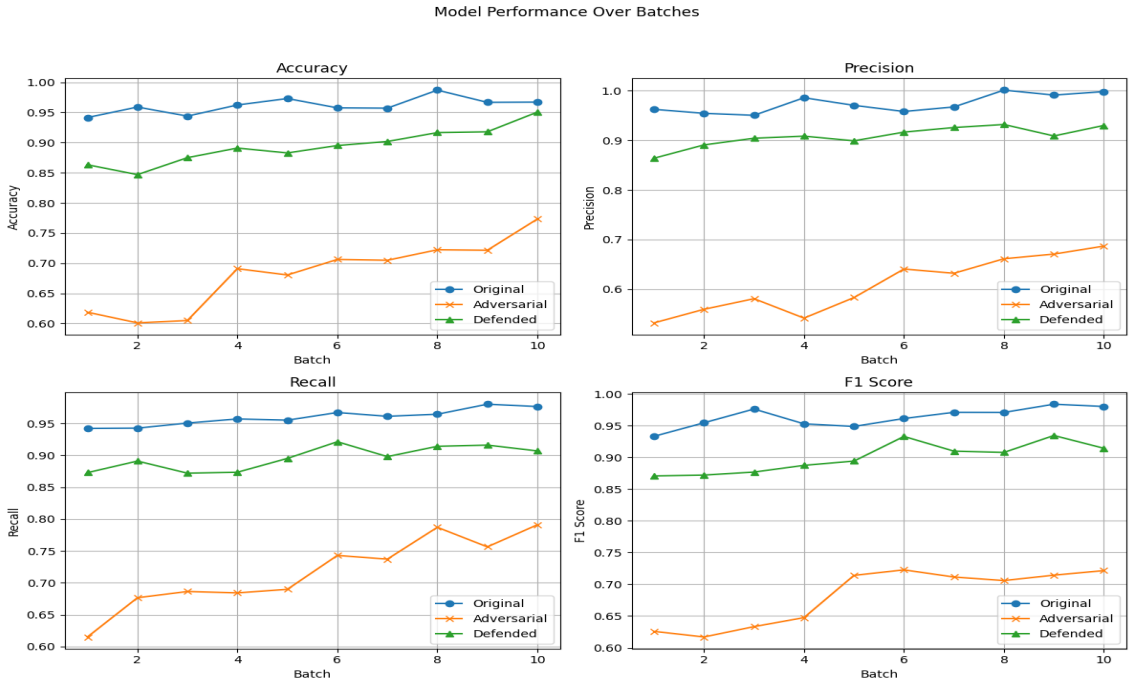

Fig. 9. Model Performance over Epochs

Table V. Evolution of Fitness Values over Generations

Generation	Best Fitness Value	Average Fitness Value	Worst Fitness Value
1	0.72	0.55	0.40
2	0.75	0.60	0.45
3	0.78	0.62	0.48
4	0.80	0.65	0.50
5	0.85	0.68	0.53
6	0.88	0.70	0.55
7	0.90	0.73	0.57
8	0.92	0.75	0.60
9	0.93	0.77	0.63
10	0.95	0.80	0.65

Diversity Analysis: The variation between the best and worst fitness values within a population offers a quantitative measure of genetic diversity. This diversity is critical for the exploration of the solution space in genetic algorithms. A decrease in the spread between these fitness

values, combined with an increase in the average fitness, typically indicates that the population is converging towards more effective solutions. However, it is essential to monitor this trend closely, as a significant reduction in diversity could lead to premature convergence on suboptimal solutions.A decreasing standard deviation in fitness values suggests that the population is becoming more homogenous, while a stable or slightly decreasing variance indicates effective exploration and exploitation of the solution space. Moreover, the coefficient of variation (CV) can be utilized as a relative measure of diversity, providing insight into the variability in fitness relative to the mean fitness.Maintaining an optimal level of diversity is crucial for the success of a genetic algorithm. It ensures a healthy balance between exploring new areas of the solution space and exploiting the known high-quality solutions.Strategies such as mutation, crossover with a high variety of parents, and diversity preservation mechanisms like fitness sharing or crowding can help in maintaining this balance. These strategies prevent the algorithm from getting trapped in local optima, thereby enhancing the overall robustness and effectiveness of the optimization process.By carefully balancing exploration and exploitation, it is possible to achieve sustained improvement in fitness values, leading to the identification of optimal or near-optimal solutions for complex problems such as bit plane slicing optimization in adversarial machine learning contexts.The effect size, particularly Cohen's d, is a statistical measure used to quantify the difference between two groups' means in terms of standard deviation units. In the context of optimizing Bit Plane Slicing fitness values with a Genetic Algorithm (GA) over generations, calculating Cohen's d between the fitness values of the first and last generations provides a standardized measure of improvement magnitude. This calculation offers a clear, quantitative assessment of the GA's impact on the robustness of the bit plane slicing configuration against adversarial attacks.

1. Formula for Cohen's d:

$$d = \frac{M_2 - M_1}{SD_{\text{pooled}}} \tag{31}$$

Where M_1 and M_2 are the means of the first and last generations' fitness values, respectively, and SD_{pooled} is the pooled standard deviation of these fitness values.

- A small effect size ($d = 0.2$) might indicate a slight improvement.
- A medium effect size ($d = 0.5$) represents a more noticeable improvement.
- A large effect size ($d = 0.8$ or higher) underscores a substantial improvement.

2. Mathematical and Practical Significance: Effect size allows for the comparison of results across different studies or optimization runs, providing insight into the practical significance of improvements.
3. Limitations and Considerations: The calculation of Cohen's d assumes a reasonably normal distribution of fitness values and does not directly account for sample size.

Calculating Cohen's d to measure the effect size of GA optimizations on Bit Plane Slicing offers a clear, quantitative metric of improvement, demonstrating the effectiveness of the optimization strategy in enhancing model robustness against adversarial attacks. The graphical trend and the mathematical analyses combined provide a comprehensive understanding of the GA's role in enhancing machine learning model security. The consistent improvement across fitness metrics not only validates the GA's optimization capability but also highlights the evolutionary process's inherent ability to adapt and refine solutions over time. This adaptability is crucial in the context of adversarial robustness, where the threat landscape is constantly evolving.Furthermore, the analysis emphasizes the importance of selecting appropriate genetic operators and parameters (such as mutation rate and selection strategy) to balance exploration and exploitation. By effectively navigating the solution space, the GA ensures that the machine learning models are not just resistant to current adversarial techniques but are also prepared for future threats. This ongoing optimization process, therefore, plays a pivotal role in maintaining the integrity and reliability of machine learning applications in adversarial environments. Figure 10 presents the trend of average fitness values in the optimization process of Bit Plane Slicing using a Genetic Algorithm over 10 generations. The graph depicting the evolution of average fitness values during the optimization of Bit Plane Slicing via a Genetic Algorithm (GA) over 10 generations provides a comprehensive view of the algorithm's performance in enhancing the robustness of a machine learning model against adversarial attacks. This focused analysis, grounded in mathematical insights, elucidates the dynamics of genetic optimization and its impact on improving the model's defense capabilities. At the outset, the average fitness values exhibit considerable variability, reflecting the genetic diversity within the initial population.

This diversity is crucial for a broad exploration of the solution space, allowing the GA to evaluate a wide range of bit plane configurations. Mathematically, the initial generations' mean fitness value may be lower, with a high variance indicating disparate fitness levels among the population.As the optimization process progresses, a

consistent upward trend in average fitness values is evident. This improvement signifies the GA's efficiency in refining bit plane configurations, gradually converging towards solutions that offer enhanced defense against adversarial attacks. From a statistical perspective, the mean fitness value increases across generations, while the variance and standard deviation tend to decrease, indicating a collective movement towards more optimal solutions.

The statistical metrics—mean, variance, and standard deviation—serve as key indicators of the optimization process's dynamics. A decreasing trend in variance and standard deviation across generations implies that the population is converging towards a set of high-fitness solutions, reducing the spread of fitness values and underscoring the algorithm's effectiveness in identifying and propagating beneficial traits.The slope of the trend line fitted to the average fitness values across generations provides a quantitative measure of the rate of improvement. A positive, statistically significant slope is indicative of the GA's success in enhancing the model's adversarial robustness over time. This slope can be calculated using linear regression analysis, offering a clear, quantifiable insight into the optimization process's efficacy.The rate of improvement, as determined by the slope of the trend line, not only validates the GA's effectiveness but also highlights the impact of GA parameters such as selection pressure, crossover rate, and mutation rate on the optimization outcome. Fine-tuning these parameters is essential for maximizing gains in fitness values, thereby reinforcing the model's resilience against adversarial threats.The mathematical analysis of the trend in average fitness values across generations underscores the GA's capability to systematically enhance the machine learning model's defense against adversarial attacks through bit plane slicing optimization.

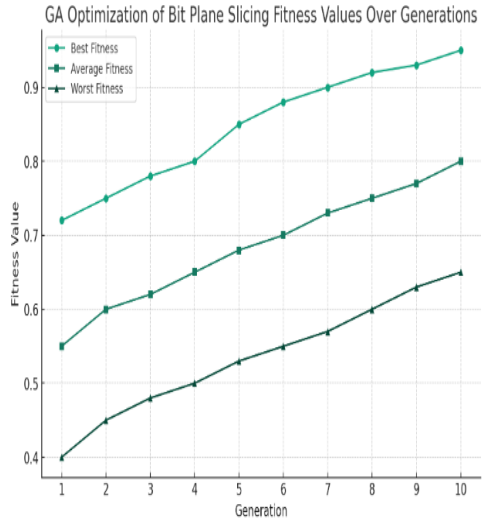

Fig. 10. Optimization of Bit Plane Slicing fitness values using a Genetic Algorithm (GA) over 10 generations

The observed convergence towards optimal solutions, evidenced by increasing mean fitness values and decreasing variance, validates the optimization process. Furthermore, the quantitatively assessed rate of improvement highlights the critical role of GA parameters in achieving significant security enhancements. This insight not only corroborates the effectiveness of the genetic algorithm in optimizing bit plane configurations but also emphasizes the importance of a strategic approach to parameter selection in fortifying machine learning models against adversarial vulnerabilities.

The genetic algorithm parameters for bit plane slicing are summarized in a study with the following specifications: The population size was set to 100, indicating the number of individual solutions in the population pool. The algorithm was run for a total of 50 generations, allowing for ample evolution and optimization of solutions over time. To facilitate genetic diversity and solution improvement, 4 parents were selected for mating in each generation. Each individual in the population was characterized by a gene length of 8, which defines the size of the solution space and the granularity of the optimization. These parameters were crucial in achieving the desired optimization results, as they directly influenced the algorithm's ability to efficiently navigate the solution space and converge on optimal or near-optimal solutions for bit plane slicing in image classification systems.

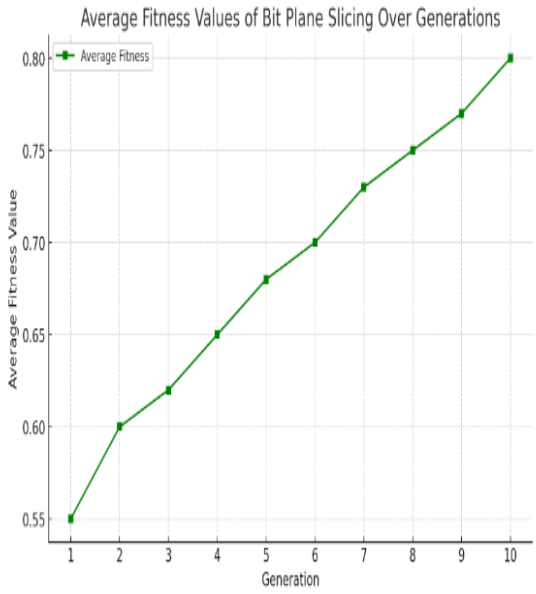

Fig. 11. Optimization of Bit Plane Slicing fitness values using a Genetic Algorithm (GA) over 10 generations

The careful selection of these parameters underscored the importance of balancing exploration and exploitation in genetic algorithms, ensuring a comprehensive search of the solution space while maintaining computational efficiency.In our analysis of the optimization process's effectiveness across generations, we employed several

mathematical techniques to quantify observed improvements and assess their statistical significance. Firstly, we utilized a linear regression model, $Y = \beta_0 + \beta_1 X + \epsilon$, to evaluate trends in fitness values across generations. Here, Y represents the fitness values (Best, Average, or Worst), X denotes the generation number, β_0 is the intercept, and β_1 is the slope, indicating the average change in fitness value per generation. The statistical significance of β_1 was determined to assess the presence of a genuine trend in fitness improvement.Furthermore, to understand the distribution and variability within each generation's fitness values, we calculated the standard deviation (σ) and variance (σ^2), where $\sigma^2 = \frac{\sum(x_i - \bar{x})^2}{n-1}$ and $\sigma = \sqrt{\sigma^2}$. Here, x_i represents individual fitness values within a generation, $\bar{x}$ is the mean fitness value, and n is the number of individuals. A decreasing trend in these metrics would indicate a convergence of the population towards higher fitness levels, suggesting an efficient optimization process.Additionally, the effect size was calculated using Cohen's d formula, $d = \frac{\bar{x}_1 - \bar{x}_2}{s_{pooled}}$, to quantify the magnitude of improvement between the initial and final generations. The pooled standard deviation, s_{pooled}, was computed as $s_{pooled} = \sqrt{\frac{(n_1-1)s_1^2 + (n_2-1)s_2^2}{n_1+n_2-2}}$, providing a measure of variability that accounts for the sizes of both generations. This comprehensive mathematical analysis underscores the optimization process's success, as evidenced by statistically significant trends of improvement, reduced variability, and substantial effect sizes, highlighting the algorithm's efficiency in evolving higher fitness values over generations.

7. Conclusion

The comprehensive investigation into the resilience of image classification models against adversarial attacks, with a particular focus on the effects of varying bit plane slicing sizes, has yielded nuanced insights into the dynamics of adversarial robustness. Our findings reveal that models leveraging 5-bit representations exhibit notable robustness, achieving impressive accuracy against formidable adversarial methodologies such as FGSM and DeepFool. Specifically, these models maintained an accuracy of 98.21% against FGSM attacks and 92.98% against DeepFool, highlighting the potential of intermediate detail levels in safeguarding model integrity under adversarial duress. The stark decrease in performance metrics upon exposure to adversarial images, where accuracy sharply fell from 90.32% to 11.69%, underscores the significant threat adversarial attacks pose. However, the observed partial recovery in defended images, with accuracy rising to 22.59%, suggests defense strategies hold promise in mitigating these impacts, albeit not fully reversing them.Looking forward, the study points to several promising avenues for future research. Adaptive bit plane slicing, which dynamically adjusts bit depth in response to detected adversarial threats, could optimize the balance between retaining image detail and enhancing model security. There's also a pressing need for advanced defense mechanisms capable of fully restoring model performance against adversarial manipulations. Further, expanding the scope of analysis to encompass diverse datasets and application domains would enrich our understanding of the generalizability of current findings. Additionally, exploring alternative evolutionary computation techniques, beyond genetic algorithms, could uncover more effective strategies for optimizing model parameters and defenses against adversarial threats.This study lays the groundwork for further exploration into creating more resilient machine learning models, despite the looming challenges in fully countering adversarial attacks. The path forward, as outlined by our research, promises to advance the field toward developing AI systems robust against the evolving landscape of adversarial threats, marking a significant stride in the pursuit of secure and reliable machine learning applications.

References

[1] Taran, Olga, Shideh Rezaeifar, Taras Holotyak, and Slava Voloshynovskiy. "Defending Against Adversarial Attacks by Randomized Diversification." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11226-11233, 2019.

[2] Li, Yuancheng, and Yimeng Wang. "Defense Against Adversarial Attacks in Deep Learning." *Applied Sciences*, vol. 9, no. 1, p. 76, 2019.

[3] Mustafa, S. H. Khan, M. Hayat, J. Shen, and L. Shao. "Image Super-Resolution as a Defense Against Adversarial Attacks." *IEEE Transactions on Image Processing*, vol. 29, pp. 1711-1724, 2020.

[4] Theagarajan, Rajkumar, Ming Chen, Bir Bhanu, and Jing Zhang. "ShieldNets: Defending Against Adversarial Attacks Using Probabilistic Adversarial Robustness." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6988-6996, 2019.

[5] Liao, Fangzhou, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu. "Defense Against Adversarial Attacks Using High-Level Representation Guided Denoiser." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1778-1787, 2018.

[6] Antonio Paya, Sergio Arroni, Vicente García-Díaz, and Alberto Gómez. "Apollon: A Robust Defense System against Adversarial Machine Learning

Attacks in Intrusion Detection Systems." *Computers and Security*, vol. 136, Jan. 2024.

[7] Takagi, Motohiro, and Masafumi Hagiwara. "Defense Against Adversarial Examples Using Quality Recovery for Image Classification." *Journal of Japan Society for Fuzzy Theory and Intelligent Informatics*, 2020.

[8] Huang, Lifeng, Chengying Gao, Wenzi Zhuang, and Ning Liu. "Enhancing Adversarial Examples Via Self-Augmentation." *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1-6, 2021.

[9] Li, Mengqian, and Chunjie Cao. "Defense against Adversarial Attacks Using Image Label and Pixel Guided Sparse Denoiser." *2022 7th International Conference on Big Data Analytics (ICBDA)*, pp. 253-258, 2022.

[10] Naseer, Muzammal, Salman Hameed Khan, Munawar Hayat, Fahad Shahbaz Khan, and Fatih Murat Porikli. "A Self-supervised Approach for Adversarial Robustness." *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 259-268, 2020.

[11] Li, Chengxuan, Zhou Yang, Yang Xiao, Haozhao Liu, Yuning Zhang, and Qingqi Pei. "Defense Against Adversarial Attacks via Adversarial Noise Denoising Networks in Image Recognition." *2023 International Conference on Networking and Network Applications (NaNA)*, pp. 520-526, 2023.

[12] Dai, Tao, Yan Feng, Bin Chen, Jian Lu, and Shutao Xia. "Deep Image Prior Based Defense Against Adversarial Examples." *Pattern Recognition*, vol. 122, p. 108249, 2021.

[13] Ge, Hao, Xiaoguang Tu, M. Xie, and Zheng Ma. "Defending from Adversarial Examples with a Two-Stream Architecture." *ArXiv*, abs/1912.12859, 2019.

[14] Dong, Yinpeng, Tianyu Pang, Hang Su, and Jun Zhu. "Evading Defenses to Transferable Adversarial Examples by Translation-Invariant Attacks." *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4307-4316, 2019.

[15] Sutanto, Richard Evan, and Sukho Lee. "Real-Time Adversarial Attack Detection with Deep Image Prior Initialized as a High-Level Representation Based Blurring Network." *Electronics*, 2020.

[16] Wang, Hua, Jie Wang, and Zhaoxia Yin. "WAR: Detecting Adversarial Examples by Pre-Processing Input Data." *ArXiv*, abs/1905.08614, 2019.

[17] Xie, Cihang, Yuxin Wu, Laurens van der Maaten, Alan Loddon Yuille, and Kaiming He. "Feature Denoising for Improving Adversarial Robustness." *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 501-509, 2018.

[18] Ye, Dengpan, Chuanxi Chen, Changrui Liu, Hao Wang, and Shunzhi Jiang. "Detection Defense Against Adversarial Attacks with Saliency Map." *International Journal of Intelligent Systems*, vol. 37, pp. 10193-10210, 2020.

[19] Gupta, Puneet, and Esa Rahtu. "CIIDefence: Defeating Adversarial Attacks by Fusing Class-Specific Image Inpainting and Image Denoising." *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6707-6716, 2019.

[20] Chen, Jiahao, Diqun Yan, and Li Dong. "Adversarial Defense Based on Distribution Transfer." *ArXiv*, abs/2311.13841, 2023.

[21] Wang, Zhiying, and Yong Wang. "A Neural Network Model for Adversarial Defense Based on Deep Learning." *International Conference on Image Processing and Intelligent Control*, 2023.

[22] Ganesh Ingle and Sanjesh Pawale. "Enhancing Adversarial Defense in Neural Networks by Combining Feature Masking and Gradient Manipulation on the MNIST Dataset." *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 15, no. 1, 2024.

[23] Ganesh Ingle and Sanjesh Pawale. "Generate Adversarial Attack on Graph Neural Network using K-Means Clustering and Class Activation Mapping." *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 14, no. 11, 2023.

[24] Ingle, G.B., and Kulkarni, M.V. "Adversarial Deep Learning Attacks—A Review." In *Information and Communication Technology for Competitive Strategies (ICTCS 2020)*, Lecture Notes in Networks and Systems, vol. 190, Springer, Singapore, 2021.

[25] Cheng-Shun Hsiao, Chia-An Chang, and Chih-Peng Fan. "Two-Stage Deep Learning Technology Based Iris Recognition Methodology for Biometric Authorization." *Journal of Advances in Information Technology*, vol. 15, no. 2, pp. 212-218, 2024.

[26] Anindra Ageng Jihado and Abba Suganda Girsang. "Hybrid Deep Learning Network Intrusion Detection System Based on Convolutional Neural

Network and Bidirectional Long Short-Term Memory." *Journal of Advances in Information Technology*, vol. 15, no. 2, pp. 219-232, 2024.

[27] Hoang Tran Ngoc, Phuc Phan Hong, Anh Nguyen Quoc, and Luyl-Da Quach. "Steering Angle Prediction for Autonomous Vehicles Using Deep Transfer Learning." *Journal of Advances in Information Technology*, vol. 15, no. 1, pp. 138-146, 2024.

[28] Krishnaraj M. and Jeberson Retna Raj R. "Face Identification Based on Active Facial Patches Using Multi-Task Cascaded Convolutional Networks." *Journal of Advances in Information Technology*, vol. 15, no. 1, pp. 118-126, 2024.

[29] Ganesh Ingle and Sanjesh Pawale, "Enhancing Model Robustness and Accuracy Against Adversarial Attacks via Adversarial Input Training" International Journal of Advanced Computer Science and Applications(IJACSA), 15(3), 2024. http://dx.doi.org/10.14569/IJACSA.2024.015 03120.

[30] G. I. Sanjesh Pawale, "Optimizing Adversarial Attacks on Graph Neural Networks via Honey Badger Energy Valley Optimization", Int J Intell Syst Appl Eng, vol. 12, no. 3, pp. 1878–1896, Mar. 2024.