Proposed method for against adversarial images based on ResNet architecture

DOI: https://doi.org/10.54654/isj.v2i22.1036

Truong Phi Ho, Pham Duy Trung, Dang Vu Hung, Nguyen Nhat Hai*

***Abstract*— Our world is becoming increasingly automated due to the application of deep learning/machine learning models to systems, but these systems are vulnerable to adversarial attacks, which create deceptive data to trick them. Without proper defenses, attackers can exploit deep learning systems in facial recognition, self-driving cars, and social media filters. Research on adversarial image generation and methods to against attacks is important. This paper proposes employing the ResNet architecture with adversarial training to against adversarial images. The model is tested on a Hybrid CIFAR-10 dataset, which is designed to improve robustness and accuracy by incorporating GAN-generated images. The proposed model achieves an accuracy of over 95%, which is better than three other state-of-the-art architectures VGG19_bn, ShuffleNetV2, and RepVGG_a2.**

***Tóm tắt*— Thế giới của chúng ta ngày càng tự động hóa do ứng dụng các mô hình học sâu/ học máy vào các hệ thống, nhưng các hệ thống này dễ bị tấn công đối kháng, tạo ra dữ liệu lừa đảo để đánh lừa chúng. Nếu không có biện pháp phòng thủ phù hợp, kẻ tấn công có thể khai thác các hệ thống học sâu trong nhận dạng khuôn mặt, xe tự lái và các bộ lọc phương tiện truyền thông xã hội. Nghiên cứu về việc tạo hình ảnh đối kháng và phương pháp chống lại các cuộc tấn công rất quan trọng. Bài báo này đề xuất sử dụng kiến trúc ResNet kết hợp đào tạo đối kháng để bảo vệ chống lại hình ảnh đối kháng. Mô hình được thử nghiệm trên tập dữ liệu Hybrid CIFAR-10, được thiết kế để cải thiện độ mạnh mẽ và độ chính xác bằng cách kết hợp hình ảnh do GAN tạo ra. Mô hình đề xuất đạt độ chính xác trên 95%, kết quả cao hơn so với 3 kiến trúc hiện đại khác là VGG19_bn, ShuffleNetV2 và RepVGG_a2.**

***Keywords*— deep learning; ResNet architecture; adversarial attack; adversarial training.**

***Từ khóa*— học sâu; kiến trúc ResNet; tấn công đối kháng; huấn luyện đối kháng.**

I. INTRODUCTION

Machine learning has rapidly advanced, driven by neural networks like convolutional and deep learning models, achieving success in fields such as image classification [1], natural language processing [2, 3], autonomous vehicles, and gaming [4, 5]. Companies like Amazon, Google, and Uber are adopting these technologies [6, 7]. However, deep learning models remain vulnerable to adversarial examples, which can subtly alter data and cause incorrect decisions, posing risks in critical applications like facial recognition and cybersecurity [8].

Adversarial examples (AEs) are manipulative techniques in deep learning designed to produce deceptive outputs, though their objectives vary. AEs are minimally altered inputs intended to deceive machine learning models into making erroneous predictions, exposing vulnerabilities in AI systems [9, 10]. These adversarial images subtly adjust input data to mislead models [11, 12]. Therefore, developing robust detection and prevention methods is essential to protect AI systems from misuse [13].

Wang et al. [14] developed a CNN-based model to distinguish between real and fake images using a dataset of 720,000 images from the LSUN database, which is a widely used benchmark dataset in computer vision for tasks like scene classification and object detection. Through data augmentation techniques like cropping, flipping, and rotation, they improved the model’s performance, achieving high detection efficiency across various test datasets. Other research, such as by Lu [15], and Gong

This manuscript was received on May 27, 2024. It was reviewed on June 18, 2024, revised on August 5, 2024 and accepted on September 24, 2024.

* Corresponding author

[16], focused on training binary classifiers to detect AEs.

Incorporating adversarial examples into the training process is a recognized strategy to enhance the robustness of neural networks. Goodfellow et al. [17] and Huang et al. [18] have advocated for generating adversarial samples during training and integrating them into the dataset, demonstrating that adversarial training can significantly improve deep neural network's resilience and act as a regularization technique that enhances model accuracy [17, 19]. However, their studies were limited to the MNIST dataset. For our research, we selected CIFAR-10 due to its greater complexity and visual diversity, offering more challenging scenarios compared to MNIST's grayscale images of handwritten digits [20].

Madry et al. (2017 in [21]) propose a robust adversarial training method, noted for its innovative approach and thorough empirical analysis, though it faces limitations such as high computational costs, limited generalizability, and scalability issues. Zheng et al. [22] introduce a method that uses transferable adversarial examples to improve training efficiency and model robustness, achieving 93.1% accuracy on CIFAR-10 under adversarial conditions. This method effectively reduces computational costs and improves generalization but may face scalability issues and reduced effectiveness across different model architectures and adversarial scenarios, with potential risks of overfitting to specific attack types.

Santurkar et al. (2022) [23] explore the effects of removing Batch Normalization (BN) in adversarial training with ResNet-50, achieving a 6% improvement in adversarial accuracy on CIFAR-10 and similar results on ImageNet. Despite this, the approach's generalizability and its impact on clean data performance remain concerns, and alternative normalization techniques are not discussed. He et al. [24] introduced the Deep Residual Network (ResNet), which enhances image recognition through residual learning, achieving 93.3% accuracy on CIFAR-10. However, ResNet's high computational demands and large model size can complicate deployment in resource-constrained environments.

Adversarial training and pre-training significantly enhance the robustness and performance of machine learning models [25]. Adversarial training improves resistance to attacks and generalization to unseen data, while pre-training accelerates convergence and boosts task-specific performance. Together, they create models that are both robust and computationally efficient [26].

In this study, we propose an adversarial training method using the Hybrid CIFAR-10 dataset, generated via a GAN-based approach targeting five models, as described by Trung et al. [27]. Our method, based on the ResNet architecture, achieves over 95% accuracy, outperforming other models. It addresses issues like high computational costs and limited generalizability by utilizing a dataset with varying noise levels. We also apply pre-training with the AdamW optimizer and CrossEntropy loss [28, 29], reducing the number of epochs and computational requirements, making the method practical for resource-constrained environments.

The paper is presented with the following structure. In the next section, we will overview of pertinent knowledge related to this study. The authors will present the specific proposed method in detail in section III. Section IV discusses the experiments and results, and section V concludes the paper.

II. BACKGROUND

A. Adversarial examples and adversarial attack

AEs involve introducing imperceptible perturbations to the original images, which remain unnoticed by human observers but are sufficient to deceive machine learning models, leading to incorrect classifications [30].

Consider a scenario where we have a model tasked with distinguishing between images of dogs and cats. Let the blue dots represent data points corresponding to dogs, while the red dots represent those corresponding to cats. During training, we adjust the decision boundary of the model to ensure that data points are correctly classified, as depicted in Figure 1. However, when crafting adversarial examples, we maintain the decision boundary fixed and manipulate the data points along the direction of steeper gradients. This adjustment swiftly shifts the data points from being confidently classified as a cat to hovering around an ambiguous decision boundary, ultimately relocating them to a position where the model confidently identifies them as dogs [17].

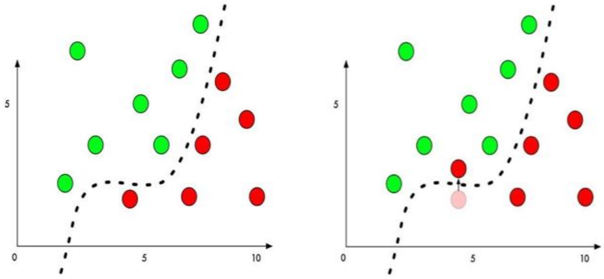

Figure 1. An adversarial attack: Green point is clean data, Red point is adversarial examples

The model depicted in Figure 2 illustrates the pseudo data generation model, comprising three primary components: the generator G, discriminator D, and target classifier C. Initially, a clean image x is fed into G to produce adversarial noise $G(x)$. Subsequently, $x + G(x)$ is forward to the discriminator C, responsible for discerning between generated samples and original clean samples. Functioning as an adversarial classification network, C determines the effectiveness of the generated samples in evading attacks.

Suppose a classification model C is trained on the dataset $X \subseteq R^n$, where n represents an image in the input set. Let (x_i, y_i) denote the $i - th$ instance in the training data, where $x_i \in X$ is trained using some unknown data and $y_i \in Y$ is the label. Despite achieving high accuracy on natural images, the objective of an attacker is to craft an adversarial example x_{adv} capable of deceiving C into producing an incorrect prediction, thereby causing C to misclassify x_i labeled as y_i to another label. The generator G leverages a clean image x and a target class label t as inputs to generate perturbations. The label t is selected based on the highest probability from the target model to classify the dataset, excluding the highest result representing the initial label.

Numerous techniques for generating adversarial samples have been documented in previous literature [17, 31, 32]. To suit our specific problem, we adopted adversarial samples generated using the methodology outlined by Trung et al. as described in [27]. This paper employs the adversarial images generation model depicted in Figure 2.

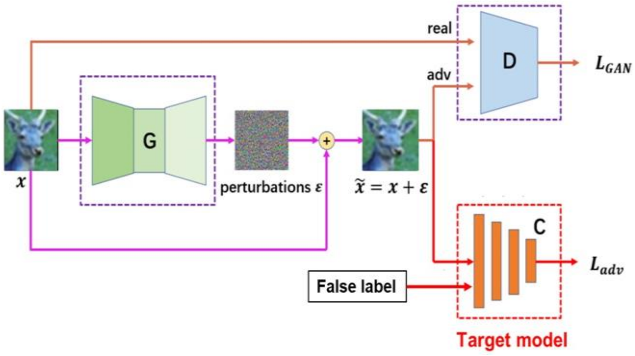

Figure 2. The GAN model [27] is used for generating adversarial images according to Trung et al

B. GAN model for creating adversarial images

The fundamental concept involves using a Generative Adversarial Network (GAN) generator to map clean samples to adversarial perturbations [33, 34], which are subsequently added to their corresponding clean samples. The discriminator's role is to differentiate between input samples and determine whether they are adversarial or not.

The author's approach differs from traditional resistance methods in several ways. Conventional techniques for generating adversarial samples typically use gradient descent algorithms to adjust pixels in a real image until it is misclassified by a machine learning model. Although this method can be effective, it has limitations. Specifically, it focuses on local pixel changes, which may not significantly alter the image's overall appearance. Consequently, this approach often fails to create robust adversarial examples.

GANs offer an alternative method capable of generating adversarial samples that deviate significantly from real images. However, GANs still maintain a degree of similarity between these adversarial samples and genuine images at the global image level. This similarity presents an advantage of GANs in crafting adversarial examples, as these nuanced differences can challenge machine learning models in distinguishing between real and generated images [10, 35].

This choice highlights the author's emphasis on diverse and robust architectures, each contributing uniquely to the field of image classification and object recognition. Trung et al. [27] outline the process of generating adversarial images in three steps:

Step 1: Using an image input, the model creates an adaptive perturbation.

Step 2: Combine the perturbation with the original image at a scale of k ratio to generate the adversarial image according to (1).

$$x_{adv} = x + (k \times perturbation), \qquad (1)$$

where x_{adv} is the adversarial image, x is the origin image, and $perturbation$ is the noise mask generated by the proposed GAN-based model with a Generator and Discriminator. This adversarial image is intended to fool the target model into misclassifying the image with a false label, as shown in Figure 2.

Step 3: Feed the newly created adversarial image into the target model. The attack is considered successful if the model produces an incorrect label. This underscores the importance of using various target models to generate a diverse range of adversarial images.

The incremental adjustment of the perturbation coefficient as in (1) allows the authors to evaluate the authenticity of the adversarial image relative to the original image step by step (details of the steps are in [27]).

We will provide an overview of the models we have selected to compare with our proposed architecture in training against adversarial attacks.

C. Target model in an adversarial attack

In an adversarial attack, the target model is the machine learning or deep learning system the attacker aims to deceive or disrupt. This model, often used for tasks such as image recognition, natural language processing, or autonomous navigation, is subjected to inputs crafted by adversarial techniques to induce incorrect outputs or behaviors.

The adversarial examples are designed to exploit the model's vulnerabilities, causing it to misclassify data, make erroneous predictions, or perform unintended actions. These attacks can significantly undermine the reliability and security of the target model, leading to potential risks in critical applications such as cybersecurity, healthcare, and autonomous systems. Ensuring the robustness and resilience of these models against adversarial attacks is a crucial challenge in the field of machine learning and artificial intelligence.

In our research, the aim of the target model is to produce a broad and varied set of adversarial examples using the CIFAR-10 dataset [36, 37], thereby aiding in the training of the detection model.

D. State-of-the-art deep learning models are used to target models and adversarial training

The authors selected three state-of-the-art image classification models: VGG19_bn [38], ShuffleNetV2 [39], RepVGG_a2 [40]. These popular CNN architectures each have unique features and trade-offs. This paper only provides an overview of these CNN models.

1. VGG19_bn [38]

Introduced in 2021 by Nicholas Carlini et al., the VGG19_bn model achieves 91.91% accuracy, showcasing its effectiveness in image classification. This variant of the VGG (Visual Geometry Group) architecture incorporates Batch Normalization layers, enhancing training performance and accelerating convergence. Renowned for its depth and use of consecutive 3×3 convolutional layers, VGG19_bn effectively captures features at various levels of the image, demonstrating good generalization across different datasets.

However, its deep structure and numerous layers require substantial computational resources and longer training times. Additionally, the model's large size, due to its many parameters, can complicate deployment on resource-constrained devices. While Batch Normalization improves training efficiency, the model may still face challenges in adapting to new variations of image data or tasks.

2. ShuffleNetV2 [39]

Achieving 93.81% accuracy, ShuffleNetV2 is optimized for computational efficiency and portability in computer vision tasks, particularly on resource-constrained devices like mobile phones. Utilizing depthwise convolution, channel shuffle, and group convolution, it balances performance and model compactness, making it fast and energy-efficient. While it offers improved efficiency over its predecessor ShuffleNetV1, with reduced computational complexity and easy implementation, it may have lower accuracy than larger networks like ResNet or EfficientNet. It may not scale well for more demanding applications. ShuffleNetV2 is ideal for lightweight and mobile-focused models but might not suit tasks requiring high accuracy or complex models.

3. RepVGG_a2 [40]

Introduced in 2021 by Xiaohan Ding and colleagues, RepVGG_a2 is a neural network architecture designed to optimize both performance and portability in computer vision tasks, achieving 94.98% accuracy. It utilizes structural re-parameterization to simplify its architecture into a 3 × 3 convolution matrix, enhancing efficiency during training and inference. The model's straightforward design relies on convolutional layers, avoiding complex operations like depthwise separable convolutions or attention mechanisms, making it easy to implement and scale.

While RepVGG_a2 delivers high performance and supports pre-trained models, it doesn't introduce novel features like those found in EfficientNet or Vision Transformers, which may lead to larger model sizes. Additionally, its simplicity may limit its effectiveness in specialized tasks requiring advanced features, and its reliance on re-parameterization could reduce interpretability.

The selected models each offer unique advantages for classifying objects on the CIFAR-10 dataset. RepVGG_a2, with its 3 × 3 convolutional design and structural re-parameterization, achieves 94.98% accuracy, balancing efficiency and performance for general tasks. ShuffleNetV2, optimized for mobile and resource-constrained environments, is compact and efficient but less precise for complex tasks. VGG19_bn delivers high accuracy on complex datasets with its deep architecture and batch normalization, though it is larger and slower.

III. THE PROPOSED MODEL ARCHITECTURE WITH ADVERSARIAL TRAINING TECHNIQUES

A. The optimization of Resnet architecture

The ResNet (Residual Network) architecture is a seminal advancement in the realm of deep convolutional neural networks, introduced by Kaiming He et al. in 2015. ResNet reached the top position at the ILSVRC 2015 competition with a top 5 error rate of only 3.57%. ResNet's innovation lies in its introduction of residual learning blocks, which enable the training of extremely deep neural networks effectively. Traditional deep networks face challenges in training as they suffer from vanishing gradients or degradation problems with increasing depth. ResNet addresses this issue by introducing skip connections, or shortcuts, which allow the network to skip layers during training. This mechanism facilitates the flow of gradients and prevents information loss, enabling the training of much deeper networks. Consequently, ResNet architectures have become widely adopted and serve as the backbone for numerous state-of-the-art computer vision tasks, including image classification, object detection, and semantic segmentation.

Furthermore, ResNet claimed the top position in the ILSVRC and COCO 2015 competitions across various tasks including ImageNet localization, ImageNet Detection, Coco segmentation, and Coco detection, thus affirming its remarkable performance and accuracy in image recognition and classification. Presently, the ResNet architecture has evolved into numerous variants with varying depths, such as ResNet-18 [41], ResNet-34, ResNet-50 [42], ResNet-101 [43], and more. Each variant is denoted by the prefix "ResNet" followed by an index indicating the specific number of layers in the architecture.

The authors of ResNet addressed the challenge of accuracy degradation in deep networks by introducing a concept called a “block”, which consists of a set of stacked layers. Assuming the input to the neural network is x and the desired output is $H(x)$, where $H(x)$ represents a complex mapping. When the model achieves high accuracy or it becomes evident that the error increases at each layer, the focus shifts to learning a similarity mapping. This involves ensuring that the input x closely resembles the output $H(x)$, to prevent accuracy degradation in subsequent stages. For a given block, the function to be approximated is $f(x)$. Instead of directly learning the underlying mapping $H(x)$, the authors proposed learning the residual mapping, denoted as $H(x) - x, f(x) =: H(x) - x$.

ResNet-50 known for its residual connections, delivers strong performance and accuracy, often surpassing other models in complexity, though it is less suited for mobile applications due to its resource demands. The primary objective of our research is to train a classifier using both clean and adversarial images from our proposed dataset and assess its generalization to other CNN-generated images. For this, we utilize ResNet-50 [24, 42], training it to withstand adversarial examples interspersed with clean images, thereby improving its robustness against such attacks.

The following section provides an overview of the model architecture, offering additional insights into the training method. The authors present detailed experiments and results of the models in Section IV.

B. Proposed model to adversarial retraining

The primary objective of our research is to create and assess a classifier capable of distinguishing between “real” and “adversarial” images using our proposed Hybrid CIFAR-10 dataset, which is designed to be resistant to adversarial attacks. Our analysis, based on the aforementioned theory, demonstrates that the ResNet architecture outperforms other models in terms of effectiveness and robustness (for comparison, we use ResNet-50 [24, 42]. The authors present the model architecture (specifically ResNet-50) and the training procedure below.

The architecture of ResNet-50 is characterized by its simplicity, featuring 50 convolutional layers dedicated to image processing. By leveraging the last convolution layer for classification, ResNet-50 achieves remarkable classification performance, making it a preferred choice in numerous image classification competitions. Moreover, its efficient training and inference capabilities are attributed to its innovative residual structure.

Given the classification nature of our dataset, ResNet-50 proves to be a suitable choice for our experiments. Figure 3 illustrates a standard ResNet-50 architecture.

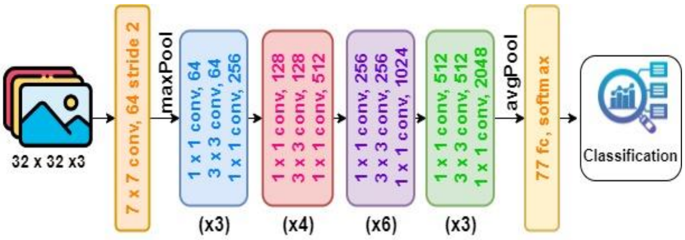

Figure 3. The architecture of ResNet-50 model

The content depicted in the image above is structured as follows:

Stage 1: Utilizes Convolution (Conv1) with 64 filters using a kernel size of (7, 7) and a stride of (2, 2). This is followed by max pooling with the same stride.

Stage 2: Comprises convolution layers with 64 filters (kernel size: (1,1)), 64 filters (kernel size: (3,3)), and 256 filters (kernel size: (1,1)), each repeated 3 times, resulting in a total of 9 layers in this stage.

Stage 3: Involves convolution layers with 128 filters (kernel size: (1,1)), 128 filters (kernel size: (3,3)), and 512 filters (kernel size: (1,1)), repeated 4 times, yielding 12 layers.

Stage 4: Features convolution layers with 256 filters (kernel size: (1,1)), 256 filters (kernel size: (3,3)), and 1024 filters (kernel size: (1,1)), repeated 6 times, resulting in 18 layers.

Stage 5: Consists of convolution layers with 512 filters (kernel size: (1,1)), 512 filters (kernel size: (3,3)), and 2048 filters (kernel size: (1,1)), repeated 3 times, resulting in 9 layers.

This cumulative configuration totals 50 layers in the deep convolution network. The author's experiments have demonstrated that ResNet effectively addresses the issue of accuracy degradation in deep neural networks, yielding excellent results in image tasks such as ImageNet and CIFAR-10. Additionally, ResNet converges faster compared to networks with a similar number of layers, owing to the absence of recurrent connections between neurons in layers, which allows for further design flexibility. Moreover, ResNet is resilient to the removal of specific layers, a characteristic significantly different from networks lacking traditional recurrent connections between neurons [44].

C. Optimizer and adjust Learning rate during Model training

The authors employ the AdamW optimizer, an improved version of the Adam optimizer that integrates weight decay directly into the update rule. This enhancement improves weight adjustments and reduces overfitting, boosting the model's generalization ability [28, 29]. In our setup, the optimizer is configured with a learning rate of 1×10^{-3}, a weight decay of 5×10^{-4}, and a batch size of 100.

Additionally, the authors apply the CosineAnnealingWarmRestarts technique [45] to periodically adjust the learning rate during training. This approach helps avoid local minima and enhances generalization by periodically restarting learning rate, promoting faster convergence and improving overall model performance. Adaptive learning rate mechanism is automatically fine-tuned throughout training, further optimizing performance.

In the ResNet-50 model, the Cross-Entropy Loss function is used, a popular choice for classification tasks. It measures the difference between predicted probabilities and actual labels, offering clear optimization. The function provides smooth gradients for efficient backpropagation, leading to faster and more stable training. It also extends well to multi-class classification through categorical cross-entropy and helps prevent vanishing gradients, especially with softmax outputs. It penalizes high-confidence incorrect predictions, reducing model overconfidence. These advantages make Cross-Entropy Loss a preferred choice for deep learning classification tasks. The Cross-Entropy Loss equation is defined as follows. For binary classification, Cross-Entropy Loss $\mathcal{L}$ is computed using (2).

$$\mathcal{L} = -(y\log(\hat{y}) + (1-y)\log(1-\hat{y})) \quad (2)$$

where y is the true label (0 or 1), $\hat{y}$ is the predicted probability of the instance being in class 1.

For multi-class classification, the Cross-Entropy Loss $\mathcal{L}$ is computed using (3).

$$\mathcal{L}(y,\hat{y}) = -\sum_{i=1}^{N} y_i \log(\hat{y_i}) \quad (3)$$

where y_i is the true label for class i (1 if the class is correct, 0 otherwise), $\hat{y_i}$ is the predicted probability for class i.

In both cases, the loss function penalizes incorrect predictions and drives the model to produce probabilities that better match the true labels. In this study, the loss function $\mathcal{L}$ is calculated as described in (3).

The following section will showcase experimental results to demonstrate the effectiveness of the proposed model and the techniques employed in this study.

IV. Experiments and Results

A. Experiments

We conduct experiments to assess the performance of the proposed model using dataset is generated by GAN method [27] based on CIFAR-10 dataset [36, 37], which comprises 158,498 color images sized at 32×32 pixels across 10 classes.

The implementation of the proposed model is carried out using Anaconda Python 3.11.3, PyTorch 1.8 framework, and CUDA 10.1 library on a computer equipped with an Intel(R) Core(TM) i9-9900 CPU @ 3.20 GHz, 64 GB

RAM, and NVIDIA GeForce RTX 2080 GPU with 8 GB VRAM.

Adversarial images are generated by applying attacks on five state-of-the-art deep learning models for image recognition using the method presented in section II-B. These generated adversarial images are then used for both training and testing purposes of the proposed model.

We train and evaluate the model through the evaluation measures below (measures are calculated accordingly [46]).

Accuracy: (Acc) is a measure assesses the performance of classification models. Accuracy is computed by dividing the number of correctly classified samples by the total number of samples, as depicted in (4).

$$Acc = \frac{TP+TN}{TP+TN+FP+FN} \tag{4}$$

Precision: a measure the proportion of predicted positive cases that are actually positive, offering insights into the rate of false positives. It is calculated using (5).

$$Precision = \frac{TP}{TP+FP} \tag{5}$$

Recall: is a measure of how many positive cases are correctly predicted, which enables analysis of false negative predictions, $Recall$ calculated as in (6).

$$Recall = \frac{TP}{TP+FN} \tag{6}$$

In equation (4), (5) and (6): (i) TP present the number of samples correctly classified as belonging to the positive class, (ii) TN denotes the number of samples correctly classified as belonging to the negative class, (iii) FP signifies the number of samples incorrectly classified as belonging to the positive class, and (iv) FN indicates the number of samples incorrectly classified as belonging to the negative class.

Loss value ($\mathcal{L}$) [47]: This measures the difference between the model's predicted results and the actual results. A lower loss value indicates a more accurate prediction. As presented above we use Cross-Entropy Loss according to (3).

We partition the data into three sets: training set, validation set, and testing set, following the distribution outlined in Table I. All images originate from CIFAR-10 or are generated by the proposed GAN model to deceive the target model. The dataset comprises 79,249 real images and 79,249 adversarial images, categorized into 10 classes. Within each category, we organize and label the data by placing real images and fake images into two separate folders, naming them "0" and "1" respectively, with "0" representing real images and "1" representing adversarial images. However, in this experiment we only choose the training dataset equivalent to the size of the CIFAR-10 dataset (about 60,000 images).

We divide the model training, validation, and testing dataset according to Table 1.

TABLE 1. THE DATASET FOR ADVERSARIAL TRAINING IS SELECTED FROM OUR HYBRID CIFAR-10 DATASET

No	CLASS NAME	TRAINING		VALIDATION		TESTING	
		Real images	*AEs*	*Real images*	*AES*	*Real images*	*AEs*
1	airplane	2,000	2,000	500	500	500	500
2	automobile	1,669	1,669	500	500	500	500
3	bird	2,028	2,028	500	500	500	500
4	cat	2,019	2,019	500	500	500	500
5	deer	2,200	2,200	500	500	500	500

No	CLASS NAME	TRAINING		VALIDATION		TESTING	
		Real images	***AEs***	***Real images***	***AES***	***Real images***	***AEs***
6	dog	2,031	2,031	500	500	500	500
7	frog	2,081	2,081	500	500	500	500
8	horse	2,257	2,257	500	500	500	500
9	ship	2,032	2,032	500	500	500	500
10	truck	2,439	2,439	500	500	500	500
TOTAL OF IMAGES		**20,756**	**20,756**	**5,000**	**5,000**	**5,000**	**5,000**

The authors illustrate the data division in Table 1 as shown in Figure 4. Figure 4 demonstrates that scaling the training data to an 8:1:1 ratio is commonly recommended.

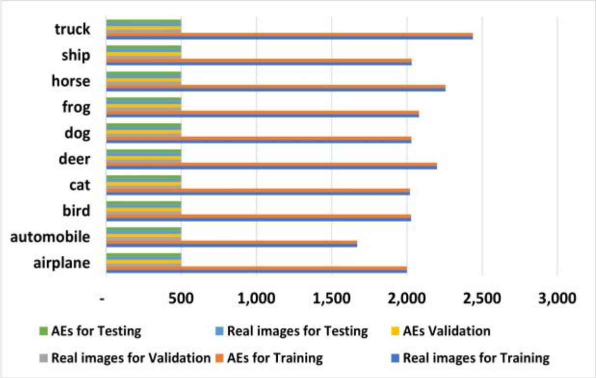

Figure 4. The chart shows the distribution of data

Conducting model training as described in the section III-C, using the dataset according to Table 1, the proposed model achieved: training accuracy achieved 99.94%, testing accuracy of 96.4%, $Precision$ achieved 95.3%, and $Recall$ achieved 95.4%, and loss value $\mathcal{L}$ achieved 0.004. Detailed results evaluating the model's accuracy and comparison with other models are provided in Section IV-B.

B. Results

The training results are presented in Table 2. Our proposed model, based on the ResNet architecture, achieves higher accuracy than the other three models after adversarial training, reaching 99.94% in training model. This highlights the effectiveness of the ResNet architecture in utilizing CNNs to extract features from digital images, thus enhancing its robustness against adversarial attacks.

In reality, attackers frequently incorporate a small proportion of adversarial images into their attack data. In the subsequent experiments, we randomly selected 1,000 images from our Hybrid CIFAR-10 dataset for testing the models, distributing them as follows: 97% real images and 3% adversarial images; 95% real images and 5% adversarial images; and 90%

TABLE 2. MODEL TRAINING RESULTS ACCORDING TO EVALUATION MEASURES: EPOCHS, LOSS VALUE ($\mathcal{L}$), ACCURACY (ACC (%)), PRECISION (%), RECALL (%)

No	**Models**	Epochs	$\mathcal{L}$	***Acc* in Training**	***Acc* of Models**	***Precision***	***Recall***
1	VGG19_bn [38]	**53**	0.030	99.41	93.8	95.3	94.0
2	ShufflenetV2 [39]	99	0.004	99.88	95.9	95.9	96.0
3	RepVGG_a2 [40, 48]	99	0.004	99.88	95.1	95.1	95.2
4	**ResNet-50**	99	**0.004**	**99.94**	95.3	95.3	95.4

real images and 10% adversarial images. The results are presented in Tables 3, 4, and 5.

TABLE 3. THE ACCURACY (%) OF THE MODELS ARE TESTED WITH 97% REAL IMAGES AND 3% ADVERSARIAL IMAGES (AES)

No	Models	Real images	AEs
1	VGG19_bn [38]	94.38	89.33
2	ShufflenetV2 [39]	95.90	92.67
3	RepVGG_a2 [40, 48]	95.09	94.33
4	**ResNet-50**	**95.28**	**94.00**

TABLE 4. THE ACCURACY (%) OF THE MODELS ARE TESTED WITH 95% REAL IMAGES AND 5% ADVERSARIAL IMAGES (AES)

No	Models	Real images	AEs
1	VGG19_bn [38]	93.81	89.33
2	ShufflenetV2 [39]	95.90	92.00
3	RepVGG_a2 [40, 48]	95.09	94.80
4	**ResNet-50**	**95.28**	**93.20**

TABLE 5. THE ACCURACY (%) OF THE MODELS ARE TESTED WITH 90% REAL IMAGES AND 10% ADVERSARIAL IMAGES (AES)

No	Models	Real images	AEs
1	VGG19_bn [38]	94.94	91.30
2	ShufflenetV2 [39]	95.83	91.70
3	RepVGG_a2 [40, 48]	95.17	94.40
4	**ResNet-50**	**96.40**	**92.80**

While the CNN adversarial training model demonstrates good generalization ability, there remains a possibility of false recognition in the images generated by the GAN model [27]. The proposed model's correct recognition rate for real images is higher than that of the other three models. Although the correct label recognition rate for adversarial images is lower than that of RepVGG_a2, the model's overall accuracy remains higher.

We compare our method with previous research and find that our results surpass those of Zheng et al. [22] by over 2% (relative to their reported 93.1%) and exceed those of He et al. [24] by 1.7% (relative to their reported 93.3%). It is important to note that He et al.'s study was conducted on the CIFAR-10 dataset without adversarial examples.

The authors also compare our method with previous studies that also utilize the ResNet architecture, including Santurkar et al., who achieved a 6% improvement [23], and Madry et al. [21], whose approach is challenging due to high computational costs. Our method addresses these issues effectively. In our comparisons, we find that our approach surpasses Zheng et al. [22] by 2% (relative to their reported 93.1%) and exceeds He et al. [24] by 1.7% (relative to their reported 93.3%). Notably, He et al.'s work was conducted on the CIFAR-10 dataset without adversarial examples.

V. CONCLUSION

In this paper, we identify the most suitable model for adversarial training through theoretical research and validate its optimization through experiments, demonstrating its superiority over three other models. This constitutes our notable contribution. We successfully implemented the training and testing process for the ResNet-50 model and conducted experiments with three additional models for comparison. This comparative analysis underscores the generalization capabilities of ResNet relative to the other architectures.

However, the study's limited detail on the remaining three models may be a constraint. This process has been a valuable learning experience, and we are dedicated to ongoing improvement in future projects. In future work, we plan to experimentally compare other models or hybrid models to enhance generalization ability and propose a model that is more optimal than the ResNet architecture. This approach can be scaled to larger datasets beyond CIFAR-10 and similar datasets.

Additionally, the method can be refined to address the challenge of adversarial images by applying it to other recognition systems, such as facial recognition or deepfake detection, which extend beyond adversarial images.

Acknowledgment

Truong Phi Ho was funded by the Master, PhD Scholarship Programme of Vingroup Innovation Foundation (VINIF), code VINIF.2024.TS.071.

References

[1] S.-C. Huang, A. Pareek, M. Jensen, M. P. Lungren, S. Yeung, and A. S. Chaudhari, "Self-supervised learning for medical image classification: a systematic review and implementation guidelines," *NPJ Digital Medicine*, vol. 6, no. 1, p. 74, 2023.

[2] X. Lu, S. Li, and M. Fujimoto, "Automatic speech recognition," *Speech-to-speech translation*, pp. 21-38, 2020.

[3] V. M. Tuan, N. X. Thang, and T. Q. Anh, "Evaluating the efficiency of Vietnamese sms spam detection techniques," *Journal of Science and Technology on Information security*, pp. 30–37, 2023.

[4] H. Wang, J. Polden, J. Jirgens, Z. Yu, and Z. Pan, "Automatic rebar counting using image processing and machine learning," in 2019 IEEE 9th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER). IEEE, 2019, pp. 900-904.

[5] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE access*, vol. 8, pp. 58 443-58 469, 2020.

[6] S. Mishra and A. R. Tripathi, "Ai business model: an integrative business approach," *Journal of Innovation and Entrepreneurship*, vol. 10, no. 1, p. 18, 2021.

[7] B. Marr, Artificial intelligence in practice: how 50 successful companies used AI and machine learning to solve problems. John Wiley & Sons, 2019.

[8] S. Hussain, P. Neekhara, M. Jere, F. Koushanfar, and J. McAuley, "Adversarial deepfakes: Evaluating vulnerability of deepfake detectors to adversarial examples," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2021, pp. 3348-3357.

[9] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 9, pp. 2805-2824, 2019.

[10] J. Zhang and C. Li, "Adversarial examples: Opportunities and challenges," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 7, pp. 2578-2593, 2019.

[11] E. Nowroozi, A. Dehghantanha, R. M. Parizi, and K.-K. R. Choo, "A survey of machine learning techniques in adversarial image forensics," *Computers & Security*, vol. 100, p. 102092, 2021.

[12] T. P. Ho, P. D. Trung, and B. T. Lam, "A novel generalized adversarial image method using descriptive features," *Journal of Science and Technology on Information security*, pp. 63-76, 2023.

[13] S. Zheng, Y. Song, T. Leung, and I. Goodfellow, "Improving the robustness of deep neural networks via stability training," in *Proceedings of the ieee conference on computer vision and pattern recognition*, 2016, pp. 4480-4488.

[14] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros, "Cnn-generated images are surprisingly easy to spot... for now," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8695-8704.

[15] J. Lu, T. Issaranon, and D. Forsyth, "Safetynet: Detecting and rejecting adversarial examples robustly," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 446-454.

[16] Z. Gong and W. Wang, "Adversarial and clean data are not twins," in Proceedings of the Sixth International Workshop on Exploiting Artificial Intelligence Techniques for Data Management, 2023, pp. 1-5.

[17] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[18] R. Huang, B. Xu, D. Schuurmans, and C. Szepesvári, "Learning with a strong adversary," *arXiv preprint arXiv:1511.03034*, 2015.

[19] Y. Wu, D. Bamman, and S. Russell, "Adversarial training for relation extraction," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 1778-1783.

[20] A. Baldominos, Y. Saez, and P. Isasi, "A

survey of handwritten character recognition with mnist and emnist," *Applied Sciences*, vol. 9, no. 15, p. 3169, 2019.

[21] A. Mkadry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *stat*, vol. 1050, no. 9, 2017.

[22] H. Zheng, Z. Zhang, J. Gu, H. Lee, and A. Prakash, "Efficient adversarial training with transferable adversarial examples," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1181-1190.

[23] H. Wang, A. Zhang, S. Zheng, X. Shi, M. Li, and Z. Wang, "Removing batch normalization boosts adversarial training," in *International Conference on Machine Learning*. PMLR, 2022, pp. 23 433–23 445.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.

[25] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *International conference on machine learning*. PMLR, 2019, pp. 7472-7482.

[26] T.-H. Wu, H.-T. Su, S.-T. Chen, and W. H. Hsu, "Revisiting semi-supervised adversarial robustness via noise-aware online robust distillation," *arXiv preprint arXiv:2409.12946*, 2024.

[27] D. T. Pham, C. T. Nguyen, P. H. Truong, and N. H. Nguyen, "Automated generation of adaptive perturbed images based on gan for motivated adversaries on deep learning models," in *Proceedings of the 12th International Symposium on Information and Communication Technology*, 2023, pp. 808-815.

[28] Z. Zhuang, M. Liu, A. Cutkosky, and F. Orabona, "Understanding adamw through proximal methods and scale-freeness," *Transactions on machine learning research*, 2022.

[29] M. Reyad, A. M. Sarhan, and M. Arafa, "A modified adam algorithm for deep neural network optimization," *Neural Computing and Applications*, vol. 35, no. 23, pp. 17 095-17 112, 2023.

[30] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[31] C. Xiao, B. Li, J.-Y. Zhu, W. He, M. Liu, and D. Song, "Generating adversarial examples with adversarial networks," *arXiv preprint arXiv:1801.02610*, 2018.

[32] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song, "Natural adversarial examples," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 15 262–15 271.

[33] M. M. Naseer, S. H. Khan, M. H. Khan, F. Shahbaz Khan, and F. Porikli, "Crossdomain transferability of adversarial perturbations," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[34] G. Jin, S. Shen, D. Zhang, F. Dai, and Y. Zhang, "Ape-gan: Adversarial perturbation elimination with gan," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3842-3846.

[35] T. Bai, J. Zhao, J. Zhu, S. Han, J. Chen, B. Li, and A. Kot, "Ai-gan: Attack-inspired generation of adversarial examples," in *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2021, pp. 2543-2547.

[36] A. Krizhevsky, G. Hinton et al., "Learning multiple layers of features from tiny images," 2009.

[37] Y. Abouelnaga, O. S. Ali, H. Rady, and M. Moustafa, "Cifar-10: Knn-based ensemble of classifiers," in 2016 *International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, 2016, pp. 1192–1195.

[38] M. Shaha and M. Pawar, "Transfer learning for image classification," in 2018 *Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 2018, pp. 656-660.

[39] Y. Martindez-Diaz, L. S. Luevano, H. Mendez-Vazquez, M. Nicolas-Diaz, L. Chang, and M. Gonzalez-Mendoza, "Shufflefacenet: A lightweight face architecture for efficient and highly-accurate face recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[40] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, "Repvgg: Making vgg-style convnets great again," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 13 733-13 742.

[41] S. Kakarwal and P. Paithane, "Automatic pancreas segmentation using resnet-18 deep learning approach," *System research and information technologies*, no. 2, pp. 104–116, 2022.

[42] B. Koonce and B. Koonce, "Resnet 34,"

Convolutional Neural Networks with Swift for Tensorflow: Image Recognition and Dataset Categorization, pp. 51–61, 2021.

[43] A. Demir, F. Yilmaz, and O. Kose, "Early detection of skin cancer using deep learning architectures: resnet-101 and inception-v3," in *2019 medical technologies congress (TIPTEKNO)*. IEEE, 2019, pp. 1-4.

[44] A. Veit, M. J. Wilber, and S. Belongie, "Residual networks behave like ensembles of relatively shallow networks," *Advances in neural information processing systems*, vol. 29, 2016.

[45] J. Bjorck, K. Q. Weinberger, and C. Gomes, "Understanding decoupled and early weight decay," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 8, 2021, pp. 6777–6785.

[46] S. Aronoff et al., "Classification accuracy: a user approach," *Photogrammetric Engineering and Remote Sensing*, vol. 48, no. 8, pp. 1299-1307, 1982.

[47] Q. Wang, Y. Ma, K. Zhao, and Y. Tian, "A comprehensive survey of loss functions in machine learning," *Annals of Data Science*, pp. 1–26, 2020.

[48] Z. Cai, X. Qiao, J. Zhang, Y. Feng, X. Hu, and N. Jiang, "Repvgg-simam: An efficient bad image classification method based on repvgg with simple parameter-free attention module," *Applied Sciences*, vol. 13, no. 21, p. 11925, 2023.

ABOUT THE AUTHORS

Truong Phi Ho

Workplace: Academy of Cryptography Techniques, Vietnam Government Information Security Commission.

Email: hotp.gvm@actvn.edu.vn

Education: Bachelor's degree in Information Technology from the Military Technology Academy in 2017, followed by a Master's degree in Information Technology from Nha Trang University in 2020. Currently, Ph.D student in Information Security at the Vietnam Academy of Cryptography Techniques, started in 2022.

Recent research direction: include Adversarial attacks and defenses, Generative AI, and Robustness of Deep learning models.

Tên tác giả: **Trương Phi Hồ**

Cơ quan công tác: Học viện Kỹ thuật mật mã, Ban Cơ yếu Chính phủ Việt Nam.

Email: hotp.gvm@actvn.edu.vn

Quá trình đào tạo: Kỹ sư Công nghệ Thông tin tại Học viện Kỹ thuật Quân sự năm 2017, Thạc sĩ Công nghệ thông tin tại Đại học Nha Trang năm 2020, Nghiên cứu sinh ngành An toàn Thông tin tại Học viện Kỹ thuật Mật Mã từ năm 2022 đến nay.

Hướng nghiên cứu hiện nay: Tấn công và Phòng thủ đối kháng, AI tạo sinh, tăng cường độ mạnh mẽ mô hình học sâu.

Pham Duy Trung

Workplace: Academy of Cryptography Techniques, Vietnam Government Information Security Commission.

Email: trungpd@actvn.edu.vn

Education: Bachelor's degree in Information Technology from Hanoi University of Science and Technology, Vietnam, in 2005. Master's degree in Information Technology and Systems from the University of Canberra, Australia, in 2012. Ph.D in Information Sciences and Engineering from the University of Canberra, Australia, in 2018.

Recent research direction: Privacy, Machine learning, and Safe machine learning.

Tên tác giả: **Phạm Duy Trung**

Cơ quan công tác: Học viện Kỹ thuật mật mã, Ban Cơ yếu Chính phủ Việt Nam.

Email: trungpd@actvn.edu.vn

Quá trình đào tạo: Kỹ sư Công nghệ Thông tin - Đại học Bách Khoa Hà Nội năm 2005, Thạc sĩ Công nghệ thông tin và Hệ thống tại Đại học Canberra - Australia năm 2012, Tiến sĩ Khoa học máy tính tại Đại học Canberra - Australia năm 2018.

Hướng nghiên cứu hiện nay: Đảm bảo riêng tư, học máy, an toàn trong học máy.

Dang Vu Hung

Workplace: Center for Information Technology and Cyber Security Monitoring, Vietnam Government Information Security Commission.

Email: hungdv@bcy.gov.vn

Education: Engineer's degree in Information Security from the Academy of Cryptography Techniques, Vietnam, in 2017. Master's degree in Information Security from the Academy of Cryptography Techniques, Vietnam, in 2019.

Recent research direction: Information Security, Network Security, Cryptography.

Tên tác giả: **Đặng Vũ Hùng**

Cơ quan công tác: Trung tâm Công nghệ thông tin và Giám sát an ninh mạng, Ban Cơ yếu Chính phủ Việt Nam

Email: hungdv@bcy.gov.vn

Quá trình đào tạo: Kỹ sư An toàn thông tin – Học viện Kỹ thuật mật mã năm 2017, Thạc sĩ An toàn thông tin – Học viện Kỹ thuật mật mã năm 2019.

Hướng nghiên cứu hiện nay: An toàn thông tin, an toàn mạng, mật mã.

Nguyen Nhat Hai

Workplace: School of Information and Communication Technology - Hanoi University of Science and Technology, Vietnam

Email: hai.nguyennhat@hust.edu.vn

Education: Bachelor's degree in Information Technology from Hanoi University of Science and Technology in Vietnam in 2005, Master's degree in Information and Applied Mathematics from Joseph Fourier University in France in 2007, and Ph.D in Information Technology, Automation, and Signal Processing from Grenoble Polytechnic University in France in 2011.

Recent research direction: include Natural language processing, Computer vision, Distributed systems (Blockchain), Wireless Sensor Networks, and Internet of Things.

Tên tác giả: **Nguyễn Nhất Hải**

Cơ quan công tác: Trường Công nghệ Thông tin và Truyền thông - Đại học Bách khoa Hà Nội, Việt Nam

Email: hai.nguyennhat@hust.edu.vn

Quá trình đào tạo: Kỹ sư Công nghệ thông tin tại trường Đại học Bách khoa Hà Nội năm 2005, Thạc sĩ Tin học – Toán ứng dụng tại Đại học Joseph Fourier - Pháp năm 2007, Tiến sĩ Công nghệ thông tin, tự động và xử lý tín hiệu tại trường Đại học Bách khoa Grenoble - Pháp năm 2011.

Hướng nghiên cứu hiện nay: Xử lý ngôn ngữ tự nhiên, thị giác máy tính, Blockchain, mạng cảm biến không dây và IoT.