# Analysis of Defense Mechanisms Against FGSM Adversarial Attacks on ResNet Deep Learning Models Using the CIFAR-10 Dataset

Krishna Aurelio Noviandri
School of Electrical Engineering and Informatics
Bandung Institute of Technology
Bandung, Indonesia
23524057@mahasiswa.itb.ac.id

Miranti Jatnika Riski
School of Electrical Engineering and Informatics
Bandung Institute of Technology
Bandung, Indonesia
23524048@mahasiswa.itb.ac.id

Yoga Hanggara
School of Electrical Engineering and Informatics
Bandung Institute of Technology
Bandung, Indonesia
23524044@mahasiswa.itb.ac.id

Nugraha Priya Utama
School of Electrical Engineering and Informatics
Center of Excellence for AI, Computer Vision, NLP, and Big Data Analytics
Bandung Institute of Technology
Bandung, Indonesia
utama@itb.ac.id

Ayu Purwarianti
School of Electrical Engineering and Informatics
Center of Excellence for AI, Computer Vision, NLP, and Big Data Analytics
Bandung Institute of Technology
Bandung, Indonesia
ayu@itb.ac.id

Abstract—Adversarial attacks threaten the reliability of deep learning models in image classification, requiring effective defense mechanisms. This study evaluates how defense distillation and adversarial training protect ResNet18 models trained on CIFAR-10 data against Fast Gradient Sign Method (FGSM) attacks. The baseline model achieves 85.01% accuracy on clean data but its accuracy falls to 19.23% when FGSM attacks at epsilon 0.3. The accuracy of defense distillation drops to 23.68% when epsilon reaches 0.3 but adversarial training maintains 0.34% accuracy at epsilon 0.25 although it reduces clean data accuracy to 57.08%. The analysis shows that classes with similar visual characteristics such as cats and dogs remain vulnerable to attacks. The study demonstrates the requirement for balanced defense approaches while indicating additional work needs to improve model robustness.

Keywords—deep learning, ResNet, adversarial attack, FGSM, CIFAR-10.

## I. INTRODUCTION

Computer Vision operates as a fundamental field of informatics which depends primarily on image classification. The application range includes medical image disease detection and autonomous vehicle traffic sign recognition and e-commerce product labeling and facial recognition systems. The deep learning method of Convolutional Neural Networks (CNNs) serves as a primary technology for image classification because it delivers high accuracy results. The standard architectural design for Computer Vision tasks involves CNNs which extract hierarchical features from images using convolutional filters to recognize complex visual patterns [1].

Deep learning models which specialize in image classification face risks of attacks. Residual Network (ResNet) represents a CNN architecture advancement through its use of residual blocks to combat the deep network vanishing gradient issue [2]. Unlike human perception, ResNet relies on pixel patterns for recognition. The modification of tiny pixel values within an image leads to substantial changes in classification results. These models face vulnerability to adversarial attacks which modify image inputs to make the classification system produce wrong results [3]. Social and economic impacts emerge from these classification errors. The incorrect predictions in finance result in major financial losses that affect banking system fraud detection operations. Autonomous vehicle traffic sign recognition system failures pose dangers to passengers and public safety during operation. Medical error diagnoses result in receiving inappropriate treatments. Failures of AI technology reduce public trust in artificial intelligence systems which obstructs its future development.

Several studies have investigated adversarial attack methods [3], [4], [5]. The Fast Gradient Sign Method (FGSM) functions as a widely recognized attack benchmark to measure model robustness through its simple yet efficient approach [6]. The FGSM enables researchers to test different attack intensities by modifying the epsilon parameter. The attack method shows success in discovering weaknesses of deep learning models including ResNet models when applied to CIFAR-10 dataset [3]. Research on defense mechanisms benefits from FGSM because its simple allows researchers to evaluate defenses without the complexities of advanced attack methods such as Projected Gradient Descent (PGD) and Carlini-Wagner (CW).

There are multiple studies about defense mechanisms to improve neural network robustness to attacks [7], [8], [9]. The training of models with adversarial examples on the dataset creates a robust model that resists attacks while maintaining generalization capabilities. The implementation of this method requires substantial resources because it needs the creation and training of adversarial datasets [10]. Several preprocessing techniques fight against adversarial effects although they diminish model accuracy so researchers need to strike a balance between robustness and accuracy loss [11]. Researchers attempt to strengthen neural networks by applying defensive distillation as one of their methods to enhance resilience [12]. The implemented defensive measures do not provide complete protection against certain adversarial attacks [13].

Despite existing research on adversarial attacks and defenses, a gap remains in understanding the effectiveness of specific defense strategies on particular architectures and datasets. For instance, adversarial training on CIFAR-10 with ResNet18 or its comparison to defense distillation. Few studies have systematically compared the effectiveness of defense distillation and adversarial training on ResNet18 using CIFAR-10. This research aims to address this gap by providing a comprehensive evaluation and direct comparison of both defense methods within a consistent framework.

The study presents a detailed comparison between defense distillation and adversarial training as defensive techniques against FGSM attacks when implemented on ResNet18 models trained on CIFAR-10 data. The study provides an in-depth assessment through cluster analysis to detect misclassification patterns which represents an innovative approach in adversarial defense studies. This study provides new insights into the trade-offs between robustness and accuracy through performance testing and error pattern evaluation which leads to suggestions for developing future defense strategies.

## II. METHODOLOGY

### A. Research Methodology

The study evaluated multiple defense mechanisms including Fast Gradient Sign Method (FGSM) to evaluate their impact on ResNet deep learning models which trained on the CIFAR-10 dataset. The methodology process is illustrated in Figure 1.

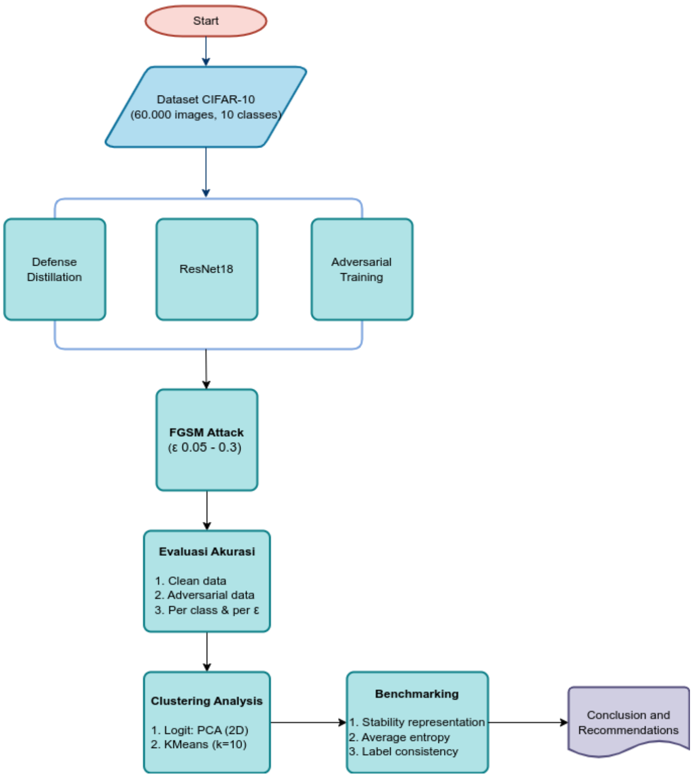


Figure 1. Research Methodology

## B. Algorithms Used

The study developing and evaluating three models which including a baseline model alongside defense distillation and adversarial training models.

### 1) *Baseline Model*

The baseline model use ResNet18 as its Residual Network architecture variant [2]. ResNet18 was chosen because ideal balance between complexity and performance for experiments that work with CIFAR-10 and other similar datasets.

The modification of the initial convolutional layer for CIFAR-10 images included using a 3x3 kernel with stride 1 and padding 1 followed by batch normalization to accommodate the small input dimensions (32x32 pixels). This modification ensures better detection of small but significant patterns.

The model consists of four layers that include two residual blocks in each of them. The initial convolution operation in each layer performs stride 2 to decrease feature dimensions. This technique both preserves vital information and reduces computational complexity effectively. The last layer includes average pooling which connects to a fully connected layer containing 10 outputs to compute CIFAR-10 class probability predictions.

### 2) *Defense Distillation*

Defense distillation uses knowledge distillation principles to improve model resistance against adversarial attacks. The training process starts with a baseline model that uses a modified softmax function with T=20 as its temperature parameter. The high temperature output produces smooth probabilities which extract detailed information about model confidence distribution across all classes. Soft labels provide deeper understanding of class relationships because they differ from traditional hard labels. The distilled model which has an identical architecture to the baseline model receives training through the soft labels. The distilled model learns a more general feature space representation through this approach which enhances its resistance to adversarial perturbations [12]. Defense distillation implementation steps:

1. A modified softmax activation function with temperature parameter T=20 should be implemented in the baseline ResNet18 model to produce soft labels that benefit the distillation process.
2. Soft labels should be generated from the baseline model outputs for the CIFAR-10 dataset.
3. Construct a distilled ResNet18 model identical to the baseline model but trained using the generated soft labels.

*3) Adversarial Training and Fast Gradient Sign Method (FGSM)*

The study implements adversarial training as a defense mechanism against adversarial attacks through the Fast Gradient Sign Method (FGSM) to produce adversarial samples. This study chose FGSM because it provides both simplicity and effectiveness which makes it a standard method for assessing deep learning model robustness against adversarial attacks [3], [6]. The method uses the model's loss function gradient to generate tiny perturbations which interfere with classification.

The FGSM perturbation is defined as follows:

$$x' = x + \varepsilon \cdot sign\big(\nabla_x J(\theta, x, y)\big)$$

Where:
$x$: Original input image
$\varepsilon$: Perturbation scale parameter
$\nabla_x J(\theta, x, y)$: Gradient of the loss with respect to input $x$
$\theta$: Model parameter
$y$: Original input label

The following steps describe how to produce adversarial samples from each minibatch:

1. Feed forward processing should be performed on the current minibatch.
2. Calculate the loss from model output and true labels.
3. Compute the gradient of the loss with respect to the input using backpropagation.
4. Generate adversarial noise by applying the sign function on gradients scaled by epsilon (e.g., 0.1, 0.2).
5. Add adversarial noise to original inputs to form adversarial samples.
6. Include adversarial samples in training minibatches.

The following examples from Figure 2 show how FGSM affects CIFAR-10 images at different epsilon values. The examples show how small pixel modifications result in incorrect classification results. The baseline ResNet18 model misclassifies images after FGSM attacks even though the visual changes remain minimal as shown in Figure 2.

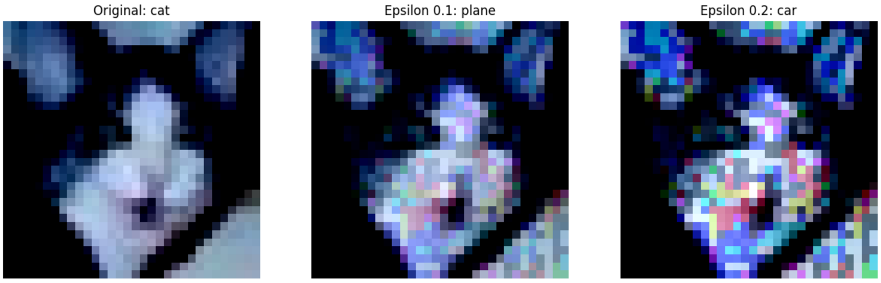


Figure 2. Examples of CIFAR-10 images before and after FGSM attacks with varying epsilon values.

Left: Original image (class: "cat").
Middle: Adversarial image with epsilon = 0.1 (predicted as "plane").
Right: Adversarial image with epsilon = 0.2 (predicted as "car").

The model receives training through combined original and adversarial data which leads to a major improvement in its resistance against FGSM attacks.

## C. Data Collection

The study uses CIFAR-10 as its dataset because it serves as a standard collection for image classification tasks. The CIFAR-10 dataset contains 60,000 color images (32x32 pixels) which are distributed equally across 10 classes (airplane, car, bird, cat, deer, dog, etc.) with 6,000 images in each class [14]. The dataset's small size and balanced class distribution facilitate effective processing and analysis. The researchers chose CIFAR-10 because it serves both image classification research and adversarial defense method testing purposes. The dataset serves as a standard tool for testing various methods including Example-Based Explanations (EBE) for Deep Neural Networks [15] and adversarial attacks such as FGSM and adversarial training [16]. The use of this standard dataset enables research reproducibility.

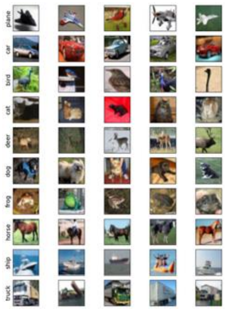


Figure 3. Example images for each class in the CIFAR-10 dataset.

### *1) Dataset Exploration*

The first step of dataset exploration checks data quality through class distribution and color intensity analysis. The visual inspection confirms that classes are distributed uniformly (Figure 3) which guarantees unbiased class representation.

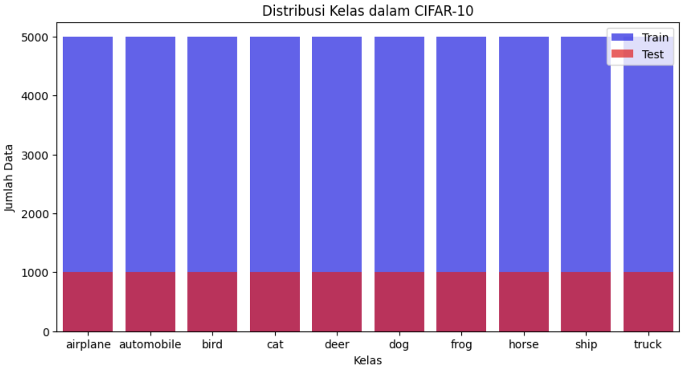


Figure 4. Class distribution in the CIFAR-10 dataset.

A color histogram (Figure 4) helps identify potential color biases influencing model performance, since certain backgrounds might become dominant visual features rather than object shape or pattern.

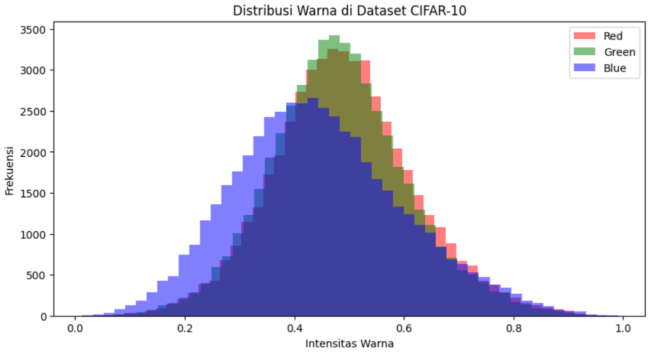


Figure 5. Color distribution histogram for the CIFAR-10 dataset.

### 2) *Data Preprocessing*

Data preprocessing improvement steps enhance model performance using data augmentation and transformations. The preprocessing parameters are detailed in Table 1.

#### a) *Data Augmentation*

The process of data augmentation creates synthetic variations in training data which helps prevent overfitting and enhances both robustness and generalization capabilities. The following techniques are used:

1. Random cropping (32x32 pixels) with padding (4 pixels), using PyTorch's transforms.RandomCrop.
2. Random horizontal flipping (probability 0.5) for varied object orientations, using transforms.RandomHorizontalFlip.
3. Random adjustments of brightness, contrast, and saturation (intensity 0–0.2), utilizing transforms.ColorJitter.

Table 1. Data preprocessing parameters.

| **Process** | **Parameters** | **Description** |
|---|---|---|
| Random Crop | Size: 32<br>Padding: 4 | Randomly crops a portion of the image, adding padding to maintain a 32×32 size. |
| Horizontal Flip | Probability: 0.5 | Horizontally flips the image randomly. |
| Color Jitter | Brightness: 0.2<br>Contrast: 0,.2<br>Saturation: 0.2 | Randomly varies the image's brightness, contrast, and saturation. |
| To Tensor | | Converts the image to a tensor with values scaled from 0 to 1. |
| Normalization | Mean: (0.4914, 0.4822, 0.4465)<br>Std: (0.2470, 0.2435, 0.2616) | Normalizes each channel of the image. |

#### b) *Data Transformation*

The process of data transformation and normalization transforms images into formats that GPUs can process efficiently:

1. Convert images to tensor format scaled between 0 and 1 using transforms.ToTensor.
2. Normalize each RGB channel using mean values (0.4914, 0.4822, 0.4465) and standard deviations (0.2470, 0.2435, 0.2616), which produces uniform data distribution and faster model convergence using transforms.Normalize.

## D. Experiments

The experimental hyperparameters, such as learning rate, optimizer, and number of epochs, were standardized across models to minimize external influences. Most hyperparameters such as optimizer, learning rate, and batch size were kept consistent across models to ensure comparability. Model-specific configurations (e.g., temperature or epsilon). To make sure the evaluation was fair, all the models were trained using the same hyperparameters. These include a batch size of 32, learning rate of 0.001, AdamW optimizer, and 50 training epochs. Specific configurations related to each defense strategy and detailed configuration is presented in Table 2.

Table 2. Hyperparameter configuration for each model.

| Parameter | Baseline | Defense Distillation | Adversarial Training |
|---|---|---|---|
| Batch Size | 32 | 32 | 32 |
| Learning Rate | 0.001 | 0.001 | 0.001 |
| Optimizer | AdamW<br>Weight decay: 0.01<br>Betas: (0.9, 0.999)<br>Eps: 1e-8 | AdamW<br>Weight decay: 0.01<br>Betas: (0.9, 0.999)<br>Eps: 1e-8 | AdamW<br>Weight decay: 0.01<br>Betas: (0.9, 0.999)<br>Eps: 1e-8 |
| Epoch | 30 | 30 | 30 |
| Additional Params | - | Temperature: 20 | Epsilon: 0.1, 0.2 |

Experiments were conducted locally on the researchers' laptops, with hardware specifications outlined in Table 3.

Table 3. Hardware specifications.

| Specification | Details |
|---|---|
| CPU | AMD Ryzen 7 5800 H with Radeon Graphics<br>8 Core<br>~3.2 Ghz |
| RAM | 32 GB DDR4<br>3200 MHz |
| GPU | NVIDIA GeForce RTX 3070 Laptop GPU<br>8 GB |
| Storage | 1TB NVME SSD |
| Operating System | Windows 11 Home 64-bit |
| CUDA | CUDA V12.5.82 |

Table 3. lists the hardware specification utilized in the experiment, such as the type of GPU, CPU, memory, and software environment.

## E. Clustering Analysis

The evaluation and experimentation phases led to clustering analysis which studied the internal representation structure of models after inference. The unsupervised clustering method was selected to assess model logit representation quality through evaluation methods that extend beyond standard confusion matrix metrics.

The clustering features consisted of pre-softmax logit outputs which were extracted from each test sample. The selection of logit outputs as features occurred because these outputs maintain class-specific information before softmax normalization takes effect.

The methods employed include:

1. Principal Component Analysis (PCA) was used to reduce the dimensionality of the logit outputs, enabling efficient processing and visualization. This method was chosen for its effectiveness and common use in exploring internal model representations.
2. K-Means clustering was applied with k=10 clusters, corresponding to the number of classes in the CIFAR-10 dataset. This method was selected for its ability to capture underlying data distribution patterns that may not be evident in a confusion matrix.

The combination of PCA with K-Means clustering according to [18] produces three robust clusters from high-dimensional data complexity. The study [19] demonstrates that

K-Means clustering produces optimal clusters which deliver valuable information for stakeholders.

### F. Benchmarking

After clustering analysis, the results were used as the baseline for benchmarking. The benchmarking process used cluster-based analysis together with entropy per cluster measurements. The calculation of entropy used class label frequency distributions from each cluster to determine cluster purity.

The evaluation of model robustness against adversarial attacks depends heavily on benchmarking because it assesses both accurate final outputs and stable internal representation (logits) when exposed to adversarial noise.

Cluster purity evaluation based on entropy was proposed in a study [20], which applied Shannon entropy to assess the quality of clusters derived from dimensionality reduction and unsupervised learning.

## III. RESULT AND DISCUSSION

### A. Implementation Steps

*1) Code Implementation*

The development of the baseline model and adversarial defense mechanisms used Python version 3.11 with PyTorch for model development and training and Matplotlib (pyplot) for visualization. The additional libraries used were NumPy and tqdm. The CIFAR-10 dataset was downloaded directly using built-in PyTorch functions. The PyTorch autograd module was used to test the adversarial attacks. The code was developed on Google Colab for ease of execution and collaboration.

*2) Testing*

The model performance evaluation was conducted using CIFAR-10's test set. Accuracy metric, calculated overall and per class, was the primary evaluation metric. This approach assesses model stability against adversarial attacks, evaluating robustness across varying intensities of FGSM-generated adversarial samples (epsilon values: 0.05, 0.1, 0.15, 0.2, 0.25, and 0.3).

### B. Results

*1) Clean Data Evaluation*

Table 4 presents the overall accuracy results for each model when tested on clean data. The baseline model together with defense distillation models reached high accuracy levels at 85.01% and 81.70% respectively. The adversarial training model delivered poor results with an accuracy rate of 57.08%.

Table 4. Overall accuracy on clean data.

| Model | Accuracy |
|---|---|
| Base Model | 85.01% |
| Defense Distillation | 81.70% |
| Adversarial Training | 57.08% |

The accuracy results for each class appear in Table 5.

Table 5. Per-class accuracy on clean data.

| | Accuracy (by class) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Model** | **plane** | **car** | **bird** | **cat** | **deer** | **dog** | **frog** | **horse** | **ship** | **truck** |
| Base Model | 86.1% | 92.5% | 77.1% | 68.3% | 85.2% | 87.5% | 87.1% | 87.8% | 87.8% | 90.7% |
| Defense Distillation | 79.5% | 91.6% | 71.8% | 74.0% | 84.8% | 71.0% | 82.0% | 83.1% | 89.5% | 89.7% |
| Adversarial Training | 64.2% | 72.4% | 42.9% | 39.2% | 48.7% | 46.4% | 62.1% | 60.9% | 69.0% | 65.0% |

The baseline model shows consistent performance across all classes but achieves the lowest accuracy in the "cat" class. The defense distillation model achieves its lowest accuracy in the "dog" class. The defense distillation model demonstrates the highest minimum per-class accuracy when compared to other models.

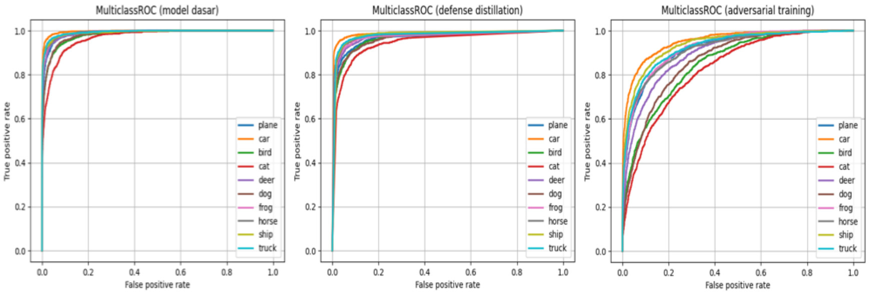
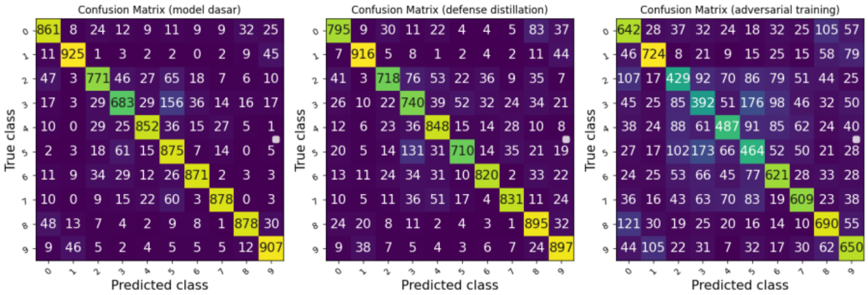


Figure 6. ROC curves for baseline, defense distillation, and adversarial training models.


Confusion Matrix (model dasar)

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 861 | 8 | 24 | 12 | 9 | 11 | 9 | 9 | 32 | 25 |
| 1 | 11 | 925 | 1 | 3 | 2 | 2 | 0 | 2 | 9 | 45 |
| 2 | 47 | 3 | 771 | 46 | 27 | 65 | 18 | 7 | 6 | 10 |
| 3 | 17 | 3 | 29 | 683 | 29 | 156 | 36 | 14 | 16 | 17 |
| 4 | 10 | 0 | 29 | 25 | 852 | 36 | 15 | 27 | 5 | 1 |
| 5 | 2 | 3 | 18 | 61 | 15 | 875 | 7 | 14 | 0 | 5 |
| 6 | 11 | 9 | 34 | 29 | 12 | 26 | 871 | 2 | 3 | 3 |
| 7 | 10 | 0 | 9 | 15 | 22 | 60 | 3 | 878 | 0 | 3 |
| 8 | 48 | 13 | 7 | 4 | 2 | 9 | 8 | 1 | 878 | 30 |
| 9 | 9 | 46 | 5 | 2 | 4 | 5 | 5 | 5 | 12 | 907 |

Confusion Matrix (defense distillation)

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 795 | 9 | 30 | 11 | 22 | 4 | 4 | 5 | 83 | 37 |
| 1 | 7 | 916 | 5 | 8 | 1 | 2 | 4 | 2 | 11 | 44 |
| 2 | 41 | 3 | 718 | 76 | 53 | 22 | 36 | 9 | 35 | 7 |
| 3 | 26 | 10 | 22 | 740 | 39 | 52 | 32 | 24 | 34 | 21 |
| 4 | 12 | 6 | 23 | 36 | 848 | 15 | 14 | 28 | 10 | 8 |
| 5 | 20 | 5 | 14 | 131 | 31 | 710 | 14 | 35 | 21 | 19 |
| 6 | 13 | 11 | 24 | 34 | 31 | 10 | 820 | 2 | 33 | 22 |
| 7 | 10 | 5 | 11 | 36 | 51 | 17 | 4 | 831 | 11 | 24 |
| 8 | 24 | 20 | 8 | 11 | 2 | 4 | 3 | 1 | 895 | 32 |
| 9 | 9 | 38 | 7 | 5 | 4 | 3 | 6 | 7 | 24 | 897 |

Confusion Matrix (adversarial training)

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 642 | 28 | 37 | 32 | 24 | 18 | 32 | 25 | 105 | 57 |
| 1 | 46 | 724 | 8 | 21 | 9 | 15 | 25 | 15 | 58 | 79 |
| 2 | 107 | 17 | 429 | 92 | 70 | 86 | 79 | 51 | 44 | 25 |
| 3 | 45 | 25 | 85 | 392 | 51 | 176 | 98 | 46 | 32 | 50 |
| 4 | 38 | 24 | 88 | 61 | 487 | 91 | 85 | 62 | 24 | 40 |
| 5 | 27 | 17 | 102 | 173 | 66 | 464 | 52 | 50 | 21 | 28 |
| 6 | 24 | 25 | 53 | 66 | 45 | 77 | 621 | 28 | 33 | 28 |
| 7 | 36 | 16 | 43 | 63 | 70 | 83 | 19 | 609 | 23 | 38 |
| 8 | 121 | 30 | 19 | 25 | 20 | 16 | 14 | 10 | 690 | 55 |
| 9 | 44 | 105 | 22 | 31 | 7 | 32 | 17 | 30 | 62 | 650 |

Figure 7. Confusion matrices for baseline, defense distillation, and adversarial training models.

The confusion matrices in Figure 7 show that the "cat" and "dog" classes are often misclassified. This is consistent with the observed increase in accuracy for the "cat" class and decrease for the "dog" class in the defense distillation model compared to other classes.

Figure 6 shows the ROC curves for each model. The baseline model generally produces sharper curves toward the left, indicating better performance. In the defense distillation model, the ROC curve for the "cat" class is closest to the diagonal, suggesting that this class remains challenging to classify accurately, despite its improved accuracy compared to the "dog" class.

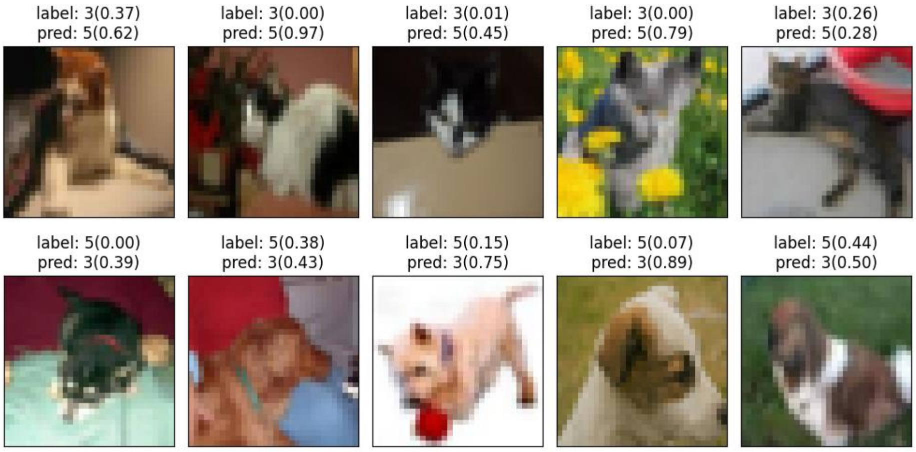


Figure 8. Examples of misclassified cat and dog images.

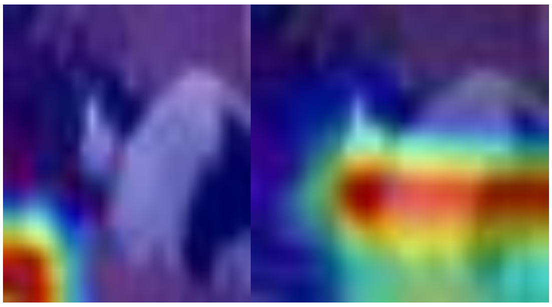


Figure 9. Grad-CAM visualizations showing focused regions in 'cat' and 'dog' classifications, highlighting attention shifts under adversarial attacks.

The analysis includes five example pictures of cats and dogs which were incorrectly identified in Figure 8. The visual similarity between cat and dog body shapes makes side-view images of these animals more likely to result in incorrect predictions. The visual similarity between body shapes of cats and dogs makes it difficult for the model to distinguish between them. The main characteristics that distinguish these animals exist in their facial features. The image shown in Figure 9 contains two cats which show either only the face or the complete body from a side perspective. Grad-CAM visualizations are used to demonstrate its focus on facial features for "cat" classification while focusing on body features for "dog" classification as shown in Figure 9. The body features dominate most images which causes the model to incorrectly classify them as "dog" because of its focus.

The baseline model delivers the highest performance results according to clean data testing. The adversarial training model demonstrates poor accuracy because training with adversarial data produces negative effects on clean data performance. The defense distillation model demonstrates the most generalization capability by achieving balanced performance across all classes especially between the often confused "cat" and "dog" classes. The models demonstrate poor ability to distinguish between "cat" and "dog" classes especially when images show animals from the side because their faces become less visible. The defense distillation model achieves better generalization which leads to more balanced performance between difficult-to-distinguish classes.

### 2) *Adversarial Attack Evaluation*

Table 6 shows the accuracy performance of each model when subjected to FGSM adversarial attacks at different epsilon values.

Table 6. Accuracy under FGSM attack (by epsilon)

| | Accuracy under FGSM attack (by epsilon) | | | | | | |
|---|---|---|---|---|---|---|---|
| **Model** | **.0** | **.05** | **.1** | **.15** | **.2** | **.25** | **.3** |
| Base Model | 85.01% | 43.39% | 35.97% | 30.31% | 25.85% | 22.47% | 19.23% |
| Defense Distillation | 81.7% | 48.97% | 40.54% | 34.04% | 29.18% | 25.99% | 23.68% |
| Adversarial Training | 57.08% | 33.43% | 48.91% | 57.03% | 60.27% | 60.34% | 58.56% |

The baseline model along with the distillation model experienced major accuracy reductions as epsilon values increased (baseline: from 85.01% to 19.23%; distillation: from 81.70% to 23.68%). The adversarial training model started with low clean accuracy at 57.08% but showed better resistance to attacks at higher epsilon values until it reached 60.34% accuracy at epsilon = 0.25 before dropping slightly at epsilon = 0.3. The trends in Figure 10 show each model's resistance to adversarial attacks.

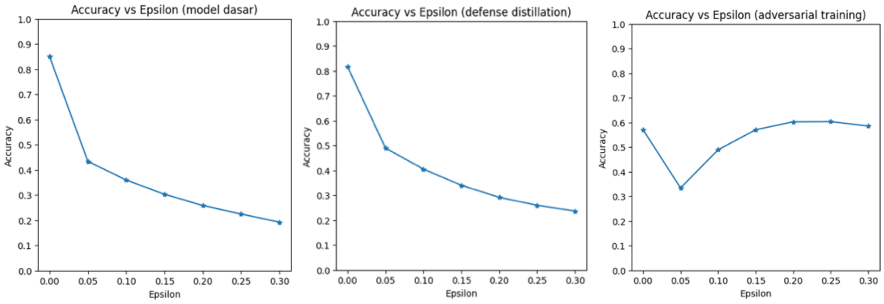


Figure 10. Accuracy against epsilon values of FGSM attack for each model.

## C. Performance Evaluation

The baseline model maintains high performance on clean data yet its accuracy plummets when subjected to adversarial testing. The substantial accuracy decrease proves that the model remains highly exposed to adversarial attacks thus requiring extra defensive measures. The model encounters difficulties when trying to identify between "cat" and "dog" classes that share similar characteristics.

The defense distillation model maintains equivalent accuracy to the baseline model on clean data while showing a minimal decrease in performance. The generalization approach implemented by this method maintains clean data performance at its original level. The method achieves balanced performance between classes that share similar features including "cat" and "dog." The model shows a notable accuracy reduction during adversarial testing but to a lesser extent than the baseline model. Defense distillation provides restricted protection against FGSM attacks according to Papernot et al. [12] who reported that such attacks can be partially effective.

The adversarial training model shows better resistance to adversarial attacks at elevated epsilon levels yet it leads to major accuracy losses on clean data. The adversarial testing results at epsilon 0.05 show that adversarial training needs to be trained with various attack intensities and hyperparameters. The obtained results match the findings of Bai et al. [10] who observed that adversarial training might not perform well against unknown attacks.

Model performance can be further improved. The model can achieve better "cat" versus "dog" classification through additional training on misclassified samples or implementing weighted loss for the "cat" class. The adversarial training model requires additional training with multiple epsilon values to enhance its performance on adversarial test data. The investigation should explore two variations of adversarial training which involve training the baseline model with adversarial data and applying weighted loss to the model.

## D. Clustering Analysis

This analysis evaluates patterns of misclassification in models that have been equipped with adversarial defense mechanisms. The main aim of this analysis is to see if misclassified samples form particular clusters and to find out data characteristics that are often misclassified. Index labels are given to 10 classes as shown in Table 7.

Table 7. Index label of class

| Index Label | Class |
|---|---|
| **0** | plane |
| **1** | car |
| **2** | bird |
| **3** | cat |
| **4** | deer |
| **5** | dog |
| **6** | frog |
| **7** | horse |
| **8** | ship |
| **9** | truck |

Table 7 shows the numbers that stand for each class in the CIFAR-10 dataset. This numbering simplifies the mapping between model output and actual class labels during clustering analysis. Using these label indices allows for consistent interpretation of clustering results across all evaluated models. For example, if a cluster has a majority index of 3, then the cluster is dominated by the cat class. This mapping also helps to analyze the label distribution per cluster and calculate entropy values in the evaluation phase.

### *1) Baseline Model*

The clustering was based on the predicted output and the class labels of the 10 classes in the CIFAR-10 dataset, as shown in Table 7. Ranging from “plane” (0) to “truck” (9). The distribution of labels in each cluster as shown in Table 8, indicates that some of the clusters are dominated by certain classes. For instance, cluster 2 is mainly comprised of “bird” samples while cluster 3 is mostly made up of “car” samples. This shows that the model groups correctly classified instances into fairly uniform clusters.

On the other hand, some clusters, such as cluster 6 and cluster 9, have a mixed class labels. These clusters most probably represent unclear or confusing feature spaces, where the model finds it hard to distinguish between the classes. Such clusters can indicate the model’s weak points.

Entropy values for each cluster as shown in Table 9 were computed to quantify label heterogeneity. Clusters with low entropy (e.g., cluster 5 and cluster 7) are composed of a single class while those with high entropy (e.g., cluster 6 and cluster 9) are of mixed class and possibly have high misclassification rates. These high-entropy clusters should be scrutinized.

Figure 11 supports these observations, showing that although some clusters are well-separated, others are either overlapping or are dispersed without a clear pattern. This supports the idea that some of the subsets of data, likely those with similar visual characteristics or insufficient separation are more likely to be misclassified.

Table 8. Label distribution per cluster.

| | Cluster | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Cluster** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| **0** | 59 | 185 | 5 | 18 | 5 | 3 | 6 | 61 | 50 | 340 |
| **1** | 1 | 0 | 8 | 107 | 99 | 351 | 18 | 348 | 1 | 0 |
| **2** | 28 | 0 | 456 | 99 | 198 | 51 | 367 | 7 | 1 | 0 |
| **3** | 37 | 455 | 0 | 0 | 0 | 0 | 0 | 0 | 58 | 178 |
| **4** | 358 | 28 | 60 | 26 | 23 | 3 | 74 | 1 | 297 | 14 |
| **5** | 195 | 72 | 3 | 1 | 0 | 0 | 0 | 0 | 526 | 30 |
| **6** | 281 | 11 | 272 | 173 | 172 | 53 | 277 | 31 | 58 | 14 |
| **7** | 1 | 241 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 414 |
| **8** | 6 | 0 | 130 | 242 | 277 | 255 | 191 | 49 | 1 | 0 |
| **9** | 34 | 8 | 66 | 334 | 226 | 284 | 67 | 502 | 6 | 10 |

Table 9. Entropy distribution per cluster.

| | Cluster |
|---|---|
| **Cluster** | **0** |
| **0** | 1.518302417 |
| **1** | 1.353652573 |
| **2** | 1.488134761 |
| **3** | 0.991121852 |
| **4** | 1.523368841 |
| **5** | 0.989830003 |
| **6** | 1.941405446 |
| **7** | 0.697201731 |
| **8** | 1.716838678 |
| **9** | 1.728903448 |

Based on Table 8 dan Table 9, among the 10 clusters generated by the baseline ResNet18 model, three clusters namely cluster 3 (“car”), cluster 5 (“ship”), and cluster 7 (“truck”) demonstrated strong internal consistency and representative, as indicated by their low entropy. These clusters could be used as markers of model strength in identifying these particular classes.

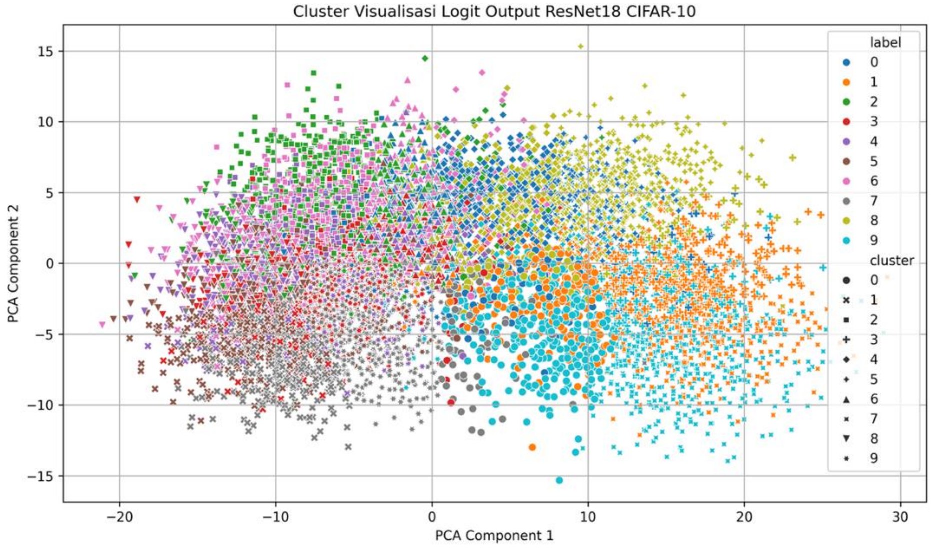


Figure 11. Cluster visualization of baseline model

Certain clusters have a variety of label distributions, as demonstrated via cluster visualization. Some clusters, like "car," "ship," and "truck," are dominated by a single label. Additionally, PCA visualization demonstrates that the distribution is not entirely class-specific. Certain classes, including dogs, cats, and deer, seem to overlap. This indicates that the model finds it difficult to distinguish between the visual characteristics of these classes because of their similarity. Although it is not ideal, the baseline model (ResNet18) generally has a very acceptable representation. 3 out of 10 clusters are pure, while the rest show a mixed class distribution.

### *2) Defense Distillation*

In the second defense mechanism, Defense Distillation, clustering again produced ten primary clusters (referencing Table 4). Label distribution as shown in Table 10.

Table 10. Label distribution per cluster.

| | **Cluster** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Cluster** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| **0** | 26 | 33 | 95 | 245 | 97 | 155 | 377 | 283 | 10 | 62 |
| **1** | 11 | 355 | 0 | 5 | 2 | 0 | 3 | 6 | 23 | 362 |
| **2** | 25 | 0 | 368 | 244 | 289 | 203 | 152 | 112 | 8 | 7 |
| **3** | 8 | 8 | 76 | 204 | 247 | 301 | 198 | 386 | 2 | 3 |
| **4** | 421 | 2 | 51 | 7 | 6 | 5 | 8 | 2 | 325 | 7 |
| **5** | 90 | 83 | 39 | 79 | 26 | 63 | 110 | 79 | 121 | 155 |
| **6** | 13 | 0 | 128 | 132 | 271 | 222 | 69 | 93 | 2 | 1 |
| **7** | 194 | 8 | 3 | 7 | 1 | 0 | 8 | 3 | 456 | 14 |
| **8** | 7 | 510 | 1 | 3 | 2 | 4 | 9 | 12 | 9 | 384 |
| **9** | 205 | 1 | 239 | 74 | 59 | 47 | 66 | 24 | 44 | 5 |

Some clusters have a dominant label distribution in one class. Cluster 1 can be seen, dominated by label 9 (“truck”). Cluster 8 (“ship”) is also filled with labels 1 (“car”) and 9 (“truck”).

Table 11. Entropy distribution per dluster.

| | **Cluster** |
|---|---|
| **Cluster** | **0** |
| **0** | 1.939873112 |
| **1** | 0.984924725 |
| **2** | 1.827700958 |
| **3** | 1.770892759 |
| **4** | 1.103135892 |
| **5** | 2.207028115 |
| **6** | 1.754010035 |
| **7** | 0.91674845 |
| **8** | 0.940658536 |
| **9** | 1.838219181 |

Cluster-wise entropy analysis showed variation across clusters (Table 11). Clusters such as cluster 1 (“car”), cluster 7 (“horse”), and cluster 8 (“ship”) demonstrated low entropy and thus higher classification consistency. In contrast, cluster 5 (“dog”) showed the highest entropy, indicating a greater likelihood of misclassification within that cluster.

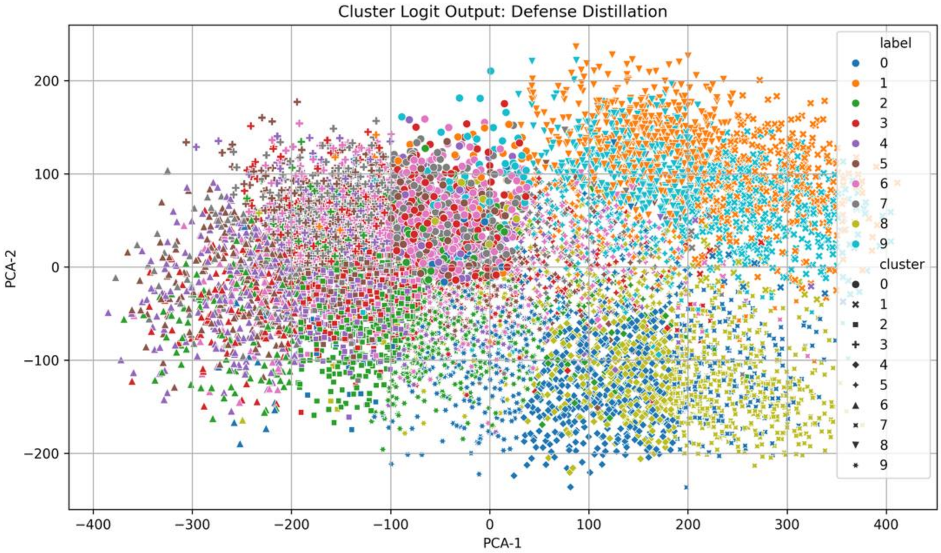


Figure 12. PCA-based clustering of logits from defense distillation model, illustrating moderate separation of classes.

PCA-based visualization (Figure 12) of these clusters revealed partial separation of class representations, with notable overlaps among visually similar classes such as “cat” vs. “dog” and “car” vs. “truck.” This underscores the limitations of the model in capturing nuanced visual differences even under a defense mechanism.

*3) Adversarial Training*

The third defense model, Adversarial Training, also resulted in ten main clusters, as aligned with Table 7. Label distribution per cluster shown in Table 12.

Table 12. Label distribution per cluster.

| | Cluster | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Cluster** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| **0** | 67 | 133 | 14 | 24 | 7 | 3 | 18 | 15 | 108 | 242 |
| **1** | 3 | 1 | 90 | 127 | 246 | 261 | 242 | 272 | 2 | 1 |
| **2** | 148 | 6 | 327 | 213 | 171 | 148 | 111 | 127 | 62 | 29 |
| **3** | 2 | 353 | 0 | 0 | 2 | 0 | 7 | 2 | 0 | 193 |
| **4** | 411 | 14 | 72 | 31 | 28 | 14 | 15 | 13 | 247 | 18 |
| **5** | 39 | 136 | 44 | 103 | 56 | 51 | 107 | 87 | 39 | 188 |
| **6** | 1 | 333 | 0 | 0 | 0 | 0 | 1 | 0 | 7 | 282 |
| **7** | 288 | 2 | 8 | 1 | 4 | 0 | 2 | 4 | 521 | 18 |
| **8** | 21 | 0 | 314 | 261 | 291 | 297 | 193 | 201 | 7 | 2 |
| **9** | 20 | 22 | 131 | 240 | 195 | 226 | 304 | 279 | 7 | 27 |

The most clusters showed different class labels yet some demonstrated strong dominance of one class. Cluster 3 together with cluster 6 contain mostly “car” and “truck” samples while cluster 7 contains mostly “plane” and “ship” examples. The class distributions in clusters 1, 2, 5 and 9 show high diversity since they contain mixed class representations.

Table 13. Entropy distribution per cluster.

| | **Cluster** |
|---|---:|
| **Cluster** | **0** |
| **0** | 1.710525056 |
| **1** | 1.757710698 |
| **2** | 2.06337266 |
| **3** | 0.772763633 |
| **4** | 1.497353959 |
| **5** | 2.160905896 |
| **6** | 0.765075666 |
| **7** | 0.878843422 |
| **8** | 1.849606474 |
| **9** | 1.941006827 |

The label distribution (Table 13) in cluster 5 shows the highest degree of heterogeneity among all clusters. The entropy values in clusters 2, 8 and 9 exceed 1.85 which shows there is a high degree of class overlap. The cluster-wise entropy measurements of the Adversarial Training model remain below those of the baseline and Defense Distillation models which indicates its output representations maintain better stability and structure.

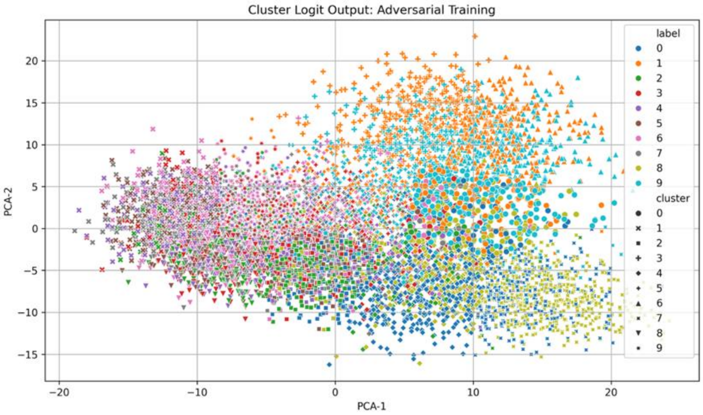


Figure 13. Clustering of adversarial training logits via PCA, showing compact and well-separated class representations.

The class distributions in clusters 5 and 9 (Figure 13) remain mixed but most clusters demonstrate growing class concentration especially for labels that attackers frequently target including label 0 (“plane”) and labels 8 (“ship”) and 9 (“truck”). The improved internal class discrimination of the Adversarial Training model becomes apparent despite its reduced accuracy on clean data.

## E. Benchmarking Clustering

Table 14. Benchmarking between models.

| Benchmarking Aspect | ResNet18 (Baseline) | Defense Distillation | Adversarial Training |
|---|---|---|---|
| Number of dominant cluster | 3 out of 10 (cluster 3, 5, 7) | 3 out of 10 (cluster 1, 7, 8) | 3 out of 10 (cluster 3, 6, 7) |
| Most heterogeneous clusters | Cluster 6, 9 | Cluster 5 | Cluster 5, 9, 2 |
| Average entropy | Medium to high | Mixed | Tends to be lower |
| Visualization | High overlap (e.g., cat/dog/deer) | Partial separation, but still overlapping | More focused, with many well-separated clusters |
| Representation stability | Unstable | Moderately stable for some classes | Most stable overall |
| Strengths | Good for initial feature representation | Helps improve cluster distribution | More robust, better at isolating adversarial target clasees |
| Weaknesses | Tends to be highly heterogeneous | Significant visual overlap remains | Lower acuracy on clean data |

The benchmarking evaluation of clustering outputs (Table 14) shows the Adversarial Training model produces the most organized and focused structure. The cluster distribution reveals both decreased total entropy and an increased number of clusters that have only one class label. The baseline ResNet18 model shows an inconsistent structure through multiple clusters containing different label distributions and higher entropy measurements. The Defense Distillation model reduces label overlaps compared to the baseline but its clustering results remain inferior to the Adversarial Training model.

PCA visualizations confirm these results. The Adversarial Training model produces distinct clusters which show strong separation between “plane,” “truck” and “ship” labels in adversarial contexts. Although the model performs poorly on uncorrupted inputs it shows outstanding resistance when facing adversarial attacks. The primary defense strategy should be Adversarial Training since it provides the best protection against adversarial attacks among the tested models.

## F. Limitation and Future Works

This study has several limitations. The research experiments were performed exclusively on the CIFAR-10 dataset which contains basic low-resolution images. The research findings have restricted applicability to real-world scenarios such as in the medical field (medical images), satellite imagery for traffic. Because they were developed using low-resolution images from a simple data structure. The study only examined model robustness through evaluation of the Fast Gradient Sign Method (FGSM) adversarial attack. The research results may not apply and cannot be justified in general against different types of complex adversarial attacks.

For further research, the study should investigate additional adversarial attack methods including transfer attacks and black-box attacks to achieve better model robustness assessment. The evaluation of defense mechanism combinations between defense distillation and adversarial training could produce more effective defense mechanisms. The research should use larger datasets with diverse content and high-resolution images to understand how models resist adversarial attacks in different scenarios.

# IV. CONCLUSION

This study shows that the Fast Gradient Sign Method (FGSM) adversarial attack significantly reduces the accuracy of the baseline ResNet18 model, dropping from 85.01% on clean data to 19.23% at an epsilon of 0.3. This sharp decline highlights the high vulnerability of deep learning models to adversarial attacks, emphasizing the need for effective defense mechanisms to ensure reliability in image classification applications. Evaluation of the two defense strategies tested, defense distillation and adversarial training reveals varied performance. Defense distillation maintains strong accuracy on clean data (81.70%) but remains vulnerable to FGSM attacks, with accuracy falling to 23.68% at epsilon 0.3. In contrast, adversarial training demonstrates robustness at higher epsilon values (up to 60.34% at epsilon 0.25) but sacrifices clean data performance, achieving only 57.08% accuracy. These findings indicate that defense distillation lacks sufficient robustness against attacks, while adversarial training trades off clean data accuracy for improved resilience. Beyond evaluating defense mechanisms based on accuracy, this study introduces a benchmarking approach using clustering analysis and per-cluster entropy to assess the stability of the models' internal logit representations. The results show that adversarial training produces more stable and focused internal representations compared to the baseline and defense distillation models, despite its lower clean data accuracy. This highlights a trade-off between clean data performance and adversarial robustness, underscoring the need for balanced defense strategies. The development and evaluation of defense distillation and adversarial training against FGSM attacks on the ResNet18 model with the CIFAR-10 dataset, along with the proposed clustering and entropy-based benchmarking approach, represent the primary contributions of this study. These provide a comprehensive analysis of two widely used defense methods in the context of FGSM attacks, supported by detailed experimental configurations and evaluation metrics.

Further refinements are needed to enhance the effectiveness of these defense strategies. Future research should explore combining defense distillation and adversarial training to achieve a better balance between accuracy and robustness. Additionally, developing new defense methods to optimize performance against a broader range of adversarial attacks, including transfer and black-box attacks, is recommended. Testing on larger, more diverse, and higher-resolution datasets is also advised to gain a more comprehensive understanding of model robustness. These efforts aim to contribute to the development of deep learning-based image classification systems that are both accurate and resilient to adversarial attacks.

## REFERENCES

[1] Y. LeCun *et al.*, "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural Comput*, vol. 1, no. 4, pp. 541–551, 1989, doi: 10.1162/neco.1989.1.4.541.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Dec. 2015, [Online]. Available: https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf

[3] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," *A conference paper at ICLR 2015*, Dec. 2014.

[4] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards Deep Learning Models Resistant to Adversarial Attacks," 2019. doi: https://doi.org/10.48550/arXiv.1706.06083.

[5] N. Carlini and D. Wagner, " Towards Evaluating the Robustness of Neural Networks ," in *2017 IEEE Symposium on Security and Privacy (SP)* , Los Alamitos, CA, USA: IEEE Computer Society, May 2017, pp. 39–57. doi: 10.1109/SP.2017.49.

[6] H. Waghela, J. Sen, and S. Rakshit, "Robust Image Classification: Defensive Strategies against FGSM and PGD Adversarial Attacks," in *2024 Asian Conference on Intelligent Technologies (ACOIT)*, 2024, pp. 1–7. doi: 10.1109/ACOIT62457.2024.10941671.

[7] J. Sen and S. Dasgupta, “Adversarial Attacks on Image Classification Models: FGSM and Patch Attacks and Their Impact,” in *Information Security and Privacy in the Digital World - Some Selected Topics*, J. Sen and J. Mayer, Eds., Rijeka: IntechOpen, 2023. doi: 10.5772/intechopen.112442.

[8] N. Carlini and D. Wagner, “Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods,” in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, in AISec ’17. New York, NY, USA: Association for Computing Machinery, 2017, pp. 3–14. doi: 10.1145/3128572.3140444.

[9] J. Sen, A. Sen, and A. Chatterjee, “Adversarial Attacks on Image Classification Models: Analysis and Defense,” 2023. doi: 10.13140/RG.2.2.29593.19044/2.

[10] T. Bai, J. Luo, J. Zhao, B. Wen, and Q. Wang, “Recent Advances in Adversarial Training for Adversarial Robustness,” *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21),* Feb. 2021, doi: 10.24963/ijcai.2021/591.

[11] N. A. S, V. Chaturvedi, and M. Shafique, “S-E Pipeline: A Vision Transformer (ViT) based Resilient Classification Pipeline for Medical Imaging Against Adversarial Attacks,” in *2024 International Joint Conference on Neural Networks (IJCNN)*, 2024, pp. 1–8. doi: 10.1109/IJCNN60899.2024.10650591.

[12] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, “Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks,” *2016 IEEE Symposium on Security and Privacy (SP),* Nov. 2015, doi: 10.1109/SP.2016.41.

[13] S. Y. Khamaiseh, D. Bagagem, A. Al-Alaj, M. Mancino, and H. W. Alomari, “Adversarial Deep Learning: A Survey on Adversarial Attacks and Defense Mechanisms on Image Classification,” 2022, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/ACCESS.2022.3208131.

[14] A. Krizhevsky, “Learning Multiple Layers of Features from Tiny Images,” 2009. [Online]. Available: https://api.semanticscholar.org/CorpusID:18268744

[15] G. Dong, H. Boström, M. Vazirgiannis, and R. Bresson, “Obtaining Example-Based Explanations from Deep Neural Networks,” 2025. doi: 10.1007/978-3-031-91398-3_32.

[16] K. Chowdhury, “Adversarial Machine Learning: Attacking and Safeguarding Image Datasets,” *Proceedings of the Fourth International Conference on Ubiquitous Computing and Intelligent Information Systems (ICUIS-2024),* Jan. 2025, doi: 10.1109/ICUIS64676.2024.10866337.

[17] K. Alomar, H. I. Aysel, and X. Cai, “Data Augmentation in Classification and Segmentation: A Survey and New Strategies,” *J Imaging,* vol. 9, no. 2, Feb. 2023, doi: 10.3390/jimaging9020046.

[18] B. Rahman, F. Fauzi, and S. Amri, “Perbandingan Hasil Klasifikasi Data Iris menggunakan Algoritma K-Nearest Neighbor dan Random Forest,” 2023. [Online]. Available: http://journalnew.unimus.ac.id/index.php/jodi

[19] E. Ilmiyah and A. Bahtiar, “PENERAPAN ALGORITMA K-MEANS CLUSTERING UNTUK MENGELOMPOKKAN DATA MAHASISWA BARU,” 2024.

[20] L. El Fattahi and E. H. Sbai, “Clustering using kernel entropy principal component analysis and variable kernel estimator,” *International Journal of Electrical and Computer Engineering*, vol. 11, no. 3, pp. 2109–2119, Jun. 2021, doi: 10.11591/ijece.v11i3.pp2109-2119