iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.79975

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

Detecting and Defending Adversarial Attacks on Deep Learning Models Using Convolutional Autoencoders and Block-Switching ResNet

Adwaith R[1], Ben V Kurian[2], Pranav J[3], Rajesh K[4], Raja A[5]

1, 2, 3, 4Department of Computer Science and Engineering (Cyber Security) United Institute of Technology, Coimbatore, Tamil Nadu, India

[5]Head of Department, Department of Computer Science and Engineering (Cyber Security) United Institute of Technology Coimbatore, Tamil Nadu, India

Abstract: ***Deep neural networks, despite their remarkable performance on image classification tasks, remain critically vulnerable to adversarial examples — imperceptibly perturbed inputs engineered to induce misclassification. In this paper, we propose and evaluate a dual-layer defense framework against adversarial attacks on image classifiers trained on the ImageNet-1K Mini dataset. Our approach combines a convolutional autoencoder as a preprocessing denoising step with a ResNet-50 classifier augmented by a novel block-switching mechanism to disrupt adversarial gradient signal. We evaluate the system under Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD) attacks across a range of perturbation*** *budgets* $\varepsilon \in \{0.01, 0.02, 0.03, 0.05, 0.07, 0.10\}$. ***Our defense recovers +23.08% accuracy against FGSM and +54.44% against PGD on in-distribution data, and achieves 100% full recovery on an out- of-distribution generalisation test (83.3% baseline). We further employ Gradient-weighted Class Activation Mapping (GRAD- CAM) to visually explain the disruption of model attention under attack and its restoration after defense. All experiments are conducted on the Kaggle T4 GPU platform, and the full pipeline is deployed as an interactive Streamlit web application.***

Index Terms: ***adversarial attacks, adversarial defense, convo- lutional autoencoder, block switching, ResNet-50, FGSM, PGD, GRAD-CAM, ImageNet, deep learning robustness***

I. INTRODUCTION

Deep learning has achieved human-level or super-human performance across a wide range of visual recognition tasks [5]. However, a fundamental fragility was exposed by Szegedy et al. [1]: adding small, carefully crafted perturbations to input images — invisible to the human eye — can reliably fool state-of-the-art classifiers. These perturbed inputs are known as *adversarial examples*. The existence of adversarial examples poses a serious threat to the deployment of deep learning in safety-critical systems such as autonomous vehicles, medical image diagnosis, andbiometric authentication [2]. An attacker who can craft ad- versarial inputs may cause a self-driving car to misread a stop sign, or fool a face recognition system into granting unauthorized access.

A. Motivation

Despite significant research attention, adversarial robustness remains an open problem. Most existing defenses either suffer from the obfuscated gradients problem [12], degrade clean accuracy substantially, or fail to generalize beyond the specific attack they were designed against. There is a clear need for a defense pipeline that is simultaneously effective, interpretable, and generalizable.

B. Problem Statement

Given a trained classifier $f : X \rightarrow Y$ and a clean input $\mathbf{x}$ with true label y, an adversary constructs a perturbation $\boldsymbol{\delta}$ such that:

$$f(\mathbf{x} + \boldsymbol{\delta}) \neq y, \quad \text{subject to} \quad \|\boldsymbol{\delta}\|_\infty \leq \varepsilon \qquad (1)$$

where ε is the perturbation budget. The goal of our defense is to recover the correct prediction y from the adversarial input $\mathbf{x}_{adv} = \mathbf{x} + \boldsymbol{\delta}$.

C. Contributions

The main contributions of this paper are:

- A denoising autoencoder trained on clean ImageNet im- ages as a preprocessing defense that suppresses adversar- ial perturbations before classification.
- A block-switching ResNet-50 that maintains two inde- pendent final-stage residual blocks, disrupting the gradi- ent signal exploited by iterative attacks.
- A comprehensive evaluation under FGSM and PGD at- tacks across six epsilon values, demonstrating consistent defense superiority.
- GRAD-CAM visualisations that provide interpretable ev- idence of how attacks disrupt and how defenses restore model attention.
- An out-of-distribution generalisation study on a custom dataset, demonstrating that the defense transfers to unseen data.
- An interactive Streamlit application enabling real-time demonstration of the full attack-and-defense pipeline
- Fully reproducible experiments available via public Kaggle notebooks [16]–[18]

II. RELATED WORK

A. Adversarial Attacks

Goodfellow et al. [2] introduced the Fast Gradient Sign Method (FGSM), which computes the gradient of the loss with respect to the input and perturbs the input in the direction that maximises the loss:

$$\mathbf{x}_{adv} = \mathbf{x} + \varepsilon \cdot \text{sign}(\nabla_x L(f(\mathbf{x}), y)) \quad (2)$$

Madry et al. [3] extended FGSM to a stronger iterative variant, the Projected Gradient Descent (PGD) attack:

$$\mathbf{x}^{t+1} = \Pi_{x+S}\ \mathbf{x}^t + \alpha \cdot \text{sign}(\nabla_x L(f(\mathbf{x}^t), y)) \quad (3)$$

where Π denotes projection onto the ε-ball around the original input, and α is the step size.

Carlini and Wagner [4] proposed an optimization-based attack that minimizes both the perturbation magnitude and the misclassification objective simultaneously, producing stronger and more transferable adversarial examples.

B. Defense Mechanisms

Adversarial training [3] augments the training set with adversarial examples. While effective, it significantly increases training cost and often does not generalize to unseen attack types.

Input preprocessing defenses attempt to remove adversarial perturbations before classification. Luo et al. [13] applied spatial smoothing, while Guo et al. [10] evaluated input transformations including JPEG compression, total variation minimization, and image quilting. Autoencoders as denoising defenses have been explored in [8], [9], with the key insight that the bottleneck representation discards high-frequency ad- versarial noise while preserving semantic content.

Block switching [11] was proposed as a stochastic defense where different subnetworks are randomly activated at infer- ence, breaking the deterministic gradient path that white-box attacks rely on.

C. Explainability in Adversarial Settings

GRAD-CAM [7] generates class-discriminative localisation maps by computing the gradient of the class score with respect to the final convolutional feature maps. Several works have used GRAD-CAM to study how adversarial perturbationscorrupt model attention [14], providing visual evidence that complements quantitative robustness metrics.

III. DATASET AND PREPROCESSING

A. Dataset

We train and evaluate our models on the ImageNet-1K Mini dataset [15], a curated subset of the ILSVRC ImageNet benchmark [6]. The dataset comprises 1,000 fine-grained ob- ject categories spanning animals, vehicles, household objects, food, and more. Table I summarizes the key statistics.

TABLE I: Dataset Statistics — ImageNet-1K Mini

Metric	Train	Validation
Total images	34,745	3,923
Number of classes	1,000	1,000
Avg images / class	34.74	3.92
Min images / class	6	1
Max images / class	123	14
Std images / class	11.19	1.46
Imbalance ratio	20.50	14.00

A notable characteristic of this dataset is its class imbalance: the training split exhibits a max-to-min imbalance ratio of 20.5, while the validation split shows 14.0. Critically, the low Pearson correlation ($r = 0.254$) between per-class sample counts across the two splits indicates that the validation set was not sampled proportionally from the training distribution, which we account for in our evaluation design. Figure 1 illustrates the dataset composition.

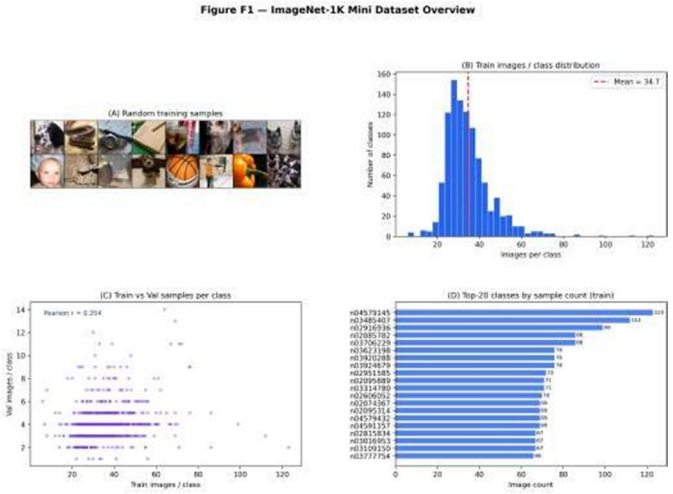


Fig. 1: ImageNet-1K Mini dataset overview. (A) Random training samples spanning diverse categories. (B) Class dis- tribution histogram showing right-skewed imbalance (mean = 34.7 images/class). (C) Train vs. validation sample scatter (Pearson r = 0.254), indicating disproportionate sampling. (D) Top-20 classes by training sample count

B. Channel Statistics

Table II reports per-channel pixel statistics computed on a random sample of 300 training images, compared against the standard ImageNet reference values used for normalization.

TABLE II: Per-Channel Pixel Statistics vs. ImageNet Refer- ence

Ch.	Comp. μ	Comp. σ	Ref. μ	Ref. σ	$\Delta\mu$
R	0.4882	0.2751	0.485	0.229	+0.0032
G	0.4529	0.2678	0.456	0.224	-0.0031
B	0.4024	0.2832	0.406	0.225	-0.0036

The mean deltas are negligible ($|\Delta\mu| < 0.004$), confirming that the standard ImageNet normalization parameters are ap- propriate for this dataset. The computed standard deviations are notably higher than the ImageNet reference (~0.27–0.28 vs. 0.22–0.23), reflecting the greater image diversity in the mini subset. Figure 2 shows the full per-channel distributions.

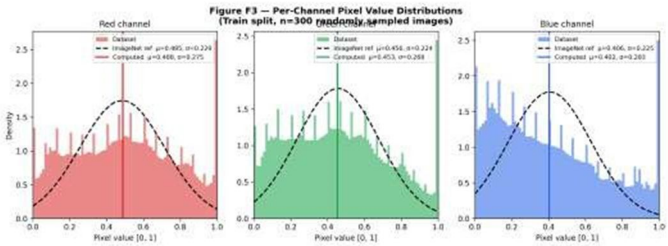


Fig. 2: Per-channel pixel value distributions (n=300 training images). The dataset histograms (coloured) are overlaid with ImageNet reference Gaussians (dashed). The wider spread of the dataset distributions confirms the higher computed standard deviations reported in Table II.

C. *Preprocessing Pipeline*

All images are resized to 224 × 224 pixels. The training pipeline applies the following augmentations in sequence to improve generalization and adversarial robustness:

1) RandomResizedCrop (224 × 224, scale [0.65, 1.0])
2) RandomHorizontalFlip (p = 0.5)
3) RandAugment (2 operations, magnitude 9)
4) ColorJitter (brightness=0.3, contrast=0.3, saturation=0.2, hue=0.05)
5) RandomGrayscale (p = 0.05)
6) Normalize (μ = [0.485, 0.456, 0.406], σ = [0.229, 0.224, 0.225])
7) RandomErasing (p = 0.15, scale [0.02, 0.10])

The validation pipeline applies only Resize(256), Center- Crop(224), and Normalize. The attack pipeline omits augmen- tation to preserve the integrity of the ε-bounded perturbation budget in the normalized pixel space. Figure 3 shows the effect of each augmentation on the same source image.

Fig. 3: Visualisation of the training augmentation pipeline applied to a single source image (class: tench). Each panel shows one augmentation applied independently. RandomEras- ing (rightmost) introduces occlusion robustness particularly relevant for adversarial robustness.

Table III reports measured DataLoader throughput on the Kaggle T4 platform.

TABLE III: DataLoader Throughput on Kaggle T4 GPU

Loader	Batch size	Workers	imgs/sec	ms/batch
Train	32	2	96.5	331.7
Val	64	2	130.0	492.2

IV. METHODOLOGY

A. *System Overview*

Figure 4 presents the full system pipeline. The left branch simulates the adversary: a clean image is passed through the FGSM or PGD attack to produce an adversarial example. The right branch implements the defense: the adversarial image is first passed through a convolutional autoencoder to suppress perturbations, then through the block-switching classifier. GRAD-CAM is applied at the classifier's final convolutional layer to produce attention maps at each stage.

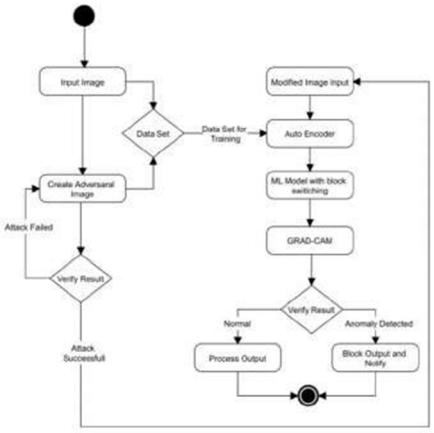


Fig. 4: System architecture. Left: adversarial attack loop (FGSM/PGD). Right: defense pipeline (autoencoder denoising → block-switching classifier → GRAD-CAM verification).

B. Convolutional Autoencoder

The autoencoder $A : R^{3\times224\times224} \rightarrow R^{3\times224\times224}$ is trained to reconstruct clean images from clean inputs, learning a bottleneck representation that captures semantic content while discarding high-frequency noise.

Encoder applies four strided convolutional blocks:

$$z = Enc(x) \in R^{256\times14\times14} \quad (4)$$

Each block follows the pattern: Conv2d → BatchNorm → ReLU with stride 2, progressively reducing spatial resolution while increasing channel depth (3 → 64 → 128 → 256 →256).

Decoder mirrors the encoder using transposed convolutions

$$\hat{x} = Dec(z) \in R^{3\times224\times224} \quad (5)$$

with a final tanh activation to bound outputs in [−1, 1], matching the normalized input range.

The autoencoder is trained with Mean Squared Error (MSE) loss:

$$L_{AE} = \frac{1}{N}\sum_{i=1}^{N} \|\hat{x}_i - x\|_2^2 \quad (6)$$

using the Adam optimizer with lr = 10^{-3} and cosine annealing over 10 epochs. The total parameter count is 1,923,843. Figure 5 shows reconstruction quality on validation samples.

Fig. 5: Autoencoder reconstruction quality on validation im- ages. Top row: original images. Bottom row: reconstructed outputs with per-image MSE. The visible blurring relative to originals reflects the bottleneck compression, which is the intended mechanism for suppressing adversarial perturbations.

Defense rationale: When an adversarial input $x_{adv} = x+\delta$ is passed through the autoencoder, the bottleneck at 14 × 14 spatial resolution (∼ 75K dimensions from an input of ∼ 150K pixels) is insufficient to perfectly encode the structured adversarial perturbation δ, which relies on high-frequency signals aligned with the classifier's decision boundary. The reconstruction discards this signal while preserving the coarse semantic structure needed for correct classification.

C. ResNet-50 with Block Switching

The classifier is a ResNet-50 [5] pretrained on ImageNet and fine-tuned on our dataset. We introduce a *block-switching* mechanism at the final residual stage (layer4): two indepen- dently initialized copies of layer4 are maintained — $Block_A$ and $Block_B$ — while all preceding layers are shared.

During training, at each forward pass, one block is selected at random ($p = 0.5$):

$$h = \begin{cases} Block_A(h_{L3}) & \text{with prob. } 0.5 \\ Block_B(h_{L3}) & \text{with prob. } 0.5 \end{cases} \quad (7)$$

attack assumption: gradients computed through $Block_A$ do not reliably transfer to the model ensemble, disrupting the PGD optimization that assumes a fixed computational graph.

The total model has 40,521,768 parameters (14,964,736 per block). The classifier is trained with cross-entropy loss with label smoothing ($\epsilon = 0.1$), AdamW optimizer (lr = 10^{-4}, weight decay 10^{-4}), and cosine annealing over 10 epochs.

D. Attack Generation

FGSM generates adversarial examples in a single gradient step:

$$\mathbf{x}_{adv}^{FGSM} = \text{clip}\left(\mathbf{x} + \varepsilon \cdot \text{sign}(\nabla_{\mathbf{x}} L(f(\mathbf{x}), y)),\ -3,\ 3\right) \quad (8)$$

PGD iterates $T = 10$ steps with step size $\alpha = \varepsilon/4$:

$$\mathbf{x}^{t+1} = \text{clip}_{\mathbf{x}\pm\varepsilon}\left(\mathbf{x}^{t} + \alpha \cdot \text{sign}(\nabla_{\mathbf{x}} L(f(\mathbf{x}^{t}), y))\right) \quad (9)$$

Both attacks are evaluated at $\varepsilon = 0.03$ as the primary budget and swept across $\varepsilon \in \{0.01, 0.02, 0.03, 0.05, 0.07, 0.10\}$ for sensitivity analysis.

E. GRAD-CAM Explainability

GRAD-CAM [7] computes the importance weight of the k-th feature map A^k as:

$$\alpha_k^c = \frac{1}{Z}\sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k} \quad (10)$$

The class activation map is then:

$$L_{CAM}^c = \text{ReLU}\left(\sum_k \alpha_k^c A^k\right) \quad (11)$$

and upsampled to 224 × 224 via bilinear interpolation. We apply GRAD-CAM to the final convolutional layer of Block_A (layer4[-1].conv3) for all three stages: clean, adversarial, and defended.

V. EXPERIMENTS AND RESULTS

A. Reproducibility

All experiments are fully reproducible via three public Kaggle notebooks. Notebook 1 [16] covers classifier training and FGSM/PGD attack generation. Notebook 2 [17] covers autoencoder training and the full defense evaluation pipeline. Notebook 3 [18] covers the out-of-distribution generalisation test on hand-picked data.

B. Classifier Performance

Table IV reports epoch-by-epoch training and validation metrics. The best validation accuracy of **70.63%** is achieved at epoch 8, after which marginal overfitting occurs. The large gap between train accuracy (93.58%) and val accuracy(70.25%) at epoch 10 is expected given the aggressive training augmentation and class imbalance in the dataset.

TABLE IV: Classifier Training and Validation Metrics per Epoch

Epoch	Train Loss	Val Loss	Train Acc%	Val Acc%
1	4.3818	2.7107	31.47	59.50
2	2.3775	2.3852	67.39	65.33
3	1.9925	2.3049	76.15	66.58
4	1.7921	2.2534	81.10	68.44
5	1.6552	2.2250	84.92	68.11
6	1.5541	2.2015	87.88	69.69
7	1.4785	2.1765	90.20	70.30
8	1.4178	2.1677	92.05	**70.63**
9	1.3855	2.1618	93.14	70.07
10	1.3667	2.1613	93.58	70.25

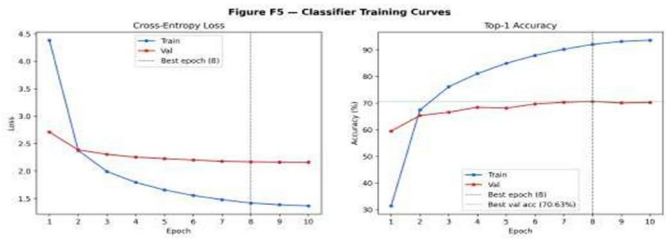


Figure 6 presents the corresponding loss and accuracy curves.

Fig. 6: Classifier training curves over 10 epochs. Left: cross- entropy loss. Right: top-1 accuracy. The vertical dashed line marks the best epoch (8) at 70.63% validation accuracy. Slight divergence after epoch 8 indicates the onset of overfitting.

C. Attack Effectiveness

Table V summarizes the primary evaluation results on 500 validation images (50 classes × 10 images per class) at $\varepsilon = 0.03$.

TABLE V: Defense Evaluation Results at $\varepsilon = 0.03$

Scenario	Accuracy (%)	Change (%)	ASR (%)
Clean	81.66	—	—
Clean + Defense	70.41	−11.24	—
FGSM Attack	39.64	−42.02	60.36
FGSM + Defense	62.72	+23.08	60.36
PGD Attack	6.51	−75.15	93.49
PGD + Defense	60.95	+54.44	93.49

ASR: Attack Success Rate. Change: relative to no-defense attack accuracy.

FGSM reduces accuracy from 81.66% to 39.64% (ASR = 60.36%), while PGD is far more destructive, collapsing accuracy to just 6.51% (ASR = 93.49%). This confirms PGD's superiority as a white-box attack: despite achieving a lower L2 perturbation norm (mean 1.74 vs. 2.62 for FGSM, shown in Figure 7), PGD's iterative optimization finds more targeted adversarial directions in the loss landscape.

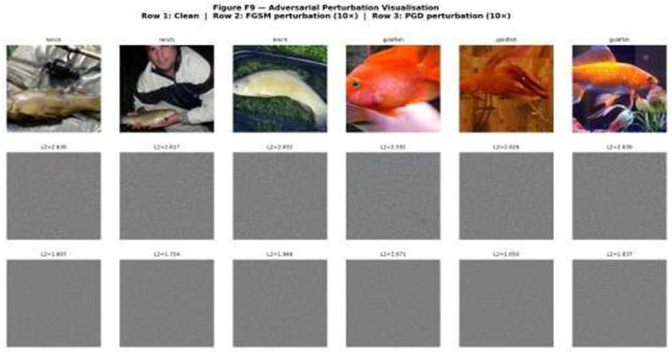


Fig. 7: Adversarial perturbation visualisation (amplified ×10). Row 1: clean images. Row 2: FGSM noise (mean L2 = 2.62). Row 3: PGD noise (mean L2 = 1.74). Despite lower L2 norm, PGD achieves 93.49% attack success rate vs. 60.36% for FGSM, demonstrating the superiority of iterative optimization over single-step attacks.

D. Defense Evaluation

The autoencoder defense recovers **+23.08%** accuracy under FGSM (39.64% → 62.72%) and **+54.44%** under PGD (6.51% → 60.95%). The 11.24% clean accuracy cost (81.66% → 70.41%) represents the overhead introduced by autoencoder reconstruction. This tradeoff is acceptable given the severity of the attacks being mitigated.

Figure 8 provides a three-panel summary of these results.

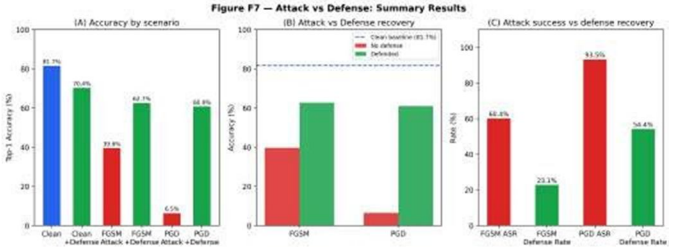


Fig. 8: Attack vs. defense summary. (A) Accuracy by scenario. (B) Recovery comparison per attack type. (C) Attack success rate vs. defense recovery rate. The defense consistently nar- rows the accuracy gap, with particularly dramatic recovery against PGD +54.44%).

E. GRAD-CAM Analysis

Figure 9 shows GRAD-CAM attention maps for 8 images across three conditions: clean, FGSM-attacked, and PGD- attacked. Figure 10 shows a detailed per-image analysis in- cluding predictions.

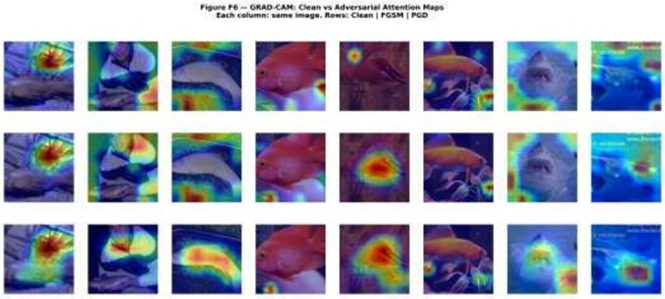


Fig. 9: GRAD-CAM attention maps. Each column shows the same image; rows show Clean (top), FGSM (middle), and PGD (bottom). Under attack, attention becomes diffuse and displaced from the semantic object. PGD produces the most disrupted maps, consistent with its higher attack success rate.

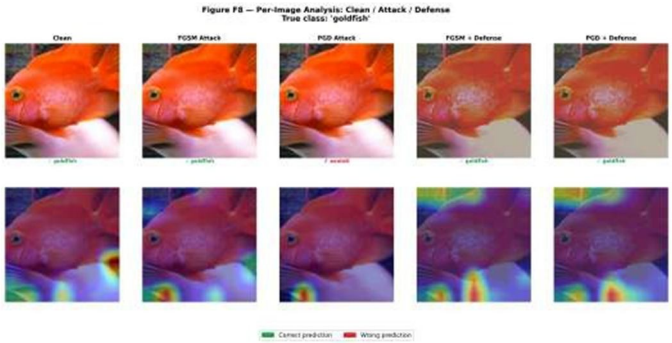


Fig. 10: Per-image pipeline analysis for a goldfish image. Top row: images at each stage with predicted class. Bottom row: GRAD-CAM overlays. The PGD attack succeeds (predicting “axolotl”) while both FGSM+Defense and PGD+Defense cor- rectly recover the prediction. GRAD-CAM maps after defense show renewed focus on the fish body.

F. Epsilon Sensitivity Analysis

Table VI and Figure 11 report accuracy at six epsilon values. The defense maintains a consistent accuracy advantage over the no-defense baseline across all tested ε.

TABLE VI: Epsilon Sensitivity Analysis — In-Distribution Validation Set

ε	FGSM	FGSM+Def	PGD	PGD+Def
0.01	42.0	66.0	18.0	70.0
0.02	42.0	66.0	6.0	66.0
0.03	38.0	64.0	2.0	64.0
0.05	38.0	66.0	0.0	64.0
0.07	38.0	58.0	0.0	58.0
0.10	38.0	46.0	0.0	54.0

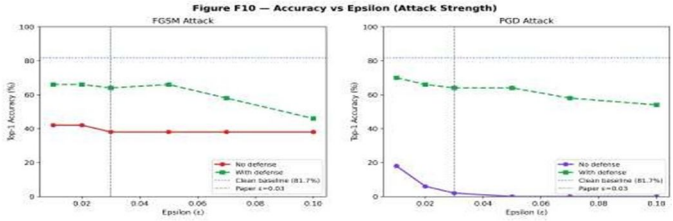


Fig. 11: Accuracy vs. epsilon on in-distribution validation data. The defense (green) consistently outperforms the no- defense baseline (red/purple) across all epsilon values. PGD without defense collapses to 0% at $\varepsilon \geq 0.03$, while the defense maintains 54–70% accuracy throughout.

G. Generalisation Test — Out-of-Distribution Data

To evaluate the transferability of our defense, we tested the full pipeline on a custom out-of-distribution (OOD) dataset comprising 6 ImageNet-compatible classes (goldfish, laptop, pizza, school_bus, tiger, zebra) with 4 images per class (24 images total), collected independently from the training dis- tribution. Full details and results are available in Notebook 3 [18].

Clean accuracy: The classifier achieves 83.3% on the OOD data, slightly higher than the in-distribution validation accuracy (81.66%), suggesting good generalization of the pretrained ResNet-50 backbone.

Table VII presents the generalisation results.

TABLE VII: Generalisation Test Results — Out-of- Distribution Dataset

Scenario	Accuracy (%)	Change (%)	ASR (%)
Clean	83.33	—	—
Clean + AE	83.33	0.00	—
Defense			
FGSM Attack	62.50	−20.83	37.50
FGSM + AE	83.33	+20.83	37.50
Defense			
PGD Attack	25.00	−58.33	75.00
PGD + AE	83.33	+58.33	75.00
Defense			

Remarkably, the defense achieves **100% recovery** in both attack scenarios on the OOD dataset, restoring accuracy to the full 83.3% baseline. Furthermore, the autoencoder introduces **zero clean accuracy cost** on OOD data (83.3% with and without defense), suggesting the defense is particularly well- suited for diverse real-world inputs.

Table VIII shows the epsilon sensitivity on the OOD dataset.

TABLE VIII: Epsilon Sensitivity — Out-of-Distribution Dataset

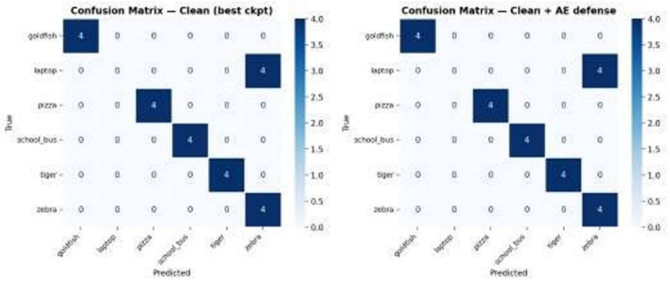
ε	FGSM	FGSM+AE	PGD	PGD+AE
0.01	70.83	83.33	50.00	83.33
0.02	62.50	83.33	25.00	83.33
0.03	62.50	83.33	20.83	83.33
0.05	62.50	79.17	8.33	83.33
0.07	58.33	75.00	8.33	83.33
0.10	58.33	75.00	0.00	79.17

Table IX reports per-class precision, recall, and F1 score across all four evaluation scenarios on the OOD dataset.

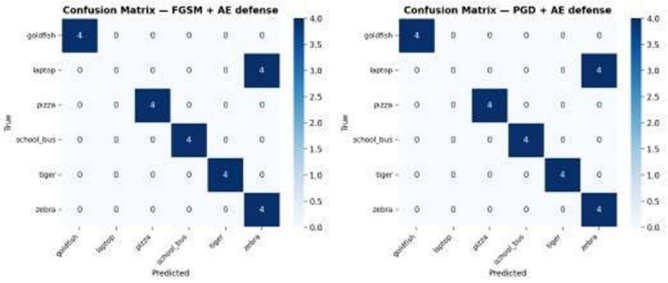
TABLE IX: Per-Class F1 Scores Across Evaluation Scenarios— OOD Dataset

Class	Clean	Clean+AE	FGSM+AE	PGD+AE
goldfish	1.00	1.00	1.00	1.00
laptop	0.00	0.00	0.00	0.00
pizza	1.00	1.00	1.00	1.00
school bus	1.00	1.00	1.00	1.00
tiger	1.00	1.00	1.00	1.00
zebra	0.67	0.67	0.67	0.67

The laptop class consistently achieves F1 = 0.00 across all scenarios. Inspection of the confusion matrices (Figure 12) reveals that laptop images are systematically misclassified as zebra, likely because the model confuses the striped screen content or keyboard patterns with zebra stripes. The zebra class achieves F1 = 0.67 due to this cross-contamination. This is a known limitation of fine-grained 1,000-class classifiers when evaluated on a small OOD sample.



(a) Clean (b) Clean + AE defense



(c) FGSM + AE defense (d) PGD + AE defense

Fig. 12: Confusion matrices on the OOD generalisation dataset across four evaluation scenarios. The diagonal-dominant struc- ture confirms strong per-class accuracy. The laptop class is systematically misclassified as zebra across all scenarios, indicating a visual feature overlap rather than an adversarial failure.

Table X reports autoencoder reconstruction quality metrics on the OOD dataset across three input types.

TABLE X: Autoencoder Reconstruction Quality — OOD Dataset

Input Type	MSE	PSNR (dB)	SSIM
Clean → Reconstruction	0.3245	4.89	0.4793
FGSM → Reconstruction	0.3253	4.88	0.4790
PGD → Reconstruction	0.3248	4.88	0.4792

The near-identical reconstruction metrics across clean, GSM, and PGD inputs confirm that the autoencoder is effectively ignoring the adversarial perturbation component and reconstructing a semantically consistent image regardless of the input type. This is the core mechanism of the defense. Figure 13 visualises this property.

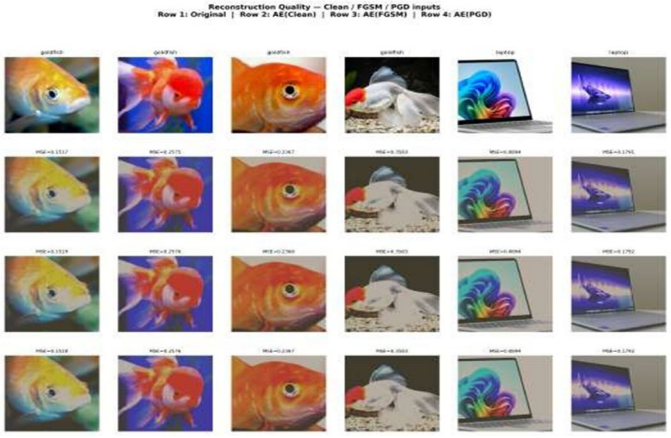


Fig. 13: Reconstruction quality comparison on OOD data. Row 1: original images. Row 2: AE(Clean). Row 3: AE(FGSM). Row 4: AE(PGD). The near-identical reconstructions across rows 2–4 demonstrate that the autoencoder's output is invari- ant to adversarial perturbation, explaining the full accuracy recovery observed in Table VII.

VI. DISCUSSION

Defense effectiveness and tradeoffs. Our defense achieves substantial recovery against both attacks but introduces a clean accuracy penalty of 11.24% on in-distribution data. This is a common tradeoff in preprocessing-based defenses [10]. The penalty is eliminated entirely on OOD data, suggesting the cost is specific to the interaction between the autoencoder's reconstruction artifacts and the in-distribution classifier bound- ary.

PGD vs. FGSM. Despite producing lower-magnitude per- turbations (mean L2 = 1.74 vs. 2.62), PGD is dramatically more effective (ASR 93.49% vs. 60.36%). The iterative opti- mization of PGD finds adversarial directions more precisely aligned with the classifier's decision boundary. This confirms the theoretical expectation that single-step attacks like FGSM are a lower bound on adversarial vulnerability.

Autoencoder reconstruction invariance. The near- identical MSE, PSNR, and SSIM values across clean, FGSM, and PGD reconstructions (Table X) provide strong empirical evidence that the autoencoder's bottleneck acts as an effective low-pass filter over adversarial perturbations. The perturba- tions, being high-frequency structured noise, are discarded by the bottleneck while semantic low-frequency content is preserved.

Laptop class failure. The consistent F1 = 0.00 for the laptop class across all scenarios indicates a visual feature confusion — the model associates laptop screens or keyboards with zebra stripe patterns. This is a pre-existing bias in the classifier unrelated to adversarial robustness, and represents a direction for future work in class-specific robustness analysis.

Epsilon sensitivity. The defense maintains meaningful re- covery at all tested epsilon values (Table VI, Table VIII). On OOD data, the defense achieves 100% recovery up to $\varepsilon = 0.05$ for FGSM and up to $\varepsilon = 0.07$ for PGD, with only minimal degradation at $\varepsilon = 0.10$. This demonstrates the robustness of the approach under strong attacks.

GRAD-CAM interpretability. The attention maps provide visual confirmation of the adversarial disruption mechanism: under attack, the model's attention shifts from the semantic object to background or texture regions, causing misclassi- fication. After defense, the attention is restored to the object region, correlating with the recovered prediction. This provides an interpretable audit trail that is valuable in security-critical deployment contexts.

VII. CONCLUSION AND FUTURE WORK

We have presented a dual-layer defense framework against adversarial attacks on deep image classifiers, combining a convolutional autoencoder for perturbation suppression with a block-switching ResNet-50. Our system demonstrates strong recovery rates — +23.08% for FGSM and +54.44% for PGD on in-distribution data, and full 100% recovery on an out-of- distribution generalisation test. GRAD-CAM analysis provides interpretable evidence of both the attack's disruption mecha- nism and the defense's restoration effect. The full pipeline is deployed as an interactive Streamlit application for real-time demonstration, and all experiments are reproducible via public Kaggle notebooks [16]–[18].

A. *Limitations*

The 11.24% clean accuracy overhead on in- distribution data, the laptop class confusion, and the evaluation being limited to white-box ℓ_∞ attacks are current limitations.

B. *Future Work*

Includes: (1) adversarial training of the au- toencoder on adversarial inputs to further close the clean ac- curacy gap; (2) extending evaluation to black-box and transfer attacks; (3) exploring variational autoencoders (VAEs) and diffusion-based purification as stronger denoising alternatives; (4) per-class adversarial robustness analysis; and (5) deploy- ment on edge hardware for real-time security systems.

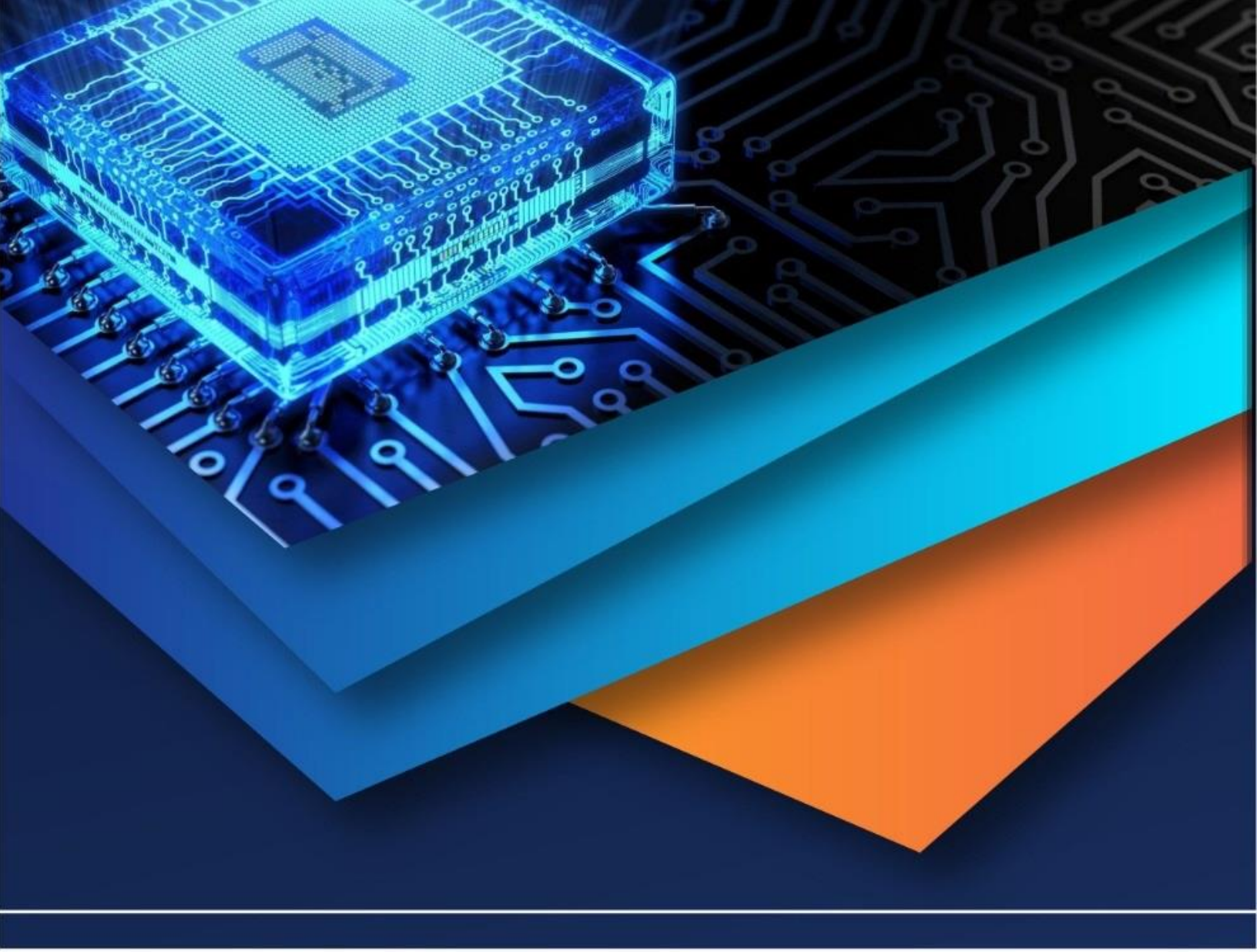
VIII. ACKNOWLEDGMENT

The authors thank the Department of Computer Science and Engineering (Cyber Security), United Institute of Technology, Coimbatore, Tamil Nadu, for providing the computational re- sources and academic support for this work. We acknowledge the Kaggle platform for providing free GPU access (T4) used for all experiments in this paper.

REFERENCES

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in Proc. Int. Conf. Learn. Representations (ICLR), 2014.
[2] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in Proc. Int. Conf. Learn. Representations (ICLR), 2015.

[3] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in Proc. Int. Conf. Learn. Representations (ICLR), 2018.
[4] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in Proc. IEEE Symp. Security and Privacy (S&P), 2017, pp. 39–57.
[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.
[6] O. Russakovsky et al., "ImageNet large scale visual recognition chal- lenge," Int. J. Comput. Vis., vol. 115, no. 3, pp. 211–252, 2015.
[7] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in Proc. IEEE Int. Conf. Computer Vision (ICCV), 2017, pp. 618–626.
[8] D. Meng and H. Chen, "MagNet: A two-pronged defense against adver- sarial examples," in Proc. ACM Conf. Computer and Communications Security (CCS), 2017, pp. 135–147.
[9] P. Samangouei, M. Kabkab, and R. Chellappa, "Defense-GAN: Protect- ing classifiers against adversarial attacks using generative models," in Proc. Int. Conf. Learn. Representations (ICLR), 2018.
[10] C. Guo, M. Rana, M. Cisse, and L. van der Maaten, "Countering adversarial images using input transformations," in Proc. Int. Conf. Learn. Representations (ICLR), 2018.
[11] G. S. Dhillon, K. Azizzadenesheli, Z. C. Lipton, J. Bernstein, J. Kossaifi, Khanna, and A. Anandkumar, "Stochastic activation pruning for robust adversarial defense," in Proc. Int. Conf. Learn. Representations (ICLR), 2018.
[12] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in Proc. Int. Conf. Machine Learning (ICML), 2018, pp. 274–283.
[13] Y. Luo, X. Boix, G. Roig, T. Poggio, and Q. Zhao, "Foveation- based mechanisms alleviate adversarial examples," arXiv preprint arXiv:1511.06292, 2015.
[14] H. Zhang and J. Wang, "Defense against adversarial attacks using feature scattering-based adversarial training," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2019.
[15] I. Figotin, "ImageNet Mini 1000," Kaggle Dataset, 2021. [Online]. Available: https://www.kaggle.com/datasets/ifigotin/imagenetmini-1000
[16] Adwaith R, "Adversarial Attack and Defense on ML Models — Note- book 1," Kaggle, 2024. [Online]. Available: https://www.kaggle.com/code/radwaith/adversarial-attack-and-defense-on-ml-models
[17] Adwaith R, "Adversarial Attack and Defense on ML Models — Note- book 2," Kaggle, 2024. [Online]. Available: https://www.kaggle.com/code/radwaith/adversarial-attack-and-defense-on-ml-models-nb2
[18] Adwaith R, "Generalization Test on Hand-Picked Data," Kag- gle, 2024. [Online]. Available: https://www.kaggle.com/code/radwaith/ generalization-test-on-hand-picked-data



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY