

University of Sheffield

Making Deep Learning Robust to Real-World Adversarial Examples



University of
Sheffield

SHUOHAN XUE

Supervisor: Dr Charith Abhayaratne

Thesis submitted to the University of Sheffield for the award of
the degree of
Doctor of Philosophy

In the

Department of Electronic and Electrical Engineering

11 July 2025

Contents

List of Abbreviations	18
1 Introduction	21
1.1 Motivation	23
1.2 Research Objectives	25
1.3 Main Contributions	25
1.3.1 Research Novelty	26
1.4 Outline	27
1.5 Publications	29
2 Literature Review	31
2.1 Notations and Preliminaries	31
2.1.1 Notations	31
2.1.2 Preliminaries	32
2.1.3 Constructing Adversarial Examples	33
2.1.4 Computing Adversarial Perturbation	33
2.2 Related Work	34
2.2.1 Adversarial Attacks	34
2.2.2 Summary of Adversarial Attacks	39
2.2.3 Review of Defence Methods	40
2.2.4 Summary of Defence Strategies	44
3 Combat AEs with Pre-processing Techniques	46
3.1 General Introduction	46
3.1.1 Motivations and Problem Formulation	48
3.2 Characterisation of DNN Gradients	50
3.2.1 Tested Probability Distributions	50

3.2.2	Goodness-of-Fit Tests	51
3.2.3	Goodness-of-Fit Test Results	54
3.2.4	Leveraging the Insights of DNN Gradients for Adversarial Attack	57
3.2.5	Performance of Gradient Modelling-Based Attack	64
3.2.6	Discussion	70
3.3	DNN Gradients Modelling-Based Defence Strategy	71
3.3.1	Real-World Scenario Application: ImageNet Dataset	74
3.3.2	Robustness Evaluation of the GMBN Defence	75
3.3.3	Discussion	79
3.3.4	Limitations	83
3.4	Real-World Scenario Application: CamVid Dataset	83
3.4.1	Experimental Setup	83
3.4.2	Experimental Results	86
3.4.3	Overall Segmentation Performance	87
3.4.4	Class-wise Performance	89
3.4.5	Summary	90
3.4.6	Limitations and Future Work	91
4	Combat AEs with Robust Features	97
4.1	General Introduction	97
4.2	Feature Robustness Analysis	99
4.2.1	Feature Extraction Methods For Comparison	99
4.2.2	Methodology	105
4.2.3	Experiment Setup	106
4.2.4	Experimental Results	111
4.3	Leveraging Robust Features to Enhance DNNs' Robustness	115
4.3.1	WSN Hybrid ResNet Architecture	116
4.3.2	Experimental Setup	120
4.3.3	Experimental Results	122
4.3.4	Discussion	129
4.4	Real-World Application in Medical Imaging: ROI-Aware ResNet Architecture	131
4.4.1	Motivation and Challenges	133
4.4.2	Methodology	133

4.4.3	Preprocessing of Volumetric Data	134
4.4.4	3D ROI-Aware ResNet Architecture	139
4.4.5	Experimental Setup	141
4.4.6	Experimental Results	142
4.4.7	Discussion	145
5	Combat AEs with AE Detection and Robust Training	149
5.1	General Introduction	149
5.1.1	Other Detection Methods for Comparison	150
5.2	Adversarial Awareness Score for AE Detection	153
5.2.1	Experimental Validation of AAS	155
5.3	AAS-Based Adversarial Detection	159
5.3.1	Detection Framework	159
5.3.2	AAS Sensitivity and Robustness Analysis	160
5.3.3	Implementation Framework	161
5.3.4	Performance Evaluation	164
5.3.5	Summary	171
5.4	AAS-Guided Jacobian Regularisation for Model Robustness	173
5.4.1	Experimental Setup	174
5.4.2	Experimental Results	177
5.4.3	Discussion	181
6	Conclusions	186
6.1	Summary of Contributions	186
6.1.1	Chapter 3	186
6.1.2	Chapter 4	187
6.1.3	Chapter 5	188
6.2	Comparison of Research Contributions	189
6.2.1	Gradients Characterisation vs. Feature Fusion	190
6.2.2	Cross-Domain Applications	191
6.2.3	Detection and Defence Strategies	192
6.2.4	Unified Perspective on Robustness	192
6.3	Limitations and Future Work	192
6.3.1	Current Research Limitations	192
6.3.2	Future Work	193

6.3.3 Ethical and Deployment Considerations	195
Appendices	213
A	214
A.1 Supplementary Material of Chapter.3	214
B	220
B.1 Supplementary Material of Chapter.4	220
C	228
C.1 Supplementary Material of Chapter.5	228

List of Figures

1.1	Demonstration of adversarial attacks on deep neural networks. Both FGSM and PGD successfully fool the ResNet-50 model with imperceptible perturbations but show different characteristics in their attack patterns.	22
1.2	Examples of deep learning applications vulnerable to adversarial attacks: (1) Physical World [1], (2) Autonomous Driving [2], (3) Medical imaging [3].	23
1.3	Conceptual framework illustrating the thesis structure, based on two core hypotheses regarding adversarial example (AE) causes. The framework branches into two pathways: External factors (cyan), attributing vulnerability to exploitable noise patterns, leading to gradient characterisation and Gradient Modelling-Based Neutralisation (GMBN) defence development; and Internal factors (orange), attributing vulnerability to intrinsic DNN limitations. The latter comprises: (1) <i>Robust Features</i> —addressing weak feature learning through hybrid networks and ROI-aware architectures; and (2) <i>Model Deficits</i> —tackling architectural shortcomings via Adversarial Awareness Score validation, detection, and guided regularisation. This structure underpins systematic investigation of complementary defence strategies, with each pathway targeting distinct aspects of adversarial vulnerability. .	27
3.1	Examples of modelling DNN gradients with six different distributions with respect to the predicted labels.	49

3.2	Goodness-of-Fit test results at the 95% and 99% confidence levels. For each subfigure, the left panel presents the overall test results, while the right panel provides a zoomed-in view to enhance clarity and interpretability. All reported test statistics are strictly positive.	56
3.3	Maximum likelihood estimation of the distribution parameters. (In subsequent steps of characterising DNN gradients, $(\mu \pm \sigma)$ from the above results are used to define parameter ranges when sampling random arrays subject to the corresponding distributions.)	57
3.4	A general overview of the proposed method for generating adversarial examples.	59
3.5	A general overview of existing methods for generating adversarial examples.	60
3.6	Degradation of overall classification performance for various attacking methods and DNN models. The figure shows a 5×3 grid where rows represent attack methods (GAMMA, Weibull, Laplacian, FGSM, PGD) and columns represent degradation metrics (Top-1 accuracy, Top-5 accuracy, ground truth confidence). . . .	66
3.7	Attack success rates comparison for the proposed attacks and existing black-box attacks.	69
3.8	Comparison of the proposed Gradient Modelling-Based Neutralisation (GMBN) and existing pre-processing defences. In GMBN, the incoming image is first passed through the DNN to obtain a predicted label and confidence score. A confidence threshold then determines whether a low or high scale perturbation will be applied. The perturbation is generated by sampling from a statistical model of DNN gradients (Gamma, Weibull, or Laplacian) and rescaling it to the chosen strength. This “neutralisation” perturbation is added to the input image, which the DNN model then reclassifies.	73
3.9	Overall classification accuracy improvement against adversarial attacks by existing pre-processing-based methods. This set of graphs visualises the data in Table 3.10.	78

3.10 Overall classification accuracy improvement against adversarial attacks by pre-processing input images with GMBN. This set of graphs visualises the data in Table 3.11.	79
3.11 Examples of using different pre-processing methods for the defence of adversarial examples (FGSM attack).	80
3.12 Examples of using different pre-processing methods for the defence of adversarial examples (PGD attack).	81
3.13 Visualisation of the defence performance of GMBN against adversarial attacks in segmentation tasks. The first three rows show the segmentation maps; the fourth row visualises IoU changes; the bar graph presents statistical data of pixel counts.	92
3.14 Visualisation of the defence performance of GMBN against adversarial attacks in segmentation tasks. The first three rows show the segmentation maps; the fourth row visualises IoU changes; the bar graph presents statistical data of pixel counts.	93
3.15 Confusion matrices for clean image and adversarial example segmentation.	94
3.16 Confusion matrices for GMBN and JPEG compression defence.	95
3.17 Confusion matrices for feature smoothing and colour reduction defence.	96
4.1 Visualisation of different feature maps extracted from a clean image and its corresponding adversarial example. DNN feature maps are visualised using Grad-CAM to highlight class-discriminative regions. SIFT and SURF maps display the ten strongest keypoints, respectively, while HOG maps are shown by superimposing oriented gradient vectors to illustrate local edge orientation patterns.	109
4.2 Visualisation of the scalogram and scattering coefficients for a clean image and its adversarial example, shown for three representative rotation angles out of eight (due to limited space). Scattering coefficients were used for feature extraction, while scalogram coefficients are included to illustrate the full wavelet transform process for academic rigour.	110

4.3	Comparison of the robustness of different feature extraction methods across all models and attacks. (Note: Smaller values indicate better robustness against adversarial attacks)	111
4.4	The architecture of the proposed hybrid network combining ResNet and Wavelet Scattering Network. The model integrates DNN features with WSN features using two fusion strategies: Rescaling+Addition and Depth Concatenation. This combination leverages WSN's stability and ResNet's hierarchical feature learning, aiming to enhance adversarial robustness.	118
4.5	Accuracy of clean images and performance trade-off on 100-class ImageNet Subset and 70% ImageNet (1000 classes) of the proposed Hybrid WSN ResNets	124
4.6	Robustness evaluation against six different adversarial attacks (A1-A6) on 100-class ImageNet subset	125
4.7	Robustness evaluation against six different adversarial attacks (A1-A6) on 70% ImageNet subset(1000-class)	126
4.8	Feature Visualisation Comparison between ResNet-101 and Hybrid WSN-ResNet-101. Set(1): Clean image correctly classified by ResNet-101 and the corresponding Grad-CAM; Set(2): AE misclassified by ResNet-101 and the Grad-CAM of AE; Set(3): Same AE correctly classified by WSN-ResNet-101 and the Grad-CAM showing its attention focus by the hybrid network.	128
4.9	An overview of the Dual-Stage Framework of the proposed method. Stage 1: Data Pre-processing; Stage 2: Feature Extraction and Prediction.	134
4.10	The Complete Pipeline of Volumetric Chest CTs Preprocessing, corresponding to Section.4.4.3 (Note that column (d) is zoomed in, thus leading to an upscaled view compared to the previous three)	135
4.11	Illustration of the pulmonary segmentation process. Rows represent partial slices of the CT volume from top to bottom.	137

4.12	An overview of the extraction procedure of slices with saliency regions. The first and second columns denote partial slices of the given volume from the top to the bottom and their corresponding segmented masks, respectively. The third column denotes the normalised area vector $l'_a()$ in VolumeRange. The parameter t is referred to as the threshold value for the saliency region. Whilst the red dotted line box denotes the slices with saliency, and the blue dotted line boxes denote slices without saliency, and thus are to be discarded. Hence, the slices within the red dotted line box are utilised to generate the volume with saliency, which corresponds to the output V_s as shown in VolumeRange.	138
4.13	RoI-aware 3D ResNet-18 architecture adapted for pulmonary feature extraction.	139
4.14	An Overview of the Baseline Model Performance and Adversarial Robustness of Regular and RoI-Aware 3D ResNet Models	143
4.15	Grad-CAM visualisation of feature activations when classifying adversarial examples. (a) shows that the RoI-aware ResNet maintains consistent activation patterns strictly within the thorax region, demonstrating focused feature learning. In contrast, (b) shows that conventional ResNet without RoI awareness exhibits scattered activation patterns beyond anatomical boundaries. This visualisation directly validates that the ROI-aware mechanism effectively constrains the network's attention to diagnostically relevant regions.	144
5.1	Illustration of Adversarial Awareness Scores (AAS) for Varied Adversarial Attacks	158
5.2	Classification performance metrics versus threshold values. The vertical line at $\tau = 0.68$ indicates the optimal threshold value that maximises overall performance.	163
5.3	Normalised Confusion Matrices for (a) AAS_F ; (b) SQ+SF (scalar Quantisation and Spatial Smoothing); (c) FeaSqr (Feature Squeeze) and (d) FreqVAE AE Detection	165
5.4	ROC curves comparing the detection performance of AAS (Ours), FreqVAE, SQ+SF and FS for adversarial example detection.	166

5.5	Comprehensive adversarial example detection pipeline. From left to right, the respective images show (1) the original input image with ResNet-18 classification results, displaying the predicted class label and confidence score. (2) Frobenius norm comparison between DNN features (extracted from ResNet-18 pool5 layer) and WSN features (from wavelet scattering transform), with their ratio (AAS_F score) used for adversarial detection. (3) AAS-based adversarial likelihood assessment showing the modified sigmoid mapping from AAS_F score to adversarial probability, with classification regions (Clean to Adversarial) and the current image's position marked. (More examples are shown in Figure.C.1 and C.2 in Appendix C due to space limit.)	167
5.6	Comprehensive evaluation of AAS performance across non-adversarial corruptions. Panels show: (a) distribution of detection outcomes, (b) sensitivity analysis, and (c) false positive rate analysis.	170
5.7	AAS-guided Jacobian regularisation mechanism. Left: Comparison of regularisation strength (λ) between conventional constant approach (blue line) and AAS-guided adaptive method (red curve) across AAS_F values. Vertical dashed lines indicate median AAS_F values for clean images and adversarial examples. Right: Three-dimensional surface showing the combined effect of Jacobian norm and AAS_F value on regularisation penalty, demonstrating how the adaptive mechanism applies stronger penalties to adversarial inputs whilst preserving performance on clean samples.	175
5.8	Architecture-specific adversarial robustness profiles under varying attack configurations (A1–A8) for four regularisation strategies. Each subplot presents the performance trajectory of a specific model architecture (ResNet-18, ResNet-50, ResNet-101, and GoogLeNet) across increasingly aggressive attack parameters, where A1–A4 represent FGSM attacks with ϵ ranging from 16/255 to 48/255, and A5–A8 denote PGD attacks with varying iterations and step sizes. The consistent superiority of AAS-guided regularisation (purple diamonds) is evident across all attack types, with particularly pronounced improvements under stronger perturbation regimes.	178

5.9	Comparative evaluation of adversarial robustness enhancement through AAS-guided Jacobian regularisation: (a) Average adversarial robustness across four regularisation strategies for different DNN architectures. The grouped bars represent the mean accuracy across all eight attack configurations (A1–A8) for baseline models, adversarial training (AT), AT with standard Jacobian regularisation (AT+JR), and AT with AAS-guided Jacobian regularisation (AT+AAS). (b) Quantitative robustness improvement achieved by three adversarial training strategies across different architectures. The grouped bars illustrate the mean accuracy gain (in percentage points) compared to undefended baseline models.	179
5.10	Comparison of adversarial perturbations required for successful attacks across ResNet-50 variants. From left to right: perturbations for standard ResNet-50, adversarially trained ResNet-50, and ResNet-50 with AT+AAS defence. Despite achieving the same attack success rate, the perturbation patterns show significant variations across models, suggesting that adversarial training may primarily shift decision boundaries rather than enhance the model’s fundamental robustness.	183
A.1	Examples of the Histograms of DNN Gradients of the Predicted Class with respect to the Input Image	215
A.2	Examples of the Fitting Curves of Using Six Distribution Models to Characterise DNN Gradients	216
A.3	Performance Improvement of GMBN for Non-Adversarially Perturbed Images (Note that the first row shows the model performance without the neutralisation perturbation added to the images, and the second row shows the model performance with the neutralisation perturbation)	217
A.4	Examples of Using Three Distributions to model DNN gradients with respect to the predicted labels	219

B.1 Demonstration of different feature extraction techniques: DNN features visualised using Grad-CAM; Hand-Crafted features SIFT and SURF showing ten strongest points respectively; HOG Features are visualised by superimposing oriented gradients to the image.	220
B.2 Scalogram and scattering coefficients for clean image and AE across right rotation angles (merely three of them are shown due to space limit)	221
B.3 Demonstration of different feature extraction techniques: DNN features visualised using Grad-CAM; Hand-Crafted features SIFT and SURF showing 10 strongest points respectively; HOG Features are visualised by superimposing oriented gradients to the image.	222
B.4 Scalogram and scattering coefficients for clean image and AE across right rotation angles (merely three of them are shown due to space limit)	223
B.5 Demonstration of different feature extraction techniques: DNN features visualised using Grad-CAM; Hand-Crafted features SIFT and SURF showing 10 strongest points respectively; HOG Features are visualised by superimposing oriented gradients to the image.	224
B.6 Scalogram and scattering coefficients for clean image and AE across right rotation angles (merely three of them are shown due to space limit)	225
B.7 Feature Visualisation Comparison between ResNet-101 and Hybrid WSN-ResNet-101-SA. Set(1): Clean image correctly classified by ResNet-101 and the corresponding Grad-CAM; Set(2): AE misclassified by ResNet-101 and the Grad-CAM of AE; Set(3): Same AE correctly classified by WSN-ResNet-101-SA and the Grad-CAM showing its attention focus by the hybrid network.	226

B.8	Feature Visualisation Comparison between ResNet-101 and Hybrid WSN-ResNet-101-SA. Set(1): Clean image correctly classified by ResNet-101 and the corresponding Grad-CAM; Set(2): AE misclassified by ResNet-101 and the Grad-CAM of AE; Set(3): Same AE correctly classified by WSN-ResNet-101-SA and the Grad-CAM showing its attention focus by the hybrid network.	227
C.1	Comprehensive adversarial example detection pipeline. From left to right, the respective images show (1) the original input image with ResNet-18 classification results, displaying the predicted class label and confidence score. (2) Frobenius norm comparison between DNN features (extracted from ResNet-18 pool5 layer) and WSN features (from wavelet scattering transform), with their ratio (AAS_F score) used for adversarial detection. (3) AAS-based adversarial likelihood assessment showing the modified sigmoid mapping from AAS_F score to adversarial probability, with classification regions (Clean to Adversarial) and the current image's position marked.	229
C.2	Comprehensive adversarial example detection pipeline. From left to right, the respective images show (1) the original input image with ResNet-18 classification results, displaying the predicted class label and confidence score. (2) Frobenius norm comparison between DNN features (extracted from ResNet-18 pool5 layer) and WSN features (from wavelet scattering transform), with their ratio (AAS_F score) used for adversarial detection. (3) AAS-based adversarial likelihood assessment showing the modified sigmoid mapping from AAS_F score to adversarial probability, with classification regions (Clean to Adversarial) and the current image's position marked.	230

C.3	Comparison of adversarial perturbations required for successful attacks across ResNet-50 variants. From left to right: perturbations for standard ResNet-50, adversarially trained ResNet-50, and ResNet-50 with AT+AAS defence. Despite achieving the same attack success rate, the perturbation patterns show significant variations across models, suggesting that adversarial training may primarily shift decision boundaries rather than enhance the model’s fundamental robustness.	231
C.4	Comparison of adversarial perturbations required for successful attacks across ResNet-50 variants. From left to right: perturbations for standard ResNet-50, adversarially trained ResNet-50, and ResNet-50 with AT+AAS defence. Despite achieving the same attack success rate, the perturbation patterns show significant variations across models, suggesting that adversarial training may primarily shift decision boundaries rather than enhance the model’s fundamental robustness.	232
C.5	Comparison of adversarial perturbations required for successful attacks across ResNet-50 variants. From left to right: perturbations for standard ResNet-50, adversarially trained ResNet-50, and ResNet-50 with AT+AAS defence. Despite achieving the same attack success rate, the perturbation patterns show significant variations across models, suggesting that adversarial training may primarily shift decision boundaries rather than enhance the model’s fundamental robustness.	233

List of Tables

2.1 Mathematical Notation Used in This Work	32
2.2 A brief comparison of the adversarial attack methods reviewed in this section.	35
2.3 A brief comparison of the adversarial defence methods reviewed in this section.	41
3.1 Distribution models utilised to characterise DNN gradients. . .	51
3.2 Goodness-of-Fit test results (95% confidence interval).	55
3.3 Goodness-of-Fit test results (99% confidence interval).	55
3.4 Parameter settings for characterising gradients in the attack algorithm.	55
3.5 Configuration parameters for different adversarial attacks. . . .	62
3.6 Performance degradation (%) under various attack methods. . .	65
3.7 Success rate (%) of black-box attacks against different DNN models under two settings.	68
3.8 Pre-processing defence methods configuration	74
3.9 Adversarial attacks configuration	75
3.10 Improvement in overall classification performance for varied pre-processing methods and DNN models (FGSM attacks). The data in this table are visualised in Figure 3.9. “QF” refers to the quality factor in JPEG compression.	76
3.11 Improvement in overall classification performance for varied pre-processing methods and DNN models (PGD attacks). The data in this table are visualised in Figure 3.10. “QF” refers to the quality factor in JPEG compression.	77
3.12 Pre-processing Defence Methods Configuration	84
3.13 Adversarial Attacks Configuration for Semantic Segmentation .	86

3.14 Overall Segmentation Performance Evaluation for Single-Class Attack	87
3.15 Overall Segmentation Performance Evaluation for Multi-Class Attack	87
3.16 Class-wise Performance Evaluation — Clean Image, Adversarial Examples, and GMBN Defence	88
3.17 Class-wise Performance Evaluation — Comparison with Other Defence Methods	88
4.1 Parameters for Adversarial Attacks	107
4.2 Parameters for Feature Extraction Methods. For WSN, parameters are specified for two 2-D Morlet [4] filter banks used in feature extraction.	108
4.3 Maximum Element-wise Difference (MED) Under Different Attacks	112
4.4 Normalised Mean Absolute Error (MAE) Under Different Attacks	113
4.5 Implementation Details	121
4.6 Adversarial Attack Parameters Configuration	121
4.7 Performance Comparison on 100-Classes ImageNet Subset - A1 to A3 Attacks	122
4.8 Performance Comparison on 100-Classes ImageNet Subset - A4 to A6 Attacks	122
4.9 Performance Comparison on Full ImageNet Dataset(70%) - A1 to A3 Attacks	123
4.10 Performance Comparison on Full ImageNet Dataset(70%) - A4 to A6 Attacks	123
4.11 Implementation Details	141
4.12 Adversarial Attack Parameters Configuration	142
4.13 Comparison of model accuracy under clean and adversarial conditions. A1-A6 represent different adversarial attacks. Bold values indicate the highest value in each column.	143
5.1 Adversarial Attack Parameters Configuration	156
5.2 Statistical Analysis of Adversarial Awareness Scores. Note: Q1 and Q3 represent the first and third quartiles, respectively. Outliers are counted from a sample size of 10,000 images per category.	157

5.3	Parameters of non-adversarial corruption methods used for AAS evaluation.	161
5.4	Comparative Detection Results on ImageNet Dataset. Note: FS refers to Feature Squeeze.	164
5.5	AAS Sensitivity and Robustness Analysis across Non-Adversarial Corruptions	169
5.6	Adversarial Attack Configuration to Evaluate Robustness of Regularised Model	176
5.7	TOP-1 Adversarial robustness evaluation of DNN models with different regularisation strategies	177
6.1	Overview of technical contributions across methods. Performance metrics are summarised in Table 6.2; here the focus is on mechanisms, computational cost, and application domains. . .	190
6.2	Quantitative performance summary of the proposed methods across datasets. Clean accuracy refers to performance on unperturbed inputs, while adversarial performance reflects recovery under the strongest tested attack.	191
A.1	GMBN Performance for Non-Perturbed Images	218

List of Abbreviations

Abbreviation	Definition
AAS	Adversarial Awareness Score
Adv	Adversarial
AE	Adversarial Example
AI	Artificial Intelligence
ALP	Adversarial Logit Pairing
AT	Adversarial Training
CAP	Community-Acquired Pneumonia
CIFAR	Canadian Institute For Advanced Research [5]
CNN	Convolutional Neural Network
COVID-19	Coronavirus Disease 2019
CPU	Central Processing Unit
CT	Computed Tomography
CV	Computer Vision
C&W	Carlini & Wagner [6]
DICOM	Digital Imaging and Communications in Medicine
DWT	Discrete Wavelet Transform
DNN	Deep Neural Network
FGSM	Fast Gradient Sign Method [7]
FS	Feature Squeezing
GMBN	Gradients Modelling-Based Neutralisation
GPU	Graphics Processing Unit
HOG	Histogram of Oriented Gradients
HU	Hounsfield Units
IoU	Intersection over Union
JPEG	Joint Photographic Experts Group

Abbreviation	Definition
MAE	Mean Absolute Error
MED	Maximum Element-wise Difference
ML	Machine Learning
MNIST	Modified National Institute of Standards and Technology [8]
MRI	Magnetic Resonance Imaging
NLP	Natural Language Processing
PGD	Projected Gradient Descent [9]
RGB	Red, Green, Blue
ROI	Region-of-Interest
SGD	Stochastic Gradient Descent
SIFT	Scale-Invariant Feature Transform
SQ	Scalar Quantisation
SF	Spatial Filtering
SURF	Speeded-Up Robust Features
VGG	Visual Geometry Group [10]
WSN	Wavelet Scattering Network [11]
YOLO	You Only Look Once

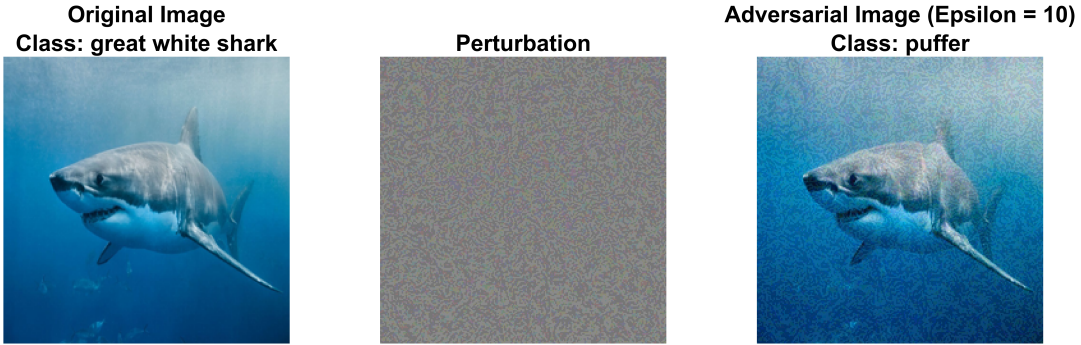
Abstract

Deep neural networks (DNNs) have achieved remarkable success across a range of real-world artificial intelligence applications. However, their vulnerability to adversarial examples poses a critical challenge, undermining their reliability and safety in domains such as autonomous driving, security, and medical imaging. Understanding and mitigating this vulnerability remain difficult due to the inherent complexity of DNN models. This thesis presents three key contributions towards improving adversarial robustness. First, we propose a Gradient Modelling-Based Neutralisation (GMBN) pre-processing defence, which leverages the statistical characteristics of DNN gradients to suppress adversarial perturbations. This approach enhances robust accuracy and consistently outperforms conventional pre-processing methods. Second, we introduce a Hybrid Wavelet Scattering and DNN Feature Network, which fuses stable hand-crafted features with deep features to improve resilience against attacks. This method achieves competitive clean accuracy and substantially better robustness on both natural and medical imaging datasets. Third, we develop the Adversarial Awareness Score (AAS), enabling the detection of adversarial and corrupted inputs and guiding Jacobian regularisation for robust model training. This approach improves detection performance and model generalisation under a range of adversarial scenarios. Experimental results demonstrate that the proposed methods achieve superior robustness against diverse attacks while maintaining strong clean accuracy. The thesis provides actionable insights into the underlying causes of adversarial vulnerability and contributes practical, reproducible strategies for deploying trustworthy deep learning in safety-critical applications.

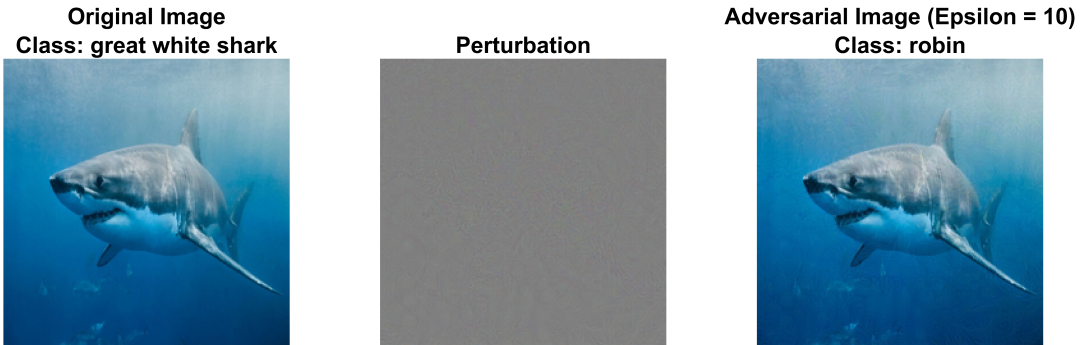
Chapter 1

Introduction

Over the past decade, deep neural networks (DNNs) have achieved unprecedented performance across a wide range of real-world applications [12]. In image classification, models like AlexNet [13], VGGNet [10], ResNet [14], EfficientNet [15], *etc.*, have outperformed human-level accuracy [16, 17, 18] on challenging datasets such as ImageNet [19], laying the foundation for applications ranging from facial recognition [20, 21, 22, 23, 24] to content moderation [25, 26, 27]. In autonomous driving, deep learning-powered perception systems enable vehicles to interpret complex driving environments. Pioneering works, such as You Only Look Once (YOLO) for object detection [28], PointNet for 3D point cloud processing [29], and Tesla’s autopilot systems [30], have significantly advanced self-driving technology. In medical imaging, convolutional neural networks (CNNs) have achieved breakthrough accuracy in tasks like tumour detection, organ segmentation, and disease diagnosis. Notable contributions include U-Net for biomedical image segmentation [31], CheXNet for pneumonia detection [32], and Google’s retinal disease detection framework [33]. Similarly, speech recognition systems have reached new heights of performance with the integration of DNNs. WaveNet [34], Deep Speech [35], and LAS (Listen, Attend and Spell) [36] are examples of architectures that have demonstrated remarkable improvements in recognising human speech. These advances have been critical for applications like virtual assistants, automated transcription, and accessibility tools. Beyond these domains, deep learning continues to foster innovations in robotics [37], natural language processing [38, 39], and other modern real-world artificial intelligence applications.



Fast Gradient Sign Method (FGSM [40]) attack on ResNet-50. The clean image of a Great White Shark (left) is misclassified as “Puffer” (right) after adding imperceptible perturbations ($\epsilon = 10$). The perturbation pattern is shown in the middle, with a constant $\frac{255}{2}$ added to enhance visibility.



Projected Gradient Descent (PGD [9]) attack on ResNet-50. The same clean image (left) is misclassified as “Robin” (right) when attacked with PGD ($\epsilon = 10$, steps=20). The more sophisticated perturbation pattern (middle) results from iterative optimisation, with a constant $\frac{255}{2}$ added to enhance visibility.

Figure 1.1: *Demonstration of adversarial attacks on deep neural networks. Both FGSM and PGD successfully fool the ResNet-50 model with imperceptible perturbations but show different characteristics in their attack patterns.*

However, Szegedy *et al.* [41, 7] revealed that existing deep neural networks (DNNs) can be vulnerable to intentionally fabricated perturbations added to images, even though the perturbations are imperceptible to human eyes. The forged perturbed image that causes the DNN to make a false prediction is called an "Adversarial Example" (see Figure 1.1). This vulnerability of DNNs has been demonstrated to cause various security and privacy threats in many real-world scenarios such as autonomous driving [42, 43, 44, 45, 46], medical imaging [47, 48, 49, 50], automatic speech recognition [51, 52, 53] and many other physical world AI applications [54, 55, 56, 57, 1]. Examples of adversarial

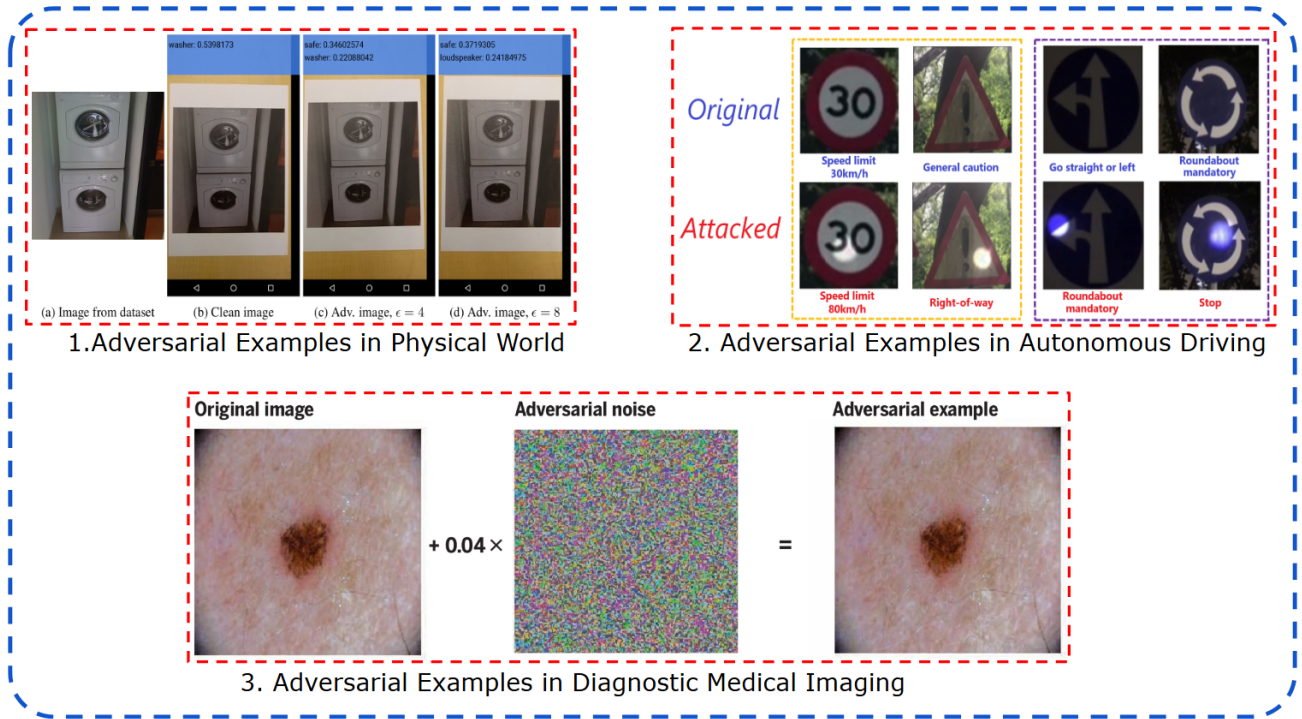


Figure 1.2: Examples of deep learning applications vulnerable to adversarial attacks: (1) Physical World [1], (2) Autonomous Driving [2], (3) Medical imaging [3].

examples in real-world applications are presented in Figure 1.2; autonomous vehicles may misinterpret traffic signs, potentially causing accidents; medical diagnosis systems could make incorrect assessments of patient conditions.

1.1 Motivation

Adversarial examples considerably downgrade trust in AI systems by challenging their reliability and robustness in real-world applications. They expose fundamental vulnerabilities in DNNs, necessitating more secure and interpretable models. By tackling this challenge, researchers aim to enhance the trustworthiness and safety of AI systems whilst pushing the boundaries of machine learning robustness and generalisation, which has triggered combat between adversarial attacks [41, 58, 6, 9, 59, 60, 61, 62, 63] and defence methods [41, 64, 1, 61, 65, 66, 62, 67]. Generally, the attacking methods focus on algorithms that efficiently construct adversarial images to fool neural

networks, whilst the defending methods mainly target building robust DNNs against adversarial examples. Unfortunately, so far, no defence method has proven thoroughly effective and robust against all adversarial attacks, as it is impractical to perform a complete and convincing evaluation of adversarial robustness [68, 69, 70]. Thus, the “arms race” between adversarial attacks and defences remains uncondusive without explicitly identifying the real root cause of the existence of adversarial examples. However, making DNNs robust against adversarial examples presents several critical challenges:

1. **Understanding Adversarial Vulnerability Mechanisms**

One of the major challenges is understanding why DNNs are vulnerable to adversarial examples. Whilst various theories have been proposed, from decision boundary proximity to non-robust feature learning, a comprehensive understanding remains elusive. This understanding is crucial for developing effective defence mechanisms.

2. **Real-world Applicability**

Many existing defence methods work well in controlled settings but fail when applied to real-world scenarios. The challenge lies in developing defence mechanisms that can handle the complexity and variability of real-world adversarial attacks whilst maintaining model performance on clean images.

3. **Computational Scalability and Efficiency**

Robust training methods often require significant computational resources and training time. Developing efficient defence mechanisms that can be practically implemented in real-world applications remains a significant challenge.

4. **Generalisation to Different Attacks**

Whilst adversarial examples have been extensively studied, no established solution fully addresses their impact across different applications and deployment scenarios. Most defence methods are merely effective for specific types of attacks under limited and controlled configurations on small-scale datasets. These methods are often not thoroughly validated on real-world, large-scale datasets, making them impractical for implementation in real-world systems.

1.2 Research Objectives

This thesis addresses the fundamental challenge of adversarial vulnerability in deep neural networks through a multi-layered defence strategy. The primary objectives are:

1. **Theoretical Understanding:** Develop new insights into the root causes of adversarial vulnerability through gradient characterisation and feature robustness analysis.
2. **Comprehensive Defence Framework:** Create complementary defence mechanisms operating at preprocessing, architectural, and training levels of the DNN pipeline.
3. **Practical Robustness Enhancement:** Design scalable solutions that maintain model performance whilst significantly improving adversarial robustness across diverse applications.
4. **Real-World Validation:** Demonstrate effectiveness across varied scenarios, including large-scale image classification and critical medical imaging applications.

1.3 Main Contributions

1. In Chapter 3, we introduce a novel theoretical framework based on DNN gradient characterisation using probability distributions. We demonstrate that adversarial vulnerability stems from DNN sensitivity to specific noise types and propose GMBN (Gradient-Modelling Based Neutralisation). This preprocessing defence generates counter-perturbations to neutralise adversarial effects.
2. In Chapter 4, we present a comprehensive feature robustness analysis revealing that hand-crafted features, particularly Wavelet Scattering Networks (WSN), exhibit superior robustness compared to DNN-learned features. Based on this insight, we develop hybrid WSN-ResNet architectures and ROI-Aware 3D ResNet for medical imaging, both achieving enhanced robustness through robust feature integration.

3. Chapter 5 introduces the Adversarial Awareness Score (AAS) framework for quantifying adversarial likelihood through feature robustness disparity. We develop both a practical detection system and an adaptive training regularisation method that dynamically adjusts based on real-time adversarial awareness during model training.

1.3.1 Research Novelty

The research presented in this thesis advances the field of adversarial machine learning through several novel contributions that distinguish it from existing work:

1. **A New Understanding of DNN Gradients in the Adversarial Context:** Unlike existing methods that focus on decision boundary geometry, our work provides the first comprehensive characterisation of DNN gradients using probability distributions. Our preprocessing-based approach represents the first method to neutralise adversarial perturbations through gradient analysis, eliminating the need for model retraining.
2. **Identifying Robust Hand-Crafted Features to Enhance Adversarial Robustness:** We present the first systematic empirical comparison between hand-crafted and DNN-learned features for adversarial robustness. Our hybrid architectural approach uniquely leverages the complementary strengths of both feature types, with modifications of the DNNs' architectures specifically integrating feature robustness characteristics.
3. **Evaluating Adversarial Awareness and Its Impact on Robustness Enhancement:** Our quantisation-based detection method differs from existing approaches by utilising feature robustness characteristics rather than statistical anomalies. Most significantly, our adversarial awareness score represents the first regularisation technique that adaptively adjusts training based on real-time adversarial detection, contrasting with fixed-schedule methods.

These contributions collectively provide a multi-layered defence strategy that addresses adversarial vulnerability from preprocessing through architectural design to training dynamics, offering a more comprehensive approach

than existing defence methods. Note that this progression allows us to examine adversarial examples from increasingly fundamental perspectives—from external perturbations to feature representations to training dynamics, whilst developing corresponding defensive strategies at each level.

1.4 Outline

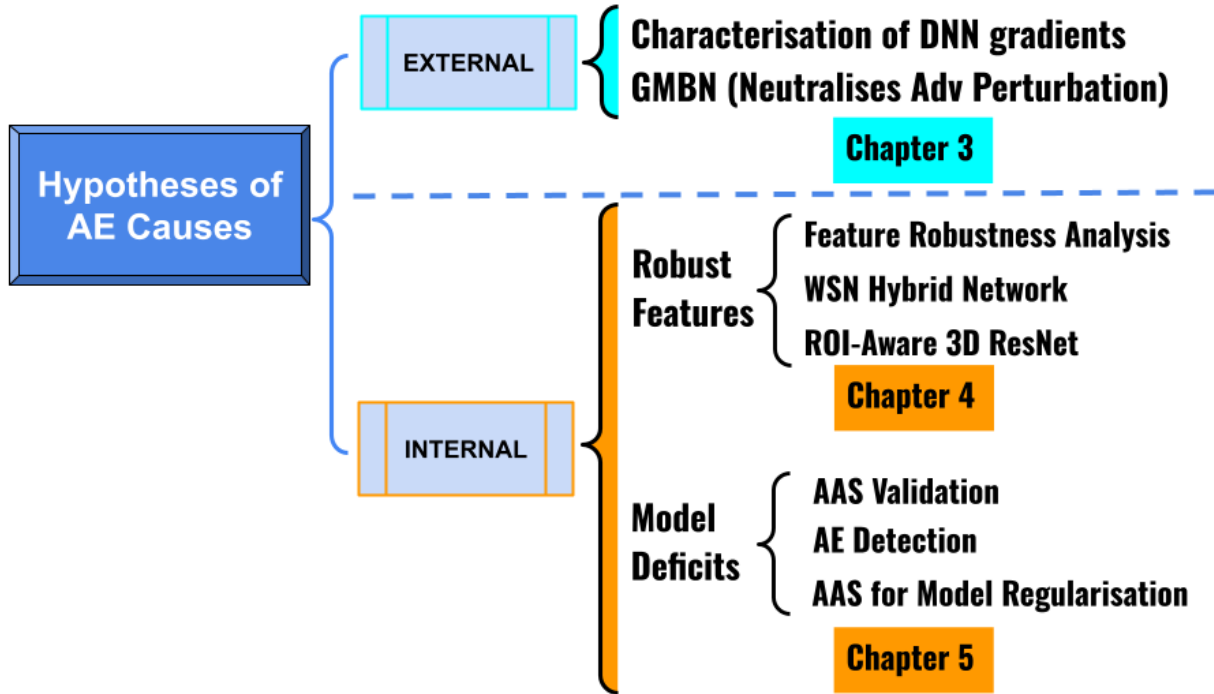


Figure 1.3: Conceptual framework illustrating the thesis structure, based on two core hypotheses regarding adversarial example (AE) causes. The framework branches into two pathways: **External factors** (cyan), attributing vulnerability to exploitable noise patterns, leading to gradient characterisation and Gradient Modelling-Based Neutralisation (GMBN) defence development; and **Internal factors** (orange), attributing vulnerability to intrinsic DNN limitations. The latter comprises: (1) **Robust Features**—addressing weak feature learning through hybrid networks and ROI-aware architectures; and (2) **Model Deficits**—tackling architectural shortcomings via Adversarial Awareness Score validation, detection, and guided regularisation. This structure underpins systematic investigation of complementary defence strategies, with each pathway targeting distinct aspects of adversarial vulnerability.

The thesis is organised as follows:

Chapter 2 introduces notations and preliminaries of this work in Section 2.1 and provides a detailed and well-categorised literature review of the existing work in the context of adversarial examples in deep learning in Section 2.2.

Chapter 3 presents our theoretical framework for gradient characterisation and its applications in developing both attack and defence methodologies. Section 3.1.1 describes our hypotheses on the existence of AEs such that adversarial perturbation is a type of external noise; Section 3.2 experimentally validates our hypotheses; Section 3.3 introduces our preprocessing-based method against AEs regarding our hypotheses and the overall performance of the proposed defence method. Section 3.2.4 evaluates the performance of the proposed defence strategy in different real-world scenarios.

Chapter 4 explores feature robustness and introduces hybrid architectural solutions for enhancing model security. Section 4.2 elaborates our second hypothesis on the existence of AEs and validates it experimentally. Section 4.3 employs the findings and introduces an architectural modification method of the DNNs to improve the models' inherent robustness against adversarial attacks. Section 4.4 introduces another architectural modification method to improve models' inherent robustness, specifically for medical imaging scenarios.

Chapter 5 describes our comprehensive detection and defence framework based on the Adversarial Awareness Score. Section 5.2 introduces a robust feature-based technique to quantify the awareness of adversarial examples, and then it is experimentally validated. Section 5.3 develops a detection method to distinguish adversarial examples from clean images effectively. Section 5.4 introduces a combined approach that uses adversarial awareness as a factor to regularise the models via adversarial training, thus improving the models' inherent robustness against attacks.

Chapter 6 concludes with a summary of our findings in Section 6.1, compares the significance and feasibility of each contribution in Section 6.2, and their implications for future research in Section 6.3.

This PhD research advances both empirical understanding and practical capabilities in defending deep learning systems against adversarial attacks. The methodologies developed herein provide concrete steps towards more secure and reliable deep learning systems for critical AI applications.

It is noteworthy that Figure 1.3 presents the conceptual framework under-

pinning this thesis, illustrating how our investigation of adversarial examples branches from two fundamental hypotheses about their causes: external factors and internal factors, into three technical chapters, each addressing specific aspects through targeted defence strategies. This hierarchical organisation ensures systematic coverage of the adversarial robustness challenge from multiple complementary perspectives.

1.5 Publications

The work presented in this thesis has resulted in the following publications:

Published

1. S. Xue and C. Abhayaratne, "Covid-19 diagnostic using 3D deep transfer learning for classification of volumetric CT scans," in *Proc. IEEE ICASSP*, 2021, pp. 1055-1059. [71]
2. S. Xue and C. Abhayaratne, "Region-of-interest aware 3D ResNet for classification of COVID-19 CT scans," *IEEE Access*, vol. 11, pp. 26883-26895, 2023. [72]

Under Review

3. S. Xue and C. Abhayaratne, "Validating the adversarial robustness of hand-crafted features and their role in defending adversarial examples," *IEEE Access*, submitted June 2025.
4. S. Xue and C. Abhayaratne, "Deep neural networks in adversarial contexts: utilising new insights to neutralise adversarial perturbation," *IEEE Trans. Neural Netw. Learn. Syst.*, submitted July 2025.

The thesis chapters correspond directly to published research contributions. Chapter 3 presents the work from publication 4, whilst Chapter 4 synthesises findings from three publications: 1, 2, and 3. Chapter 5 further develops the hand-crafted features research from publication 3, focusing on robustness and adversarial considerations. Noticeably, at the time of completing this revised

thesis on **11th July 2025**, publications [3](#) and [4](#) were under review with their respective journals. For the most current publication status and any future publications arising from this research, readers are directed to the [author's GitHub repository](#).

Chapter 2

Literature Review

Adversarial robustness has emerged as a critical research area in deep learning due to the vulnerability of deep neural networks (DNNs) to adversarial attacks. These vulnerabilities compromise the reliability of DNNs in safety-critical applications, thereby necessitating robust solutions. This chapter provides a comprehensive review of existing works that focus exclusively on adversarial examples, categorised into adversarial attack mechanisms, defence methods, and theoretical foundations that address the root causes of adversarial vulnerabilities.

2.1 Notations and Preliminaries

This section establishes the mathematical foundations necessary for understanding adversarial examples in deep learning. We begin by defining the notation used throughout this thesis, followed by a formal description of deep neural networks and the process of generating adversarial examples.

2.1.1 Notations

For clarity and precision, the mathematical notation used throughout this thesis is defined in Table [2.1](#).

Table 2.1: *Mathematical Notation Used in This Work*

Notation	Description
$\mathcal{X}$	The set of training data.
$\mathcal{Y}$	Output space (set of possible class labels).
D	The deep neural network.
x	An image from the training data, typically a 3D matrix where pixel values are integers in the range $[0, 255]$.
r	The adversarial perturbation, which is a 3D matrix with the same dimensions as x .
x'	The adversarial example, generated by adding the perturbation r to x .
y	The categorical class label, generated by one-hot decoding the output probability vector.
ϵ	A hyperparameter that controls the scale of the perturbation.
$\mathcal{L}$	Loss function.
$\mathcal{L}(x, y)$	The loss function used to train the deep neural network.
$\nabla_x \mathcal{L}(x, y)$	The gradient of the loss function $\mathcal{L}(x, y)$ with respect to the input image x .
∇_θ	Gradient with respect to parameters θ .

2.1.2 Preliminaries

We begin by explaining the problem from a mathematical perspective. For a given pre-trained DNN, $D_\theta : \mathcal{X} \rightarrow \mathcal{Y}$, the model can be regarded as a mapping from input data $\mathcal{X}$ to an output vector $\mathcal{Y} = \{1, 2, \dots, k\}$, where k is the number of classes. The parameter θ denotes the model's learnable parameters (*i.e.*, convolutional filter sizes, weights, biases, *etc.*), which are typically optimised during training.

The loss function of the model is denoted as $\mathcal{L}(D_\theta(x_i), y_i)$, where x_i is an input image and y_i is the numerical label. Conventionally, training a DNN is framed as an optimisation problem that minimises the average loss over the training set $\{x_i \in \mathcal{X}, y_i \in \mathcal{Y}\}, i = 1, 2, \dots, m$, where m is the total number of training images:

$$\underset{\theta}{\text{minimise}} \quad \frac{1}{m} \sum_{i=1}^m \mathcal{L}(D_\theta(x_i), y_i), \quad (2.1)$$

This optimisation problem is typically solved using gradient descent, where the first-order gradients $\nabla_\theta \mathcal{L}(D_\theta(x_i), y_i)$ are computed and used to update θ in each epoch until convergence. The parameter update rule during training is given by:

$$\theta \leftarrow \theta - \frac{\alpha}{\mathcal{B}} \sum_{i \in \mathcal{B}} \nabla_{\theta} \mathcal{L}(D_{\theta}(x_i), y_i), \quad (2.2)$$

where α is the learning rate (step size) and $\mathcal{B}$ denotes a mini-batch sampled from the training set. The gradient is critical in computing adversarial perturbations, as it significantly influences the loss function.

2.1.3 Constructing Adversarial Examples

The logic behind constructing an adversarial example is straightforward. Consider an input image x , predicted by the DNN D_{θ} with label y . The objective of an adversarial attack is to induce misclassification by increasing the loss associated with the correct class label. A common approach is to add a perturbation δ to x . Let the original loss be $\mathcal{L}_0 = \mathcal{L}(D_{\theta}(x), y)$, and the loss after perturbation be $\mathcal{L}_{\text{new}} = \mathcal{L}(D_{\theta}(x + \delta), y)$. The problem is then framed as:

$$\text{find } \delta \text{ such that } \mathcal{L}_{\text{new}} > \mathcal{L}_0, \quad \|\delta\|_p \leq \epsilon, \quad (2.3)$$

where $\|\delta\|_p$ denotes the ℓ_p norm (typically ℓ_2 or ℓ_{∞}) of the perturbation, and ϵ controls its magnitude. When $\mathcal{L}_{\text{new}}$ is sufficiently greater than $\mathcal{L}_0$, the network's prediction may be diverted to another label $y' \in \mathcal{Y}$. Thus, an adversarial example $x' = x + \delta$ is successfully generated, as shown below:

$$y' = \arg \max(F_{\theta}(x + \delta)); \quad y' \neq y. \quad (2.4)$$

2.1.4 Computing Adversarial Perturbation

As described in Equation (2.2), a DNN is optimised using gradient descent. Therefore, a natural strategy to construct adversarial examples is to reverse this process by ascending the gradient. The perturbation r can be computed as:

$$r = \alpha \text{ sign}(\nabla_{\theta} \mathcal{L}(D_{\theta}(x'), y)), \quad (2.5)$$

where α controls the step size. However, since adversarial perturbations are typically expected to be imperceptible to humans, the perturbation r must be

small. Hence, the constraint $\|r\|_p \leq \epsilon$ is applied before adding r to x .

It is important to note that this process represents the fundamental principle for crafting adversarial examples—namely, maximising the likelihood of misclassification in the target model. Nonetheless, practical adversarial attacks may differ in various aspects, which are discussed in the following section.

2.2 Related Work

This section reviews the existing literature on adversarial attacks and defence methods.

2.2.1 Adversarial Attacks

This subsection comprehensively reviews established adversarial attack methods, analysing their mechanisms and mathematical foundations using the notation defined previously. A brief comparison of the adversarial attack methods reviewed in this section is presented in Table 2.2 to provide a clear overview of their characteristics and computational requirements.

Fast Gradient Sign Method (FGSM) [7, 41]. FGSM is one of the earliest and most fundamental adversarial attacks. It generates adversarial examples in a single step by following the gradient of the loss function. Given an input image x and its corresponding true label y , FGSM constructs the adversarial example x' as:

$$x' = x + \epsilon \cdot \text{sign}(\nabla_x \mathcal{L}(D_\theta(x), y)), \quad (2.6)$$

where ϵ controls the magnitude of the perturbation. This method exploits the linearity of deep networks in high-dimensional spaces by shifting input features in the direction that maximises the loss.

Projected Gradient Descent (PGD) [9]. PGD can be viewed as a multi-step variant of FGSM, performing iterative optimisation. Starting from the original image $x^{(0)} = x$, PGD updates the adversarial example as follows:

Table 2.2: A brief comparison of the adversarial attack methods reviewed in this section.

Attack Method	Knowledge	Mechanism	Computational Overhead
FGSM [41]	White-Box	Gradient	Low
PGD [9]	White-Box	Iterative	High
C&W [6]	White-Box	Gradient	High
DeepFool [58]	White-Box	Iterative	Moderate
MIM [73]	White-Box	Gradient	Moderate
One Pixel [74]	Black-Box	Metaheuristic	High
APGDt [59]	White-Box	Iterative	High
JSMA [60]	White-Box	Greedy	Moderate
Square Attack [75]	Black-Box	Iterative	Moderate
Perceptual [76]	White-Box	Gradient	High
AutoAttack++ [77]	Grey-Box	Gradient	High
Patch-Wise [78]	Black-Box	Greedy	Low
Distribution-Aware [79]	Black-Box	Gradient	High
Frequency-Tuned [80]	White-Box	Gradient	Low
Adaptive Boundary [81]	White-Box	Iterative	High
Transferable Mixture [82]	Black-Box	Gradient	High

Mechanism definitions:

Gradient – uses gradients of the model’s loss function with respect to the input to craft perturbations. *Iterative* – applies multiple steps of gradient-based updates to refine the adversarial example gradually. *Metaheuristic* – uses gradient-free optimisation strategies such as genetic algorithms or evolutionary methods to search for adversarial examples based only on model outputs. These are especially useful in black-box settings. *Greedy* – makes locally optimal changes to the input in a step-by-step manner based on immediate effectiveness, without full optimisation.

$$x^{(t+1)} = \Pi_{x+\mathcal{S}} \left(x^{(t)} + \alpha \cdot \text{sign}(\nabla_x \mathcal{L}(D_\theta(x^{(t)}), y)) \right), \quad (2.7)$$

where $\Pi_{x+\mathcal{S}}$ denotes the projection onto the allowed perturbation set $\mathcal{S}$ (typically an ℓ_∞ ball of radius ϵ centred at x), and α is the step size. PGD performs T iterations to obtain a stronger adversarial example.

Carlini & Wagner (C&W) Attack [6]. The C&W attack formulates adversarial example generation as an optimisation problem:

$$\text{minimise}_\delta \|\delta\|_2 + c \cdot f(x + \delta), \quad (2.8)$$

$$\text{subject to } x + \delta \in [0, 1]^n, \quad (2.9)$$

where $f(x')$ is an objective function ensuring misclassification:

$$f(x') = \max \left(\max_{i \neq t} \{Z(x')_i\} - Z(x')_t, -\kappa \right), \quad (2.10)$$

Here, $Z(x')$ represents the logits (pre-softmax outputs), t is the target class, and κ controls the confidence margin. The parameter c balances the trade-off between perturbation magnitude and attack success.

DeepFool [58]. DeepFool iteratively approximates the decision boundary using a linearisation technique. For a binary classifier, the perturbation at iteration i is:

$$r_i = -\frac{f(x_i)}{\|\nabla f(x_i)\|_2^2} \nabla f(x_i), \quad (2.11)$$

and the update step is:

$$x_{i+1} = x_i + r_i, \quad (2.12)$$

This process is repeated until the perturbed image crosses the decision boundary, resulting in misclassification.

Momentum Iterative Method (MIM) [73]. MIM integrates momentum into the iterative gradient-based attack process to stabilise the update direction and escape poor local optima. The update is defined as:

$$g_{t+1} = \mu \cdot g_t + \frac{\nabla_x \mathcal{L}(D_\theta(x^{(t)}), y)}{\|\nabla_x \mathcal{L}(D_\theta(x^{(t)}), y)\|_1}, \quad (2.13)$$

$$x^{(t+1)} = x^{(t)} + \alpha \cdot \text{sign}(g_{t+1}), \quad (2.14)$$

where μ is the momentum decay factor, and g_t accumulates the velocity vector in the gradient direction.

One Pixel Attack [74]. This attack uniquely modifies only a single pixel using differential evolution to optimise the pixel's location and RGB values. The objective is formulated as:

$$\text{minimise}_{p,c} \mathcal{L}(D_\theta(x_{p,c}), y), \quad (2.15)$$

where p denotes the pixel position and c its colour values. The perturbed image $x_{p,c}$ differs from x only at position p .

Auto-PGD (APGDt) [59]. APGDt extends PGD by incorporating automatic step-size adjustment and momentum restarts. For a target class t , the loss is defined as:

$$\mathcal{L}_{\text{APGDt}}(x') = Z(x')_t - \max_{i \neq t} Z(x')_i, \quad (2.16)$$

and the update step is:

$$x^{(t+1)} = \Pi_{x+\mathcal{S}} \left(x^{(t)} + \gamma_t \cdot \text{sign}(g_t) \right), \quad (2.17)$$

where γ_t is an adaptive step size that evolves based on the loss trajectory:

$$\gamma_{t+1} = \begin{cases} \gamma_t \cdot (1 + \alpha) & \text{if improved in last } w \text{ steps,} \\ \gamma_t \cdot (1 - \alpha) & \text{otherwise.} \end{cases} \quad (2.18)$$

Jacobian-based Saliency Map Attack (JSMA) [60]. JSMA selectively perturbs features by computing the Jacobian matrix of the network. The saliency map $S(x, t)$ for target class t is defined as:

$$S(x, t)[i] = \begin{cases} 0 & \text{if } \frac{\partial D_t}{\partial x_i} < 0 \text{ or } \sum_{j \neq t} \frac{\partial D_j}{\partial x_i} > 0, \\ \frac{\partial D_t}{\partial x_i} \cdot \left| \sum_{j \neq t} \frac{\partial D_j}{\partial x_i} \right| & \text{otherwise} \end{cases}, \quad (2.19)$$

The attack modifies pixels with the highest saliency scores:

$$x'_i = \min(\max(x_i + \theta, 0), 1), \quad (2.20)$$

where θ is the perturbation magnitude.

Square Attack [75]. This is a gradient-free method employing random search with square-shaped updates. It iteratively perturbs square patches:

$$x^{(t+1)} = \Pi_{x+\mathcal{S}}(x^{(t)} + \delta_t \cdot M_t), \quad (2.21)$$

where M_t is a randomly selected square mask, and δ_t is the perturbation magnitude.

Perceptual Attack [76]. This method optimises perturbations in the perceptual CIELAB colour space:

$$\text{minimise}_\delta \|\delta\|_{LAB} \quad \text{subject to } D_\theta(x + \delta) \neq y, \quad (2.22)$$

where $\|\cdot\|_{LAB}$ measures perceptual difference.

AutoAttack++ [77]. An ensemble-based attack that dynamically adjusts its strategy:

$$x' = \arg \max_{\|x'-x\|_p \leq \epsilon} \sum_{i=1}^K w_i \mathcal{L}_i(D_\theta(x'), y), \quad (2.23)$$

where w_i are learnable weights and $\mathcal{L}_i$ are different loss components.

Patch-Wise Attack [78]. This method decomposes the image into patches and optimises each independently:

$$x'_p = x_p + \arg \max_{\|\delta_p\|_\infty \leq \epsilon} \mathcal{L}(D_\theta(x'_{<p} \cup \delta_p \cup x'_{>p}), y), \quad (2.24)$$

where p denotes the patch index.

Distribution-Aware Attack [79]. This method models the distribution of effective perturbations:

$$\delta \sim q_\phi(x, y) = \mathcal{N}(\mu_\phi(x), \sigma_\phi(x)), \quad (2.25)$$

where ϕ are learnable parameters.

Frequency-Tuned Attack [80]. This method operates in the frequency domain

using the Fourier transform:

$$x' = x + \mathcal{F}^{-1}(\alpha \odot \mathcal{F}(\nabla_x \mathcal{L}(D_\theta(x), y))), \quad (2.26)$$

where $\mathcal{F}$ denotes the Fourier transform and α are frequency weights.

Adaptive Boundary Attack [81]. This method iteratively refines perturbations along the decision boundary:

$$x^{(t+1)} = x^{(t)} + \eta_t \cdot \nabla_B(x^{(t)}), \quad (2.27)$$

where ∇_B estimates the normal direction of the decision boundary.

Transferable Mixture Attack [82]. This approach combines multiple perturbation patterns into a weighted mixture:

$$x' = x + \sum_{i=1}^M \lambda_i r_i, \quad (2.28)$$

where r_i are perturbation bases and λ_i are learnable coefficients.

Each of the above methods represents a distinct approach to generating adversarial examples, offering different trade-offs between attack success rate, perturbation sparsity, perceptibility, and computational cost.

2.2.2 Summary of Adversarial Attacks

Based on their access to model gradients, the reviewed adversarial attacks can be categorised into two main classes:

Gradient-Based Attacks. These attacks, also known as white-box attacks, directly utilise the model’s gradient information $\nabla_x \mathcal{L}(D_\theta(x), y)$ to construct adversarial perturbations. They typically aim to solve the following optimisation problem:

$$\text{maximise}_{\|\delta\|_p \leq \epsilon} \mathcal{L}(D_\theta(x + \delta), y). \quad (2.29)$$

This category includes several well-established methods. FGSM employs the sign of the gradient for a single-step perturbation. PGD and APGDt perform iterative optimisation using projected gradients. DeepFool approximates decision boundaries via local gradients. MIM introduces momentum into the gradient update. The C&W attack solves a constrained optimisation problem

using precise gradient information. When full access to the model is available, these attacks typically achieve high success rates, as they can accurately compute the optimal perturbation direction.

Gradient-Estimation Attacks. Also known as black-box attacks, these methods do not rely on direct access to model gradients. Instead, they estimate gradient information using various techniques, such as:

$$\tilde{\nabla}_x \mathcal{L}(D_\theta(x), y) \approx \mathbb{E}_{\delta \sim \mathcal{D}} [\mathcal{L}(D_\theta(x + \delta), y) \cdot \delta]. \quad (2.30)$$

Representative methods in this category include the Square Attack, which performs random search using square-shaped updates; the One Pixel Attack, which leverages differential evolution to modify a single pixel; and the Patch-Wise Attack, which perturbs localised regions to estimate gradients. The Distribution-Aware Attack learns the distribution of effective perturbations, while the Transferable Mixture Attack combines multiple perturbation patterns to enhance transferability. The Frequency-Tuned Attack operates in the frequency domain to estimate impactful perturbations.

Despite differing in gradient accessibility, both categories share the common objective of identifying minimal perturbations that cause misclassification:

$$\delta^* = \arg \min_{\delta} \{ \|\delta\|_p : D_\theta(x + \delta) \neq y, \|\delta\|_p \leq \epsilon \}. \quad (2.31)$$

The key distinction lies in their optimisation strategies: gradient-based attacks directly solve the objective using true gradients, whereas gradient-estimation attacks rely on approximations. The choice between these approaches is often guided by the practical constraints of the attack scenario, such as model accessibility, computational cost, and required success rate.

2.2.3 Review of Defence Methods

This section comprehensively reviews influential defence strategies against adversarial attacks, analysing their mechanisms and mathematical foundations using the established notation. Table 2.3 summarises the defence methods discussed in this section, offering a concise reference for their applicability and overhead.

Table 2.3: *A brief comparison of the adversarial defence methods reviewed in this section.*

Defence Method	Computational Overhead	Effective Against	Mechanism
Min-Max AT [83]	High	White-Box	Adversarial training
TRADES [84]	High	White-Box	Regularisation-based adversarial training
Defensive Distillation [64]	Moderate	White-Box	Knowledge distillation
Feature Denoising [85]	Low	White-Box	Feature purification
Gradient Regularisation [86]	Moderate	White-Box	Gradient smoothing
ALP [87]	High	White-Box	Logit pairing for adversarial robustness
Ensemble Defence [62]	Very High	Both	Model ensembling
Input Preprocessing [88]	Low	White-Box	Input transformation
Randomised Denoising [89]	Low	Both	Randomisation and denoising
ComDefend [90]	Moderate	White-Box	Autoencoder-based input reconstruction
HGD [91]	Moderate	White-Box	High-level representation denoising
PD-Denoiser [92]	Moderate	White-Box	Progressive denoising network
ShieldNet [93]	Moderate	White-Box	Frequency-domain feature masking

Min-Max Adversarial Training [83]. This defence formulates the training process as a min-max optimisation problem, where the model is trained to minimise the loss against the worst-case perturbation:

$$\min_{\theta} \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[\max_{\|\delta\| \leq \epsilon} \mathcal{L}(D_{\theta}(x + \delta), y) \right], \quad (2.32)$$

where the inner maximisation identifies the most effective perturbation within the allowed set, and the outer minimisation adjusts the model parameters to be robust to such perturbations.

TRADES Defence [84]. TRADES introduces a formulation that explicitly captures the trade-off between natural accuracy and robustness:

$$\min_{\theta} \mathbb{E}_{(x,y)} [\mathcal{L}(D_{\theta}(x), y) + \lambda \cdot \text{KL}(p_{\theta}(x) \| p_{\theta}(x + \delta))], \quad (2.33)$$

where KL denotes the Kullback–Leibler divergence between the predictions for clean and adversarial inputs, and λ controls the balance between accuracy and robustness.

Defensive Distillation [64]. Defensive distillation improves robustness by training a student model to mimic the soft output probabilities of a teacher model. The teacher is first trained using a high temperature $T > 1$ in the softmax function:

$$q_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)}, \quad (2.34)$$

where z_i is the logit for class i , and q_i is the softened class probability. The student is then trained on these soft labels using the same temperature, and deployed with $T = 1$ during inference.

The softened labels encode richer inter-class information, encouraging the student to learn smoother decision boundaries. This reduces sensitivity to small perturbations and increases resistance to adversarial attacks.

Feature Denoising [85]. This method incorporates denoising operations directly into the network to suppress adversarial noise in the feature space:

$$\hat{f}(x) = f(x) - \mathcal{D}(f(x)), \quad (2.35)$$

where $f(x)$ denotes the feature map and $\mathcal{D}$ represents a denoising operation, such as non-local means filtering.

Gradient Regularisation [86]. This method penalises large input gradients during training to reduce the model’s sensitivity to adversarial perturbations:

$$\mathcal{L}_{\text{total}}(\theta) = \mathcal{L}(D_{\theta}(x), y) + \lambda \cdot \|\nabla_x \mathcal{L}(D_{\theta}(x), y)\|_2^2, \quad (2.36)$$

thereby encouraging smoother decision boundaries.

Adversarial Logit Pairing (ALP) [87]. ALP enforces similarity between the logits of clean and adversarial examples:

$$\mathcal{L}_{\text{ALP}}(\theta) = \mathcal{L}(D_{\theta}(x), y) + \mathcal{L}(D_{\theta}(x + \delta), y) + \lambda \cdot \|z(x) - z(x + \delta)\|_2^2, \quad (2.37)$$

where $z(x)$ denotes the pre-softmax logits of the model.

Ensemble Defence [62]. This method improves robustness by combining predictions from multiple models:

$$y = \arg \max \left(\frac{1}{p} \sum_{i=1}^p D_i(x) \right), \quad (2.38)$$

where p is the number of models in the ensemble.

Input Preprocessing [88]. This approach applies transformation functions to the input before classification:

$$y = D_{\theta}(g(x)), \quad (2.39)$$

where $g(\cdot)$ represents preprocessing operations such as JPEG compression, feature squeezing, or image denoising.

Randomised Denoising [89]. This defence introduces stochasticity by applying randomised denoising operations at various layers of the network. The clean version of the input is computed as:

$$x_{\text{clean}} = x + \alpha \cdot \text{Uniform}(-1, 1) \odot \text{Denoise}(x), \quad (2.40)$$

where α controls the randomisation strength, and $\text{Denoise}(\cdot)$ denotes a denoising function selected randomly from a set of candidate filters (*e.g.*, non-local means, bilateral filter). Here, $\text{Uniform}(-1, 1)$ represents a matrix of random values sampled independently from a continuous uniform distribution over the interval $[-1, 1]$, with the same dimensions as the output of $\text{Denoise}(x)$. The

operator $\odot$ denotes element-wise multiplication, introducing controlled noise into the denoised features to create stochasticity during inference and increase robustness against adversarial attacks.

ComDefend [90]. This method uses a compression–reconstruction network pair to remove adversarial patterns:

$$x_{\text{compressed}} = f_{\text{comp}}(x), \quad x_{\text{clean}} = f_{\text{recon}}(x_{\text{compressed}}), \quad (2.41)$$

where f_{comp} retains key visual features while discarding noise, and f_{recon} reconstructs the denoised image.

HGD [91]. High-level Guided Denoiser (HGD) leverages feature representations from a pretrained model:

$$\mathcal{L}_{\text{HGD}} = \|g(x_{\text{adv}}) - x\|_2^2 + \lambda \|\phi(g(x_{\text{adv}})) - \phi(x)\|_2^2, \quad (2.42)$$

where $g(\cdot)$ is the denoising network and $\phi(\cdot)$ extracts semantic features, ensuring both pixel-level and perceptual consistency.

PD-Denoiser [92]. This progressive denoising strategy iteratively refines the input:

$$x^{(t+1)} = x^{(t)} + \text{ReLU}(\epsilon - \|f_{\theta}(x^{(t)}) - x^{(t)}\|_{\infty}) \cdot f_{\theta}(x^{(t)}), \quad (2.43)$$

where f_{θ} is a learned denoising module and ϵ controls the threshold for allowable perturbation removal per step.

ShieldNet [93]. ShieldNet applies a wavelet-based multi-scale preprocessing pipeline:

$$x_{\text{clean}} = \mathcal{W}^{-1}(h_{\theta}(\mathcal{W}(x))), \quad (2.44)$$

where $\mathcal{W}$ and $\mathcal{W}^{-1}$ denote wavelet decomposition and reconstruction, and h_{θ} learns to clean multi-resolution coefficients.

2.2.4 Summary of Defence Strategies

The reviewed defence methods can be broadly categorised into four main strategies:

Training-based defences modify the learning objective to improve ro-

bustness. These include min-max adversarial training [83], TRADES [84], MART [94], Feature Scattering [95], and FAT [96].

Architecture modification approaches involve changes to the network structure itself. Representative methods include feature denoising [85], gradient regularisation [86], LAT [97], AdvProp [98], and Resiliency [99].

Preprocessing-based defences apply input transformations to remove adversarial noise. These can be further divided into: Basic transformations: such as input preprocessing [88], PixelDefend [100], and BitDepth [101]; Advanced denoising: such as ComDefend [90], HGD [91], and ShieldNet [93].

Hybrid approaches combine elements from the above strategies. Notable examples include PD-Denoiser [92], DefenseGAN [102], ALP [87], and NRP [103], which integrate iterative refinement with input transformation techniques.

Each defence strategy presents trade-offs between computational cost, clean-image accuracy, and robustness. The optimal choice depends on the expected threat model, available resources, and application-specific requirements.

Chapter 3

Combat AEs with Pre-processing Techniques

3.1 General Introduction

This chapter investigates the efficacy of pre-processing-based techniques in neutralising the effects of adversarial examples (AEs). To date, no widely accepted consensus has been reached on the existence of AEs, and existing studies demonstrate that it is arduous to train a deep neural network (DNN) that is completely robust to adversarial attacks. Accordingly, this study develops a pre-processing-based defence strategy as an external procedure for the input image before it is fed into the DNN model, aiming to mitigate the effects of AEs.

First, a new perspective on DNN gradients is provided, as these gradients represent one of the most critical factors in achieving successful adversarial attacks and can be characterised by specific probabilistic distribution models. It is then experimentally demonstrated that, by utilising this insight into DNN gradients, the adversarial robustness of DNNs can be both degraded and enhanced. Based on these findings, a novel black-box attacking approach is proposed that uses only label information to construct AEs successfully. Subsequently, a pre-processing-based defence strategy is proposed to neutralise the effects of adversarial perturbations. The proposed attacking and defence methods are compared with benchmark attacks and pre-processing-based defence approaches. A systematic performance analysis across varied attack scenarios shows that the proposed attacking method achieves perform-

ance comparable to benchmark attacks, while the proposed defence strategy demonstrates superior results among pre-processing-based defences.

The main contributions of this chapter are as follows:

1. **Characterisation of DNN Gradients:** Comprehensive analysis using multiple goodness-of-fit tests demonstrates that DNN gradients follow specific distribution models. This finding offers a new theoretical foundation for analysing and improving adversarial robustness.
2. **Insights into Gradient-Based Adversarial Robustness:** Leveraging the identified gradient patterns, a label-only black-box attack is proposed that generates adversarial perturbations by exploiting the statistical characterisation of DNN gradients. This attack significantly reduces the robustness of various DNN models, further validating the gradient-based vulnerability hypothesis.
3. **GMBN: A Gradient-Based Defence Strategy:** A novel pre-processing-based defence strategy, termed Gradient-Modelling Based Neutralisation (GMBN), is introduced. GMBN applies gradient distribution properties to generate neutralising perturbations that effectively mitigate adversarial effects. Experiments across multiple datasets demonstrate its competitive defensive capabilities compared with existing pre-processing approaches, while preserving classification accuracy. The method provides universal protection against diverse attack types without requiring prior knowledge of the target model.

The remainder of this chapter is organised as follows: Section 3.1.1 introduces the initial motivation for addressing the problem of AEs and the underlying hypotheses regarding their existence. Section 3.2 experimentally demonstrates the characterisation of DNN gradients using different distribution models. A label-only black-box adversarial attack is then proposed to validate these hypotheses. Section 3.3 explains how these insights are leveraged to mitigate adversarial perturbations using a novel defence strategy, evaluated in both large-scale image classification and semantic segmentation tasks for autonomous driving. Finally, Section 3.3.3 summarises the key contributions of this chapter.

3.1.1 Motivations and Problem Formulation

Despite extensive research on AEs, the underlying cause of DNN vulnerability remains unclear, as discussed in Section 2.2. Rather than merely identifying the inherent opacity of AEs, this work proposes that DNNs are intrinsically vulnerable to a specific type of noise, which is not stochastic but can be deliberately constructed. This noise is designed to maximise the probability of misclassification while minimising its magnitude. Existing studies [7, 41, 1, 6, 104, 105, 106, 58] indicate that DNN gradients are central to the successful construction of AEs. In this context, gradients typically refer to the first-order derivatives of the loss function with respect to the input, which directly indicate how to modify the input to maximise the loss. Consequently, gradient information is vital for inducing misclassification.

As described in Section 2.1, for a pre-trained DNN D_θ with an input image $x \in \mathcal{X}$, a successful adversarial attack must satisfy Eq. (2.4). In white-box attacks, where attackers have complete access to the DNN, solving Eq. (2.3) is straightforward. In practical black-box settings, attackers may query the DNN to obtain outputs (*e.g.*, confidence scores) and iteratively refine adversarial examples. Substitute models trained to replicate the target DNN’s behaviour enable the use of white-box techniques to craft transferable AEs, ultimately solving Eq. (2.3) and exposing the vulnerability of gradient-based optimisation. However, existing defences lack effective mechanisms to prevent Eq. (2.3) from being satisfied. Fundamentally, as long as training relies on gradient descent, adversaries can obtain or approximate gradients $\nabla_\theta \mathcal{L}(D_\theta(x'), y')$ to compute perturbations r that satisfy Eq. (2.4).

Let the set $\mathcal{Z}$ denote an unidentified type of noise, and $\mathcal{A}$ denote the set of adversarial perturbations r . It is hypothesised that $\mathcal{A}$ is a subset of $\mathcal{Z}$, such that

$$r \in \mathcal{A}; \mathcal{A} \subset \mathcal{Z}, \quad (3.1)$$

where $\omega \in \mathcal{Z}$ is the unidentified noise. The objective is to increase $\mathcal{L}(D_\theta(x+\omega), y)$ until $y' \neq y$ while ensuring $\|\delta\|_p \leq \epsilon$. In short, $\omega \in \mathcal{Z}$ is hypothesised to be the actual cause of adversarial examples. Thus, gradient-descent-optimised DNNs may be inherently vulnerable to any $\omega \in \mathcal{Z}$, not just $r \in \mathcal{A}$.

Examples of gradient histograms are shown in Figure 3.1 and Figure A.1 in Appendix A, which illustrate gradients of the target DNN with respect to the

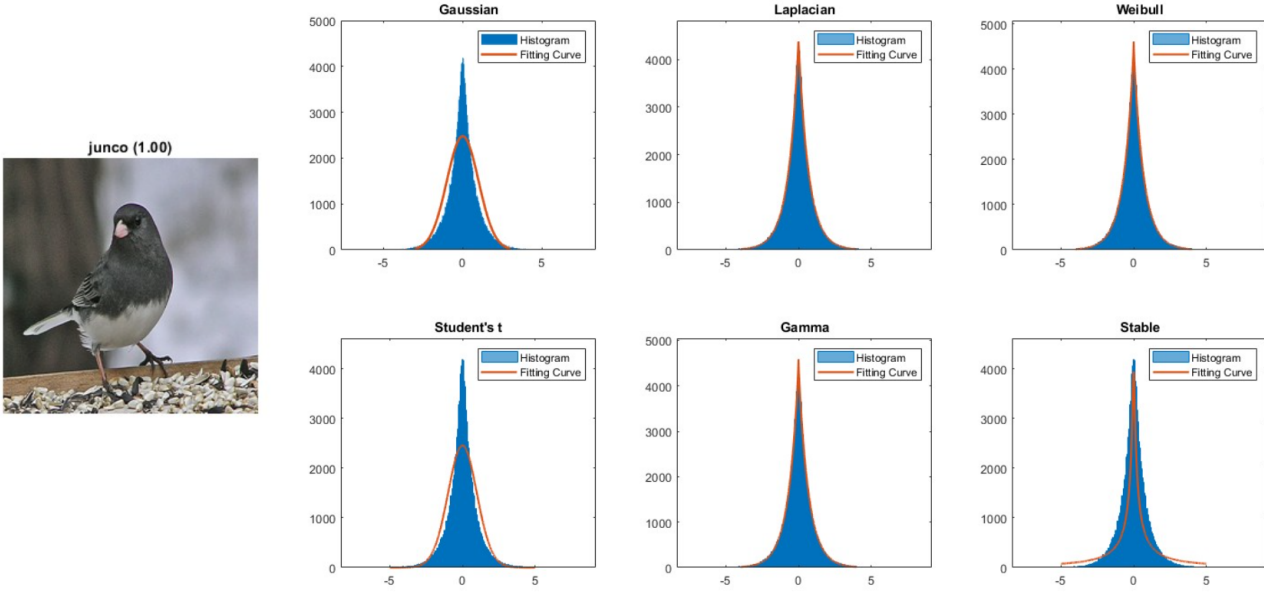


Figure 3.1: Examples of modelling DNN gradients with six different distributions with respect to the predicted labels.

true label. These histograms indicate that DNN gradients follow similar distributions. It is hypothesised that $\omega \subset \mathcal{Z}$ constitutes an intrinsic vulnerability of DNNs, which may explain why AEs are difficult to eliminate entirely.

The histograms show that DNN gradients often exhibit symmetry, heavy tails, and a sharp peak near the mean. The next step, therefore, is to determine the unknown distribution $\mathcal{W}$. Given the hypothesis that $\omega \in \mathcal{Z}$ represents a specific type of structured noise, it is assumed that

$$\omega \sim \mathcal{W}(u), \quad (3.2)$$

where u denotes the parameters of $\mathcal{W}$. Identifying $\mathcal{W}$ requires modelling these values mathematically and using them to attack DNNs. The hypothesis can be validated if the following are experimentally confirmed:

1. **Distribution Model Identification:** For any DNN gradients $\nabla_{\theta} \mathcal{L}(D_{\theta}(x_i), y_i)$, there exists a specific distribution $\mathcal{W}(u)$ such that:

$$\nabla_{\theta} \mathcal{L}(D_{\theta}(x_i), y_i) \sim \mathcal{W}(u), \quad (3.3)$$

where u denotes the parameters of $\mathcal{W}$.

2. **Validation of Adversarial Effects:** Once $\mathcal{W}(u)$ is identified, noise ω is generated using the inverse cumulative distribution function of various candidate distributions. If attacks using ω achieve a misclassification rate comparable to that of $r \in \mathcal{A}$, then it can be concluded that DNNs are intrinsically vulnerable to $\omega \sim \mathcal{W}$.

In summary, this work aims not only to describe the adversarial phenomenon but also to investigate its underlying cause by proposing that DNNs are inherently sensitive to a broader class of structured noise. This noise is not arbitrary but follows statistical patterns derived from gradient behaviour. Therefore, the central research question is whether adversarial vulnerability is an intrinsic consequence of gradient statistics rather than solely a product of explicit optimisation attacks. By modelling $\mathcal{W}$ and generating perturbations sampled from it, it is possible to assess whether such noise alone can produce comparable adversarial effects, thereby reframing AEs as a subset of a broader vulnerability in gradient-descent-based learning.

3.2 Characterisation of DNN Gradients

In this section, the hypotheses described in Section 3.1.1 are experimentally validated. To identify $\mathcal{W}(u)$, various probability distributions are employed for modelling. Existing work [107, 108, 106] typically considers only the standard Gaussian distribution $\mathcal{N}(0, 1)$ to characterise DNN gradients. This study extends the modelling approach by employing six distribution models with varied parameters, followed by Goodness-of-Fit tests to evaluate how well DNN gradients conform to these distributions.

3.2.1 Tested Probability Distributions

Numerous examples of histograms of DNN gradients with respect to the predicted labels are shown in Figure 3.1 and in Figure A.1 in Appendix B. It can be observed that the unidentified distribution $\mathcal{W}(u)$ exhibits characteristics of symmetry, light tails, and positive kurtosis. Accordingly, six probability distribution models are considered for modelling DNN gradients. The Student's

t-distribution [109] is included due to its higher peaks and heavier tails. The Laplacian [110] (double exponential) distribution and Stable [111] distribution are considered for their sharper peaks and heavier tails compared to the Gaussian distribution. The Stable distribution refers to a class of probability distributions and generally lacks a closed-form PDF, instead being defined by its characteristic function:

$$\exp \left(it\delta - |\gamma t|^\alpha \left(1 - i\beta \frac{t}{|t|} \tan \left(\frac{\pi\alpha}{2} \right) \right) \right), \quad (3.4)$$

It is symmetric with sharper peaks and heavier tails only when $\alpha = 0.5$ and $\beta = 0$. Thus, in practical implementation, its parameters are fixed to these values. To achieve sharp peaks and heavy tails, the GAMMA [112] and Weibull [113] distributions are also used. Since the original GAMMA and Weibull distributions are asymmetric and model non-negative data, their PDFs are modified to make them symmetric when fitting histograms of DNN gradients. Details of the distribution models are provided in Table 3.1.

Table 3.1: *Distribution models utilised to characterise DNN gradients.*

Distribution	PDF	Parameters
Gaussian	$f(t) = \frac{1}{\sigma\sqrt{2\pi}} \exp \left[-\frac{1}{2} \left(\frac{t-\mu}{\sigma} \right)^2 \right]$	μ, σ
Student's t	$f(t) = \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\nu\pi} \Gamma(\frac{\nu}{2})} \left(1 + \frac{t^2}{\nu} \right)^{-\frac{\nu+1}{2}}$	ν
GAMMA (sym.)	$f(t) = \frac{\beta^\alpha t ^{\alpha-1} e^{-\beta t }}{\Gamma(\alpha)}$	α, β
Laplacian	$f(t) = \frac{1}{2b} \exp \left(-\frac{ t-\mu }{b} \right)$	μ, b
Weibull (sym.)	$f(t) = \frac{b}{a} \left(\frac{ t }{a} \right)^{b-1} \exp[-(t /a)^b]$	a, b
Stable	See Equation (3.4)	$\alpha, \beta, \gamma, \delta$

3.2.2 Goodness-of-Fit Tests

In this experiment, four Goodness-of-Fit tests are used to statistically validate the hypothesised distribution $\mathcal{W}(u)$: Negative Log-Likelihood [114], Chi-Square Test [115] (χ^2), Kolmogorov–Smirnov test [116] (KS test), and Anderson–Darling test [117] (AD test). Both 95% and 99% confidence intervals

Algorithm 1 Modelling DNN Gradients and Goodness-of-Fit Tests

Input: Dataset $\mathcal{X}$,
Pre-trained DNN D_θ
Images $\{x_1, x_2, x_3, \dots, x_i\} \in \mathcal{X}$
Iteration number $j \leftarrow 1$
Maximum iteration number $\mathcal{I}$
Statistical model $\mathcal{W}$ parameterised by u
Likelihood loss function $\mathcal{L}(\mathcal{W}(u), \nabla_\theta \mathcal{L}())$
for $j = 1 : \mathcal{I}$ **do**
 $y_j \leftarrow \arg \max(D_\theta(x_j))$
 Calculate $\nabla_\theta \mathcal{L}(D_\theta(x_j), y_j)$
 Normalise $\nabla_\theta \mathcal{L}(D_\theta(x_j), y_j)$
 Calculate the mean μ and standard deviation σ of $\nabla_\theta \mathcal{L}(D_\theta(x_j), y_j)$
 Calculate u using μ and σ
 Initialise $\omega \leftarrow \text{icdf}(\mathcal{W}(u))$
 Vectorise ω and $\nabla_\theta \mathcal{L}()$
 Perform Goodness-of-Fit tests on ω and $\nabla_\theta \mathcal{L}()$
end for
 Return Goodness-of-Fit test results

(CIs) are calculated for all tests to ensure rigorous evaluation. The dataset used is a random selection of 100,000 images from ImageNet [19]. The DNNs used for gradient extraction include ResNet-50 [14], GoogLeNet [118], and SqueezeNet [119]. Their mathematical formulations are explained in the following.

Negative Log-Likelihood

Given a set of observed DNN gradients $\{\nabla_1, \nabla_2, \dots, \nabla_i\}$, the Negative Log-Likelihood test evaluates whether these gradients follow a hypothesised probability distribution. For a hypothesised distribution with probability density function (PDF) $p(\mathcal{W} | u)$, where u represents the distribution parameters, the objective function is:

$$\text{NLL}(u) = - \sum_{i=1}^n \log p(\nabla_i | u), \quad (3.5)$$

where lower NLL values indicate a better fit to the hypothesised distribution.

Chi-Square Test

The Chi-Square test (χ^2) measures the significance of the difference between expected and observed frequencies. In this experiment, it is defined as:

$$\chi^2 = \sum_{i=1}^n \frac{(O_i - E_i)^2}{E_i}, \quad (3.6)$$

where O_i represents the observed frequency of gradients in bin i , and $E_i = n p(\nabla_i | u)$ is the expected frequency under the hypothesised distribution with parameters u . The test statistic follows a chi-square distribution with $(n - k)$ degrees of freedom, where k is the number of estimated parameters in the hypothesised distribution. A smaller χ^2 value indicates a better fit between the observed gradients and the hypothesised distribution.

Kolmogorov–Smirnov Test

The Kolmogorov–Smirnov test (KS test) is a nonparametric test comparing the empirical cumulative distribution function (CDF) of the observed gradients with a hypothesised probability distribution. The test statistic is the maximum absolute difference between the empirical CDF and the hypothesised CDF:

$$\mathcal{D}_{\text{KS}} = \max(|f_n(\nabla_i) - f(\nabla_i | u)|), \quad (3.7)$$

where $f_n(x)$ is the empirical CDF of the observed gradients and $f(x | u)$ is the CDF of the hypothesised distribution with parameters u . A smaller $\mathcal{D}_{\text{KS}}$ value indicates a better fit.

Anderson–Darling Test

The Anderson–Darling test assesses whether a given sample comes from a specified distribution and is particularly sensitive to the tails. The AD test statistic is:

$$A^2 = -n - \frac{1}{n} \sum_{i=1}^n (2i - 1) [\ln f(\nabla_i | u) + \ln(1 - f(\nabla_{n-i+1} | u))], \quad (3.8)$$

where $f(\nabla_i | u)$ is the CDF of the hypothesised distribution $\mathcal{W}$ with parameters u , and the gradients $\{\nabla_i\}$ are arranged in ascending order. A smaller A^2 value indicates a better fit.

3.2.3 Goodness-of-Fit Test Results

Tables 3.2 and 3.3 present the results of the Goodness-of-Fit tests at 95% and 99% CIs, respectively. Lower values correspond to better fitting performance; results are shown as absolute values for clarity. To maintain experimental rigour, the mean and standard deviation are reported for each test category across the dataset of 100,000 images. For clearer insights, error-bar graphs are provided in Figure 3.2. Note that each subfigure in Figure 3.2 contains two panels: the left shows overall performance for a test category, and the right is a zoomed-in view of the best-fit models.

For the Negative Log-Likelihood test, the GAMMA distribution achieves the best-fitting performance with $4.87 (\pm 0.28)$ at 95% CI and $6.67 (\pm 0.54)$ at 99% CI. Laplacian and Weibull demonstrate comparable performance at $4.89 (\pm 0.27)$ and $4.87 (\pm 0.28)$, respectively, at 95% CI. However, at 99% CI the values are $9.69 (\pm 0.47)$ and $7.37 (\pm 0.48)$, respectively, as illustrated in Figure 3.2 (a) and (b). For the Chi-Square test, Laplacian, GAMMA, Weibull, and Stable yield overall good fitting performance, though GAMMA and Weibull achieve lower mean values; Figure 3.2 (c) and (d) show overlap among these models at both 95% and 99% CIs. For the KS test, the best-fit models are GAMMA and Weibull. Although Laplacian yields comparable performance of $0.026 (\pm 0.014)$ and $0.045 (\pm 0.024)$ at 95% and 99% CIs, respectively, it does not achieve adequate overlap with GAMMA and Weibull, as indicated in Figure 3.2 (e) and (f). For the AD test, the results fail to converge for the Gaussian and Student's t distributions; thus, the values are not shown in Tables 3.2 and 3.3. Laplacian, GAMMA, and Weibull exhibit stronger overall fitting performance than the Stable distribution at both CIs. It is worth noting that Laplacian has larger means and standard deviations, as shown in Figure 3.2 (g) and (h); nevertheless, due to complete overlap, it is considered as good as GAMMA and Weibull here.

Based on the above, Laplacian, GAMMA, and Weibull distributions yield the best modelling performance for DNN gradients, whereas Gaussian and

Student's t fail to fit adequately. The Stable distribution exhibits acceptable modelling performance but does not match the best three. Therefore, the experiments demonstrate that DNN gradients $\nabla_{\theta}\mathcal{L}(D_{\theta}(x_i), y_i)$ are subject to a distribution $\mathcal{W}(u)$ that is most plausibly Laplacian, GAMMA, or Weibull, as indicated by our Goodness-of-Fit tests.

Table 3.2: Goodness-of-Fit test results (95% confidence interval).

Distribution	95% Confidence Interval			
	NLL ($\times 10^4$)	χ^2 ($\times 10^3$)	KS	AD
Gaussian	21.33 (± 1.87)	181.83 (± 116.54)	0.062 (± 0.017)	N/A
Student's t	17.12 (± 1.85)	148.37 (± 114.56)	0.052 (± 0.016)	N/A
Laplacian	4.89 (± 0.27)	3.43 (± 6.50)	0.026 (± 0.014)	0.13 (± 0.16)
GAMMA	4.87 (± 0.28)	1.28 (± 1.26)	0.010 (± 0.005)	0.021 (± 0.026)
Weibull	4.87 (± 0.28)	1.65 (± 2.40)	0.012 (± 0.007)	0.026 (± 0.045)
Stable	7.22 (± 0.26)	6.18 (± 7.32)	0.18 (± 0.001)	12.08 (± 0.53)

Table 3.3: Goodness-of-Fit test results (99% confidence interval).

Distribution	99% Confidence Interval			
	NLL ($\times 10^4$)	χ^2 ($\times 10^3$)	KS	AD
Gaussian	34.76 (± 3.67)	347.30 (± 217.93)	0.14 (± 0.029)	N/A
Student's t	28.59 (± 2.86)	434.53 (± 188.72)	0.11 (± 0.032)	N/A
Laplacian	9.69 (± 0.47)	7.69 (± 15.61)	0.045 (± 0.024)	0.27 (± 0.37)
GAMMA	6.67 (± 0.54)	2.48 (± 3.19)	0.015 (± 0.013)	0.052 (± 0.078)
Weibull	7.37 (± 0.48)	4.32 (± 4.90)	0.026 (± 0.012)	0.067 (± 0.15)
Stable	10.56 (± 0.49)	19.37 (± 17.08)	0.28 (± 0.003)	34.68 (± 1.16)

Table 3.4: Parameter settings for characterising gradients in the attack algorithm.

GAMMA	$\alpha \in [1.04, 1.13], \beta \in [0.65, 0.67]$
WEIBULL	$a \in [0.72, 0.77], b \in [1.0, 1.1]$
LAPLACIAN	$\mu \in [-0.5, 0.5], b \in [0.70, 0.73]$

Table 3.4 and Figure 3.3 demonstrate the **maximum likelihood estimation** (MLE) [120] of the distribution parameters in the Goodness-of-Fit tests. This step determines the parameters of the distribution models that best fit the

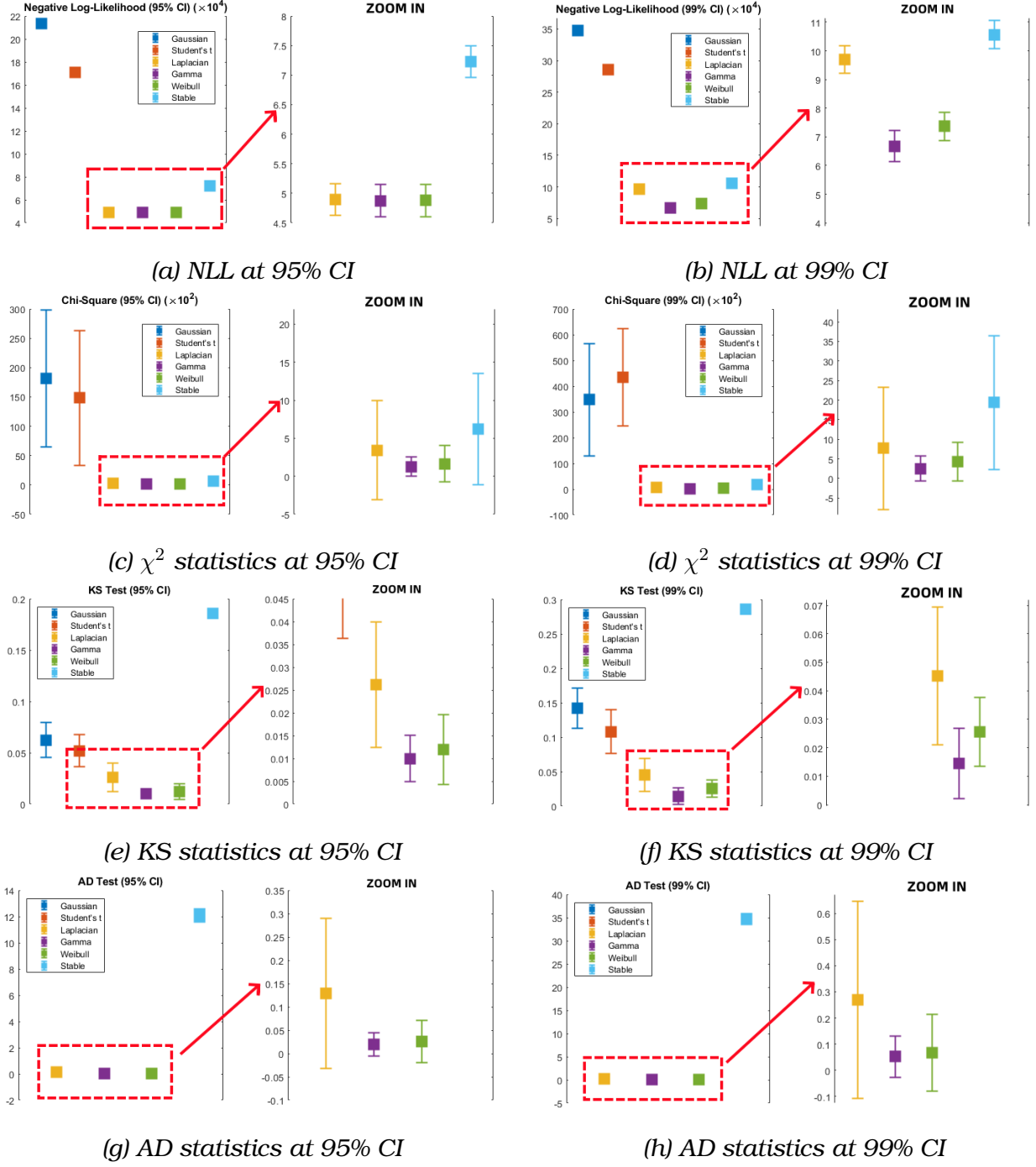


Figure 3.2: Goodness-of-Fit test results at the 95% and 99% confidence levels. For each subfigure, the left panel presents the overall test results, while the right panel provides a zoomed-in view to enhance clarity and interpretability. All reported test statistics are strictly positive.

DNN gradients. These results are then used to characterise DNN gradients in subsequent implementations of attack and defence algorithms.

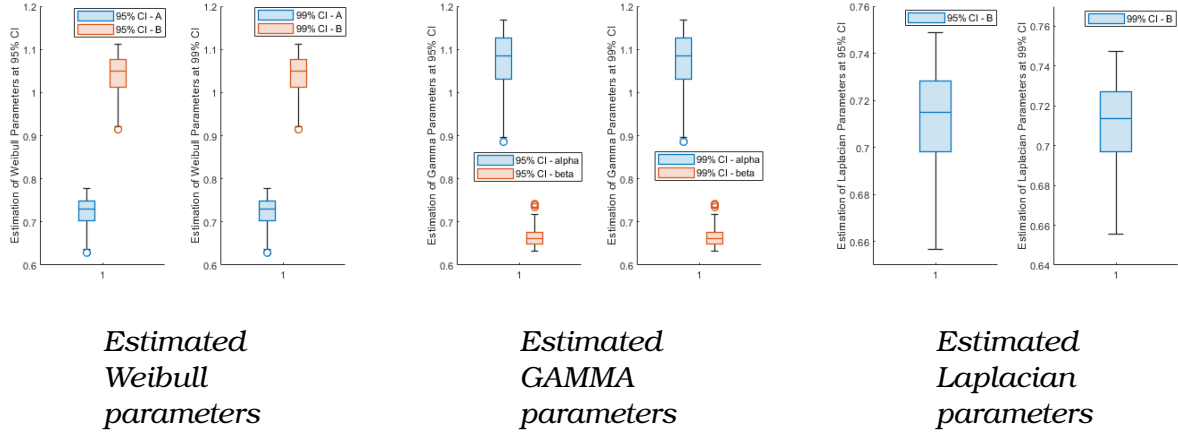


Figure 3.3: Maximum likelihood estimation of the distribution parameters. (In subsequent steps of characterising DNN gradients, $(\mu \pm \sigma)$ from the above results are used to define parameter ranges when sampling random arrays subject to the corresponding distributions.)

Summary

This section validates the first hypothesis proposed in Section 3.1.1: DNN gradients can be characterised by specific probability distributions. The experimental results show that Laplacian, GAMMA, and Weibull distributions provide the best fit for DNN gradients, while Gaussian and Student’s t distributions fail to model the gradients effectively. These findings provide a foundation for understanding the inherent characteristics of DNN gradients and their role in adversarial vulnerability. In the next section, the hypothesis that modelling DNN gradients can be utilised to degrade the adversarial robustness of DNN models is validated (see Section 3.1.1).

3.2.4 Leveraging the Insights of DNN Gradients for Adversarial Attack

As discussed in Section 3.2, DNN gradients $\nabla_{\theta} \mathcal{L}(D_{\theta}(x_i), y_i)$ can be modelled by a specific distribution that is experimentally demonstrated to be Laplacian, GAMMA, or Weibull. In this section, it is demonstrated that these insights can be utilised to degrade the adversarial robustness of DNNs. A black-box adversarial attack algorithm with only label information is proposed. Randomly generated $\omega \sim \mathcal{W}(u)$ is used to replace DNN gradients $\nabla_{\theta} \mathcal{L}(D_{\theta}(x_i), y_i)$ when

performing adversarial attacks. That is, the attacker is no longer required to access or train a substitute DNN to estimate gradient information. Instead, the gradient information is modelled by specific distribution models; in other words, the targeted DNN is attacked with a completely external random array of numbers $\omega \sim \mathcal{W}$. The approach is elaborated in Algorithm 2. To provide an interpretable overview of the methodological framework, a diagram of the proposed attack is shown in Figure 3.4, and the method is compared with existing approaches in Figure 3.5.

Algorithm 2 Label-Only Gradient Modelling Attack Algorithm

Input: Input image $x \in \mathcal{X}$,
Pre-trained DNN D_θ
Predicted label $y \leftarrow \arg \max(D_\theta(x))$
Confidence score $c \leftarrow D_\theta(x, y)$
Perturbation constraint $\epsilon \in [32, 64]$
Iteration number $j \leftarrow 1$
Maximum iteration number $\mathcal{I}$
Pre-defined distribution model $\mathcal{W}(u)$

STEP-1
while $c \leq c' \wedge j \leq \mathcal{I}$ **do**
 Gradient modelling $\omega \leftarrow \text{icdf}(\mathcal{W}(u))$
 $r \leftarrow \text{rescale } \omega \text{ into } [-\epsilon/4, \epsilon/4]$
 Calculate the new confidence score $c' \leftarrow D_\theta(x + r, y)$
 $j \leftarrow j + 1$
end while

STEP-2
while $y' = y \wedge j \leq \mathcal{I}$ **do**
 $r \leftarrow r + \text{round}(\frac{2j}{\mathcal{I}} + 1) r \text{ sign}(\omega)$
 $\text{clipped}(r) \leftarrow \max(-\epsilon, \min(\epsilon, r))$
 $y' \leftarrow \arg \max(D_\theta(x + r))$
 $j \leftarrow j + 1$
end while

Return: r

Adversarial example: $x' \leftarrow x + r$

Otherwise: the attack fails

An input image x from the dataset $\mathcal{X}$ is predicted by the targeted DNN D_θ with label y and the corresponding confidence score c . A maximum iteration number $\mathcal{I}$ is pre-determined to prevent the algorithm from exceeding a reasonable runtime. The first step synthesises an array of gradients that has a high

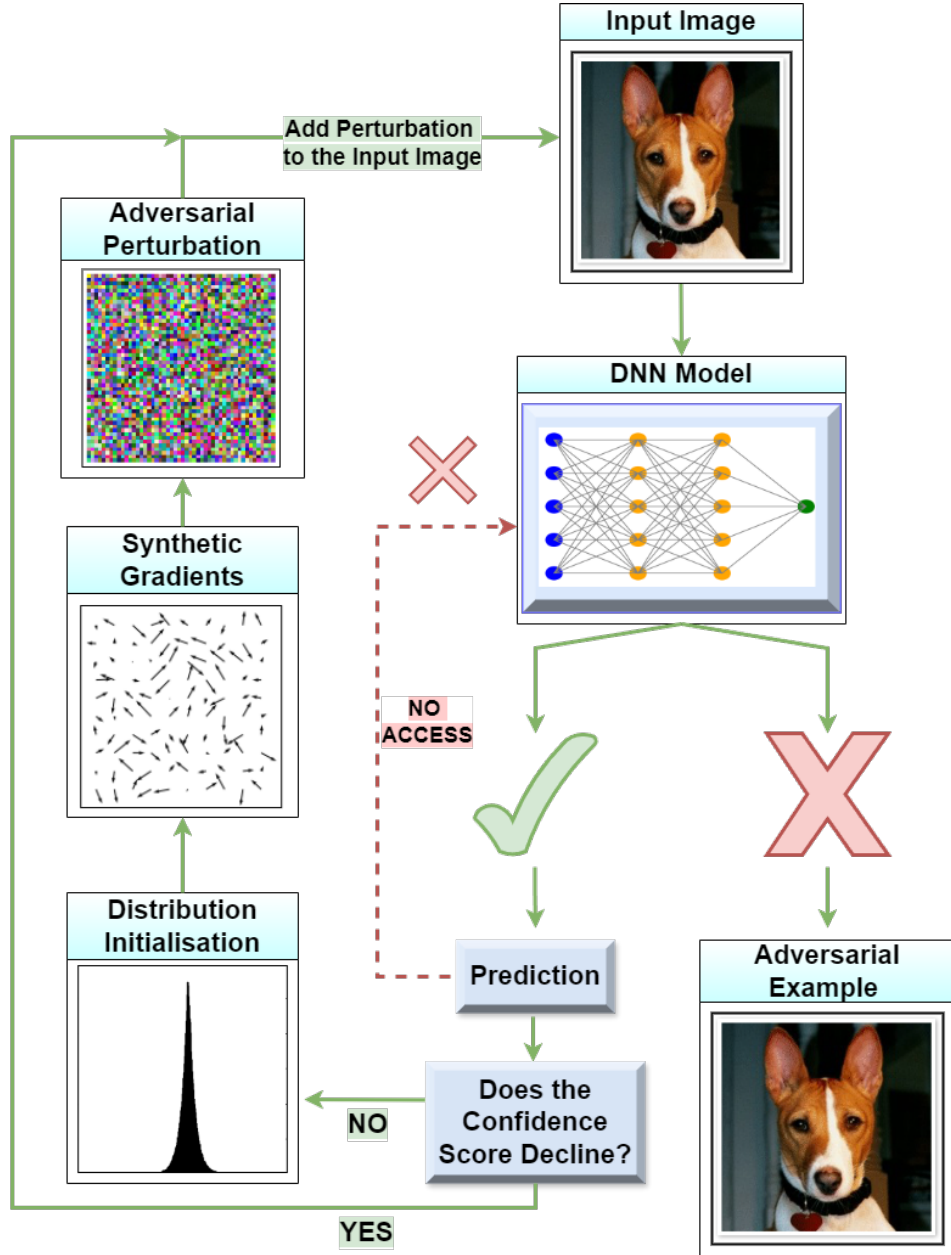


Figure 3.4: A general overview of the proposed method for generating adversarial examples.

chance of causing a larger loss: ω is initialised using the ICDF of $\mathcal{W}(u)$ with the same size as the input image x . Note that the parameters u for $\mathcal{W}$ can no longer be calculated using the mean and standard deviation of the gradients, since the gradients are inaccessible; instead, the parameters u are treated as critical components of the ablation study. The perturbation r is then computed

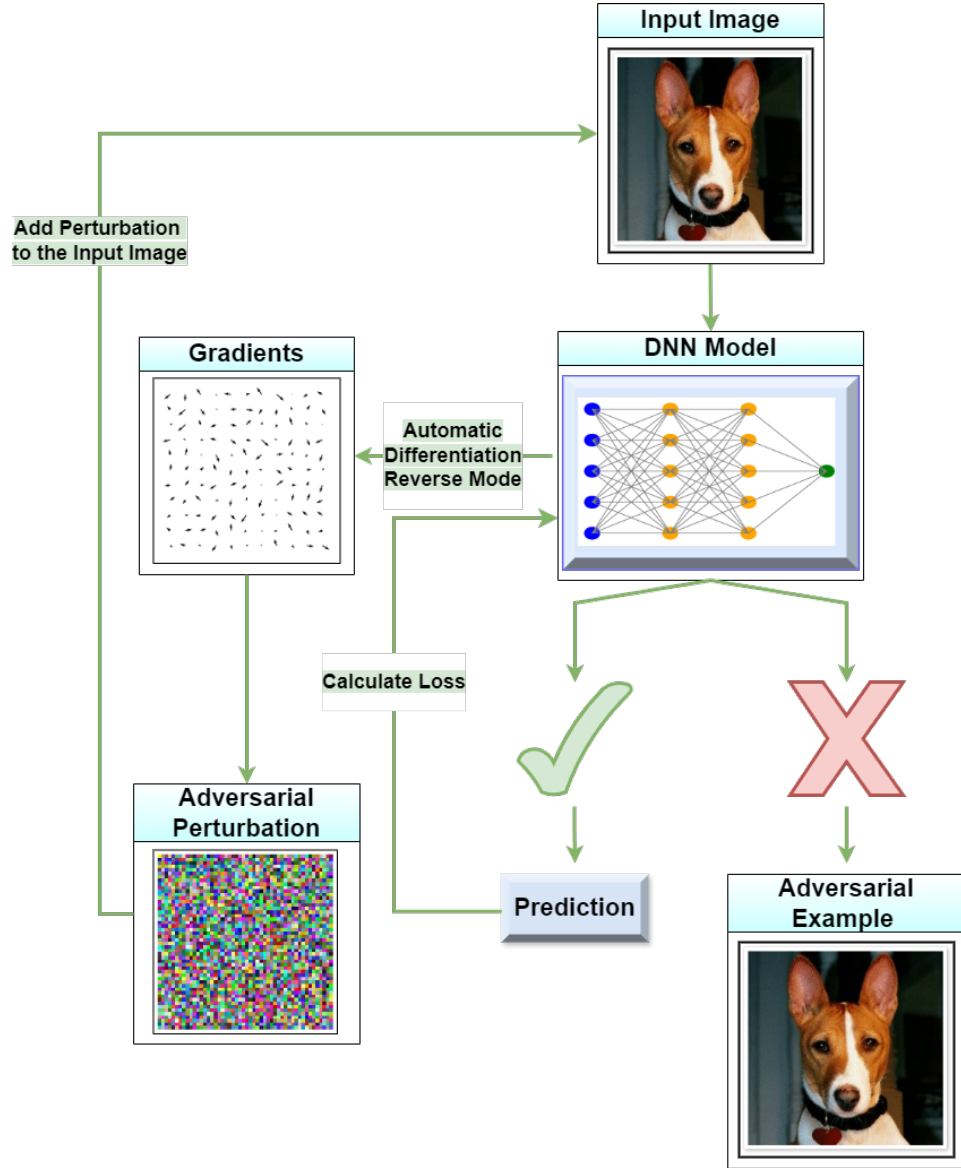


Figure 3.5: A general overview of existing methods for generating adversarial examples.

by rescaling ω into $[-\epsilon/4, \epsilon/4]$, where ϵ is the perturbation constraint. This rescaling sets an appropriately large perturbation level, thereby degrading the confidence score c' . Note that $D_\theta(x + r, y)$ is a constant representing the confidence score with respect to the original prediction y .

The second step constructs the adversarial perturbation with the synthetic gradients ω . In each iteration, the perturbation r is enhanced by the rounded value $(\frac{2j}{T} + 1)$, which restrains the perturbation while j is small and increases

it when j is large. r is then clipped within $[-\epsilon, \epsilon]$. The algorithm outputs the adversarial example x' once misclassification occurs ($y' \neq y$). Otherwise, if j exceeds $\mathcal{I}$ while $y' = y$, the attack fails. The experimental setups are explained in Section 3.2.4, and the corresponding results are presented in Section 3.2.5.

Experimental Setups of the Attack Algorithm

As discussed in Section 3.2, the GAMMA, Weibull, and Laplacian models demonstrate the best fitting performance with the DNN gradients. In the experiments, these three distributions are respectively used to synthesise gradients for adversarial perturbation construction to test which achieves the best attacking performance (see Algorithm 2). To reduce complexity in practice, an ensemble variant randomly uses one of the three to initialise the gradients in Step 1. Experimentally acquired values from the Goodness-of-Fit tests (Table 3.4 and Figure 3.3) are used to initialise $\mathcal{W}(u)$ in Step 1 of Algorithm 2.

Dataset

The ImageNet dataset [19] and publicly available pre-trained DNNs are utilised: VGG-19 [10], ResNet-50 [14], GoogLeNet [118], SqueezeNet [119], InceptionV3 [121], and NASNet-Large [122]. When implementing and testing the attack and defence algorithms, 100 000 images are randomly selected from ImageNet that are predicted by the target DNN with high confidence ($\max(D_\theta(x_i, y_i)) \geq 0.50$). The selected images are from various classes to ensure experimental rigour and prevent bias.

Attack Configuration

Table 3.5 illustrates the configurations used for comparison. FGSM and PGD represent white-box attacks, and ZOO [105] and Square Attack [75] represent black-box attacks. The primary aim is not to demonstrate superior attack performance but to validate the hypothesis that DNN gradient modelling can effectively downgrade model robustness. Showing that the gradient-modelling attacks achieve comparable performance to established methods provides evidence that adversarial vulnerability is intrinsically linked to gradient characteristics, thereby validating the second hypothesis in Section 3.1.1.

Table 3.5: Configuration parameters for different adversarial attacks.

Attack Method	Parameter	Setting
FGSM [7]	ϵ	16, 32
PGD [9]	ϵ	16
	Number of iterations	25, 50
	Step size (α)	0.5
ZOO [105]	Binary search steps	10
	Learning rate	0.01
Square [75]	Initial probability	0.1

Notes: Black-box attacks use two additional settings:

Setting-1: 500 queries, 30 iterations.

Setting-2: 1000 queries, 50 iterations.

Evaluation Metrics

The performance of each method is evaluated using the following metrics:

- **Top-1 classification accuracy:** measures how often the model's most confident prediction (i.e., the class with the highest probability) matches the true label, defined as:

$$\text{Top-1 Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}. \quad (3.9)$$

- **Top-5 classification accuracy:** measures whether the true label is among the five classes with the highest predicted probabilities, defined as:

$$\text{Top-5 Accuracy} = \frac{\text{Number of Predictions Where True Label in Top-5}}{\text{Total Number of Predictions}}. \quad (3.10)$$

- **Prediction confidence score:** the probability assigned by the DNN to a specific predicted class, indicating how confident the model is in its prediction. For a given DNN D and input x , if the predicted class is y , the confidence score is:

$$\text{Confidence Score} = D(x, y), \quad y = \arg \max(D(x)), \quad (3.11)$$

where $D(x, y)$ is the output softmax probability of the predicted class y .

Computational Resources and Reproducibility

To facilitate the reproducibility of all experiments presented in this thesis, the computational environment is documented in this subsection unless otherwise specified in the relevant experimental setup descriptions. All experiments reported in Chapters 3, 4, and 5 were conducted using the hardware, software, and libraries listed below. Note that the experiments were implemented interchangeably in MATLAB and Python, depending on the methodological requirements and computational demands. Upon the formal conferral of the PhD degree, the source code associated with this research will be made publicly available via the author's [GitHub repository](#). Readers are invited to consult this repository for the most up-to-date and maintained version of the implementation.

1. Hardware Configuration

- **GPU:** NVIDIA RTX 3090Ti, driver version 535.104.05
- **CPU:** AMD Ryzen 9 3900X @ 3.80 GHz (12 cores / 24 threads)

2. Software Environment

- **Operating System:** Ubuntu 22.04 LTS (64-bit)
- **CUDA Toolkit:** 12.1
- **cuDNN:** 8.9.0
- **Python:** 3.10.12
- **MATLAB:** R2020b–R2024b

3. Python Libraries for Deep Learning

- **PyTorch:** 2.2.0
- **torchvision:** 0.17.0
- **torchattacks:** 3.5.1

4. MATLAB Toolboxes Utilised

- Deep Learning Toolbox
- Wavelet Toolbox

- Image Processing Toolbox
- Statistics Toolbox
- Signal Processing Toolbox
- Medical Imaging Toolbox

5. Reproducibility Considerations

- All experiments were executed on a dedicated GPU workstation to maintain consistent computational resources.
- Random seeds were fixed where applicable (`seed(0)` in Python and `rng(42)` in MATLAB) to minimise stochastic variation in model training.

3.2.5 Performance of Gradient Modelling-Based Attack

Comparison with Gradient-Based White-Box Attacks

Table 3.6 presents the performance of the proposed gradient-modelling attacks (using pre-defined GAMMA, Weibull, and Laplacian models) compared with benchmark attacks (FGSM and PGD) across six DNN architectures, reported using Top-1 accuracy degradation, Top-5 accuracy degradation, and ground-truth confidence degradation.

- **Top-1 accuracy degradation:** The proposed attacks demonstrate significant degradation. The Weibull-based attack achieves the strongest performance among the proposed methods, causing degradation of 14.6% to 24.7% at $\epsilon = 16$, and 27.8% to 35.1% at $\epsilon = 32$. The GAMMA attack shows comparable performance with 15.5% to 22.6% at $\epsilon = 16$ and 28.7% to 35.7% at $\epsilon = 32$. While white-box PGD achieves higher degradation (31.5% to 50.3% at $\epsilon = 16$), the label-only attacks remain competitive.
- **Top-5 accuracy degradation:** Similar patterns are observed. The Weibull attack yields 18.7% to 29.7% at $\epsilon = 16$, increasing to 31.5% to 44.7% at $\epsilon = 32$. GAMMA follows closely with 18.3% to 28.1% at $\epsilon = 16$ and 32.6% to 41.5% at $\epsilon = 32$.

Table 3.6: Performance degradation (%) under various attack methods.

Attack	Model	$\epsilon = 16$			$\epsilon = 32$		
		Top-1	Top-5	Conf.	Top-1	Top-5	Conf.
GAMMA	VGG19	18.4	25.6	22.7	31.6	38.1	35.1
	ResNet50	22.1	27.5	29.5	35.3	39.3	42.8
	GoogLeNet	20.4	23.3	21.7	33.2	37.5	36.5
	SqueezeNet	19.5	22.3	20.0	29.4	36.6	35.6
	InceptionV3	22.6	28.1	26.1	35.7	41.5	40.2
	NASNet-Large	15.5	18.3	17.5	28.7	32.6	33.8
Weibull	VGG19	21.6	26.2	24.2	33.1	41.1	43.1
	ResNet50	24.7	29.7	32.0	35.1	44.7	44.1
	GoogLeNet	22.1	25.6	22.1	33.8	40.2	42.2
	SqueezeNet	19.0	25.7	27.1	27.7	34.8	33.5
	InceptionV3	22.9	24.6	25.6	31.6	38.5	41.5
	NASNet-Large	14.6	18.7	17.5	27.8	31.5	35.2
Laplacian	VGG19	22.5	25.3	29.8	32.8	42.5	42.0
	ResNet50	16.9	23.8	26.4	33.1	39.3	41.0
	GoogLeNet	15.5	19.9	23.8	30.7	30.4	32.1
	SqueezeNet	18.6	17.7	19.9	30.7	37.8	41.1
	InceptionV3	27.4	33.3	36.4	34.0	39.9	42.4
	NASNet-Large	12.3	16.7	19.0	25.4	30.1	33.6
FGSM	VGG19	35.3	45.9	42.2	51.5	54.4	57.1
	ResNet50	41.7	45.0	43.5	53.5	60.8	55.5
	GoogLeNet	35.5	41.1	36.6	50.1	49.4	48.0
	SqueezeNet	32.8	36.1	43.7	46.0	49.1	52.5
	InceptionV3	42.0	45.2	46.9	55.2	59.4	58.8
	NASNet-Large	27.4	30.5	32.9	42.2	46.4	48.6
PGD	VGG19	46.5	50.4	49.7	59.0	60.6	57.1
	ResNet50	49.1	57.0	53.9	63.9	69.3	61.5
	GoogLeNet	45.5	51.2	47.6	54.6	59.2	60.0
	SqueezeNet	39.3	50.8	46.1	55.9	58.6	62.5
	InceptionV3	50.3	48.8	48.7	65.4	66.3	60.8
	NASNet-Large	31.5	34.8	36.5	44.7	48.9	49.6

Notes: Top-1/Top-5: accuracy degradation for Top-1/Top-5 prediction accuracy; Conf.: ground truth confidence score degradation; ϵ : perturbation level. Higher values indicate stronger attack performance.

- **Confidence degradation:** The Weibull attack achieves 17.5% to 32.0% at $\epsilon = 16$ and 33.5% to 44.1% at $\epsilon = 32$. Similar patterns are observed for GAMMA (17.5% to 29.5% at $\epsilon = 16$) and Laplacian (19.0% to 36.4% at $\epsilon = 16$).

These results demonstrate the effectiveness of the gradient-modelling at-

Performance Degradation Under Various Attack Methods

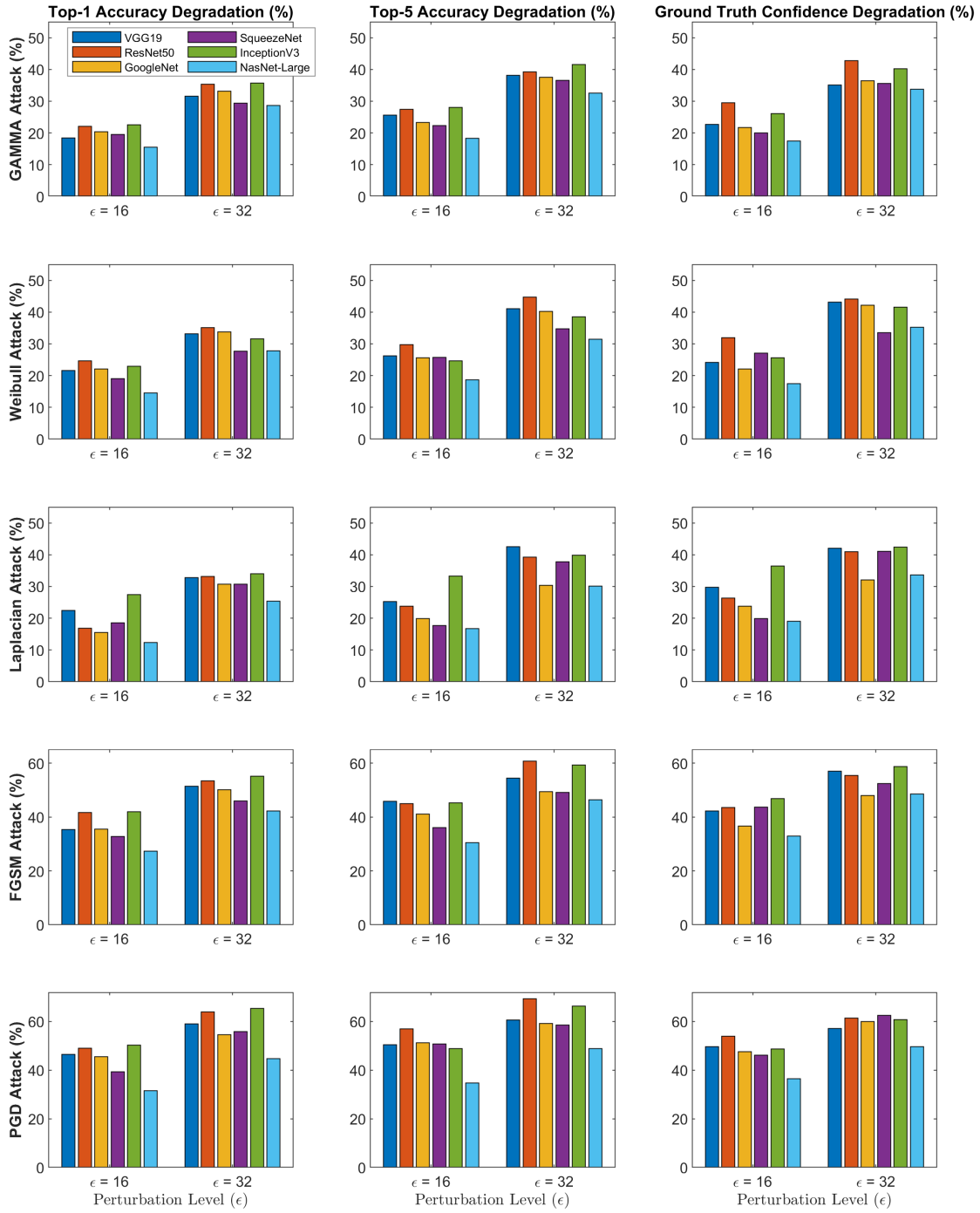


Figure 3.6: Degradation of overall classification performance for various attacking methods and DNN models. The figure shows a 5×3 grid where rows represent attack methods (GAMMA, Weibull, Laplacian, FGSM, PGD) and columns represent degradation metrics (Top-1 accuracy, Top-5 accuracy, ground truth confidence).

tacks, thereby validating the hypotheses on DNN gradients. While not outperforming white-box attacks with direct access to gradients, the proposed label-only black-box methods achieve substantial degradation across all metrics, underscoring the significance of the theoretical insights into gradient modelling.

Comparison with Query-Limited Black-Box Attacks

Figure 3.7 and Table 3.7 present a comprehensive comparison of success rates for the proposed black-box attacks (GAMMA, Weibull, and Laplacian) with existing methods (ZOO and Square Attack) under two settings across six DNN architectures.

- **Setting-1 (query-limited):** With 500 queries and 30 iterations, the proposed attacks demonstrate superior performance. The Weibull-based approach achieves the highest success rates, ranging from 60.8% to 67.2% with $\epsilon = 32$, improving to 68.1% to 73.1% with $\epsilon = 48$. GAMMA shows comparable effectiveness with 56.6% to 63.5% ($\epsilon = 32$) and 64.2% to 69.8% ($\epsilon = 48$). Laplacian maintains strong performance with 55.4% to 61.8% ($\epsilon = 32$) and 62.3% to 68.2% ($\epsilon = 48$). In contrast, ZOO achieves 36.3% to 41.5% ($\epsilon = 32$) and Square Attack reaches only 26.2% to 31.4% ($\epsilon = 32$).
- **Setting-2 (extended queries):** With 1000 queries and 50 iterations, the proposed attacks remain strong. The Weibull attack achieves 68.5% to 74.1% ($\epsilon = 32$) and 71.3% to 76.2% ($\epsilon = 48$). GAMMA follows with 63.5% to 70.5% ($\epsilon = 32$) and 65.1% to 72.6% ($\epsilon = 48$). The Laplacian method yields 63.0% to 69.1% ($\epsilon = 32$) and 67.7% to 72.8% ($\epsilon = 48$). While baseline methods improve with additional queries, they still lag significantly.

It is emphasised that demonstrating superior attack performance is not the primary aim. Rather, these results further validate the hypothesis that DNN gradients can be characterised by specific probability distributions for compromising model robustness. The consistent performance of the distribution-based attacks provides strong evidence that modelling DNN gradients through pre-defined distributions enables effective attacks without direct access to model gradients. This finding supports the second hypothesis (see Sec-

Table 3.7: Success rate (%) of black-box attacks against different DNN models under two settings.

Attack	Model	Setting-1		Setting-2	
		$\epsilon = 32$	$\epsilon = 48$	$\epsilon = 32$	$\epsilon = 48$
GAMMA	VGG19	63.2	69.1	70.1	72.3
	ResNet50	62.8	68.5	69.4	71.8
	GoogLeNet	61.1	67.2	68.2	70.4
	SqueezeNet	63.5	69.8	70.5	72.6
	InceptionV3	61.8	68.0	68.7	70.8
	NASNet-Large	56.6	64.2	63.5	65.1
Weibull	VGG19	66.8	72.8	73.8	75.9
	ResNet50	66.1	72.4	73.2	75.2
	GoogLeNet	64.8	71.0	71.8	73.9
	SqueezeNet	67.2	70.1	74.1	74.2
	InceptionV3	65.5	71.6	72.4	74.5
	NASNet-Large	60.8	68.1	68.5	71.3
Laplacian	VGG19	61.5	67.9	68.8	72.6
	ResNet50	60.8	67.2	68.1	71.9
	GoogLeNet	59.2	65.8	66.5	70.4
	SqueezeNet	61.8	68.2	69.1	72.8
	InceptionV3	60.1	66.4	67.2	71.2
	NASNet-Large	55.4	62.3	63.0	67.7
ZOO	VGG19	41.2	46.3	43.8	51.8
	ResNet50	40.5	45.6	43.1	51.2
	GoogLeNet	39.1	44.2	41.8	49.8
	SqueezeNet	41.5	46.8	44.2	52.1
	InceptionV3	40.2	45.1	42.6	50.4
	NASNet-Large	36.3	41.4	39.5	47.1
Square	VGG19	31.2	31.5	38.2	40.6
	ResNet50	30.5	30.8	37.5	39.8
	GoogLeNet	29.1	29.3	36.1	38.4
	SqueezeNet	31.4	31.8	38.5	40.8
	InceptionV3	29.8	30.1	36.8	39.1
	NASNet-Large	26.2	25.9	33.1	35.3

Notes: Setting-1: 500 queries, 30 iterations; Setting-2: 1000 queries, 50 iterations.

tion 3.1.1) that DNNs are intrinsically vulnerable to perturbations derived from specific probability distributions.

Attack Success Rate (%) Comparison

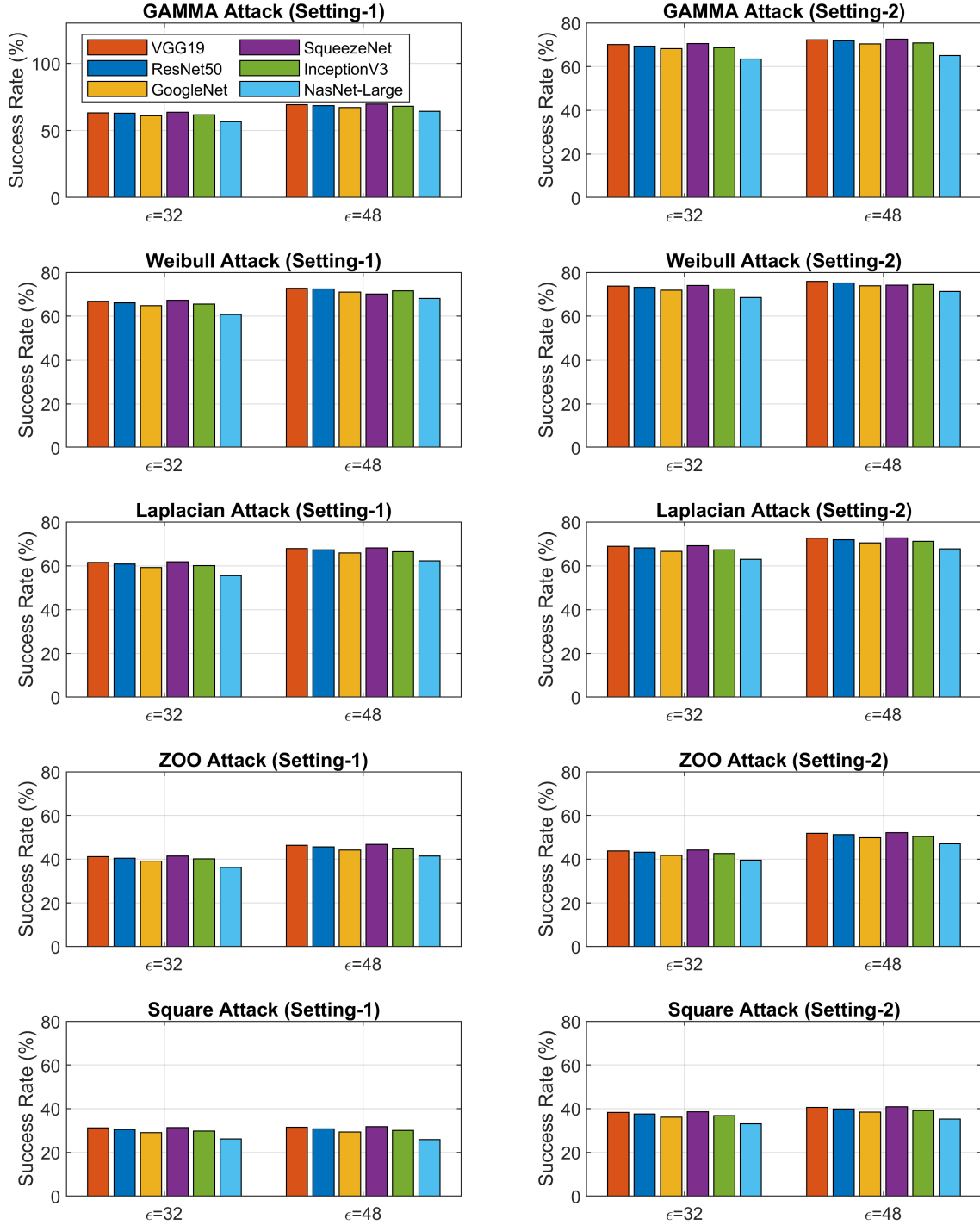


Figure 3.7: Attack success rates comparison for the proposed attacks and existing black-box attacks.

3.2.6 Discussion

Section 3.2.3 provides comprehensive statistical analysis using Goodness-of-Fit tests to demonstrate that DNN gradients can be characterised by specific probability distributions, particularly Laplacian, GAMMA, and Weibull, and Section 3.2.5 shows that gradient-modelling attacks achieve performance comparable to gradient-based white-box attacks, while significantly outperforming query-based black-box attacks across settings.

The primary objective is not to propose an attack outperforming existing methods. Rather, the results validate the hypotheses on DNN gradients described in Section 3.1.1. The findings provide new insight into why DNNs are vulnerable to adversarial examples (AEs): adversarial vulnerability is intrinsic to models optimised via gradient descent, where gradients follow specific distributions that can be efficiently modelled by external attackers. Addressing this vulnerability may therefore require additional external processes to mitigate adversarial perturbations, rather than fundamental architectural changes alone.

Limitations

Several limitations should be noted:

1. **Attack configurations:** For fair comparison across methods, configurations and parameters are unified. Deviations from original defaults may affect baseline performance relative to their reported results.
2. **Parameter sensitivity:** The attack success rates in Table 3.7 are constrained by the pre-defined query limits and iteration counts. Performance could vary under different settings, as reported in the original papers [7, 9, 105, 75].
3. **Comparative framework:** To ensure consistency and rigour, only established attacks are included. While enabling direct comparison, this may not reflect the optimal performance achievable by these methods under bespoke configurations.

Overall, despite these limitations, the consistent performance advantages shown by the proposed methods across different metrics and experimental

settings validate the hypotheses on gradient modelling in the context of adversarial robustness. In the next section, a defence strategy is proposed to combat adversarial examples by utilising insights into DNN gradients to neutralise input images, thereby leading to correct predictions.

3.3 DNN Gradients Modelling-Based Defence Strategy

Building upon the earlier discussion of gradient modelling for adversarial attacks, this section explores its use in improving the adversarial robustness of DNNs. As outlined in Section 3.2, DNN gradients can be characterised by Gamma, Weibull, and Laplacian distribution models, and can be exploited to degrade model robustness. Here, a gradient modelling-based neutralisation (GMBN) defence strategy is proposed. This method leverages insights into DNN gradients to generate a neutralisation perturbation, applied as a pre-processing step to the input image, with the aim of restoring correct predictions. The approach also enhances the confidence score of correctly classified clean images. The strategy is detailed in Algorithm 3 and illustrated in Figure 3.8, which also highlights differences from existing pre-processing-based methods.

In this defence approach, all input images are assumed to be potentially adversarially perturbed; therefore, a pre-processing stage initialises the image with gradient-modelling-based randomisation. The goal is to attenuate adversarial effects non-deterministically, rather than denoising the image. This is achieved by adding a neutralisation perturbation to assist the DNN in making correct predictions, rather than by removing specific image components. For a suspicious input image x , predicted by a DNN D_θ with label y and confidence score c , a threshold T_c is defined: $c \geq T_c$ denotes sufficient confidence, while $c < T_c$ denotes insufficient confidence. Sufficiently confident predictions are initialised with a lower level of neutralisation, whereas insufficiently confident ones receive a higher level. For each iteration, the parameters u of $\mathcal{W}$ are regenerated to introduce randomness and increase the chance of attenuating adversarial effects. Additionally, when $c < T_c$, an alternating factor $(-1)^j$ is applied to the perturbation. The experimental setup and results for this strategy are described in Section 3.3.1, with bar graphs illustrating the outcomes in Figures 3.9 and 3.10.

Algorithm 3 Gradients Modelling-Based Neutralisation (GMBN) for Adversarial Perturbation

Input: Input image $x \in \mathcal{X}$, pre-trained DNN D_θ

Predicted label $y \leftarrow \arg \max(D_\theta(x))$

Confidence score $c \leftarrow D_\theta(x, y)$

Confidence score threshold T_c

Perturbation constraint $\epsilon \in [16, 32]$

Iteration number $j \leftarrow 1$

Maximum iteration number $\mathcal{I}$

Pre-defined distribution model $\mathcal{W}(u)$

for $j = 1 : \mathcal{I}$ **do**

if $c \geq T_c$ **then**

 Initialise distribution model $\mathcal{W}(u)$

 Gradients modelling: $\omega \leftarrow \text{icdf}(\mathcal{W}(u))$

 Rescale ω into $[-\frac{\epsilon}{4}, \frac{\epsilon}{4}]$ to obtain ω'

$x_r \leftarrow x + \omega'$

$y_r \leftarrow \arg \max(D_\theta(x_r))$

$c_r \leftarrow \max(D_\theta(x_r))$

else

 Initialise distribution model $\mathcal{W}(u)$

 Gradients modelling: $\omega \leftarrow \text{icdf}(\mathcal{W}(u))$

 Rescale ω into $[-\frac{\epsilon}{2}, \frac{\epsilon}{2}]$ to obtain ω'

$x_r \leftarrow x + (-1)^j \omega'$

$y_r \leftarrow \arg \max(D_\theta(x_r))$

$c_r \leftarrow \max(D_\theta(x_r))$

end if

end for

Return: x_r, y_r, c_r

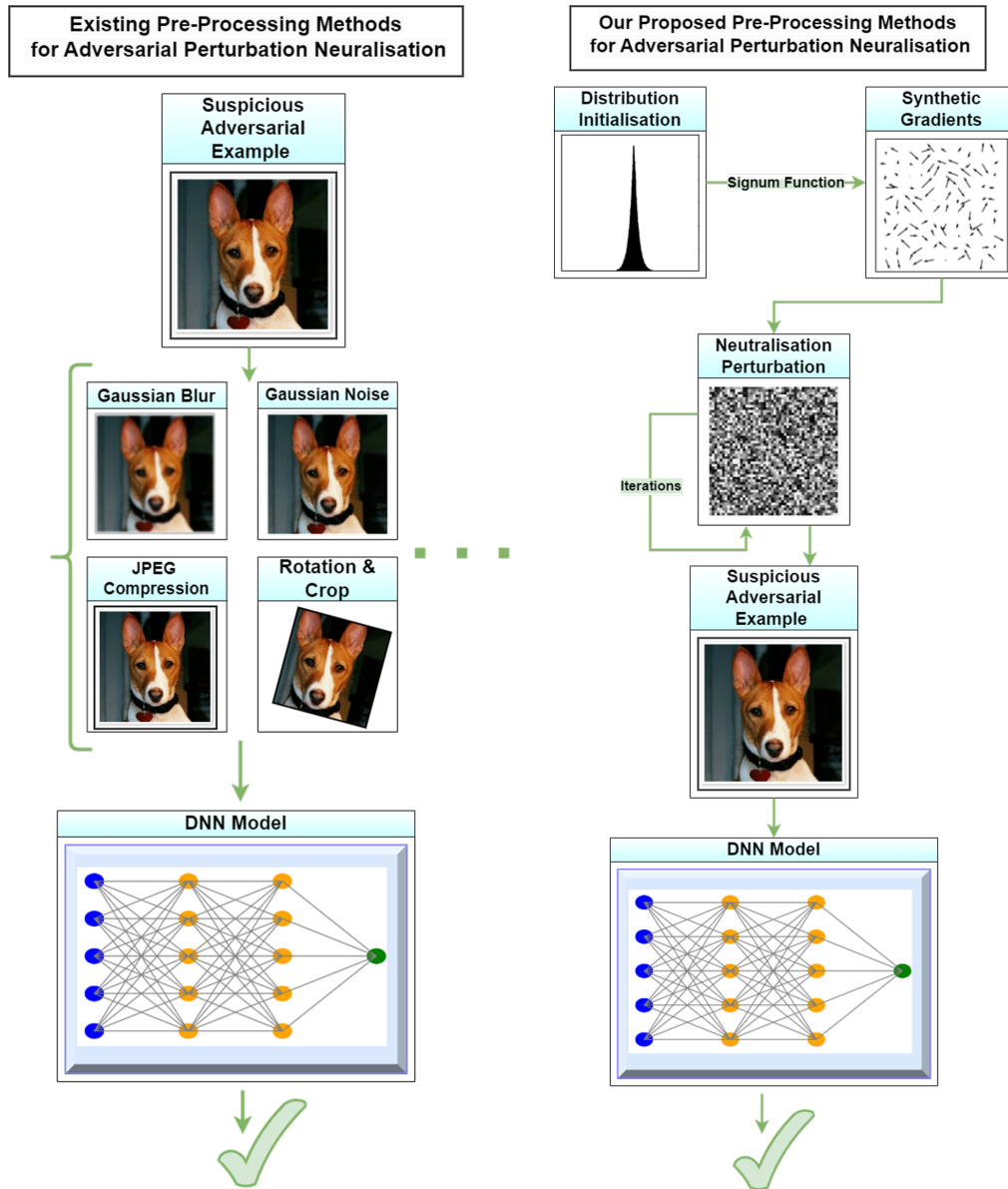


Figure 3.8: Comparison of the proposed Gradient Modelling-Based Neutralisation (GMBN) and existing pre-processing defences. In GMBN, the incoming image is first passed through the DNN to obtain a predicted label and confidence score. A confidence threshold then determines whether a low or high scale perturbation will be applied. The perturbation is generated by sampling from a statistical model of DNN gradients (Gamma, Weibull, or Laplacian) and rescaling it to the chosen strength. This “neutralisation” perturbation is added to the input image, which the DNN model then reclassifies.

3.3.1 Real-World Scenario Application: ImageNet Dataset

To evaluate the effectiveness of the proposed pre-processing method in mitigating adversarial effects, a set of experiments was conducted comparing GMBN with established techniques, as shown in Figure 3.8. The objective was to assess how each method defends against different attack scenarios, focusing on improvements in DNN robustness for image classification.

Experimental Setup

1. **Dataset:** A subset of the ImageNet dataset [19] containing 100,000 randomly selected images was used, ensuring balanced class representation while maintaining computational feasibility.
2. **Pre-trained DNN models:** Six diverse pre-trained architectures were tested: VGG-19 [10], ResNet-50 [14], GoogLeNet [118], SqueezeNet [?], Inception-v3 [121], and NASNet-Large [122]. This diversity supports a comprehensive examination of robustness across architectures.
3. **Pre-processing defence methods for comparison:** Following the literature reviewed in Chapter 2, JPEG compression and feature smoothing were selected as reference methods. Configuration details are shown in Table 3.8.

Table 3.8: Pre-processing defence methods configuration

Category	Method	Parameters
JPEG compression	JPEG-90	Quality factor: 90%
	JPEG-70	Quality factor: 70%
	JPEG-50	Quality factor: 50%
Feature smoothing	Gaussian blur	Kernel: 3×3 , $\sigma = 1.5$
	Median filtering	Kernel: 3×3

4. **Adversarial attacks:** Configurations are given in Table 3.9, covering FGSM and PGD with varying ϵ and α parameters. This range allows testing under benchmark settings.

Table 3.9: *Adversarial attacks configuration*

Attack	ϵ	α	Iterations	Index
FGSM	16/255	—	—	A1
FGSM	32/255	—	—	A2
PGD	16/255	0.25	50	A3
PGD	32/255	0.25	50	A4

Note: ϵ and α values are in pixel scale (0–255).

As shown in Figure 3.8, a comparative analysis of pre-processing-based defence methods was conducted on six popular DNN models against two well-known adversarial attacks. To validate the effectiveness and robustness of the proposed strategy, GMBN was compared with several existing pre-processing methods verified to be effective against adversarial attacks, namely JPEG compression and feature squeezing (Gaussian blur and median filtering). Implementation details are provided in Table 3.8. The dataset and training environment follow those used for the attacking algorithms described in Section 3.2.4. In the experiments, FGSM [7] attacks with perturbation levels ϵ of 16 and 32, and PGD [9] attacks with perturbation levels ϵ of 16 and 32, were used to test the pre-processing methods.

3.3.2 Robustness Evaluation of the GMBN Defence

Tables 3.10 and 3.11 demonstrate the improvement achieved by GMBN across six DNN models against FGSM and PGD attacks under different conditions. The results provide insight into how pre-processing-based defence methods improve performance against strong adversarial attacks.

Across all attacking scenarios, GMBN demonstrates the most substantial improvements in adversarial robustness compared with other pre-processing methods, especially in Top-5 accuracy when the perturbation level is large. For example, under FGSM with $\epsilon = 16$, GMBN (lower and higher scale) improves the Top-5 accuracy of DNN models by an average of 50.6% and 65.95%, compared with 48.88% for the next best method, JPEG compression (QF = 50%). With $\epsilon = 32$ for FGSM, GMBN (higher scale) also yields an average Top-5 improvement of 19.1%, compared with 14.5% for Gaussian blurring. Likewise, under PGD with $\epsilon = 32$, GMBN (lower and higher scale) increases Top-5 accuracy by an average of 32.8% and 31.1%, compared with 27.2% and 25.7% for

Table 3.10: Improvement in overall classification performance for varied pre-processing methods and DNN models (FGSM attacks). The data in this table are visualised in Figure 3.9. “QF” refers to the quality factor in JPEG compression.

Pre-processing methods	DNN models	Improvement in overall classification performance			
		FGSM ($\epsilon = 16$)		FGSM ($\epsilon = 32$)	
		Top-1 accuracy	Top-5 accuracy	Top-1 accuracy	Top-5 accuracy
JPEG compression (QF = 90%)	VGG-19	9.4%	33.9%	2.8%	5.5%
	ResNet-50	14.6%	39.0%	3.4%	4.5%
	GoogLeNet	10.7%	37.6%	3.5%	4.6%
	SqueezeNet	8.8%	36.6%	2.1%	6.0%
	Inception-v3	6.8%	42.0%	2.7%	5.3%
	NASNet-Large	9.2%	43.4%	2.6%	4.3%
JPEG compression (QF = 70%)	VGG-19	22.7%	45.8%	3.0%	8.3%
	ResNet-50	21.9%	37.6%	4.0%	8.6%
	GoogLeNet	27.4%	32.3%	3.7%	8.5%
	SqueezeNet	27.8%	47.4%	2.5%	9.2%
	Inception-v3	14.7%	39.1%	3.2%	8.7%
	NASNet-Large	24.3%	31.6%	2.9%	6.6%
JPEG compression (QF = 50%)	VGG-19	29.9%	45.8%	3.8%	6.0%
	ResNet-50	31.7%	45.8%	5.7%	5.8%
	GoogLeNet	33.9%	52.9%	5.1%	6.0%
	SqueezeNet	39.3%	53.5%	3.3%	7.0%
	Inception-v3	20.0%	51.9%	4.5%	6.7%
	NASNet-Large	31.4%	43.4%	3.6%	4.9%
Gaussian blur	VGG-19	18.0%	56.8%	5.5%	15.4%
	ResNet-50	15.6%	51.2%	3.3%	14.4%
	GoogLeNet	18.4%	53.6%	5.3%	14.6%
	SqueezeNet	24.4%	42.3%	2.8%	13.1%
	Inception-v3	26.2%	37.3%	4.1%	15.1%
	NASNet-Large	21.5%	49.3%	2.7%	14.4%
Median filtering	VGG-19	23.3%	52.2%	3.1%	14.0%
	ResNet-50	22.8%	46.1%	4.9%	13.0%
	GoogLeNet	17.8%	49.0%	4.6%	12.0%
	SqueezeNet	20.1%	38.3%	3.4%	9.5%
	Inception-v3	23.6%	35.1%	5.4%	10.9%
	NASNet-Large	23.9%	39.7%	3.8%	15.5%
GMBN (lower scale)	VGG-19	29.3%	41.3%	6.2%	14.2%
	ResNet-50	28.0%	65.7%	4.1%	12.1%
	GoogLeNet	24.0%	40.0%	5.5%	13.0%
	SqueezeNet	26.4%	61.1%	4.4%	9.6%
	Inception-v3	31.0%	47.4%	2.7%	10.4%
	NASNet-Large	32.7%	48.1%	3.8%	11.3%
GMBN (higher scale)	VGG-19	21.4%	59.2%	3.9%	21.5%
	ResNet-50	18.0%	70.8%	4.6%	20.5%
	GoogLeNet	28.5%	55.4%	4.4%	21.7%
	SqueezeNet	28.5%	74.2%	3.9%	16.2%
	Inception-v3	37.3%	67.1%	4.1%	16.4%
	NASNet-Large	34.3%	69.0%	4.1%	18.3%

JPEG compression (QF = 70% and 50%), respectively. In summary, for Top-1 accuracy, all pre-processing methods exhibit comparable improvements across attacking scenarios. For Top-5 accuracy, GMBN demonstrates the best overall

Table 3.11: Improvement in overall classification performance for varied pre-processing methods and DNN models (PGD attacks). The data in this table are visualised in Figure 3.10. “QF” refers to the quality factor in JPEG compression.

Pre-processing methods	DNN models	Improvement in overall classification performance			
		PGD ($\epsilon = 16$)		PGD ($\epsilon = 32$)	
		Top-1 accuracy	Top-5 accuracy	Top-1 accuracy	Top-5 accuracy
JPEG compression (QF = 90%)	VGG-19	14.2%	27.9%	9.9%	20.0%
	ResNet-50	12.3%	23.6%	9.3%	16.8%
	GoogLeNet	12.7%	16.4%	13.2%	21.7%
	SqueezeNet	15.7%	22.3%	12.5%	24.4%
	Inception-v3	14.8%	32.4%	9.9%	20.1%
	NASNet-Large	12.1%	20.8%	13.4%	28.0%
JPEG compression (QF = 70%)	VGG-19	28.6%	39.2%	16.9%	25.9%
	ResNet-50	21.4%	35.1%	19.3%	29.6%
	GoogLeNet	24.6%	42.1%	14.1%	21.7%
	SqueezeNet	23.1%	45.7%	18.8%	29.5%
	Inception-v3	21.1%	38.8%	24.8%	28.8%
	NASNet-Large	24.9%	35.4%	21.0%	27.6%
JPEG compression (QF = 50%)	VGG-19	27.4%	29.1%	17.6%	27.4%
	ResNet-50	26.1%	39.3%	22.5%	31.5%
	GoogLeNet	32.5%	44.6%	22.7%	28.8%
	SqueezeNet	28.0%	45.7%	15.2%	23.4%
	Inception-v3	20.3%	36.8%	14.2%	22.1%
	NASNet-Large	28.4%	36.8%	19.8%	21.1%
Gaussian blur	VGG-19	27.8%	27.1%	19.6%	24.1%
	ResNet-50	23.1%	24.8%	24.5%	20.0%
	GoogLeNet	17.6%	33.5%	17.9%	22.6%
	SqueezeNet	24.1%	42.0%	19.1%	23.2%
	Inception-v3	29.5%	33.5%	13.7%	23.1%
	NASNet-Large	24.3%	23.0%	15.7%	16.9%
Median filtering	VGG-19	25.7%	33.6%	17.2%	23.0%
	ResNet-50	34.6%	32.9%	13.9%	16.9%
	GoogLeNet	18.7%	33.8%	15.2%	18.4%
	SqueezeNet	18.3%	34.5%	11.9%	13.6%
	Inception-v3	24.2%	25.3%	16.9%	18.1%
	NASNet-Large	14.9%	32.7%	12.8%	14.1%
GMBN (lower scale)	VGG-19	27.3%	24.4%	15.0%	35.2%
	ResNet-50	21.9%	27.9%	17.6%	28.6%
	GoogLeNet	25.3%	25.7%	19.9%	37.5%
	SqueezeNet	28.8%	27.1%	18.9%	33.3%
	Inception-v3	23.7%	34.0%	11.2%	29.7%
	NASNet-Large	27.6%	33.2%	16.0%	32.5%
GMBN (higher scale)	VGG-19	26.9%	33.9%	16.3%	30.0%
	ResNet-50	36.3%	36.8%	19.6%	29.9%
	GoogLeNet	34.5%	49.2%	19.5%	30.4%
	SqueezeNet	23.7%	38.9%	17.0%	28.1%
	Inception-v3	27.7%	33.3%	13.7%	31.1%
	NASNet-Large	31.9%	35.6%	23.7%	36.8%

performance compared with JPEG compression and feature squeezing; noticeably, JPEG compression tends to rank second against FGSM, while feature squeezing exhibits performance comparable to GMBN against PGD. Several

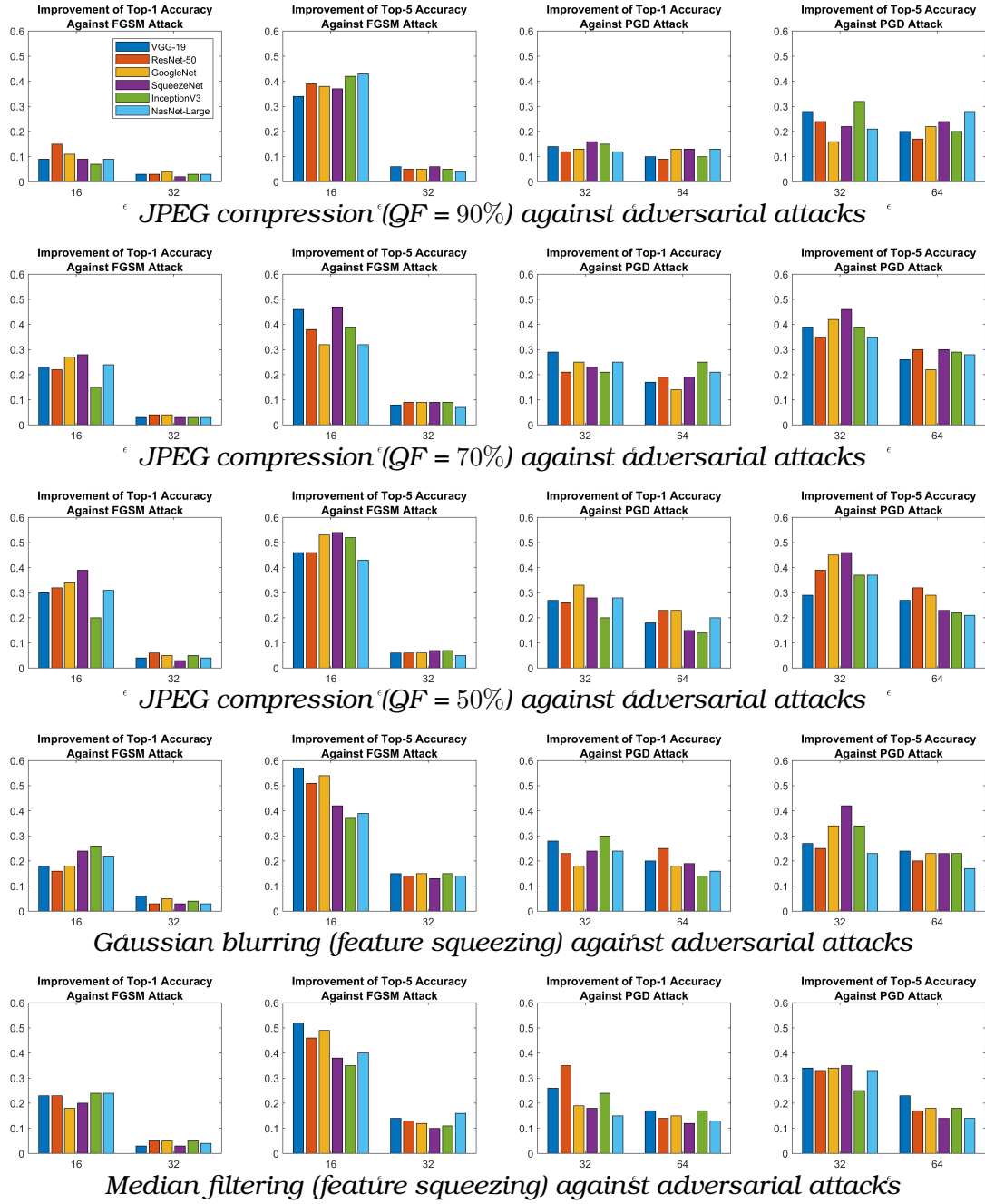


Figure 3.9: Overall classification accuracy improvement against adversarial attacks by existing pre-processing-based methods. This set of graphs visualises the data in Table 3.10.

qualitative examples are provided in Figures 3.11 and 3.12, with additional examples in Appendix A, Figure A.4, providing a broader comparison of defence performance.

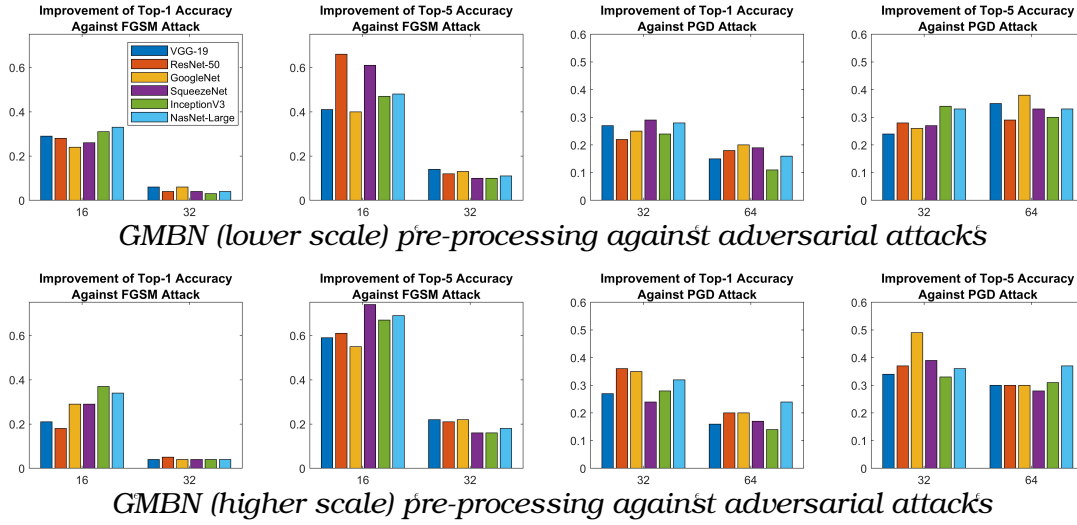


Figure 3.10: Overall classification accuracy improvement against adversarial attacks by pre-processing input images with GMBN. This set of graphs visualises the data in Table 3.11.

Moreover, GMBN was evaluated on non-adversarial (clean) images. Across the six DNN models, GMBN enhances Top-1 accuracy, Top-5 accuracy, and confidence scores, indicating that the neutralisation perturbation can assist classification with greater confidence. Results are provided in Appendix A, Table A.1 and Figure A.3.

3.3.3 Discussion

In the preceding sections, a new insight into DNN gradients was revealed: the gradients can be characterised using Gamma, Weibull, and Laplacian distribution models (see Section 3.2). It was then demonstrated that this insight can be leveraged to either degrade or enhance adversarial robustness (see Section 3.2.4). Subsequently, experiments showed that neutralisation perturbations based on synthetic gradients can mitigate the impact of adversarial perturbations (see Section 3.3). This section discusses the significance of these results and outlines limitations and potential future work.

A new understanding of DNN gradients

To the best of current knowledge, this is the first study to investigate precise modelling of DNN gradients and to conduct comprehensive experiments valid-

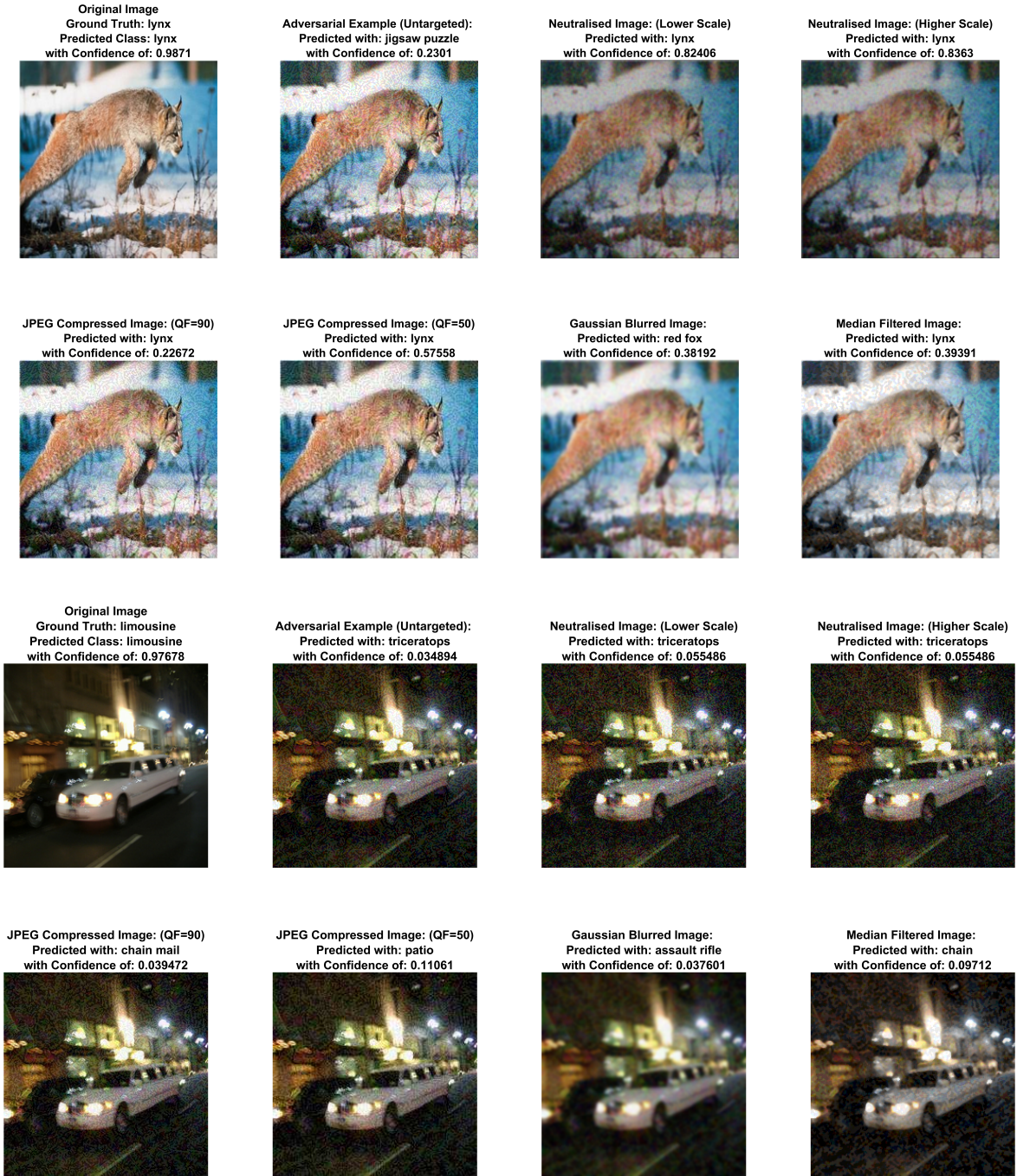


Figure 3.11: Examples of using different pre-processing methods for the defence of adversarial examples (FGSM attack).

ating the fit of different distributions. Conventionally, successful adversarial attacks require access to the gradients of the DNN output with respect to the input image, as the gradients are essential for constructing adversarial per-



Figure 3.12: Examples of using different pre-processing methods for the defence of adversarial examples (PGD attack).

turbations. In black-box scenarios, where access to the target model is limited or absent, attackers often train a substitute model to estimate gradients, or rely on other sophisticated approaches. The analysis in Section 3.2 supports the

hypothesis that DNN gradients can be characterised by specific distribution models, offering a simpler alternative to computationally expensive approaches. Section 3.2.4 further demonstrates that adversaries may construct adversarial examples with only label information by exploiting synthetic gradients. Overall, the crucial factor in constructing adversarial examples, $\nabla_{\theta}\mathcal{L}(D_{\theta}(x_j), y_j)$, can be approximated by Gamma, Weibull, or Laplacian distributions. Although synthetic gradients are generally weaker than exact gradients computed from the model, repeated generation can still effect a significant drop in prediction confidence and induce misclassification. This discovery exposes a potential adversarial weakness and suggests that external input pre-processing may serve as an effective defence.

A new insight into combating adversarial examples

The aim of this study is not to propose a state-of-the-art black-box attack, but to reveal an internal characteristic of DNN gradients and show how it can be used to both downgrade and enhance robustness. These findings suggest a practical pathway for combating adversarial examples:

1. DNN gradients are the most critical factor in efficiently constructing adversarial perturbations because they directly manipulate the output of DNNs, maximising the likelihood of misclassification. When attackers lack model access, they typically estimate gradients through complex or computationally expensive means. This study demonstrates that gradients can be characterised by probability distributions with specific parameters. While synthetic gradients may be less effective per attempt, repeated trials can still achieve a distinct drop in prediction confidence and cause misclassification.
2. As yet, there is no universally accepted explanation for the intricate nature of adversarial examples in deep learning; thus, a single definitive solution remains challenging. Since DNN gradients can be characterised and exploited to construct adversarial perturbations, the converse is also viable: synthetic gradients may be used to construct neutralisation perturbations as a pre-processing defence. The results in Section 3.3.2 show that the proposed GMBN achieves markedly better Top-5 accuracy improvements than other pre-processing defences and exhibits competitive Top-1

improvements. Rather than attempting to eradicate adversarial perturbations, GMBN introduces a neutralisation perturbation that mitigates their effect, leading to correct predictions.

3.3.4 Limitations

Several limitations are noted:

1. **Distribution model selection:** Although three distributions were identified that effectively characterise DNN gradients, the theoretical foundation for their suitability remains to be established.
2. **Model architecture:** While results were consistent across different architectures, the impact of broader architectural variations on gradient distributions was not exhaustively studied. A systematic investigation could yield deeper insight.
3. **Computational power:** The analysis focused on benchmark attacks due to limited computational resources. Further research should evaluate robustness against a wider range of attack types.

3.4 Real-World Scenario Application: CamVid Dataset

To evaluate the effectiveness of the proposed pre-processing method in mitigating adversarial effects, as described in Section 3.3, a comprehensive set of experiments was conducted comparing the performance of the approach with established pre-processing techniques for image segmentation tasks. The primary objective is to determine how well the method defends against adversarial attacks in a real-world scenario, thereby providing a thorough evaluation of the proposed GMBN.

3.4.1 Experimental Setup

As this is a different deep learning task, the experimental setups are described in detail below for clarity.

1. **Dataset:** The CamVid dataset is employed, containing 701 images of urban scenes with pixel-wise semantic labels for 11 classes. The dataset is split into 367 training images, 101 validation images, and 233 test images. Each frame is accompanied by high-quality semantic labels, making the dataset a recognised benchmark for assessing segmentation robustness.
2. **Pre-trained Segmentation Models:** Due to computational constraints and to ensure a thorough evaluation, the focus is on the DeepLabV3+ architecture with two backbone networks: **DeepLabV3+ ResNet-18** and **DeepLabV3+ ResNet-50**.
3. **Pre-processing Defence Techniques for Comparison:** As summarised in Table 3.12, three categories of pre-processing defence techniques, reviewed in Chapter 2, are implemented to evaluate their effectiveness against adversarial attacks in semantic segmentation. JPEG compression is applied at three quality levels (90%, 70%, and 50%). Feature smoothing techniques include Gaussian blur with a 3×3 kernel (standard deviation: 1.5) and median filtering with a 3×3 neighbourhood. Additionally, colour quantisation is performed by reducing the number of colours to eight.

Table 3.12: *Pre-processing Defence Methods Configuration*

Category	Method	Parameters
JPEG Compression	JPEG-90	Quality factor: 90%
	JPEG-70	Quality factor: 70%
	JPEG-50	Quality factor: 50%
Feature Smoothing	Gaussian Blur	Kernel size: 3×3 Standard deviation: 1.5
	Median Filtering	Kernel size: 3×3
Colour Quantisation	Colour Reduction	Number of colours: 8

4. **Evaluation Metrics:** Performance is evaluated using standard semantic segmentation metrics:

- **Global Accuracy:** Overall pixel-wise classification accuracy:

$$\text{Global Accuracy} = \frac{\sum_{i=1}^{n_c} TP_i}{\sum_{i=1}^{n_c} (TP_i + FP_i)}, \quad (3.12)$$

where TP_i and FP_i represent true positives and false positives for class i , and n_c is the total number of classes.

- **Mean Accuracy:** The average of per-class accuracies:

$$\text{Mean Accuracy} = \frac{1}{n_c} \sum_{i=1}^{n_c} \frac{TP_i}{TP_i + FN_i}, \quad (3.13)$$

where FN_i denotes false negatives for class i .

- **MeanIoU:** The mean intersection over union:

$$\text{MeanIoU} = \frac{1}{n_c} \sum_{i=1}^{n_c} \frac{TP_i}{TP_i + FP_i + FN_i}, \quad (3.14)$$

- **WeightedIoU:** A class-frequency weighted IoU:

$$\text{WeightedIoU} = \sum_{i=1}^{n_c} w_i \cdot \frac{TP_i}{TP_i + FP_i + FN_i}, \quad (3.15)$$

where w_i is the frequency weight of class i .

- **MeanBFScore:** The boundary F1 score:

$$\text{BFScore} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \quad (3.16)$$

where precision and recall are computed specifically for boundary pixels.

These metrics collectively assess segmentation quality, overall accuracy, and class-wise balance.

5. **Adversarial Attacks Configuration:** Table 3.13 lists the targeted PGD attacks adapted for semantic segmentation. Unlike image classification, these attacks operate at the pixel level, aiming to erase specific classes by misclassifying them as other classes. Three attack types are considered: single-class, two-class, and three-class erasure, with target classes randomly selected from the 11 semantic categories in the CamVid dataset [123]. A constant step size of 1 and 30 iterations are used for all attacks, with perturbation constraints $\epsilon = 16/255$ and $\epsilon = 32/255$.

Table 3.13: *Adversarial Attacks Configuration for Semantic Segmentation*

Attack	Target	ϵ	Step Size	Iterations	Index
PGD	Single Class	16/255	1	30	A1
		32/255	1	30	A2
	Two Classes	16/255	1	30	A3
		32/255	1	30	A4
	Three Classes	16/255	1	30	A5
		32/255	1	30	A6

Notes: Target classes are randomly selected for each attack from the CamVid dataset’s 11 semantic classes. The attacks modify the segmentation output such that targeted classes are misclassified as other classes, effectively erasing them from the prediction.

Unlike image classification tasks where adversarial attacks aim to cause misclassification of the entire image, semantic segmentation requires a different attack approach focusing on pixel-wise predictions. As shown in Table 3.13, targeted PGD attacks are employed to erase selected classes from the segmentation output. Three types of attacks with increasing complexity are conducted: single-class erasure, two-class erasure, and three-class erasure. Due to computational constraints and the dense prediction nature of segmentation tasks, a constant step size of 1 and 30 iterations are used for all attack scenarios. For each attack type, two perturbation constraints ($\epsilon = 16/255$ and $\epsilon = 32/255$) are evaluated to assess defence performance under different attack strengths. This configuration enables a comprehensive evaluation of robustness against class-wise targeted semantic attacks.

3.4.2 Experimental Results

The defence method is evaluated against both single-class and multi-class adversarial attacks on semantic segmentation tasks. The evaluation includes overall segmentation performance and class-wise analysis. Tables 3.14 and 3.15 present the results for single-class and multi-class attacks, respectively, while Tables 3.16 and 3.17 show the detailed class-wise performance. To provide an intuitive view of the defence performance of the proposed GMBN, Figures 3.13 and 3.14 visualise the neutralisation effects of adversarial per-

turbations by displaying segmentation maps of clean images, adversarial examples, and neutralised adversarial examples at both image and pixel levels.

Table 3.14: Overall Segmentation Performance Evaluation for Single-Class Attack

Methods	Global Accuracy	Mean Accuracy	MeanIoU	Weighted IoU	Mean BFScore
Clean Image	0.9035	0.8716	0.6703	0.8413	0.7260
AE	0.4359	0.2764	0.1724	0.3077	0.2331
GMBN (Proposed)	<u>0.8326</u>	<u>0.8064</u>	<u>0.5903</u>	<u>0.7597</u>	<u>0.6146</u>
JPEG	0.6265	0.6459	0.4459	0.5931	0.4874
Colour Reduction	0.4923	0.5246	0.3089	0.3965	0.3577
Feature Smoothing	0.6682	0.6780	0.4866	0.5980	0.4819

Table 3.15: Overall Segmentation Performance Evaluation for Multi-Class Attack

Methods	Global Accuracy	Mean Accuracy	MeanIoU	Weighted IoU	Mean BFScore
Clean Image	0.9035	0.8716	0.6703	0.8413	0.7260
AE	0.4359	0.2764	0.1724	0.3077	0.2331
GMBN (Proposed)	<u>0.7888</u>	<u>0.7993</u>	<u>0.5567</u>	<u>0.7234</u>	<u>0.6098</u>
JPEG	0.5465	0.5659	0.3759	0.5131	0.4074
Colour Reduction	0.4123	0.4446	0.2389	0.3165	0.2777
Feature Smoothing	0.5882	0.5980	0.4066	0.5180	0.4019

Notes: Results combine DeepLabV3+ with ResNet-18 and ResNet-50 backbones. **Bold** indicates clean image performance; underlined values indicate the best defence performance. Multi-class attack results combine performance across two-class and three-class attacks.

3.4.3 Overall Segmentation Performance

For clean images, the segmentation model achieves baseline performance with 90.35% global accuracy and 67.03% mean IoU. Under adversarial attack, performance degrades substantially, with global accuracy dropping to 43.59% and mean IoU to 17.24%.

The proposed GMBN defence significantly outperforms other pre-processing approaches. For single-class attacks, GMBN achieves 83.26% global accuracy and 59.03% mean IoU, outperforming JPEG compression (62.65%, 44.59%), colour

Table 3.16: *Class-wise Performance Evaluation — Clean Image, Adversarial Examples, and GMBN Defence*

Class	Clean Image			AE			GMBN (Proposed)		
	PA	IoU	BFS	PA	IoU	BFS	PA	IoU	BFS
Sky	0.963	0.943	0.943	0.934	0.611	0.621	0.848	0.688	0.625
Building*	0.884	0.865	0.749	0.007	0.007	0.102	0.630	0.471	0.388
Pole	0.842	0.338	0.648	0.253	0.048	0.237	0.149	0.076	0.186
Road*	0.925	0.902	0.762	0.023	0.023	0.136	0.677	0.569	0.474
Pavement	0.919	0.838	0.846	0.831	0.344	0.336	0.624	0.446	0.355
Tree	0.804	0.729	0.736	0.341	0.195	0.250	0.311	0.201	0.262
SignSymbol	0.889	0.489	0.600	0.462	0.054	0.209	0.181	0.070	0.182
Fence	0.642	0.221	0.472	0.000	0.000	0.000	0.000	0.000	0.005
Car	0.944	0.834	0.696	0.614	0.238	0.180	0.491	0.301	0.230
Pedestrian*	0.942	0.526	0.702	0.004	0.003	0.055	0.154	0.087	0.224
Bicyclist	0.772	0.248	0.517	0.003	0.001	0.741	0.000	0.000	0.000

* indicates target classes. PA: class-wise pixel accuracy; BFS: mean BF score.

Bold indicates target-class values.

Table 3.17: *Class-wise Performance Evaluation — Comparison with Other Defence Methods*

Class	JPEG			Feature Smoothing			Colour Reduction		
	PA	IoU	BFS	PA	IoU	BFS	PA	IoU	BFS
Sky	0.882	0.762	0.736	0.805	0.756	0.708	0.929	0.689	0.642
Building*	0.568	0.451	0.401	0.645	0.496	0.413	0.044	0.044	0.131
Pole	0.221	0.081	0.277	0.232	0.110	0.312	0.258	0.047	0.246
Road*	0.633	0.575	0.399	0.659	0.607	0.396	0.025	0.025	0.185
Pavement	0.673	0.485	0.437	0.619	0.484	0.427	0.827	0.353	0.353
Tree	0.333	0.192	0.217	0.362	0.169	0.225	0.446	0.235	0.277
SignSymbol	0.385	0.109	0.201	0.423	0.160	0.221	0.519	0.059	0.206
Fence	0.000	0.000	0.006	0.000	0.000	0.019	0.023	0.004	0.078
Car	0.574	0.320	0.205	0.592	0.332	0.201	0.655	0.251	0.201
Pedestrian*	0.165	0.122	0.252	0.196	0.158	0.251	0.005	0.005	0.125
Bicyclist	0.332	0.087	0.088	0.514	0.196	0.201	0.000	0.000	0.000

* indicates target classes. PA: class-wise pixel accuracy; BFS: mean BF score.

Bold indicates target-class values.

reduction (49.23%, 30.89%), and feature smoothing (66.82%, 48.66%). In multi-class scenarios, GMBN achieves 78.88% global accuracy and 55.67% mean IoU, whereas conventional pre-processing methods remain comparatively weak against multi-class attacks (e.g., JPEG: 54.65% global accuracy; feature smoothing: 58.82%; colour reduction: 41.23%).

Overall, these results demonstrate that the gradient-based neutralisation

approach provides markedly better robustness to adversarial perturbations than other pre-processing methods, particularly in complex multi-class attack settings. The consistent performance of GMBN across both single- and multi-class attacks supports its effectiveness in maintaining semantic segmentation accuracy by neutralising adversarial noise.

3.4.4 Class-wise Performance

A class-wise evaluation of the proposed GMBN method and other pre-processing defences is presented in Tables 3.16 and 3.17. The three target classes ("*Building*", "*Road*", and "*Pedestrian*") are chosen for adversarial attacks. Baseline performance on clean images shows high accuracy across all classes, with pixel accuracy ranging from 0.642 to 0.963 and IoU from 0.221 to 0.943. For the target classes, adversarial attacks cause severe degradation. For example, the "*Building*" class accuracy drops from 0.884 to 0.007, "*Road*" from 0.925 to 0.023, and "*Pedestrian*" from 0.942 to 0.004.

The GMBN method demonstrates consistent improvement across most classes compared with other defence methods. For instance, it recovers "*Building*" accuracy to 0.630 and "*Road*" accuracy to 0.677, while JPEG compression achieves 0.568 and 0.633 respectively, and feature smoothing achieves 0.645 and 0.659 respectively. All methods perform substantially better than colour reduction. However, objects occupying a smaller proportion of the image (*e.g.*, "*Pole*", "*SignSymbol*") remain challenging for all defence methods, suggesting that preserving the semantic accuracy of minor-proportion objects while neutralising adversarial perturbations requires further improvement. Notably, non-target classes also benefit from GMBN, indicating an advantage in neutralising adversarial effects on global semantic accuracy. For example, the "*Sky*" and "*Car*" classes maintain relatively high recovery rates under GMBN despite not being direct targets.

To provide deeper insight into how pre-processing methods affect segmentation under adversarial attack from a class-wise perspective, confusion matrices are visualised in Figures 3.15, 3.16, and 3.17. The clean image confusion matrix (Figure 3.15(a)) shows strong diagonal dominance, indicating good class predictions across all categories. Under adversarial attack (Figure 3.15(b)), severe misclassification of the target classes is observed, with predictions dis-

tributed across multiple incorrect classes. The GMBN method (Figure 3.16(a)) effectively restores the diagonal dominance for the target classes, outperforming JPEG compression (Figure 3.16(b)), feature smoothing (Figure 3.17(a)), and colour reduction (Figure 3.17(b)).

3.4.5 Summary

The experimental results demonstrate that GMBN provides better overall neutralisation of adversarial perturbations compared with existing pre-processing methods in both single- and multi-class scenarios.

Several aspects of the proposed GMBN distinguish it from other pre-processing-based methods. First, while traditional methods show reduced effectiveness in preserving semantic information in the single-class scenario compared with the multi-class scenario, GMBN maintains stable segmentation accuracy for both attack conditions.

Second, GMBN exhibits balanced performance across different semantic categories, whereas other pre-processing methods show uneven class-wise performance, with particularly poor results for minor-proportion objects. In contrast, GMBN preserves both global and local semantic information more consistently.

Nonetheless, all pre-processing methods, including GMBN, show limited effectiveness in recovering segmentation accuracy for small and less-represented objects such as *"Pole"* and *"SignSymbol"*. This highlights an inherent challenge in neutralising adversarial perturbations while preserving fine semantic details, requiring further investigation.

Third, the complexity of semantic segmentation tasks can reduce defence performance. Unlike image classification, where gradients share the same dimensions as the input image ($\text{height} \times \text{width} \times 3$), segmentation gradients have dimensions of $\text{height} \times \text{width} \times \text{channel} \times \text{number of classes}$. In these experiments with 11 semantic classes, gradient dimensionality is considerably larger, increasing computational complexity and making effective neutralisation while preserving semantic information more challenging. This is particularly evident in class-wise results, where smaller objects with fewer pixels are more difficult to recover with accurate pixel-wise classification.

The class-wise results in Tables 3.16 and 3.17 should therefore be inter-

preted cautiously. The aim is to provide insight into how different methods affect specific semantic categories, rather than to represent general defence capabilities. These results relate to a specific case in which "*Building*", "*Road*", and "*Pedestrian*" were deliberately targeted, and thus the overall effectiveness should be judged from the aggregated performance metrics in Tables 3.14 and 3.15.

3.4.6 Limitations and Future Work

Despite promising results, several limitations of the study should be noted:

1. **Computational Constraints:** Due to limited computational resources, evaluation was restricted to DeepLabV3+ with ResNet-18 and ResNet-50 backbones. Future work could explore the method's effectiveness across a wider range of architectures and deeper networks.
2. **Defence Method Comparison:** Although several pre-processing methods were compared, other defence strategies were not included. Broader comparison with state-of-the-art approaches could provide further insight.
3. **Class-wise Performance Variability:** While GMBN delivers strong overall performance, class-wise results reveal inconsistencies. Certain targeted classes recover well, whereas others show limited improvement.
4. **Semantic Complexity:** Semantic segmentation presents unique challenges compared with image classification, requiring pixel-level predictions while preserving spatial and contextual relationships. The varying class-wise performance suggests that further research is necessary to enhance adversarial robustness in segmentation tasks.

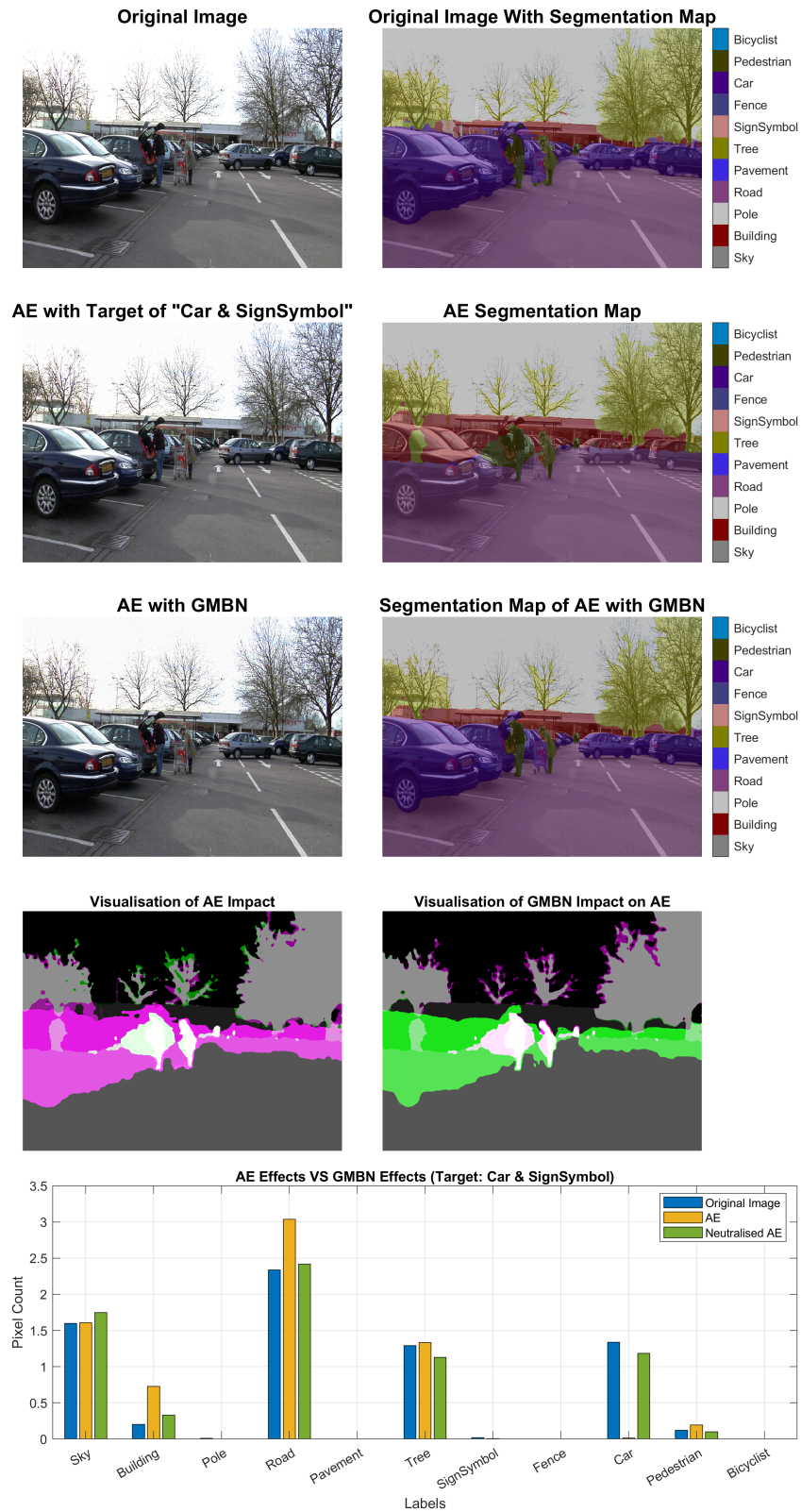


Figure 3.13: Visualisation of the defence performance of GMBN against adversarial attacks in segmentation tasks. The first three rows show the segmentation maps; the fourth row visualises IoU changes; the bar graph presents statistical data of pixel counts.

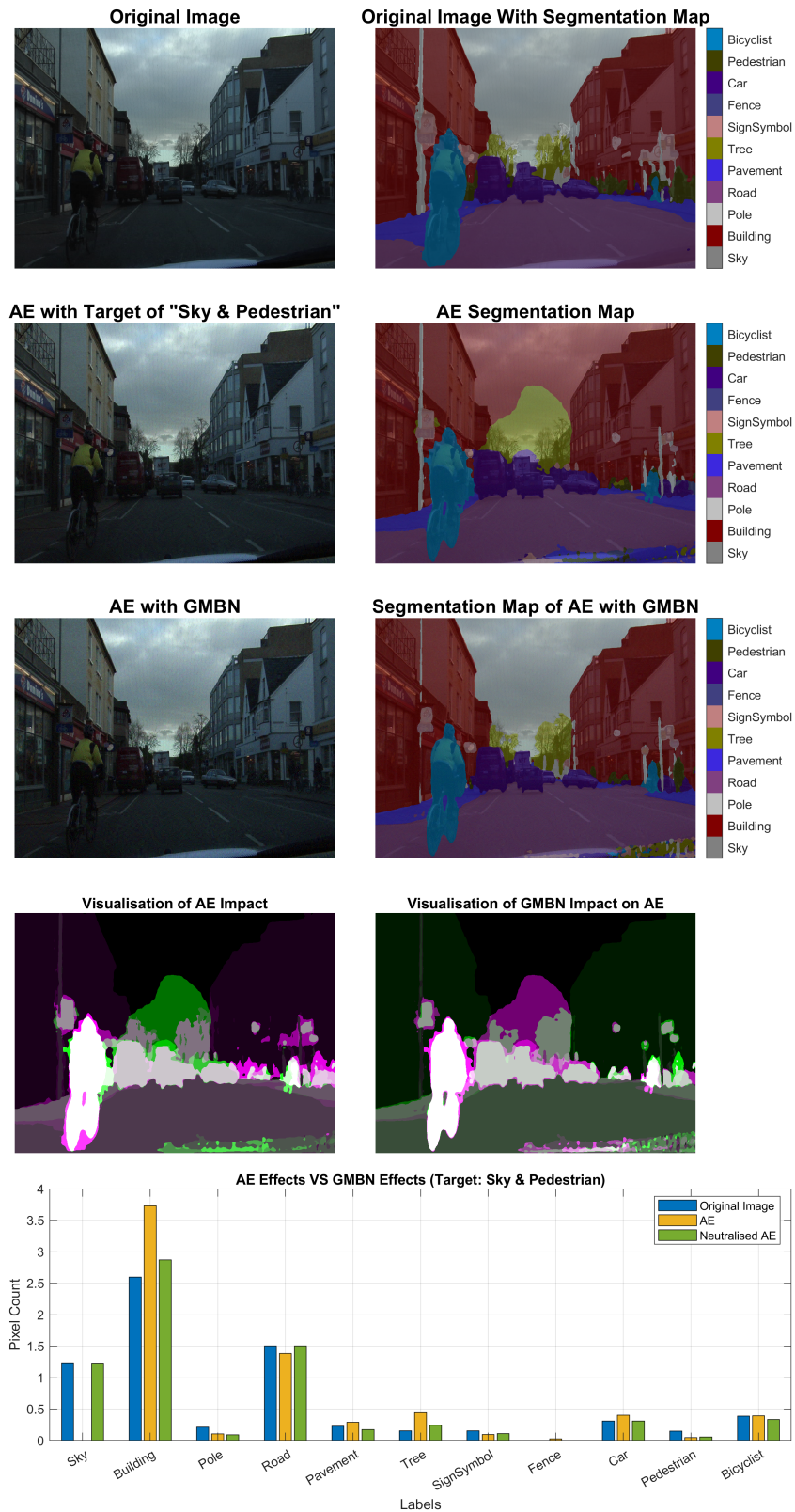


Figure 3.14: Visualisation of the defence performance of GMBN against adversarial attacks in segmentation tasks. The first three rows show the segmentation maps; the fourth row visualises IoU changes; the bar graph presents statistical data of pixel counts.

	Sky	Building	Pole	Road	Pavement	Tree	Sign Symbol	Fence	Car	Pedestrian	Bicyclist
Sky	96.31%	0.45%	1.03%	0.00%	0.00%	2.02%	0.02%	0.00%	0.18%	0.00%	0.00%
Building	0.40%	88.36%	3.30%	0.00%	0.50%	0.76%	2.49%	0.19%	2.11%	1.76%	0.13%
Pole	0.12%	3.75%	84.17%	0.03%	0.49%	3.99%	3.33%	0.02%	2.32%	1.76%	0.02%
Road	0.00%	0.00%	0.08%	92.52%	3.39%	0.00%	0.00%	0.01%	3.93%	0.03%	0.04%
Pavement	0.00%	0.24%	1.00%	5.38%	91.88%	3.63E-05	3.39E-05	0.04%	0.39%	0.95%	0.11%
Tree	4.16%	2.65%	6.01%	0.00%	0.03%	80.38%	3.18%	0.13%	3.26%	0.18%	0.01%
Sign Symbol	4.14E-05	4.21%	3.81%	0.00%	0.00%	1.72%	88.87%	0.03%	1.00%	0.33%	0.02%
Fence	0.00%	12.62%	19.74%	0.00%	1.19%	0.11%	0.00%	64.22%	2.11%	0.00%	0.00%
Car	9.17E-05	1.98%	1.13%	0.22%	0.06%	0.04%	0.11%	0.10%	94.38%	1.28%	0.68%
Pedestrian	0.00%	0.23%	1.04%	9.68E-06	1.53%	0.07%	0.00%	0.00%	0.92%	94.22%	2.01%
Bicyclist	0.00%	0.00%	1.96%	0.00%	0.22%	0.00%	0.00%	0.00%	5.57%	15.03%	77.21%

(a) Clean Image

	Sky	Building	Pole	Road	Pavement	Tree	Sign Symbol	Fence	Car	Pedestrian	Bicyclist
Sky	93.44%	9.81E-05	0.65%	4.63E-05	0.30%	0.97%	1.64%	0.04%	2.94%	0.00%	0.00E+00
Building	24.87%	0.72%	13.40%	4.18E-05	5.44%	12.47%	23.43%	0.18%	18.67%	0.45%	0.37%
Pole	11.51%	0.13%	25.28%	0.00%	15.05%	12.19%	17.16%	0.02%	18.67%	8.09E-06	0.00%
Road	0.20%	0.16%	0.09%	2.32%	36.29%	0.54%	0.06%	1.78%	57.83%	0.02%	0.71%
Pavement	0.09%	0.09%	2.98%	0.02%	83.11%	0.45%	0.36%	0.11%	12.64%	0.04%	1.06E-03
Tree	23.51%	0.24%	5.58%	0.00%	0.82%	34.06%	11.85%	0.60%	23.34%	1.25E-05	0.00%
Sign Symbol	15.36%	0.82%	8.40%	0.00%	2.49%	16.23%	46.16%	0.14%	9.98%	0.01%	0.41%
Fence	0.00%	0.00%	41.43%	0.00%	0.00%	53.28%	5.29%	0.00%	0.00%	0.00%	0.00%
Car	2.63%	0.38%	2.60%	0.02%	22.18%	2.78%	7.29%	0.27%	61.43%	0.17%	0.24%
Pedestrian	3.87E-05	1.94E-05	18.49%	0.00%	31.74%	9.09%	8.63%	0.14%	30.86%	0.40%	0.64%
Bicyclist	0.00%	0.00%	0.58%	0.00%	2.86%	0.74%	32.50%	0.00%	61.14%	1.91%	0.27%

(b) Adversarial Example

Figure 3.15: Confusion matrices for clean image and adversarial example segmentation.

	Sky	Building	Pole	Road	Pavement	Tree	Sign Symbol	Fence	Car	Pedestrian	Bicyclist
Sky	84.83%	12.55%	0.16%	0.28%	0.24%	1.15%	0.31%	3.90E-04	0.43%	4.18E-05	0.00%
Building	10.46%	62.99%	2.33%	0.25%	2.51%	7.77%	4.14%	0.50%	8.08%	0.92%	0.05%
Pole	7.93%	38.40%	14.85%	1.46%	8.02%	9.68%	5.83%	0.25%	9.07%	4.45%	0.07%
Road	0.24%	1.07%	0.09%	67.73%	9.31%	1.99%	0.60%	0.57%	18.25%	0.14%	0.00E+00
Pavement	0.09%	8.95%	2.66%	13.28%	62.41%	0.76%	0.88%	0.06%	9.50%	1.42%	0.00E+00
Tree	10.72%	36.69%	1.44%	1.05%	0.48%	31.11%	4.72%	0.59%	12.98%	0.22%	0.00E+00
Sign Symbol	2.96%	59.48%	1.41%	0.28%	1.28%	7.55%	18.05%	0.59%	7.09%	1.31%	0.02%
Fence	9.16E-04	27.87%	17.11%	0.00%	0.00%	53.28%	0.53%	0.00%	1.12%	0.00%	0.00%
Car	1.87%	21.66%	0.76%	17.46%	4.06%	2.26%	0.53%	0.14%	49.07%	2.16%	0.04%
Pedestrian	0.02%	23.53%	2.70%	6.06%	17.04%	7.26%	0.64%	1.55%	25.83%	15.37%	0.00%
Bicyclist	0.00%	42.26%	0.00%	0.00%	2.46%	1.31%	1.56%	0.00%	44.14%	8.27%	0.00%

(a) GMBN (Proposed)

	Sky	Building	Pole	Road	Pavement	Tree	Sign Symbol	Fence	Car	Pedestrian	Bicyclist
Sky	88.21%	8.82%	0.46%	0.00%	0.34%	0.93%	0.06%	0.00E+00	1.13%	2.17E-05	0.04%
Building	6.33%	56.78%	4.97%	0.07%	2.11%	9.85%	8.06%	0.92%	10.14%	0.57%	0.19%
Pole	3.15%	34.42%	22.14%	0.00%	5.86%	9.80%	5.91%	1.14%	16.89%	0.46%	0.23%
Road	0.13%	0.85%	0.17%	63.30%	9.57%	1.38%	0.65%	1.96%	21.78%	0.19%	5.04E-05
Pavement	0.03%	5.61%	2.43%	8.38%	67.30%	1.17%	0.57%	0.09%	14.05%	0.34%	3.99E-04
Tree	10.40%	33.52%	2.17%	0.01%	0.45%	33.31%	4.43%	0.00%	15.39%	0.07%	2.39E-03
Sign Symbol	0.49%	44.66%	1.33%	0.05%	1.16%	7.03%	38.48%	0.00%	5.84%	0.40%	0.57%
Fence	0.00E+00	37.65%	11.93%	0.00%	0.00%	49.50%	0.92%	0.00%	0.00%	0.00%	0.00%
Car	1.86%	15.44%	0.97%	9.39%	3.94%	7.80%	1.23%	0.05%	57.38%	0.96%	0.98%
Pedestrian	0.00%	18.21%	5.18%	0.63%	17.08%	5.94%	0.73%	0.22%	35.47%	16.53%	0.00%
Bicyclist	0.00%	0.00%	18.76%	0.00%	0.05%	3.36%	4.10%	0.00%	40.55%	0.00%	33.18%

(b) JPEG Compression

Figure 3.16: Confusion matrices for GMBN and JPEG compression defence.

	Sky	Building	Pole	Road	Pavement	Tree	Sign Symbol	Fence	Car	Pedestrian	Bicyclist
Sky	80.54%	16.51%	0.64%	0.00%	0.17%	1.07%	0.15%	0.00E+00	0.91%	3.96E-05	7.69E-05
Building	1.97%	64.53%	3.11%	0.04%	1.07%	13.54%	4.85%	0.84%	9.54%	0.31%	0.19%
Pole	0.91%	33.44%	23.24%	0.00%	2.86%	13.20%	5.44%	0.76%	19.81%	0.09%	0.24%
Road	0.05%	0.76%	0.07%	65.85%	8.97%	2.36%	0.48%	0.86%	20.48%	0.10%	2.17E-04
Pavement	1.52E-05	5.86%	1.62%	8.05%	61.93%	4.13%	0.07%	0.17%	17.60%	0.50%	6.13E-04
Tree	6.94%	36.01%	1.36%	1.72E-05	0.14%	36.15%	4.76%	0.00%	14.40%	0.06%	1.73E-03
Sign Symbol	0.18%	39.11%	0.97%	0.00%	0.22%	8.55%	42.34%	0.00%	7.82%	0.22%	0.59%
Fence	0.00E+00	43.70%	24.55%	0.00%	0.00%	16.51%	2.08%	0.00%	4.79%	8.36%	0.00%
Car	0.94%	13.97%	0.62%	7.20%	1.54%	14.96%	0.57%	0.04%	59.24%	0.63%	0.29%
Pedestrian	0.00%	16.05%	1.75%	0.32%	7.61%	12.81%	0.75%	0.33%	40.82%	19.55%	0.00%
Bicyclist	0.00%	0.01%	7.82%	0.00%	0.12%	5.40%	2.35%	0.00%	32.81%	0.13%	51.36%

(a) Feature Smoothing

	Sky	Building	Pole	Road	Pavement	Tree	Sign Symbol	Fence	Car	Pedestrian	Bicyclist
Sky	92.93%	0.02%	0.98%	0.00%	0.04%	1.50%	1.52%	1.34E-05	3.01%	0.00E+00	0.00%
Building	16.54%	4.39%	14.35%	0.00%	4.63%	15.20%	24.57%	0.37%	19.08%	0.45%	0.42%
Pole	7.93%	0.19%	25.81%	0.00%	12.55%	13.93%	18.17%	0.17%	21.12%	0.05%	0.07%
Road	1.51E-05	6.23E-06	0.07%	2.49%	36.56%	0.09%	0.02%	0.52%	59.60%	0.00%	6.44E-03
Pavement	0.00%	0.11%	1.83%	1.07%	82.70%	0.51%	0.19%	0.09%	13.38%	0.01%	1.12E-03
Tree	17.03%	0.04%	5.55%	0.00%	0.38%	44.58%	11.71%	0.12%	20.61%	0.00%	0.00E+00
Sign Symbol	8.54%	0.05%	9.28%	0.00%	1.83%	17.89%	51.88%	2.48E-05	10.13%	0.00%	0.40%
Fence	0.00E+00	0.00%	38.14%	0.00%	0.09%	48.49%	10.97%	2.31%	0.00%	0.00%	0.00%
Car	1.07%	0.21%	2.85%	3.96E-05	20.32%	3.38%	6.36%	0.04%	65.50%	0.03%	0.23%
Pedestrian	0.00%	9.68E-06	19.92%	0.00%	26.46%	12.35%	10.57%	0.46%	28.98%	0.51%	0.74%
Bicyclist	0.00%	0.00%	0.00%	0.00%	4.18%	2.12%	29.15%	0.00%	64.56%	0.00%	0.00%

(b) Colour Reduction

Figure 3.17: Confusion matrices for feature smoothing and colour reduction defence.

Chapter 4

Combat AEs with Robust Features

4.1 General Introduction

While deep neural networks (DNNs) extract high-dimensional and complex features from images and consistently outperform conventional hand-crafted feature methods in computer vision tasks [124, 125, 126], hand-crafted features possess a vital advantage: strong invariance to various distortions and transformations [127, 128, 129, 130, 131, 132]. This characteristic can be crucial for improving the robustness of DNN models.

In this chapter, we hypothesise that the phenomenon of adversarial examples (AEs) arises from the lack of robustness in the features learned by deep learning models. To examine this hypothesis, we conduct a detailed feature robustness analysis and present experimental evidence supporting it. Based on our findings, we then perform comprehensive experiments to combat AEs through both feature-level and architectural modifications in a variety of real-world scenarios.

Our main contributions are as follows:

1. **Feature Robustness Comparison:** To provide the basis for our hypothesis, we first investigate the layer-by-layer adversarial effects on DNN features by visualising channel-wise variations. We then evaluate the robustness of deep learning features versus several hand-crafted features under adversarial attacks. Our analysis shows that Wavelet Scattering Network (WSN) [11, 133] features exhibit superior robustness against adversarial perturbations, offering valuable insights into improving a

model’s inherent resistance to such attacks.

2. **Hybrid Network Architecture with Feature Fusion:** To further enhance the adversarial robustness of DNN models, we propose a hybrid network architecture—*WSN-ResNet*—which combines WSN and DNN features to achieve both strong classification accuracy and improved robustness. We introduce two dedicated feature fusion strategies: *rescaling addition* and *depth concatenation*. Before fusion, the WSN features are adjusted with learnable scaling and bias parameters. The proposed hybrid networks are then rigorously evaluated under benchmark adversarial attacks.
3. **ROI-Aware 3D ResNet for Medical Imaging:** In medical imaging—particularly for prognosis prediction using 3D CT/MRI volumetric scans—robustness is critical due to the sensitivity of real-world clinical applications. The WSN-ResNet is designed for 2D images and cannot directly handle 3D volumetric data. To address this limitation, we propose a novel *Region-of-Interest (ROI)-Aware 3D ResNet* architecture, purpose-built for COVID-19 prognosis from volumetric CT/MRI scans. The design introduces additional layers that apply masking not only to the input volume but also to intermediate feature maps, thereby reducing feature-layer-level adversarial effects. This enables the model to outperform standard 3D networks in both clean accuracy and adversarial robustness. Furthermore, we propose a preprocessing pipeline for CT scans that uses a saliency coefficient to select lung slices, ensuring the network learns from diagnostically critical thoracic regions and thus mitigating adversarial perturbations.

The roadmap for this chapter is as follows: Section 4.2 presents a comprehensive robustness analysis of deep learning versus hand-crafted features to identify the most robust representation for defending against adversarial attacks. Section 4.3 introduces a novel architectural modification that integrates WSN features with DNN features to enhance inherent robustness. Section 4.4 presents an ROI-aware architecture tailored to medical imaging, aimed at improving both model performance and robustness.

4.2 Feature Robustness Analysis

This section presents a comprehensive comparative analysis of the adversarial robustness of DNN features and traditional hand-crafted features. The experimental evaluation assesses their robustness on different DNNs under benchmark adversarial attacks in different scenarios, aiming to determine the feature with the best robustness against adversarial attacks.

4.2.1 Feature Extraction Methods For Comparison

To evaluate the adversarial robustness of various feature representations, four hand-crafted feature extraction techniques are employed: **SIFT** [134], **SURF** [135], **HOG** [136], and **WSN** [11, 133]. SIFT and SURF are popular hand-crafted feature extraction methods that detect local key points from the images, while HOG focuses on global structural information. WSN is more complex than the former three as it generates multi-scale hierarchical and translation-invariant features. Each method is described in detail as follows.

Scale-Invariant Feature Transform (SIFT)

Scale-Invariant Feature Transform (SIFT) [134] is a widely used hand-crafted feature extraction method that primarily detects local key points from an image. SIFT features are scale and rotation invariant, which makes them robust to geometric changes. Key points are defined by detecting the extrema of the image such as the local maximum or minimum. SIFT computes a local gradient histogram within a region around each key point, resulting in a 128-dimensional descriptor. SIFT is effective in object recognition due to its robustness to geometric transformations and the advantage of capturing local details. However, this method can be computationally expensive. The feature extraction procedures can be expressed as follows. Given an input image $x(m, n)$ (the image is converted into grey if it is RGB), the SIFT features are computed as follows:

1. Scale-Space Construction

The scale-space is constructed by convolving the image with Gaussian kernels:

$$L_S(m, n, \sigma) = G(m, n, \sigma) * x(m, n), \quad (4.1)$$

This produces progressively smoothed versions of the image for different scale values σ .

The Gaussian kernel is defined as:

$$G(m, n, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{m^2+n^2}{2\sigma^2}}. \quad (4.2)$$

This kernel applies isotropic smoothing, suppressing high-frequency details at a given scale.

2. Difference of Gaussians

$$R(m, n, \sigma) = L_S(m, n, k\sigma) - L_S(m, n, \sigma), \quad (4.3)$$

This efficiently approximates the scale-normalised Laplacian of Gaussian, highlighting blob-like structures stable across scales. Here k is a constant multiplication factor for scale (commonly $k = \sqrt{2}$).

3. Keypoint Localisation

Potential keypoint (m, n, σ) is identified as a local extrema of $R(m, n, \sigma)$ in both space and scale, which can be localised by computing the Hessian matrix:

$$H = \begin{bmatrix} \frac{\partial^2 R}{\partial m^2} & \frac{\partial^2 R}{\partial m \partial n} & \frac{\partial^2 R}{\partial m \partial \sigma} \\ \frac{\partial^2 R}{\partial m \partial n} & \frac{\partial^2 R}{\partial n^2} & \frac{\partial^2 R}{\partial n \partial \sigma} \\ \frac{\partial^2 R}{\partial m \partial \sigma} & \frac{\partial^2 R}{\partial n \partial \sigma} & \frac{\partial^2 R}{\partial \sigma^2} \end{bmatrix}. \quad (4.4)$$

The Hessian encodes second-order derivatives, enabling precise localisation and edge-response elimination.

4. Orientation Assignment

The gradient magnitude $\nabla_g(m, n)$ and orientation $\nabla_o(m, n)$ of $L_S(m, n, \sigma_k)$ (where σ_k is the scale at the detected keypoint) at each key point can be computed:

$$\nabla_g(m, n) = \sqrt{\left(\frac{\partial L_S}{\partial m}\right)^2 + \left(\frac{\partial L_S}{\partial n}\right)^2}, \quad \nabla_o(m, n) = \arctan\left(\frac{\frac{\partial L}{\partial n}(m, n, \sigma_k)}{\frac{\partial L}{\partial m}(m, n, \sigma_k)}\right). \quad (4.5)$$

These values provide a consistent orientation reference, ensuring rotation invariance.

5. Feature Descriptor

Around each key point, a region is divided into sub-regions. Within each sub-region S , gradients are weighted and binned by orientation:

$$H_j = \sum_{(m',n') \in S} w(m',n') \nabla_g(m',n') \delta(\nabla_o(m',n') - \nabla_j^o), \quad (4.6)$$

This forms orientation histograms for each sub-region, with weights favouring gradients near the keypoint.

Concatenating these histograms from all sub-regions yields a feature vector:

$$f_{\text{SIFT}} = [H_1, H_2, \dots, H_N], \quad (4.7)$$

The descriptor is the concatenation of all local orientation histograms.

$$\|f_{\text{SIFT}}\|_2 = 1, \quad (4.8)$$

The final descriptor is L_2 -normalised to improve invariance to illumination changes. The resulting $f_{\text{SIFT}} \in \mathbb{R}^{128}$ is computed from a 16×16 window around each keypoint.

Speeded-Up Robust Features (SURF)

Speeded-Up Robust Features (SURF) [135] is a fast alternative to SIFT, designed to reduce computational cost while maintaining robustness to scale and rotation changes. SURF detects key points using a Hessian-based approach, focusing on regions with high-intensity changes. It computes descriptors by analysing the distribution of Haar wavelet responses in the neighbourhood of each key point. SURF is computationally more efficient than SIFT but compromises robustness to noise and high-frequency perturbations compared to SIFT. The SURF features are computed as follows:

1. Integral Image Construction

The integral image is computed as:

$$I_x(m, n) = \sum_{m'=1}^m \sum_{n'=1}^n x(m', n'). \quad (4.9)$$

2. Scale Space Construction

Using box filters B_σ and the integral image, the approximated determinant of Hessian is:

$$L_B(m, n, \sigma) = B_{mm}(\sigma)B_{nn}(\sigma) - (wB_{mn}(\sigma))^2, \quad (4.10)$$

where B_{mm}, B_{nn}, B_{mn} are box filter approximations of second-order Gaussian derivatives, and w is a weight factor.

3. Interest Point Detection

Points are localised using non-maximum suppression in a $3 \times 3 \times 3$ neighbourhood:

$$H_B = \begin{bmatrix} \frac{\partial^2 L_B}{\partial m^2} & \frac{\partial^2 L_B}{\partial m \partial n} \\ \frac{\partial^2 L_B}{\partial m \partial n} & \frac{\partial^2 L_B}{\partial n^2} \end{bmatrix}. \quad (4.11)$$

4. Orientation Assignment

The Haar wavelet responses in both directions are:

$$\begin{aligned} r_m(m, n) &= H_m(m, n, \sigma) * I_x(m, n), \\ r_n(m, n) &= H_n(m, n, \sigma) * I_x(m, n), \end{aligned} \quad (4.12)$$

where H_m and H_n are Haar wavelets.

5. Feature Descriptor

The SURF descriptor is constructed by:

$$f_{\text{SURF}} = [\sum r_m, \sum r_n, \sum |r_m|, \sum |r_n|]_{4 \times 4}, \quad (4.13)$$

$$\|f_{\text{SURF}}\|_2 = 1, \quad (4.14)$$

where $f_{\text{SURF}} \in \mathbb{R}^{64}$ is the final normalized descriptor.

Histogram of Oriented Gradients (HOG)

Histogram of Oriented Gradients (HOG) [136] is a feature extraction technique that captures the structural information in an image by encoding the distribution of gradient orientations. The feature extraction process has three steps. First, the image is divided into a grid of regions; the second step computes

a histogram of gradients for each region; the third step groups those histograms into normalised overlapping blocks thereby enhancing robustness to illumination changes. HOG is advantageous in capturing shape and texture information, making it popular in object detection tasks. However, it is limited in its sensitivity to deformations compared to SIFT and SURF. HOG features are computed as follows:

1. Gradient Computation

The image gradients are computed:

$$\begin{aligned} g_m(m, n) &= x(m+1, n) - x(m-1, n), \\ g_n(m, n) &= x(m, n+1) - x(m, n-1), \end{aligned} \quad (4.15)$$

$$\begin{aligned} \|\nabla\|(m, n) &= \sqrt{g_m(m, n)^2 + g_n(m, n)^2}, \\ \varphi(m, n) &= \arctan\left(\frac{g_n(m, n)}{g_m(m, n)}\right). \end{aligned} \quad (4.16)$$

2. Cell Histograms

For each cell C , orientation histograms are computed:

$$H_c(k) = \sum_{(m,n) \in C} w(m, n) \|\nabla\|(m, n) \delta(\varphi(m, n) - \varphi_k), \quad (4.17)$$

where $w(m, n)$ is a weighting function, φ_k are orientation bins, and δ is the indicator function.

3. Block Normalisation

For each block B containing multiple cells, the concatenated histograms are normalised:

$$f_B = \frac{[H_{c1}, H_{c2}, H_{c3}, H_{c4}]}{\sqrt{\|[H_{c1}, H_{c2}, H_{c3}, H_{c4}]\|_2^2 + e}}, \quad (4.18)$$

where e is a small constant to avoid division by zero.

4. Final Descriptor

The final HOG descriptor is the concatenation of all normalised block features:

$$f_{\text{HOG}} = [f_{B1}, f_{B2}, \dots, f_{B_N}], \quad (4.19)$$

$$\|f_{\text{HOG}}\|_2 = 1, \quad (4.20)$$

where f_{HOG} is typically a high-dimensional vector.

Wavelet Scattering Transform for Feature Extraction

The Wavelet Scattering Transform or Wavelet Scattering Networks (WSN) [11, 133] use predefined wavelet and scaling filters to capture both local and global information, thereby generating robust and distortion-invariant features from time series and image data for machine/deep learning applications. Unlike DNNs, where convolution filters are learned during training, wavelet scattering filters are predetermined, where the scattering coefficients are invariant to translations and small deformations while retaining discriminative feature representations of the input. WSN is advantageous in capturing high-dimensional complex feature representations while maintaining interpretability compared to DNN features. However, since WSN relies on fixed wavelet filters for robust feature extraction, it is difficult to adapt to deep learning tasks. Specifically, the output dimensions of WSN are affected by parameters such as invariance scales, filter banks and rotation numbers, which require substantial experiments to be acquired. The wavelet scattering transform can be expressed through the following equations:

Given an input image x , the image is first convolved with a family of wavelet filters $\psi(j, k)$, where j denotes the scaling number and k denotes the orientation number. The first-order wavelet transform is given by:

$$U_1 = |x * \psi_{j,k}|, \quad (4.21)$$

where $*$ denotes the convolution operation, and $|\cdot|$ represents the modulus operation. To capture hierarchical representations for higher orders, the wavelet transform is recursively applied to the output of the previous step:

$$U_m = |U_{m-1} * \psi_{j',k'}|, \quad (4.22)$$

where U_m represents the m -th order scattering coefficients, and $\psi_{j',k'}$ denotes the wavelet filters at scale j' and orientation k' . To improve invariance to translations, the scattering coefficients are convolved with a scaling function

ϕ_S :

$$S_m = U_m * \phi_S, \quad (4.23)$$

where S_m represents the m -th order scattering coefficients after low-pass filtering. To extract robust, translation-invariant features, the scattering coefficients across all orders are aggregated:

$$f_{m'} = \{S_0, S_1, S_2, \dots, S_{m'}\}, \quad (4.24)$$

where f is the set of extracted features, and m' is the maximum order included. To summarise, the complete scattering transform for feature extraction using wavelet scattering transform can be denoted as:

$$f = \bigcup_{m=0}^{m'} (|x * \psi_{j_1, k_1} * \dots * \psi_{j_m, k_m}| * \phi_S), \quad (4.25)$$

The final step obtains a feature matrix by applying pooling operations (*e.g.*, averaging or max pooling) for dimension reduction:

$$f = \text{mean}(f_{m'}). \quad (4.26)$$

f is the final output feature matrix.

4.2.2 Methodology

The detailed procedures of the feature robustness analysis can be described as follows:

- Prepare the dataset $\mathcal{X}$ to generate AES, load an input image $x_i \in \mathcal{X}$ and a pre-trained DNN D_θ that is trained with the dataset $\mathcal{X}$, the ground truth prediction is given by $y_i = D_\theta(x_i)$.
- Use benchmark attacks such as FGSM, PGD and CW attacks to generate AEs x'_i in different attacking conditions, predicted by the target DNN $y'_i = D_\theta(x'_i)$, if $y'_i \neq y_i$, output the AE x'_i .
- Extract features f_c and f_a from clean and adversarial images x_i and x'_i , respectively, using varied feature extraction methods as described in section 4.2.1.

- Calculate the robustness metrics for each feature type to quantify the sensitivity to adversarial perturbations.

The detailed procedures of the experiment are described in Algorithm 4.

Algorithm 4 Feature Robustness Analysis

Input: image x , pre-trained DNN D_θ .

Configure Attacking Parameters: Parameters vary for different attacks

Compute Adversarial Perturbation $r : x' = x + r, y' \leftarrow \arg \max(D_\theta(x'))$,

Ensure: $y' \neq y$,

Configure Parameters for varied Feature Extraction Methods

Extract features from x and x' using varied Extraction Methods:

Obtain $f_c^1, f_c^2, \dots, f_c^m$ and $f_a^1, f_a^2, \dots, f_a^m$,

(m corresponds to the number of feature extraction methods utilised)

Compute robustness metrics: **MED** and **MAE** for $f_c^1, f_c^2, \dots, f_c^m; f_a^1, f_a^2, \dots, f_a^m$.

4.2.3 Experiment Setup

Dataset

A subset of the ImageNet dataset [19] is utilised for the experiment stage, and the subset contains 10 000 images, which are randomly selected from the dataset to balance computational efficiency with a sufficient representation of content variety and quality.

Pre-trained DNN Models

To assess feature robustness across the spectrum of DNN architectures, a diverse set of pre-trained DNN models is used, which includes **ResNet-50** [14], **VGG-16** [10], **Inception-v3** [121], **GoogLeNet** [118], **EfficientNet** [15] and **NASNet-Large** [122]. This extensive selection of models allows a comprehensive examination of feature robustness across different architectures, thereby providing a better understanding of the factors influencing adversarial robustness.

Evaluation Metrics

In this experiment, two metrics are utilised to measure feature robustness, which can be expressed as follows:

- **Maximum Element-wise Difference (MED):**

$$\text{MED} = \max_i |f_{\text{clean}}^i - f_{\text{adv}}^i|, \quad (4.27)$$

where f_{clean} and f_{adv} represent the extracted feature of clean and adversarial images, respectively. MED highlights the distinct impact of adversarial perturbations by identifying the maximum feature deviation across all components, offering insight into the significant impact of adversarial perturbation.

- **Mean Absolute Error (MAE):**

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |f_{\text{clean}}^i - f_{\text{adv}}^i|, \quad (4.28)$$

where n denotes the dimensionality of the feature vectors. MAE provides an average measure of deviation, representing overall stability in the feature representations by capturing the cumulative impact of perturbations on feature values.

Adversarial Attacks Configuration

Table 4.1: *Parameters for Adversarial Attacks*

Attack	Parameter	Value	AEs Generated
FGSM	Epsilon (ϵ)	[32, 64]	4,000
PGD	Epsilon (ϵ)	[16, 32]	4,000
	Step Size (α)	0.25	
	Iterations	[50, 100]	
CW	Confidence	25	2,000
	Learning Rate	0.01	
	Iterations	[100, 250]	

To thoroughly evaluate feature robustness, three benchmark adversarial attacks are considered, which include FGSM [7], PGD [9] and CW attack [6]. These attacks are implemented in a variety of attacking conditions and the ϵ is the primary parameter to control the level of the adversarial perturbation.

Note that for each attack, the parameters vary; thus, detailed configurations and parameter settings for each specific attack are exhibited in Table.4.1 for better understanding.

Parameter Settings

Table 4.2: *Parameters for Feature Extraction Methods. For WSN, parameters are specified for two 2-D Morlet [4] filter banks used in feature extraction.*

Method	Parameter	Value
SIFT	Descriptor Size	128
SURF	Descriptor Size	64
HOG	Cell Size	8×8
	Block Size	2×2
	Orientation Bins	9
WSN	Invariance Scale	[112 112]
	Number of Rotations	[8 8]

Table 4.2 illustrates the parameter settings for all feature extraction methods utilised in the experiment. The parameters are pre-defined to ensure a complete comparison between DNN features and hand-crafted features. The default parameter settings are utilised for SIFT, SURF and HOG feature extraction as they are optimised for the trade-off between robustness and computational overheads.

Adaption of Wavelet Scattering For Feature Extraction

As previously mentioned in section.4.2.1, WSN requires adaptation for specific tasks. For the experiment of feature robustness analysis, the WSN algorithm is adapted as follows:

1. The input image is resized to the required input size of the DNN, typically $224 \times 224 \times 3$ (exceptions exist such as GoogleNet for $227 \times 227 \times 3$ and NASNet for $331 \times 331 \times 3$).

2. The wavelet scattering transform parameters are configured, where the main parameters include *Invariance Scale* = 112, *Number of Rotation* = [8 8], and utilisation of two 2-D *Morlet* [4] filter banks.
3. The scattering coefficients are computed using the process described in section.4.2.1.
4. The dimensions of the output feature matrix are modified to align with the requirements by using operations such as averaging or pooling. In this experiment, averaging is applied to the fourth dimension of the feature matrix and max pooling to the former three dimensions to generate a compact feature matrix with the dimension of $3 \times 3 \times 681$.

Visualisation of Different Features



Figure 4.1: Visualisation of different feature maps extracted from a clean image and its corresponding adversarial example. DNN feature maps are visualised using Grad-CAM to highlight class-discriminative regions. SIFT and SURF maps display the ten strongest keypoints, respectively, while HOG maps are shown by superimposing oriented gradient vectors to illustrate local edge orientation patterns.

Different feature extraction methods utilised in the experiment are illustrated through varied visualisation techniques, as shown in Figures.4.1 and

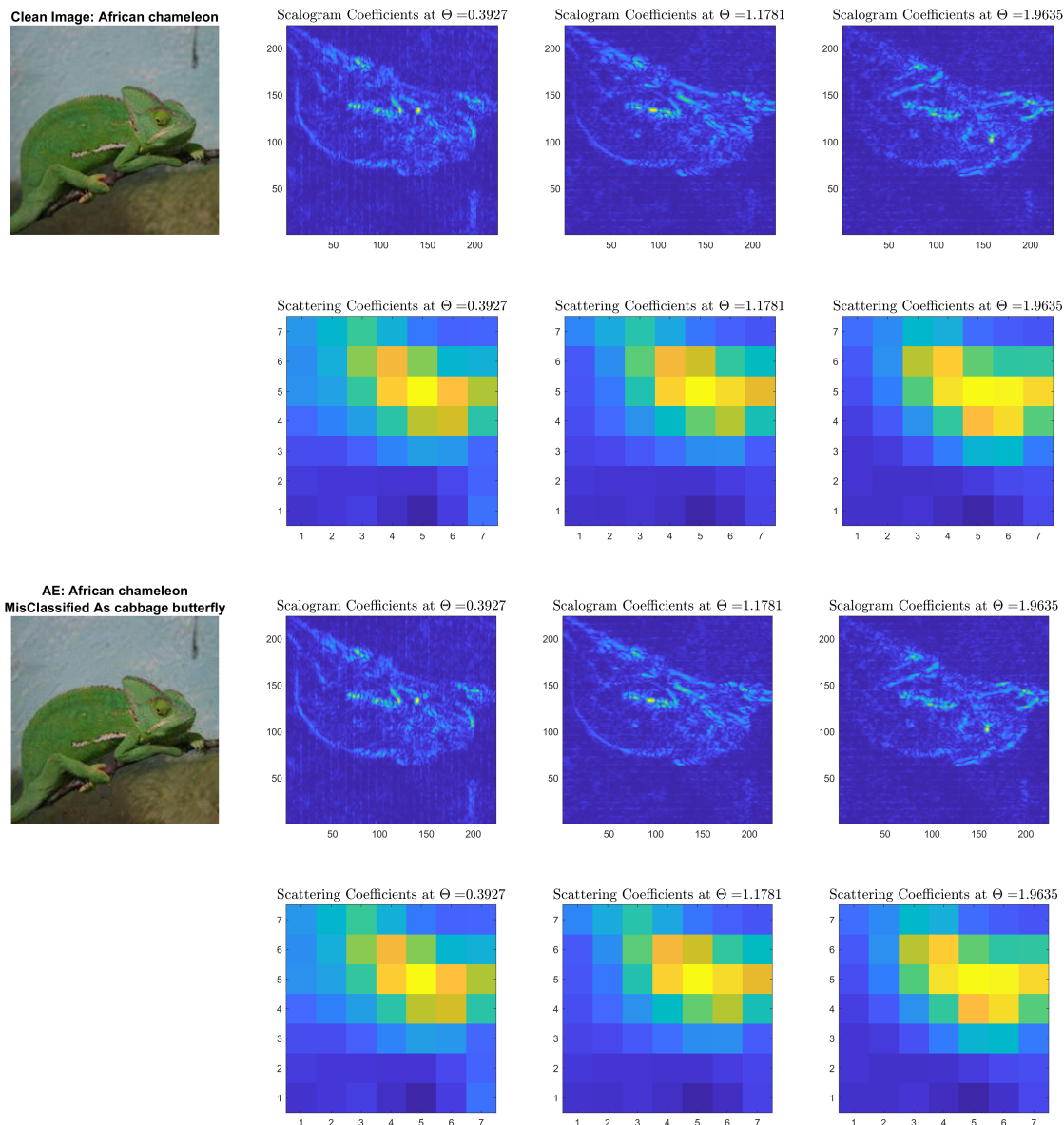


Figure 4.2: Visualisation of the scalogram and scattering coefficients for a clean image and its adversarial example, shown for three representative rotation angles out of eight (due to limited space). Scattering coefficients were used for feature extraction, while scalogram coefficients are included to illustrate the full wavelet transform process for academic rigour.

4.2. These visualisations aim to provide intuitive interpretability of the analysis. It is important to note that the actual features utilised in the experiments differ from these visualised versions.

For the DNN features, Gradient-Weighted Class Activation Mapping (Grad-

CAM) [137] is employed to visualise the regions of interest (ROI) of the network’s predictive decision. For traditional computer vision approaches, SIFT and SURF features are demonstrated by overlaying their ten strongest feature points onto the clean images and AEs. The HOG features are visualised by superimposing the oriented gradients onto the image, showing how the method captures local features through gradient distributions.

The WSN feature matrices are challenging to visualise for human comprehension due to their mathematical complexity. Instead of directly visualising the features, their characteristics are demonstrated through the scalogram and scattering coefficients across different rotation angles in the scattering transform. As shown in Figures.4.2, these visualisations provide insights into how the WSN extract image information at different scales and rotation orientations. The scalogram coefficients exhibit the detailed wavelet decomposition, whilst the scattering coefficients exhibit robust and low-pass representation of these features. For additional examples and illustrations of these feature extraction methods, please refer to Figures.B.1, B.2, B.3, B.4, B.5 and B.6 in Appendix B.

4.2.4 Experimental Results

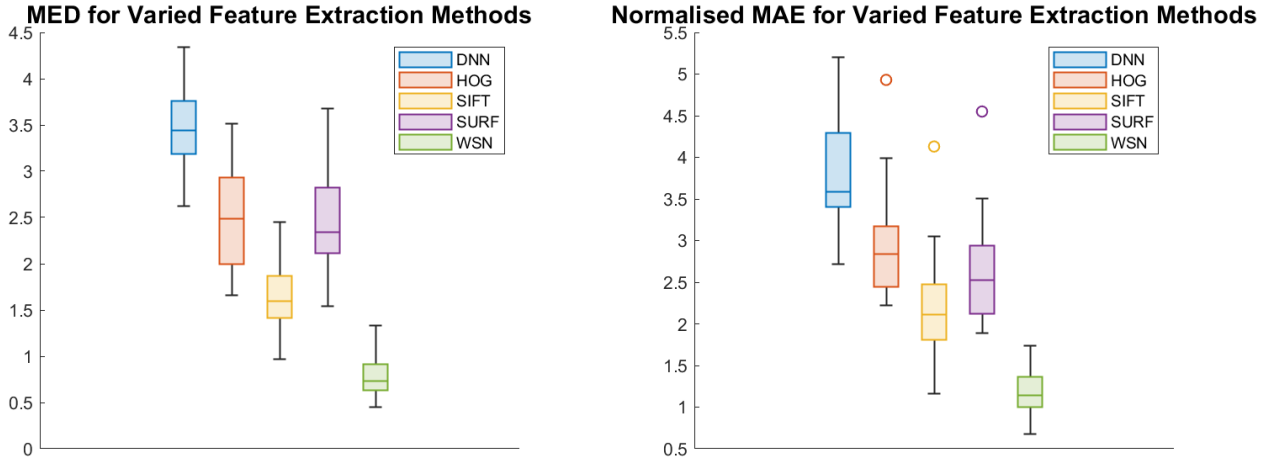


Figure 4.3: Comparison of the robustness of different feature extraction methods across all models and attacks. (**Note:** Smaller values indicate better robustness against adversarial attacks)

Tables.4.3 and 4.4 illustrate the MED and normalised MAE values across different feature extraction methods and DNN models. The distribution of the

Table 4.3: Maximum Element-wise Difference (MED) Under Different Attacks

Feature	Model	FGSM		PGD		CW
		$\epsilon = 16$	$\epsilon = 32$	$\epsilon = 16$	$\epsilon = 32$	
DNN	VGG-16	3.36	3.52	2.96	3.52	2.98
	ResNet-50	3.28	3.98	3.25	3.57	3.25
	GoogLeNet	3.46	4.34	3.03	3.76	3.40
	EfficientNet	3.55	3.92	2.94	3.82	3.19
	InceptionV3	3.54	3.94	2.75	3.77	2.62
	NASNet-Large	3.55	3.78	2.99	3.42	3.29
SIFT	VGG-16	1.64	1.79	1.41	1.22	1.41
	ResNet-50	1.60	1.81	1.61	2.45	1.68
	GoogLeNet	1.58	2.02	1.89	2.07	2.33
	EfficientNet	2.32	1.48	0.97	1.46	1.59
	InceptionV3	1.08	1.72	1.33	1.29	1.49
	NASNet-Large	1.90	1.87	1.43	1.35	1.52
SURF	VGG-16	2.82	2.36	2.27	1.82	2.26
	ResNet-50	2.25	2.73	2.42	3.44	2.73
	GoogLeNet	2.30	3.45	2.08	3.68	2.91
	EfficientNet	2.94	2.33	1.54	2.11	2.23
	InceptionV3	1.89	2.40	1.69	1.98	1.77
	NASNet-Large	2.87	3.04	2.31	2.36	2.45
HOG	VGG-16	2.17	2.99	1.66	1.73	2.39
	ResNet-50	2.93	2.52	2.29	3.09	2.53
	GoogLeNet	2.98	2.45	2.28	2.75	2.00
	EfficientNet	2.68	1.66	1.98	2.07	2.94
	InceptionV3	2.27	3.51	1.79	1.77	3.31
	NASNet-Large	2.64	3.25	2.55	1.93	2.62
WSN	VGG-16	0.73	0.69	0.47	0.52	0.62
	ResNet-50	0.89	0.63	0.45	0.73	1.29
	GoogLeNet	0.99	0.71	0.67	0.77	1.33
	EfficientNet	0.93	1.09	0.63	0.77	0.62
	InceptionV3	0.54	0.63	0.53	0.52	0.74
	NASNet-Large	0.91	1.12	0.75	0.75	0.63

data is also shown in Figure.4.3 for a clearer view. It is worth emphasising that for both metrics, lower values indicate better robustness against adversarial perturbations. Several key observations can be summarised based on these extensive experimental results:

1. WSN features demonstrate superior robustness compared to other fea-

Table 4.4: Normalised Mean Absolute Error (MAE) Under Different Attacks

Feature	Model	FGSM		PGD		CW
		$\epsilon = 16$	$\epsilon = 32$	$\epsilon = 16$	$\epsilon = 32$	
DNN	VGG-16	4.47	4.55	3.47	3.45	3.31
	ResNet-50	4.33	4.31	3.42	3.40	3.69
	GoogLeNet	4.13	4.29	3.50	3.62	4.35
	EfficientNet	3.80	3.09	2.80	2.72	4.06
	InceptionV3	3.47	3.60	2.76	2.95	3.41
	NASNet-Large	4.74	5.20	3.58	3.66	3.11
SIFT	VGG-16	2.34	3.05	1.81	2.50	2.14
	ResNet-50	2.47	2.94	1.98	2.30	2.11
	GoogLeNet	2.28	2.82	2.11	2.16	2.43
	EfficientNet	1.97	1.73	1.35	1.16	1.68
	InceptionV3	1.97	1.50	1.91	1.83	2.83
	NASNet-Large	2.56	4.13	1.68	1.88	1.45
SURF	VGG-16	2.77	2.94	2.47	3.01	2.50
	ResNet-50	3.29	2.92	2.18	2.49	2.11
	GoogLeNet	2.88	3.39	2.87	2.13	2.80
	EfficientNet	2.34	2.11	1.89	2.56	2.12
	InceptionV3	3.04	1.90	2.04	2.31	3.08
	NASNet-Large	3.51	4.55	2.61	1.99	2.07
HOG	VGG-16	2.93	3.60	2.70	3.21	2.90
	ResNet-50	2.67	3.68	2.99	3.80	2.28
	GoogLeNet	3.35	3.99	2.22	2.60	3.17
	EfficientNet	2.38	2.53	2.73	3.14	2.44
	InceptionV3	2.57	2.79	2.34	2.24	3.05
	NASNet-Large	3.12	4.93	2.43	2.24	2.88
WSN	VGG-16	1.19	1.32	1.27	1.57	1.57
	ResNet-50	1.44	1.74	1.36	1.73	1.10
	GoogLeNet	1.30	1.47	0.95	1.02	1.03
	EfficientNet	1.08	1.08	0.76	0.70	1.33
	InceptionV3	0.72	1.02	0.68	0.70	0.99
	NASNet-Large	1.18	1.22	1.00	1.02	1.42

tures, achieving the lowest MED and MAE across all DNN models and adversarial attacks, with MED values ranging from 0.45 to 1.33 and normalised MAE values ranging from 0.68 to 1.74.

2. DNN features demonstrate the highest sensitivity to adversarial attacks across all DNN models, with MED values ranging from 3.28 to 3.55 and

normalised MAE values ranging from 3.47 to 4.74.

3. For traditional hand-crafted features, SIFT features exhibit moderate robustness with MED ranging from 1.08 to 2.32 and MAE ranging from 1.35 to 2.94. Whereas SURF and HOG features demonstrate less robustness, with MED of [1.54 3.04] and [1.66 3.51], MAE of [1.89 4.55] and [2.22 4.93], respectively.
4. DNN features exhibit the highest vulnerability since adversarial attacks are intentionally crafted through end-to-end optimisation. In contrast, hand-crafted features do not offer such direct optimisation paths.

Several noteworthy perspectives deserve particular emphasis to ensure a correct understanding of the results and findings:

1. **Objectives of the Feature Robustness Analysis:** This experiment specifically focuses on comparing the adversarial robustness of different feature extraction methods, rather than evaluating the robustness of various DNN architectures. The primary aim is to improve DNN models' adversarial robustness by using robust features. Hence, while different DNN architectures are utilised in the experiments, model-wise performance comparisons are not discussed as they fall outside the scope of the research objectives.
2. **Implementation Considerations:** In the practical implementation of adversarial attacks, an optimised while-loop construction mechanism is employed for constructing adversarial perturbations, meaning the AE is immediately output once the misclassification occurs. This approach aims to enhance computational efficiency while maintaining attack effectiveness. It is important to note that due to these settings, higher ϵ values in perturbation level do not necessarily cause increased feature sensitivity.
3. **Model-Specific AE Construction:** For each row in the results tables, the specified DNN model serves as the exclusive model for constructing adversarial examples. Hence, hand-crafted feature extraction must be performed independently for each model's AEs, as these perturbations are unique and model-dependent. This setting ensures a fair and

comprehensive evaluation of feature robustness across different attack scenarios.

4. **Limitations:** The selection of feature extraction techniques in this study was constrained by computational power and, more importantly, aligned with the research objectives. Since this analysis aims to provide a theoretical foundation for subsequent research, not all other possible feature extraction methods are considered as they would be both computationally expensive and tangential to the primary research goals.

Summary

The findings from this experiment provide insights that inform the development of robust feature extraction techniques for DNN model robustness improvement, which are explored in subsequent sections of this thesis. In conclusion, the comprehensive analysis demonstrates that WSN features represent the most robust feature extraction method against adversarial attacks. These key insights make WSN features a promising choice for applications requiring adversarial robustness. The next section explores how to leverage these insights to improve DNN models' adversarial robustness through feature fusion and architectural modifications.

4.3 Leveraging Robust Features to Enhance DNNs' Robustness

Based on the comprehensive feature robustness analysis in Section 4.2, these findings are leveraged to enhance the adversarial robustness of DNNs. The analysis demonstrated that amongst various feature extraction methods, WSN features consistently outperform other feature types in terms of robustness against adversarial perturbations. Given this clear advantage, the focus is placed on using WSN features to improve the adversarial robustness of DNN models. The contributions of this section are outlined as follows:

1. A novel hybrid WSN ResNet architecture is presented to improve the adversarial robustness of DNNs by integrating WSN features with DNN features, while maintaining decent model classification performance.

2. To facilitate the integration of DNN features and WSN features, two tailor-made feature fusion approaches are introduced: scaling addition and depth-concatenation. These approaches are specifically designed to maintain the salient characteristics of both feature types whilst achieving a smooth combination.
3. To achieve a good trade-off between overall model performance and adversarial robustness, the hybrid ResNets are retrained on a 100-class subset of ImageNet and 70% of the whole ImageNet dataset (1000-class), respectively. The experimental results demonstrate that the overall adversarial robustness of the hybrid architecture has significantly improved compared to the original ResNets.

4.3.1 WSN Hybrid ResNet Architecture

As previously mentioned in Chapter.2 and Section.3.1.1, no established work has identified the root cause of AE, thereby proposing an effective method to solve the problem. In the previous Chapter.3, it was hypothesised that adversarial perturbation is a specific noise type that DNNs are spontaneously vulnerable to. In this Section, it is hypothesised that the root cause of AEs is due to the inherent deficit of the features learnt by the DNN models. To clarify, let $D_\theta : \mathcal{X} \rightarrow \mathcal{Y}$ be a DNN with parameters θ , where $\mathcal{X}$ is the input space, and $\mathcal{Y}$ is the output space. Let f_x represent the features learned for input x . For clean samples, the DNN learns discriminative features such that:

$$|f_{x_i} - f_{x_j}| \gg 0, \quad \text{when } y_i \neq y_j. \quad (4.29)$$

However, for adversarial examples $x' = x + r$, there exists a perturbation r where $\|r\| \leq \epsilon$ such that:

$$|f_x - f_{x'}| \gg 0, \quad \text{even when } \|r\| \leq \epsilon, \quad (4.30)$$

which maximises the chance of misclassification, such that:

$$y \neq y', \quad y = \arg \max(D_\theta(x)) \text{ and } y' = \arg \max(D_\theta(x')), \quad (4.31)$$

where the sensitivity of learned features against adversarial perturbation has been tested in the previous Section.4.2 where it is concluded that f_{WSN} is the feature type that demonstrates the best robustness, which can be denoted as

$$|f_{\text{WSN}}^x - f_{\text{WSN}}^{x'}| \approx 0, \quad \text{when } \|r\| \leq \epsilon. \quad (4.32)$$

Hence, it is hypothesised that the existence of AE is caused by the lack of robustness of the features learnt by the DNNs. While DNNs excel at learning discriminative complex features during the training process, these learnt features are experimentally validated to be highly sensitive to adversarial perturbations in the analysis in Section.4.2. Hence, the enhancement of overall robustness by integrating DNN features with WSN features is fundamentally motivated by those key insights. However, WSN features have traditionally been limited in their utilisation for deep learning tasks due to several challenges:

- **Feature Representation:** WSN decomposes images into multi-scale components depending on frequency bands and orientations, whereas DNN features are typically data-driven and task-specific, high-level representations of the image. This distinct difference makes these two feature types vary in their semantic representations.
- **Incompatibility in Combination:** The multi-scale nature of WSN decomposition generates features at varying scales, which cannot be mathematically combined with the DNN features since their scales and dimensionalities are substantially different. A straightforward approach to combining DNN and WSN features may result in a computational error or feature dominance.
- **Limited Performance of WSN Features:** Though WSN features are experimentally validated to be robust, no established study has verified that WSN can achieve performance comparable to DNNs in complex multi-class real-world scenario image classification tasks(*i.e.* ImageNet [19], CamVid [123], Volumetric CT/MRI, *etc.*).

To address these challenges, a novel hybrid WSN ResNet architecture with two tailor-made feature fusion techniques specifically designed to integrate

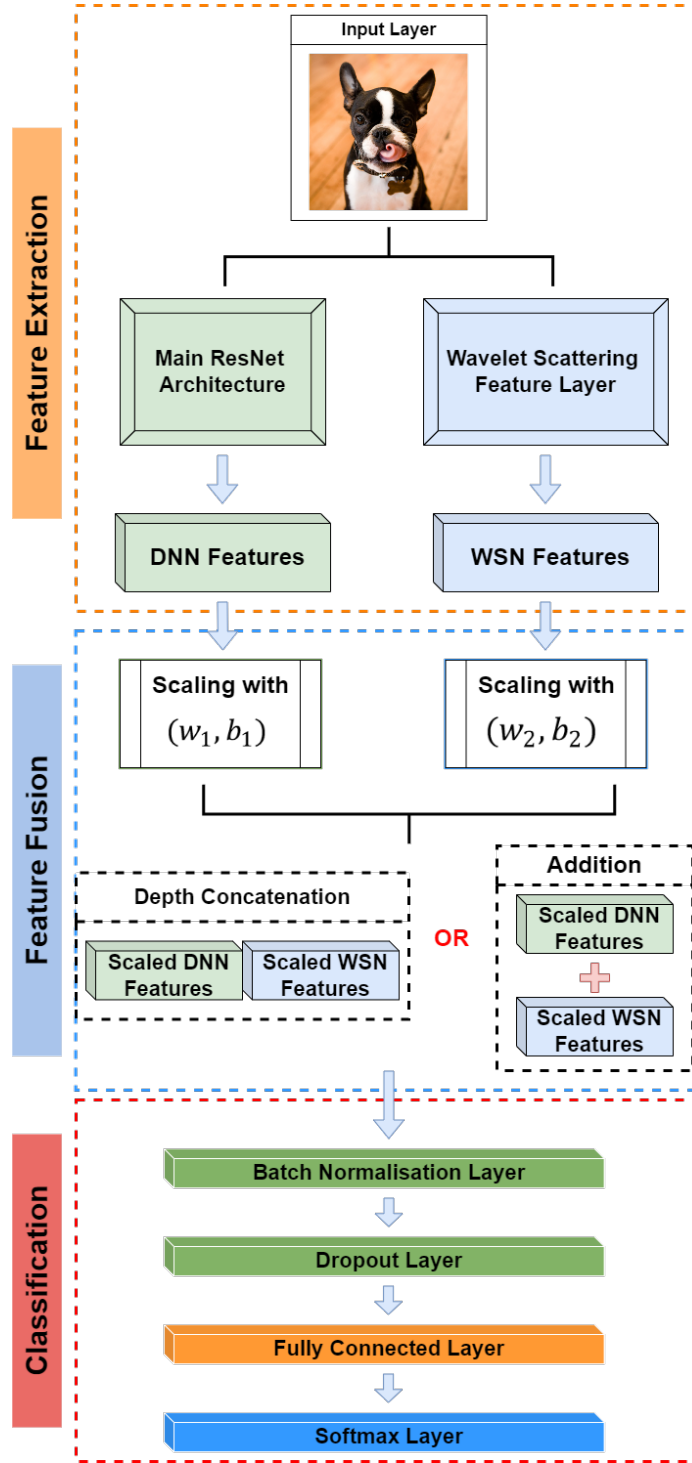


Figure 4.4: The architecture of the proposed hybrid network combining ResNet and Wavelet Scattering Network. The model integrates DNN features with WSN features using two fusion strategies: Rescaling+Addition and Depth Concatenation. This combination leverages WSN's stability and ResNet's hierarchical feature learning, aiming to enhance adversarial robustness.

WSN and DNN representations whilst maintaining the performance of the original DNN is proposed, named Hybrid WSN ResNet. Figure.4.4 illustrates the proposed hybrid architecture as well as two feature fusion methods, detailing the main contribution of this work. The design of the hybrid architecture includes the following technical mechanisms:

1. **Dual Feature Extraction:** The whole spectrum of Pre-trained ResNet models(ResNet-18, 50 and 101) is utilised as the backbones to generate DNN features; meanwhile, a customised wavelet scattering layer is utilised to extract WSN features, where the layer performs a 2D scattering transform to the image with Morlet wavelets, generating a feature matrix of scattering coefficients. The feature matrix is typically 4-D; max pooling is used to the first to third dimensions and averaging to the fourth dimension, thereby reducing its dimensions into a feature vector.
2. **Learable Parameters for Feature Balance:** To avoid dominance by either feature type, four learnable parameters ω_1 , ω_2 , b_1 and b_2 are introduced to adjust the scale and offset of DNN and WSN features respectively before fusion. This design prevents the overwhelming of either feature type.
3. **Adaptive Dimensionality Adjustment of WSN Features:** For the depth-concatenation feature fusion approach, the length of the WSN feature vector is resized to one-fourth of the DNN feature vector. This ensures that WSN features do not overwhelm the fused representation, thereby achieving a balance between robustness and classification accuracy.
4. **Transfer Learning of the Hybrid Network:** To improve the transfer learning performance of the hybrid network, the classification layers are rearranged. The fused features are processed through the following sequence:
 - **Batch Normalization Layer:** Ensures the fused feature are normalised.
 - **Dropout Layer** (50% *Dropout Rate*): To reduce overfitting by randomly dropping a proportion of the neurons during training.

- **Fully Connected Layer:** For mapping the high-dimensional fused representation to the class probabilities.
 - **Softmax Layer:** Outputs class probabilities for the final prediction.
5. **Feature Fusion Approach:** To leverage the strengths of both DNN and WSN features whilst minimise their limitations, two tailor-made feature fusion approaches are proposed:

- **Scaling and Addition**

$$f_a = (\omega_1 \cdot f_{DNN} + b_1) + (\omega_2 \cdot f_{WSN} + b_2), \quad (4.33)$$

- **Depth Concatenation**

$$f_c = \text{Concat}[(\omega_1 \cdot f_{DNN} + b_1), (\omega_2 \cdot f_{WSN} + b_2)], \quad (4.34)$$

where ω_1 , ω_2 and b_1 , b_2 are learnable weights and biases to scale DNN features f_{DNN} and f_{WSN} before the step of feature fusion. These methods preserve the distinct characteristics of both feature types while allowing for feature integration.

4.3.2 Experimental Setup

Experiments are conducted based on the following configurations:

Datasets

The experiments are conducted on two distinct datasets:

- 100-class ImageNet subset randomly selected from the original ImageNet dataset
- The complete 1000-class ImageNet dataset, using 70% of the data for training due to limited computational power

All experiments are conducted using MATLAB2024a with RTX3090Ti GPU. For reproducibility, a fixed random seed default is used across all experiments. The hybrid network architecture is constructed by using the *Deep Network Designer* within the deep learning toolbox of MATLAB.

Table 4.5: *Implementation Details*

Training Parameter	Setting
Optimiser	Adam
Initial Learning Rate	10^{-3} , decreased by 0.1 every 500 epochs
Squared Gradient Decay Factor	0.99
Batch Size	128
Training Duration	> 5000 epochs

To thoroughly evaluate the adversarial robustness of the proposed hybrid networks, a comprehensive set of adversarial attacks with varying parameters is implemented. Table 5.1 exhibits the attacking parameters of FGSM and PGD used in the experiments.

Table 4.6: *Adversarial Attack Parameters Configuration*

Attack Type	ϵ	α	Iterations	Attack Index
FGSM	16/255	-	-	A1
FGSM	32/255	-	-	A2
PGD	16/255	0.25	50	A3
PGD	16/255	0.25	100	A4
PGD	32/255	0.25	50	A5
PGD	32/255	0.25	100	A6

Note: ϵ and α values are in pixel scale (0-255)

For FGSM attacks(A1, A2), two perturbation magnitudes (ϵ) are employed to test the networks' robustness. The selected values of $\epsilon = \{16, 32\}$ ensure a sufficient success rate whilst preventing the adversarial perturbation from being perceptible to human eyes. The PGD attacks(A3 TO A6) are configured with two perturbation levels ($\epsilon = \{16, 32\}$) and a constant step size ($\alpha = 0.25$). Each configuration is tested with both 50 and 100 iterations.

These six attack settings (A1-A6) aim to provide a comprehensive evaluation framework. The FGSM attacks serve as a baseline for single-step perturbations, whilst the PGD configurations represent more sophisticated multi-step attacks, which enables the assessment of the robustness of the hybrid networks across a spectrum of adversarial threats.

10,000 adversarial examples are generated for each attack type (A1 to A6). CW attacks are excluded due to their computational complexity and relatively lower attack strength compared to the benchmark.

4.3.3 Experimental Results

Table 4.7: Performance Comparison on 100-Classes ImageNet Subset
- A1 to A3 Attacks

Architecture	Clean	Trade-off	A1(%)		A2(%)		A3(%)	
	Acc(%)	(%)	Acc	Imp	Acc	Imp	Acc	Imp
ResNet-18	94.3	-	28.5	-	22.8	-	21.8	-
+ Scaling Addition	87.5	-6.8	73.2	44.7	71.8	49.0	71.6	49.8
+ Depth-Concat	87.2	-7.1	75.1	46.6	73.9	51.1	73.9	52.1
ResNet-50	95.8	-	32.2	-	26.4	-	26.1	-
+ Scaling Addition	88.9	-6.9	75.5	43.3	73.8	47.4	74.2	48.1
+ Depth-Concat	88.0	-7.8	77.2	45.0	75.6	49.2	76.1	50.0
ResNet-101	96.2	-	30.8	-	25.1	-	24.8	-
+ Scaling Addition	91.5	-4.7	76.1	45.3	74.4	49.3	74.8	50.0
+ Depth-Concat	90.2	-6.0	77.8	47.0	76.2	51.1	76.7	51.9

Attack Settings:
FGSM (A1,A2): $\epsilon=16,32$
PGD (A3): $\epsilon=16$, $\alpha=0.25$, iter=50

Table 4.8: Performance Comparison on 100-Classes ImageNet Subset
- A4 to A6 Attacks

Architecture	Clean	Trade-off	A1(%)		A2(%)		A3(%)	
	Acc(%)	(%)	Acc	Imp	Acc	Imp	Acc	Imp
ResNet-18	94.3	-	19.2	-	17.1	-	15.5	-
+ Scaling Addition	87.5	-6.8	70.3	43.6	68.8	51.7	67.2	51.7
+ Depth-Concat	87.2	-7.1	72.5	45.8	71.1	54.0	69.4	53.9
ResNet-50	95.8	-	23.7	-	21.2	-	19.8	-
+ Scaling Addition	88.9	-6.9	72.9	38.2	71.4	50.2	70.1	50.3
+ Depth-Concat	88.0	-7.8	74.8	40.1	73.2	52.0	72.0	52.2
ResNet-101	96.2	-	22.2	-	19.9	-	18.4	-
+ Scaling Addition	91.5	-4.7	73.5	38.6	72.1	52.2	70.8	52.4
+ Depth-Concat	90.2	-6.0	75.4	40.5	73.9	54.0	72.5	54.1

Attack Settings:
PGD (A4): $\epsilon=16$, $\alpha=0.25$, iter=100
PGD (A5,A6): $\epsilon=32$, $\alpha=0.25$, iter=50,100

Results on 100-class ImageNet Subset

It is envisaged that the WSN ResNets could slightly downgrade their model performance compared to their baseline models due to the limitations of WSN

Table 4.9: Performance Comparison on Full ImageNet Dataset(70%) - A1 to A3 Attacks

Architecture	Clean	Trade-off	A1(%)		A2(%)		A3(%)	
	Acc(%)	(%)	Acc	Imp	Acc	Imp	Acc	Imp
ResNet-18	74.2	-	28.2	-	22.9	-	21.8	-
+ Scaling Addition	67.7	-6.5	53.8	25.6	52.2	29.3	52.8	31.0
+ Depth-Concat	66.4	-7.8	57.2	29.0	55.8	32.9	56.2	34.4
ResNet-50	78.9	-	31.5	-	25.9	-	24.9	-
+ Scaling Addition	71.5	-7.4	62.9	31.4	61.4	35.5	62.2	37.3
+ Depth-Concat	70.2	-8.7	66.6	35.1	65.2	39.3	65.9	41.0
ResNet-101	82.1	-	30.8	-	25.1	-	24.4	-
+ Scaling Addition	74.4	-7.7	65.2	34.4	63.8	38.7	64.6	40.2
+ Depth-Concat	73.1	-9.0	67.5	36.7	66.2	41.1	66.9	42.5

Attack Settings:
FGSM (A1,A2): $\epsilon=16,32$
PGD (A3): $\epsilon=16, \alpha=0.25, \text{iter}=50$

Table 4.10: Performance Comparison on Full ImageNet Dataset(70%) - A4 to A6 Attacks

Architecture	Clean	Trade-off	A1(%)		A2(%)		A3(%)	
	Acc(%)	(%)	Acc	Imp	Acc	Imp	Acc	Imp
ResNet-18	74.2	-	19.5	-	17.1	-	15.4	-
+ Scaling Addition	67.7	-6.5	51.4	30.9	50.2	33.1	48.8	33.4
+ Depth-Concat	66.4	-7.8	54.8	34.3	53.5	36.4	52.2	36.8
ResNet-50	78.9	-	22.7	-	20.2	-	18.5	-
+ Scaling Addition	71.5	-7.4	60.8	35.1	59.5	39.3	58.1	39.6
+ Depth-Concat	70.2	-8.7	64.5	38.8	63.2	43.0	61.8	43.3
ResNet-101	82.1	-	22.2	-	19.8	-	18.1	-
+ Scaling Addition	74.4	-7.7	63.2	34.0	61.9	42.1	60.5	42.4
+ Depth-Concat	73.1	-9.0	65.5	36.3	64.2	44.4	62.8	44.7

Attack Settings:
PGD (A4): $\epsilon=16, \alpha=0.25, \text{iter}=100$
PGD (A5,A6): $\epsilon=32, \alpha=0.25, \text{iter}=50,100$

features. It is observed in Table 4.7 and Fig. 4.5 that for the 100-class ImageNet subset, the original ResNet models demonstrate better clean accuracy ranging from 94.3% (ResNet-18) to 96.2% (ResNet-101), compared to 87.2%(WSN ResNet-18+DC) to 91.5%(WSN ResNet+SA) of the hybrid networks, resulting in an accuracy degradation of -7.8% (WSN ResNet-50+DC) to -4.7% (WSN ResNet-101+SA). Moreover, the proposed hybrid network architectures demonstrate significant improvements across all attacks(A1 to A6). As shown in Figure. 4.6 and detailed in Tables 4.7 and 4.8, baseline ResNet models exhibit severe

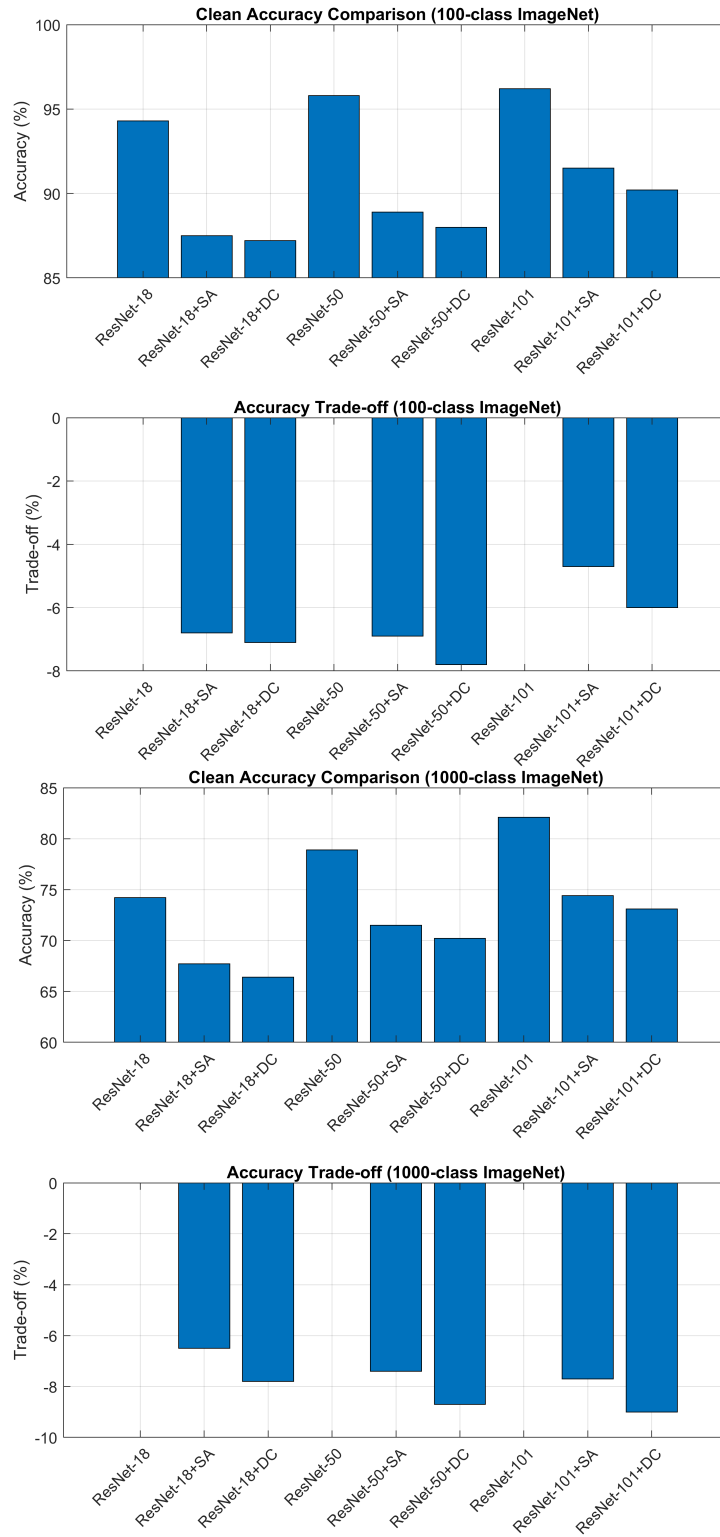


Figure 4.5: Accuracy of clean images and performance trade-off on 100-class ImageNet Subset and 70% ImageNet (1000 classes) of the proposed Hybrid WSN ResNets

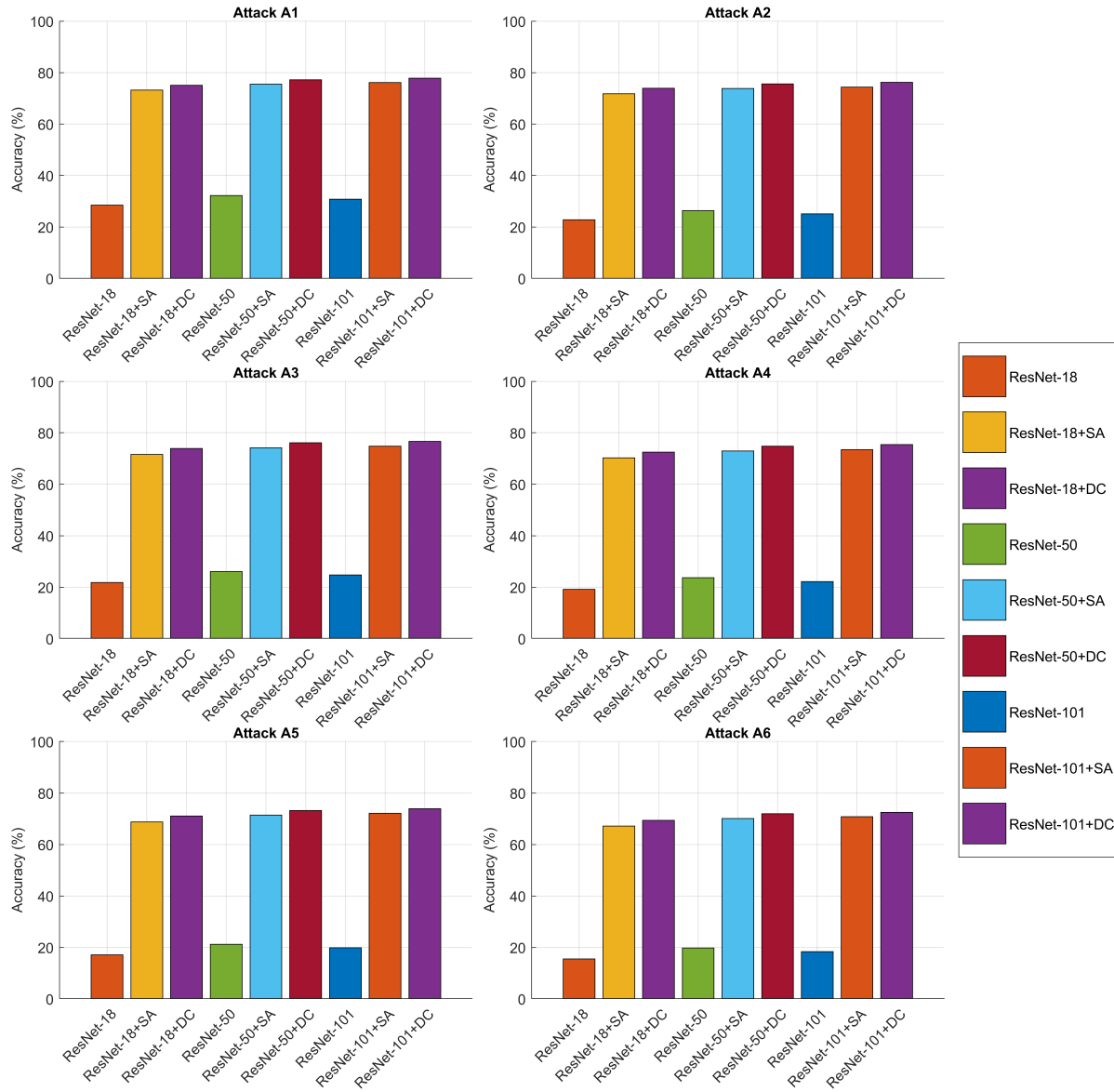


Figure 4.6: Robustness evaluation against six different adversarial attacks (A1-A6) on 100-class ImageNet subset

vulnerability to adversarial attacks, with accuracy dropping to between 15.5% and 32.2%, whereas the hybrid variants demonstrate great robustness, with the WSN ResNet with Depth-Concatenation feature fusion modification (WSN ResNet+DC) consistently outperforming those with Scaling Addition feature fusion modification (WSN ResNet+SA). For attacks A1-A3 (Table 4.7), WSN ResNet+DC improves adversarial robustness by 45.0% to 52.1%, achieving classification accuracy of AEs between 73.9% and 77.8% compared to the baseline ResNets. Under stronger attacks A4-A6 (Table 4.8), it achieves improvements of 40.5% to 54.1%, with classification accuracy of AEs ranging from 69.4% to

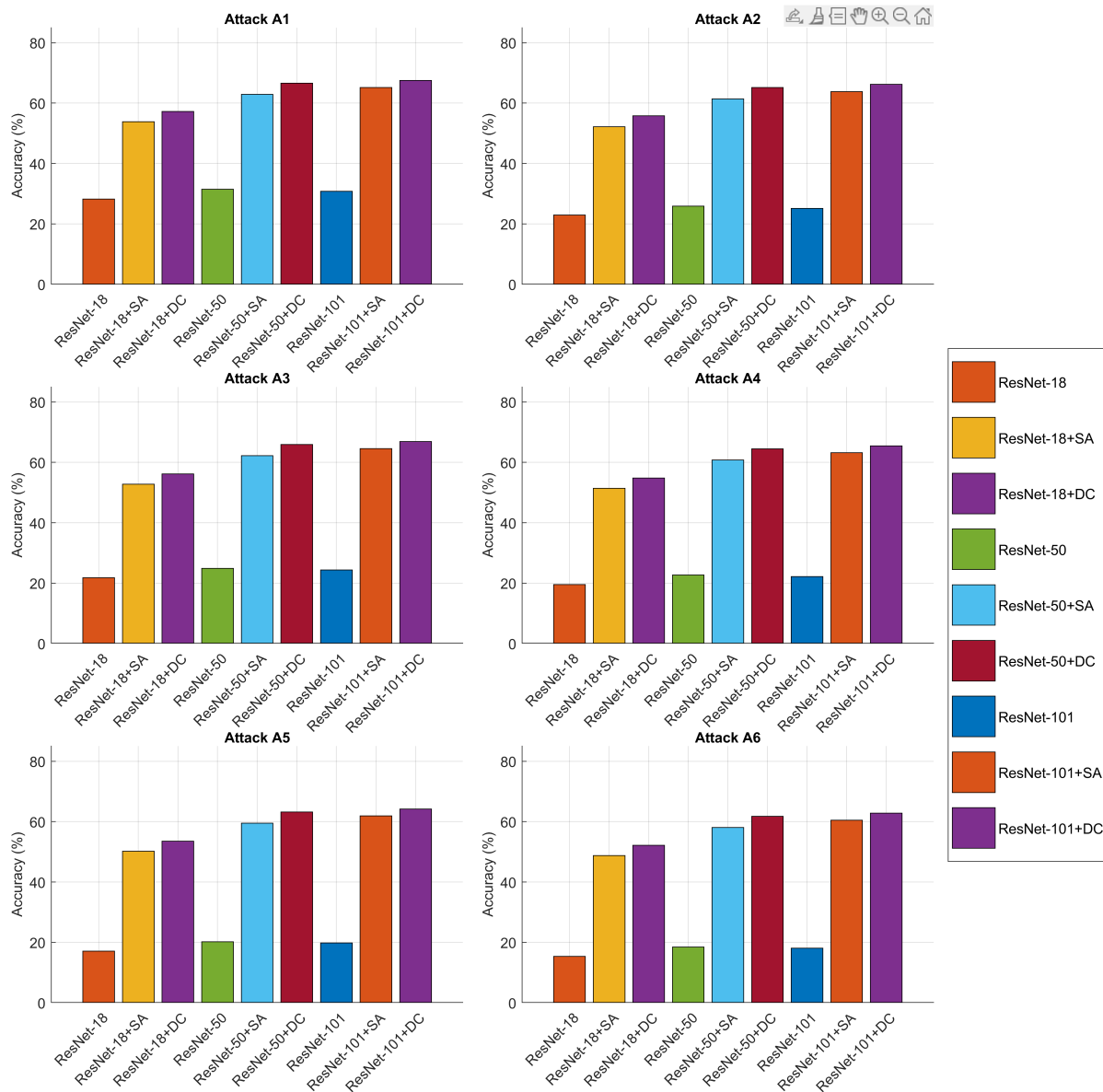


Figure 4.7: Robustness evaluation against six different adversarial attacks (A1-A6) on 70% ImageNet subset(1000-class)

75.4%.

Results on 70% ImageNet (1000-class)

For the large-scale evaluation on 70% of the 1000-class ImageNet dataset, Table 4.9 and Figure. 4.5 show that baseline models naturally achieve lower model performance in classifying clean images (74.2% to 82.1%) due to the increased complexity, whilst the hybrid modifications demonstrate an acceptable accuracy reduction of -9.0% to -6.5% in clean images. Furthermore,

as illustrated in Figure 4.7 and detailed in Tables 4.9 and ??, the hybrid modifications exhibit strong adversarial robustness on the large-scale dataset, while baseline models remain vulnerable against varied adversarial attacks with accuracies of 15.4% to 31.5%. For attacks A1-A3, the WSN ResNet+DC modifications achieve robustness improvements of 29.0% to 42.5%, achieving overall classification accuracy of AEs between 55.8% and 67.5%. For attacks A4-A6, they achieve robustness improvements of 34.3% to 44.7%, with classification accuracy of AEs ranging from 52.2% to 65.5%. The WSN ResNet+SA modifications, though slightly less effective, still achieve significant robustness improvements between 48.8% and 65.2%.

Key Observations

The experimental results demonstrate several noticeable characteristics of the proposed Hybrid WSN ResNets modifications:

- Generally, deeper hybrid network modifications exhibit better adversarial robustness and lower accuracy trade-offs in clean images.
- The WSN ResNet+DC modifications demonstrate slightly better adversarial robustness than The WSN ResNet+SA modifications, whereas there is more decline in clean image classification.
- The proposed hybrid modifications achieve significant improvement in adversarial robustness across both small and large-scale datasets, with acceptable compromise in classification accuracy in clean images.
- The proposed hybrid modifications demonstrate overall good adversarial robustness against all attack types(A1 to A6), demonstrating their potential effectiveness in addressing future, more challenging adversarial attacks.

In addition, several examples of the predictions and feature visualisations of baseline ResNet models and their hybrid-modified ones are shown, as illustrated in Figure.4.8. More examples are shown in Figures.B.7 and B.8 in Appendix.B. These figures clearly illustrate the excitation maps of clean images and AEs of original and hybrid ResNet models, respectively, which provide an intuitive view of the differences in feature extraction patterns.

Clean Image: tobacco shop
ResNet-101 (0.8)



AE (Misclassified As loggerhead)
With Confidence of 0.99



Clean Image: tobacco shop
Hybrid WSN ResNet-101(0.76)



AE: tobacco shop
Hybrid WSN ResNet-101(0.63)



Grad-CAM of Clean Image



Grad-CAM of AE



Grad-CAM of Clean Image by
Hybrid WSN ResNet-101



Grad-CAM of AE by
Hybrid WSN ResNet-101



Clean Image: umbrella
ResNet-101 (0.99)



AE (Misclassified As tape player)
With Confidence of 0.99



Clean Image: umbrella
Hybrid WSN ResNet-101(0.75)



AE: umbrella
Hybrid WSN ResNet-101(0.7)



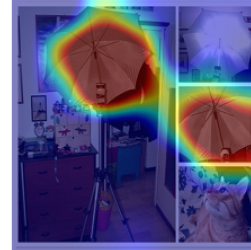
Grad-CAM of Clean Image



Grad-CAM of AE



Grad-CAM of Clean Image by
Hybrid WSN ResNet-101



Grad-CAM of AE by
Hybrid WSN ResNet-101



Figure 4.8: Feature Visualisation Comparison between ResNet-101 and Hybrid WSN-ResNet-101. Set(1): Clean image correctly classified by ResNet-101 and the corresponding Grad-CAM; Set(2): AE misclassified by ResNet-101 and the Grad-CAM of AE; Set(3): Same AE correctly classified by WSN-ResNet-101 and the Grad-CAM showing its attention focus by the hybrid network.

4.3.4 Discussion

The experimental results in Section 4.3.3 reveal several significant insights regarding the effectiveness of the proposed architectural modifications for improving adversarial robustness in DNN models. Thus, in this section, these findings are discussed by considering both the contributions and limitations of the methods as follows:

1. **Impact of DNN Model Depth on Its Overall Performance:** Across all attacking scenarios and datasets, it is distinctly shown that deeper baseline ResNet model modifications achieve better experimental results in both classification accuracy in clean images and AEs. Moreover, deeper architectures produce DNN features with higher dimensions (ResNet-50, 101 by 2048×1 and ResNet-18 by 512×1), whilst the WSN feature vectors in the experimental settings have fixed dimensions of 681×1 . This suggests that DNN features extracted from deeper architectures are critical in capturing more complex features but also providing inherent adversarial robustness.
2. **Scaling Addition VS Depth-Concatenation:** While both modifications significantly improve adversarial robustness, WSN ResNets with Depth-Concatenation feature fusion consistently outperform those with Scaling Addition architecture and attack-wise, whereas compromising acceptably more accuracy in classifying clean images. This suggests that maintaining multiple feature representations through depth concatenation could provide more robustness of the non-DNN feature type than scaling addition. However, it must be mentioned that a decent trade-off between feature types could only be experimentally acquired; thus, it could be a challenging task for researchers to address in any future work that includes the integration of DNN features and other feature types.
3. **Prediction and Feature Visualisation Analysis:** As shown in Fig. 4.8, B.7 and B.8, the original ResNet-101 shows distinct feature excitation area drift when predicting clean images and AEs. In contrast, the hybrid WSN ResNet-101 does not maintain the same attention to images as its baseline counterpart but exhibits a highly intersected excitation area to clean images and AEs. This indicates the enhanced robustness of feature

extraction against adversarial perturbation. Even though the excitation areas of the proposed Hybrid ResNet differ from the baseline ResNet, the consistency between clean and AEs indicates that the model has learned more adversarial perturbation-resistant features. This also aligns with the hypothesis in Section 4.3 that adversarial vulnerability could derive from non-robust features learnt by DNN models.

4. **Inherent Adversarial Robustness of WSN Features:** The proposed hybrid WSN ResNet architectures demonstrate significant improvements in adversarial robustness with acceptable trade-offs in clean image accuracy, validating the theoretical analysis of feature robustness presented in Section 4.2. These results demonstrate that while DNN models achieve remarkable performance in most deep-learning tasks through end-to-end optimised feature extraction, the integration of sophisticated, hand-crafted features, such as WSN features, can make critical contributions to addressing adversarial robustness challenges. By combining traditional signal processing techniques with DNN features, the hybrid approach enhances the robustness of learned representations, leveraging advantages that mere DNN models may not naturally develop through training alone.
5. **Practical Implications:** Based on the experimental results of 100-class and 1000-class datasets, it is observed that the performance metrics decline with increased task complexity, but the improvements in robustness remain significant. The effectiveness against various attacks suggests great applicability in real-world scenarios where there is scarce information on the attacks. Moreover, with stable improvements across different architectures, this study provides flexibility in choosing baseline models for other researchers according to their computational power considerations.

Limitations

While the proposed approach demonstrates significant improvements in adversarial robustness, several limitations must be mentioned:

1. **Computational Resource Constraints:** Due to computational power

limitations, this study focused exclusively on ResNet architectures for hybrid modifications. Moreover, the study was constrained to training on 70% of the ImageNet dataset rather than the complete dataset. Future work with access to more substantial computational resources could explore the effectiveness of the hybrid approach across a broader range of architectures and the full ImageNet dataset.

2. **Attack Configuration Scope:** While the evaluation demonstrates robust performance against various attacks, the study was limited to specific pre-defined attacking parameters to produce clear and precise experimental results. Further investigation could explore a wider range of attack types and parameters to provide a more comprehensive evaluation of robustness. The current results, while promising, represent performance under a specific set of attack configurations.
3. **Feature Fusion Methodology:** The current implementation explores only two feature fusion methods (Scaling Addition and Depth-Concatenation) to adapt the hybrid networks. There is potential to investigate more sophisticated fusion techniques that might better leverage the complementary strengths of WSN features and learned representations. Alternative fusion strategies could potentially offer better trade-offs between clean accuracy and adversarial robustness.
4. **Feature Extraction Alternatives:** While WSN features have proven effective in the hybrid approach, other sophisticated feature extraction methods could be explored. Future research could investigate alternative hand-crafted features or hybrid approaches that might simultaneously enhance both model performance and adversarial robustness. This could lead to more optimal solutions that better balance the trade-off between accuracy and defence capabilities.

4.4 Real-World Application in Medical Imaging: ROI-Aware ResNet Architecture

Different real-world scenarios are considered in this PhD research. In this section, adversarial robustness is investigated in the context of deep learning

for medical imaging, particularly for using volumetric CT scans for AI-assisted COVID-19 prognosis, where both prognosis accuracy and model robustness are vital. Noticeably, due to the unique characteristics of 3D volumetric CT scans, previous defence methods against adversarial attacks are incompatible with the DNN models utilised in this case. To address these challenges, a novel 3D ROI-Aware ResNet architecture designed for robust prognosis of COVID-19 using chest CT scans is proposed. The innovative aspect of this architecture is the introduction of the region-of-interest (ROI)-Aware techniques, which employ adaptive masking not only to the input volumetric CT scans but also to the intermediate convolutional feature maps within the network. This technique ensures that the architecture remains focused on critical lung regions, alleviating the impact of adversarial perturbations. Notably, this study’s fundamental research originates from work in the second year of the PhD; the original papers have been published in [71, 72] as mentioned in Section.1.5. The main contributions of this work are twofold:

1. A new volumetric pre-processing method specifically designed for 3D volumetric chest CT scans is proposed, which effectively converts the raw CT volumes into a standardised format that is ideal for the input of 3D DNN models. The pre-processing method comprises many procedures that discard diagnostically irrelevant voxels and non-salient slices, thereby simultaneously improving computational efficiency and maintaining robust features to be learnt by the 3D models.
2. A novel 3D ROI-Aware ResNet architecture designed for the prognosis of COVID-19 using volumetric chest CT scans is proposed. The proposed architecture introduces a novel region-of-interest (ROI)-Aware technique to alleviate the effects of adversarial perturbations layer-by-layer adaptively. Unlike the Hybrid WSN ResNet models that achieve a trade-off between robustness and model performance, as previously discussed in Section.4.3, these proposed architectures achieve the best robustness whilst maintaining great classification performance.

4.4.1 Motivation and Challenges

Deep learning in medical imaging faces unique challenges regarding adversarial robustness [138, 139, 140, 141, 142, 143, 144]. Volumetric medical data, such as 3D CT scans, contains rich spatial information that must be reliable and robust for critical diagnostic decisions. However, in the context of adversarial robustness in medical imaging, there are several significant challenges compared to natural image data in several aspects:

1. **Lack of High-Calibre Medical Datasets:** High-quality dataset collection in medical imaging requires expertise (*e.g.*, radiologists, pathologists), which could be costly and time-consuming, causing an insufficiency of high-quality datasets. In addition, medical imaging datasets are often imbalanced and lack diversity compared to natural image datasets (*e.g.*, ImageNet). This makes it difficult to train DNN models that are robust to AEs across varied medical scenarios.
2. **Data Confidentiality:** Medical images contain sensitive patient information; thus, privacy and security are vital considerations when developing AI-powered auxiliary models for medical scenarios. Therefore, it is imperative to ensure the deep learning model is sufficiently robust against any potential adversarial attacks.
3. **Volumetric Data Characteristics:** Unlike natural images, volumetric CT/MRI scans are typically generated through medical sensors and required to maintain diagnostic features without any artificial alterations, resulting in unique data characteristics that previous adversarial defence methods in Section 3.3 may not be fully applicable (Adding GMBN perturbation).

4.4.2 Methodology

To address these challenges and thus improve the adversarial robustness of deep learning in real-world medical imaging scenarios, a novel 3D ROI-Aware ResNet architecture with a tailor-made pre-processing approach specifically designed for volumetric data is proposed. The technical details are elaborated as follows. The proposed framework is structured into two sequential stages:

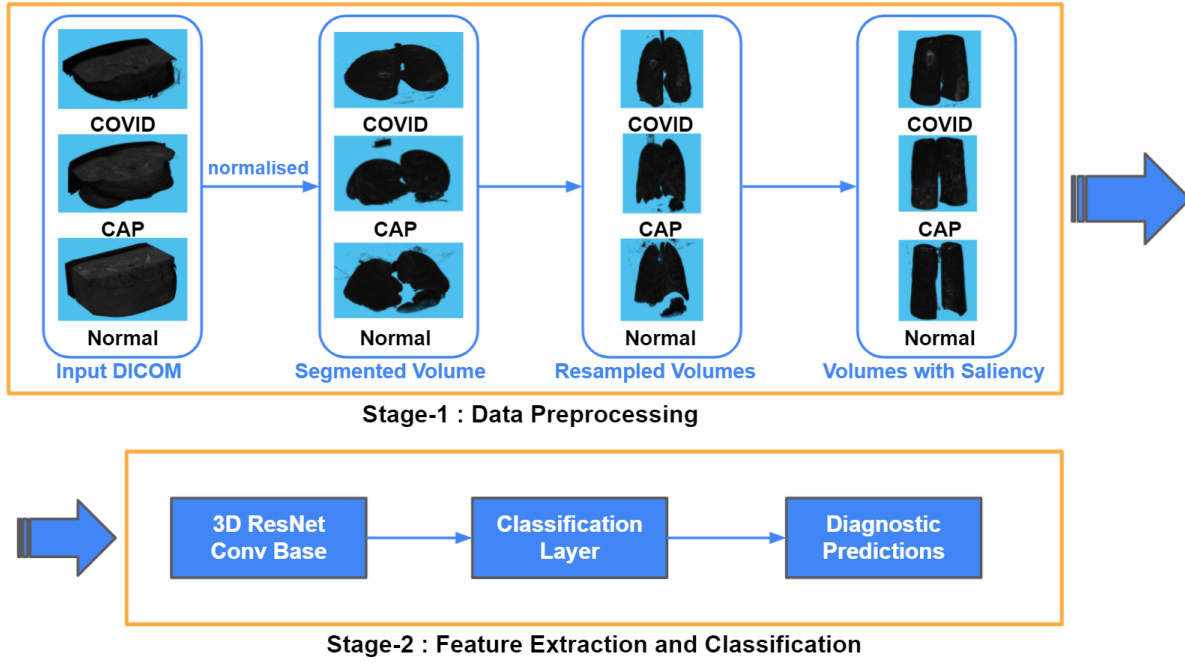


Figure 4.9: An overview of the Dual-Stage Framework of the proposed method. Stage 1: Data Pre-processing; Stage 2: Feature Extraction and Prediction.

(1) volumetric data preprocessing and (2) feature extraction and classification via a transfer learning approach based on a RoI-aware 3D ResNet architecture. Figure.4.9 illustrates the conceptual pipeline. In the first stage, a novel preprocessing technique tailored for 3D medical imaging is introduced, elaborated in Section.4.4.3. In the second stage, the refined volumetric data undergo feature extraction and classification using a modified 3D ResNet model designed to focus on the region of interest (RoI). Details of this stage are provided in Section.4.4.4.

4.4.3 Preprocessing of Volumetric Data

The primary aim of the preprocessing stage is to segment regions of interest (RoI) within 3D volumetric data, excluding irrelevant or non-contributory features that may impair model performance. The preprocessing workflow involves three main steps: (1) pulmonary segmentation, (2) volumetric resampling, and (3) extraction of salient slices. These steps are elaborated below.

To provide a better understanding, the complete preprocessing pipeline for volumetric CT scans is demonstrated in Figure.4.10 by showing each stage using thorax CT volumes from the input scans. This visualisation clearly exhibits how the preprocessing method systematically transforms raw CT volumes into standardised representations for the robust input of the 3D DNN models.

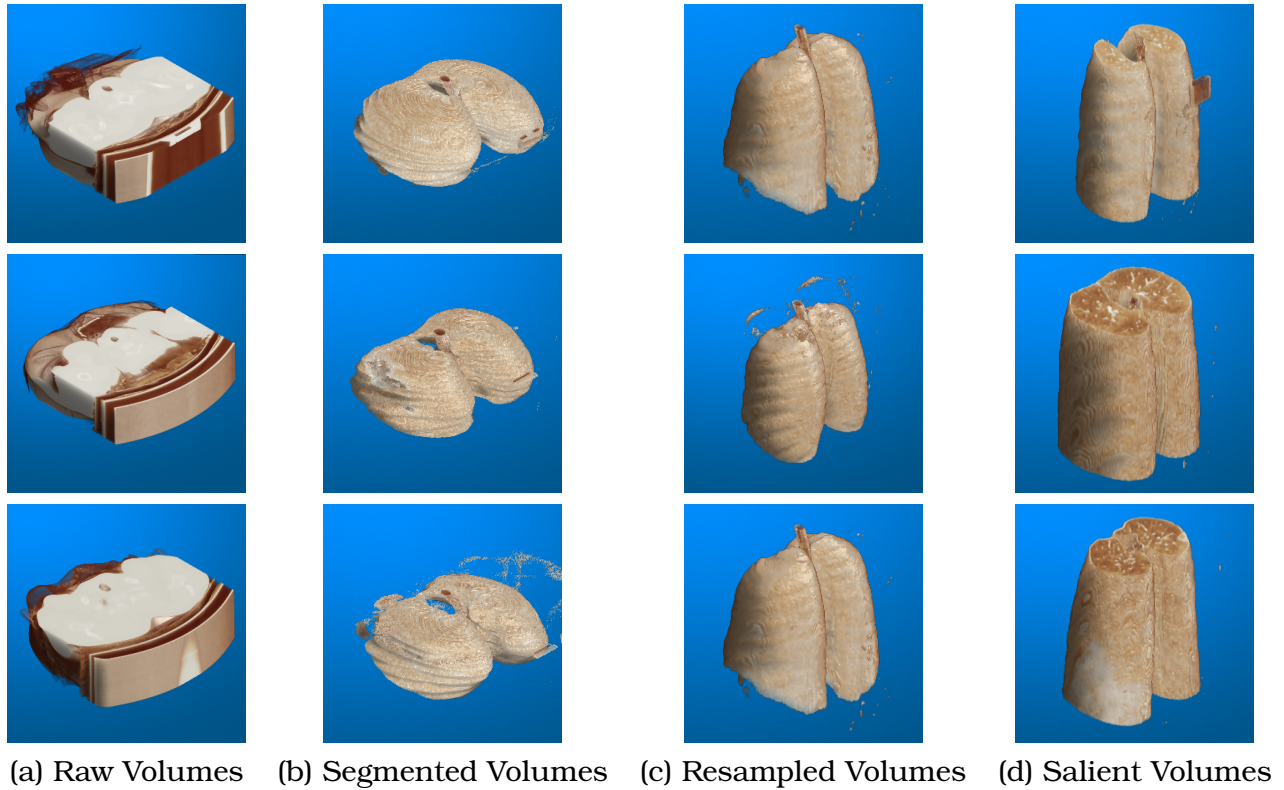


Figure 4.10: *The Complete Pipeline of Volumetric Chest CTs Preprocessing, corresponding to Section.4.4.3 (Note that column (d) is zoomed in, thus leading to an upscaled view compared to the previous three)*

Pulmonary Segmentation

Pulmonary segmentation is critical for extracting regions containing lung tissue, ensuring that only relevant information is retained for diagnosis. This step involves several processes:

1. Conversion of Digital Imaging and Communications in Medicine (DICOM) files to 3D volumes.

2. Normalisation of pixel intensities using Hounsfield Units (HU) to a range of $[0, 1]$ through linear transformation, as shown below:

$$V_n = (V_i - m) \frac{M' - m'}{M - m} + m', \quad (4.35)$$

where m and M are the original minimum and maximum values of the input volume V_i , and m' and M' are the target normalisation limits.

3. Enhancement of contrast for segmentation accuracy by saturating extreme pixel values. The contrast enhancement process is implemented by constraining extreme intensity values within the volume. Specifically, the highest 1% of pixel intensities are saturated to normalise the intensity distribution, which provides a more robust representation of the tissue structures.
4. Binarisation, morphological operations (*e.g.*, inversion, border clearing, hole filling), and subtraction of background regions to isolate the lungs. The resulting segmentation process is illustrated in Figure.4.11.

Volumetric Resampling

After segmentation, the physical dimensions of the volume are standardised by resampling both the in-plane resolution and the distance between slices. This resampling process transforms the original 3D volume into a standardised coordinate system using specific pixel spacing and slice thickness scaling factors. Examples of resampled volumes are shown in Figure.4.10.

Salient Slice Extraction

Since not all slices in a CT volume contain diagnostically relevant information, an automated method to extract slices containing robust features is introduced, particularly focusing on regions with salient lung tissues. This approach effectively selects slices with saliency from the very top or bottom of the scan that could contribute to the robust diagnostic process. A threshold operation is applied to identify slices above a pre-defined saliency threshold t . The algorithm for salient slice extraction is detailed in Algorithm.5, where the

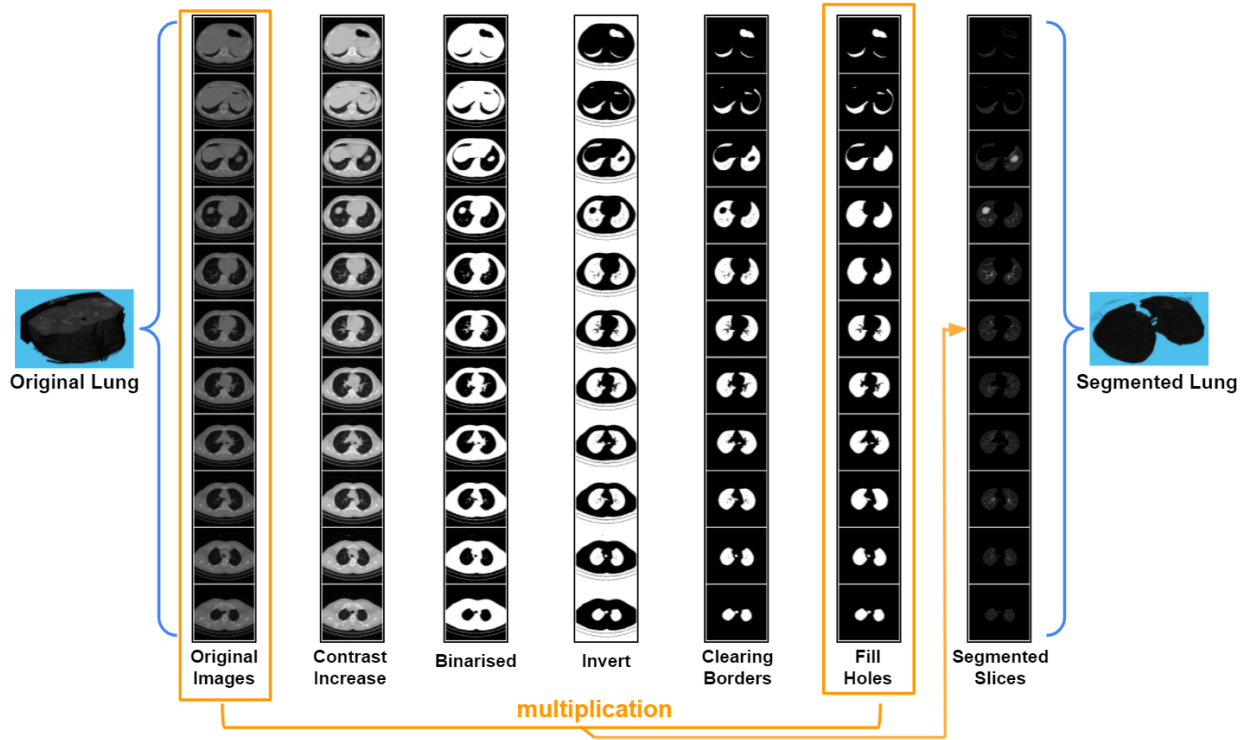


Figure 4.11: Illustration of the pulmonary segmentation process. Rows represent partial slices of the CT volume from top to bottom.

Algorithm 5 Slices with Saliency Extraction

- 1: **Input:** V_R , the input volume with dimensions of M, N, K ; $V_m(i)$, the binary segmented mask for the slice i ; t , the saliency thresholding value.
 - 2: **Output:** V_s , the volume with salient features; l'_a , the column vector of the slice-wise lung areas of the volume.
- Require:** $t \in [0, 1]$
- 3: **for** $i = 1 : K$ **do**
 - 4: $l_a(i) \leftarrow \frac{\sum(V_m(i))}{MN}$.
 - 5: **end for**
 - 6: **Normalising:** $l'_a \leftarrow \frac{l_a}{\text{Max}(l_a)}$.
 - 7: **Thresholding:** $l'_a \leftarrow [l'_a \geq t]$.
 - 8: $k_t \leftarrow$ index of the first slice s.t. $l'_a \geq t$.
 - 9: $K_n \leftarrow \text{length}(l'_a)$.
 - 10: $V_s \leftarrow V_R\{M, N, [k_t : (k_t + K_n)]\}$.
 - 11: **Return** V_s, l'_a .
-

final output volumes contain only the selected salient slices. The details of

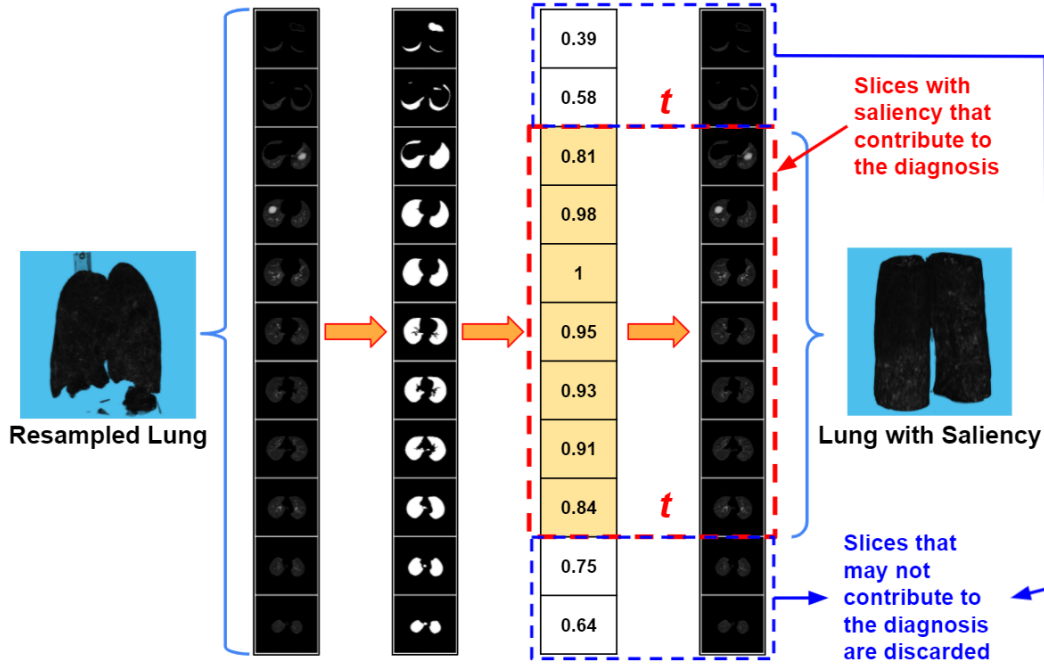


Figure 4.12: An overview of the extraction procedure of slices with saliency regions. The first and second columns denote partial slices of the given volume from the top to the bottom and their corresponding segmented masks, respectively. The third column denotes the normalised area vector $l'_a()$ in VolumeRange. The parameter t is referred to as the threshold value for the saliency region. Whilst the red dotted line box denotes the slices with saliency, and the blue dotted line boxes denote slices without saliency, and thus are to be discarded. Hence, the slices within the red dotted line box are utilised to generate the volume with saliency, which corresponds to the output V_s as shown in VolumeRange.

the algorithm are as follows.

Given a resampled volume, V_R , with dimensions $M \times N \times K$, a column of binary masks containing voxels of 0 and 1 is first generated. The area of the thorax is then calculated by the proportion of 1s in the mask. The column of lung area, denoted by l_a , is then normalised to $[0 \ 1]$. Next, the normalised lung area column l'_a is thresholded by a pre-defined value, t . Thus, by applying the threshold, it returns a column of indices of slices with salient features, denoted by $l'_a()$. The final step is to reconstruct the output volume V_s by using slices with saliency regions. Figure.4.12 shows the mechanism of extraction of the slices with robustness and saliency. Several examples of volumes before

and after salient slice extraction are illustrated in Figure.4.10, demonstrating how this process preserves diagnostically robust features for the input of 3D DNN models.

4.4.4 3D ROI-Aware ResNet Architecture

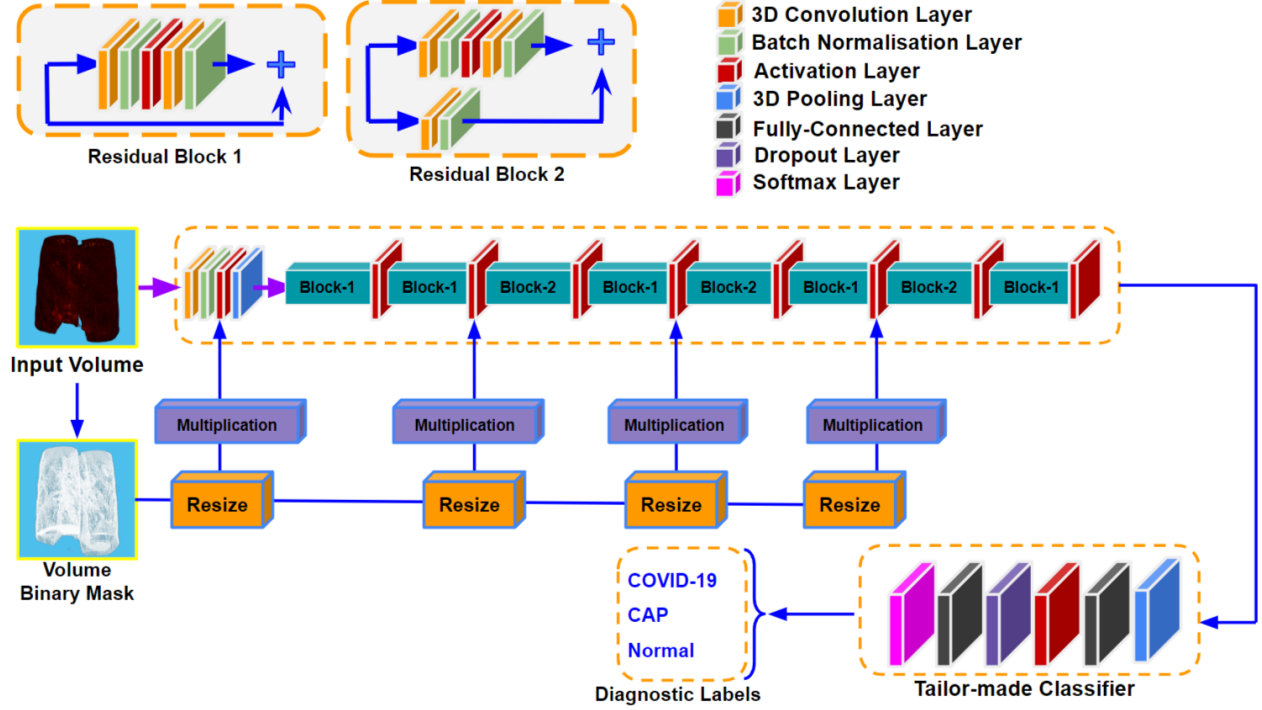


Figure 4.13: RoI-aware 3D ResNet-18 architecture adapted for pulmonary feature extraction.

In this stage, a novel RoI-aware 3D ResNet for feature extraction and classification is introduced. Transfer learning is applied to a pre-trained 3D ResNet model, which is fine-tuned to focus on pulmonary RoIs. Additional customisations include masking feature maps at intermediate layers to enforce RoI awareness. The final classifier outputs predictions for three classes: COVID-19, normal, and CAP. Augmentation strategies such as random cropping, flipping, and resampling are applied to increase dataset diversity whilst preserving robust volumetric integrity. The technical details are elaborated as follows:

- Convolution layers in 2D ResNet models [14] are designed to extract features from 2D images, limiting their capability to planar feature com-

putation and learning. However, medical image analysis often requires capturing volumetric representations from 3D data. 3D convolution layers are employed to extract volumetric features from chest CT scans to address this. The 3D ResNet architecture extends the 2D ResNet filters to the third dimension, enabling volumetric feature learning. This study evaluates three variants: 3D ResNet-18, 3D ResNet-50, and 3D ResNet-101 [145].

- A non-linear activation function follows each convolution layer in the network to generate feature maps. This non-linearity enables the network to learn complex representations. The ReLU activation function is employed, which applies element-wise activation by thresholding negative values to zero.
- Feature maps are down-sampled using pooling layers whilst preserving salient features. The 3D pooling operation integrates features within a 3D window through either average or maximum computations. The implementation uses max-pooling as the initial 3D pooling layer, followed by average-pooling at the classifier stage.

Despite input volumes being preprocessed to remain robust and salient features (See Section.4.4.3), conventional 3D convolution and pooling operations can extract features from regions outside the RoI, thereby causing non-robust features to be propagated to the subsequent layers. To address this limitation, a novel adaptation of the 3D ResNet architecture that constrains feature learning within the RoI is proposed. Figure.4.13 illustrates this RoI-aware architecture using 3D ResNet-18 as an example. The architecture of 3D ResNet-50 and ResNet-101 follows similar principles but with increased network depth.

The RoI Awareness Mechanism

The RoI awareness is implemented through binary volume masks. First, the input volume V_s is binarised using a threshold value p to generate mask V_m :

$$V_m(x, y, z) = \begin{cases} 1 & V_s(x, y, z) \geq p, \\ 0 & V_s(x, y, z) < p. \end{cases} \quad (4.36)$$

The activation outputs $V_a(L_j)$ at layer j are then masked through element-wise multiplication with the resized mask V'_m :

$$V'_a(L_j) = V'_m \odot V_a(L_j), \quad (4.37)$$

To preserve significant features, RoI masking is applied only to the first 4 – 6 activation layers (depending on the depth of ResNet models), as deeper layers undergo substantial downsampling.

Transfer Learning and Data Augmentation

The available pre-trained 3D ResNets in MATLAB are trained on MRI brain images, lacking learned features specific to chest CT scans. Therefore, all convolution-based parameters are set as trainable to enable transfer learning on chest CT data. The network terminates in a custom ternary classifier for categorising scans into COVID-19, normal, and CAP classes. To address limited training data, several augmentation strategies are implemented: 1) volume resampling with varying k_1 and k_2 , 2) random cropping, 3) random rotation, and 4) random vertical flipping. To preserve slice saliency, augmentation is applied in 2D before volume construction. Given the computational demands of volumetric data, augmentation is performed prior to training to optimise computational efficiency.

4.4.5 Experimental Setup

Table 4.11: *Implementation Details*

Training Parameter	Setting
Optimiser	Adam
Initial Learning Rate	10^{-3} , decreased by 0.1 every 500 epochs
Squared Gradient Decay Factor	0.99
Batch Size	16
Training Duration	> 5000 epochs

The implementation details of the transfer learning of the 3D ResNets are illustrated in Table 4.11. All experiments are conducted using MATLAB2024a with RTX3090Ti GPU. The hybrid network architecture is constructed by using the *Deep Network Designer* within the deep learning toolbox of MATLAB. 3D

ResNet 18, 50 and 101 with and without RoI-Aware architectural modification are trained, respectively, to provide a comprehensive evaluation of their model performance as well as adversarial robustness.

Additionally, to thoroughly evaluate the adversarial robustness of the proposed hybrid networks, a comprehensive set of adversarial attacks with varying parameters is implemented. Table 4.12 exhibits the attacking parameters of FGSM and PGD used in the experiments.

Table 4.12: *Adversarial Attack Parameters Configuration*

Attack Type	ϵ	α	Iterations	Attack Index
FGSM	16/255	-	-	A1
FGSM	32/255	-	-	A2
PGD	16/255	0.5	50	A3
PGD	16/255	0.5	100	A4
PGD	32/255	0.5	50	A5
PGD	32/255	0.5	100	A6

Note: ϵ and α values are in pixel scale (0-255)

For FGSM attacks(A1, A2), two perturbation magnitudes (ϵ) are employed to test the networks' robustness. The selected values of $\epsilon = \{16, 32\}$ ensure a sufficient success rate whilst preventing the adversarial perturbation from being perceptible to human eyes. The PGD attacks(A3 TO A6) are configured with two perturbation levels ($\epsilon = \{16, 32\}$) and a constant step size ($\alpha = 0.5$). Each configuration is tested with both 50 and 100 iterations. These six attack settings (A1-A6) aim to provide a comprehensive evaluation framework. 30 AEs are generated for each attack type (A1 to A6). CW attacks are excluded due to their computational complexity and relatively lower attack strength compared to the benchmark.

4.4.6 Experimental Results

To evaluate the robustness of the proposed RoI-aware approach, extensive experiments are conducted across different ResNet architectures (ResNet-18, ResNet-50, and ResNet-101) under both clean and adversarial conditions. Table 4.13 presents a comprehensive comparison of model performance with and without RoI awareness. Moreover, the overall effect of the ROI-aware mechanism on model robustness is demonstrated in Figure.4.14. While main-

Model	RoI-aware	Clean Acc.	Classification Accuracy of AEs					
			A1	A2	A3	A4	A5	A6
ResNet-18	No	80.0%	61.2%	60.9%	64.0%	60.3%	60.9%	64.0%
ResNet-18	Yes	79.2%	64.8%	70.7%	71.3%	68.6%	64.1%	59.4%
ResNet-50	No	86.2%	66.9%	62.6%	69.0%	66.8%	63.8%	69.0%
ResNet-50	Yes	86.9%	69.7%	70.6%	81.3%	74.8%	68.7%	67.8%
ResNet-101	No	89.2%	61.2%	62.8%	66.9%	63.4%	67.4%	66.9%
ResNet-101	Yes	90.0%	69.7%	70.9%	82.1%	69.5%	70.3%	68.4%

Table 4.13: Comparison of model accuracy under clean and adversarial conditions. A1-A6 represent different adversarial attacks. **Bold** values indicate the highest value in each column.

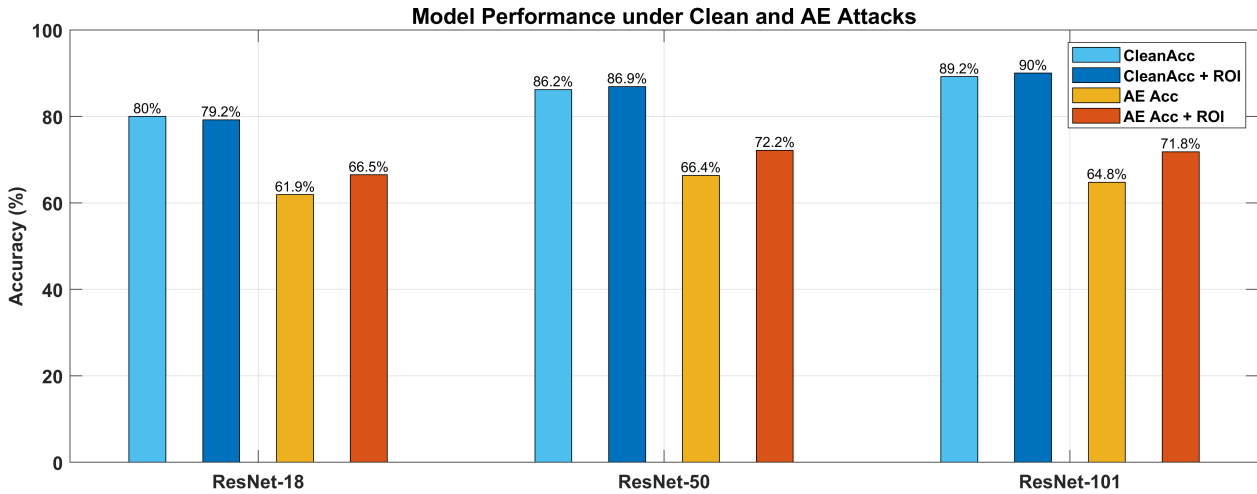


Figure 4.14: An Overview of the Baseline Model Performance and Adversarial Robustness of Regular and RoI-Aware 3D ResNet Models

taining comparable clean accuracy (90.0% vs 89.2% for ResNet-101), the ROI-aware mechanism substantially improves adversarial robustness across all architectures. Particularly, ROI-aware ResNet-101 demonstrates consistent improvement against various adversarial attacks, achieving up to 82.1% accuracy under adversarial conditions compared to 66.9% without ROI-awareness. This is similarly observed in ResNet-50 and ResNet-18, suggesting that input-wise and model-wise robust features provide inherent robustness against adversarial perturbations without compromising clean performance.

To further demonstrate the effectiveness of the ROI-aware mechanism, feature activations are visualised using Grad-CAM in Figure.4.15. When classifying AEs, the ROI-aware ResNet maintains consistent activation patterns

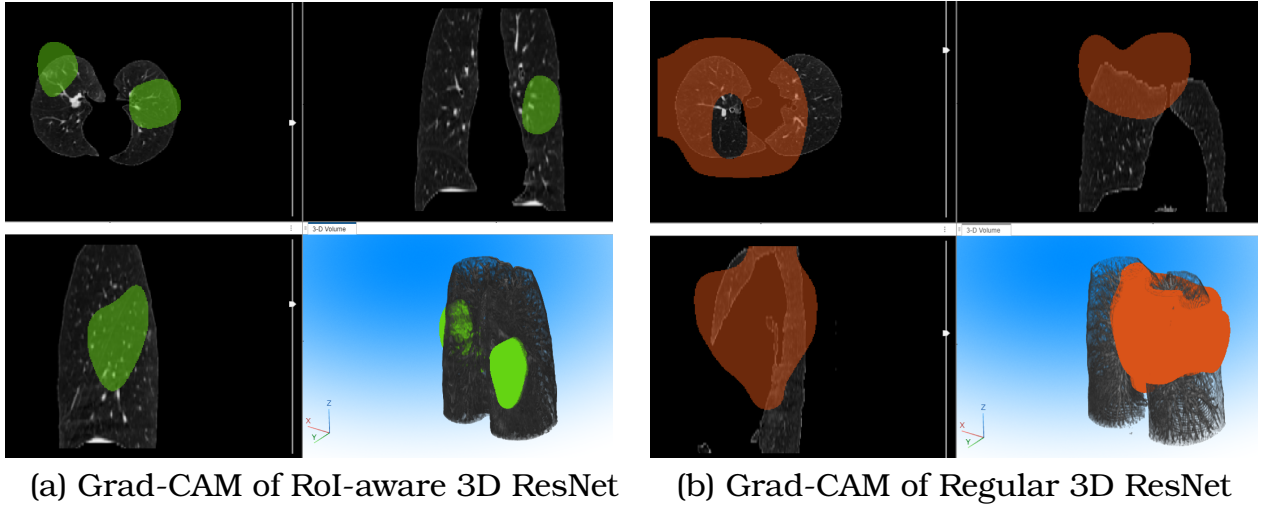


Figure 4.15: Grad-CAM visualisation of feature activations when classifying adversarial examples. (a) shows that the RoI-aware ResNet maintains consistent activation patterns strictly within the thorax region, demonstrating focused feature learning. In contrast, (b) shows that conventional ResNet without RoI awareness exhibits scattered activation patterns beyond anatomical boundaries. This visualisation directly validates that the ROI-aware mechanism effectively constrains the network’s attention to diagnostically relevant regions.

within the thorax region, whilst the conventional ResNet shows scattered activations outside the lung areas. This directly visualises how the ROI-aware mechanism maintains robust feature learning by focusing on anatomically relevant regions even under adversarial attacks.

Clean Performance

Under clean conditions, a consistent improvement in accuracy is observed as the network depth increases. The RoI-aware ResNet-101 achieves the highest clean accuracy of 90.0%, demonstrating deeper architectures’ effectiveness in capturing complex features. Notably, the RoI-aware variants of ResNet-50 and ResNet-101 show marginal improvements over their non-RoI-aware counterparts (86.9% vs 86.2% for ResNet-50, and 90.0% vs 89.2% for ResNet-101). Those performances align with the original paper of this work [71, 72].

Adversarial Robustness

The robustness of each model is evaluated against six different adversarial attacks (A1-A6). Several key observations emerge from the analysis:

- **RoI-aware Enhancement:** Across most architectures and attacks, RoI-aware models demonstrate superior robustness compared to their standard counterparts. This improvement is particularly evident in the ResNet-50 and ResNet-101 architectures.
- **Attack Resilience:** The RoI-aware ResNet-101 shows remarkable resilience, achieving the highest accuracy under most attack scenarios: - Against A1 and A2 attacks, it maintains accuracies of 69.7% and 70.9% respectively - Under A3 attack, it achieves the best performance with 82.1% accuracy - For A5, it demonstrates superior robustness with 70.3% accuracy
- **Architecture-specific Performance:** Deeper networks generally exhibit better adversarial robustness when equipped with RoI-awareness, except that - ResNet-50 with RoI-awareness shows notable improvement against A4 attack, achieving 74.8% accuracy - ResNet-50 without RoI-awareness performs best against A6 attack at 69.0% accuracy
- **Consistency of Improvement:** The RoI-aware variants consistently outperform their standard counterparts across most attack types, with improvements ranging from 3-15 percentage points, suggesting that the RoI-aware mechanism provides a robust defence against various adversarial perturbations.

These results demonstrate that incorporating RoI-awareness not only improves clean accuracy but also enhances model robustness against adversarial attacks. The consistent performance improvements across different architectures and attack types validate the effectiveness of the proposed approach in developing more robust deep learning models.

4.4.7 Discussion

The experimental results provide several important insights into the effectiveness of RoI-aware architectures for improving both model performance and

adversarial robustness. Here, the key findings and their broader implications for deep learning security and medical image analysis are discussed.

1. Impact of RoI-Awareness on Model Performance

The integration of RoI-awareness into ResNet architectures demonstrates a clear pattern of performance enhancement, particularly in deeper networks. The progressive improvement in clean accuracy from ResNet-18 (79.2%) to ResNet-101 (90.0%) with RoI-awareness suggests that deeper architectures can better leverage the localised attention mechanism to capture relevant features. However, it is noteworthy that the performance gain is not uniform across all architectures. Whilst ResNet-18 shows a slight decrease in clean accuracy with RoI-awareness (80.0% to 79.2%), both ResNet-50 and ResNet-101 exhibit improvements, indicating that RoI-awareness may require sufficient network capacity to be effectively utilised.

2. The ROI-aware Mechanism

The ROI-aware mechanism consistently improves adversarial robustness across all network architectures, with particularly notable gains in deeper networks. The improvement in adversarial scenarios (average 7.05% for ResNet-101) significantly outweighs the clean accuracy gains (0.8%), indicating that constraining feature learning to anatomically relevant regions effectively mitigates adversarial vulnerabilities. This suggests that a great proportion of adversarial perturbations are inherently filtered out by the ROI-aware mechanism.

3. Architectural Depth and Robustness

The correlation between network depth and adversarial robustness becomes evident when examining the performance across different attacks. ResNet-101 with RoI-awareness consistently achieves superior performance, particularly against sophisticated attacks like A3 (82.1%). This suggests that deeper networks with RoI-aware mechanisms can better maintain their decision boundaries under adversarial perturbations.

4. Attack-Specific Robustness

Different architectures show varying levels of robustness to specific attacks. For instance, RoI-aware ResNet-50's superior performance against

A4 (74.8%) indicates that certain architectural configurations may be particularly effective against specific types of adversarial perturbations. This observation highlights the importance of considering attack-specific defences in model design.

5. Consistency of Model Performance

The consistent improvement in adversarial robustness across multiple attack types suggests that the ROI-aware mechanism provides a generalisable defence mechanism. This is particularly valuable in real-world applications where the nature of potential attacks may be unknown.

Practical Implications Regarding Adversarial Robustness in Real-World Applications

The findings have several important implications for the deployment of deep learning models in medical imaging applications, particularly concerning robustness and reliability:

1. **Robust Feature Learning:** The two-stage approach achieves state-of-the-art performance through complementary robustness enhancement. The tailor-made preprocessing pipeline provides robust input volumes by preserving diagnostically relevant regions, whilst the ROI-aware mechanism ensures layer-wise robust feature learning through anatomically-guided masking. This is particularly crucial for medical imaging applications where feature extraction must be constrained to diagnostically contributing regions. This again validates the hypothesis that the cause of AE could be due to the lack of robust features learnt by the DNNs as previously mentioned in Section.4.2.
2. **Task-Specific Considerations:** The proposed ROI-aware mechanism is specifically designed for medical imaging tasks where clear anatomical priors exist. Unlike natural image classification tasks (*e.g.*, ImageNet[19] task as in the previous Section.4.3) where region-of-interest definitions are mostly unavailable, medical imaging provides well-defined anatomical references for feature map masking. This task specificity, whilst limiting broader applicability, ensures robust and interpretable learning in the medical scenarios.

3. **Model Selection Trade-offs:** Whilst deeper networks with ROI-aware mechanisms demonstrate superior robustness, the computational costs must be considered. The marginal improvements between ResNet-50 (72.15% mean adversarial accuracy) and ResNet-101 (71.82%) suggest that ROI-aware ResNet-50 might offer an optimal balance between robustness and efficiency for clinical deployment.

Limitations and Future Work

Despite the promising results, several limitations of this work are worth mentioning:

1. **Limited Adversarial Evaluation:** Due to the computational overhead involved in generating 3D adversarial examples and dataset constraints, the evaluation was limited to 180 adversarial examples in total. This relatively small sample size, coupled with computational constraints, also restricted the strength of adversarial attacks that could be implemented. Future work could explore more comprehensive robustness evaluation with larger sets of adversarial examples and stronger attack parameters.
2. **Incomplete Ablation Studies:** Conducting exhaustive ablation studies could be time-consuming regarding training 3D networks. Particularly, models without both preprocessing and ROI-aware mechanisms were unable to be evaluated, which would have provided more complete validation of the robustness of the preprocessed features. A comprehensive ablation study should be conducted for future investigation.
3. **Architectural Refinements:** The current ROI-aware ResNet architecture could be further optimised through layer-wise improvements. Future research could explore more efficient masking strategies at different network depths, potentially incorporating adaptive attention mechanisms that improve the learning process.

Chapter 5

Combat AEs with AE Detection and Robust Training

5.1 General Introduction

As discussed in the previous Section 4.3 of Chapter 4, Wavelet Scattering Network (WSN) features have experimentally demonstrated great robustness against adversarial perturbation (see Section 4.2). In this chapter, the authors propose a novel quantisation approach that utilises WSN features and DNN features to generate an AE likelihood, namely Adversarial Awareness Score (AAS), aiming to detect adversarial examples by exploiting the robustness of WSN features. In addition, the authors propose another novel Jacobian regularisation method that leverages the AAS mechanism to add a penalty factor to the loss function, to improve the model's ability to correctly predict adversarial examples. To summarise, the contributions are threefold:

1. **Quantisation for Adversarial Awareness:** This work introduces a novel quantisation method called Adversarial Awareness Score (AAS), which exploits the robustness of Wavelet Scattering Network (WSN) features compared to Deep Neural Network (DNN) features to define an adversarial awareness value. This value is determined by calculating the ratio of the Frobenius norms of DNN and WSN features. A high adversarial awareness value signals a greater likelihood that the input image is adversarial.
2. **AE Detection Using Adversarial Awareness Score:** The AAS mechanism

is extended to a practical adversarial detection application by converting raw AAS values into a percentage-based likelihood score. This interpretable metric quantifies the probability of an input being adversarial, making it well-suited for real-world detection tasks.

3. **Model Regularisation Using Adversarial Awareness Score:** An innovative model enhancement technique is introduced that integrates AAS into Jacobian regularisation during adversarial training. This adaptive penalty approach amplifies the loss for inputs with high AAS values, thereby improving the model’s capacity to distinguish between clean and adversarial examples. The method guides the learning process towards more robust feature representations.

These contributions collectively advance the state-of-the-art in adversarial detection and models’ robustness enhancement, providing theoretical insights and practical tools for enhancing the robustness of DNN models.

5.1.1 Other Detection Methods for Comparison

1. **FreqVAE [146]** This method introduces a data reconstruction-based framework for adversarial example detection in black-box settings. A reconstruction model R , such as an autoencoder, is trained to reconstruct input data x by minimising the reconstruction loss:

$$\mathcal{L}_{\text{recon}} = \|x - \hat{x}\|_2^2, \quad \hat{x} = R(x), \quad (5.1)$$

where $\hat{x}$ represents the reconstructed input. The detection mechanism evaluates the consistency of predictions between the original input and its reconstruction, computed as:

$$\text{score}_{\text{consistency}} = \|f(x) - f(\hat{x})\|_1, \quad (5.2)$$

where $f(\cdot)$ is the black-box model’s output. An input is classified as adversarial if $\text{score}_{\text{consistency}} > T$, where T is a predefined threshold. This approach effectively detects adversarial examples without requiring access to the model’s internal structure, making it suitable for black-box

applications. While the method demonstrates robustness against attacks such as FGSM and PGD, its effectiveness depends on the reconstruction quality and threshold selection.

2. **SQ+SF** [147] the detection mechanism combines scalar quantisation and spatial smoothing to identify AEs effectively. In principle, scalar quantisation reduces the perturbations in the input by mapping pixel intensities within a specific interval to a single representative value. The quantised image x_{SQ} is defined as:

$$x_{SQ} = \lfloor \frac{x}{\Delta} \rfloor \cdot \Delta, \quad (5.3)$$

where Δ is the interval size determined adaptively based on the entropy of the image. The entropy H of an image is computed as:

$$H = - \sum_{i=0}^{255} p_i \log_2(p_i), \quad (5.4)$$

where p_i is the normalised frequency of pixel intensity of the image x . Spatial smoothing reduces sharp intensity transitions introduced by adversarial perturbations. The smoothed image $f_{SQ-SF}(m, n)$ is computed as:

$$x_{SS} = (w * x) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) x(m-s, n-t), \quad (5.5)$$

where $w(s, t)$ is the filter mask, and a and b are the spatial dimensions of the mask. Their detection filter combines scalar quantisation and spatial smoothing based on the entropy of the input. For a pixel (m, n) , the output pixel $x_{SQ+SF}(m, n)$ is determined as:

$$x_{SQ+SF}(m, n) = \begin{cases} x_{SQ}, & \text{if } |x_{SQ} - x| \leq |x_{SS} - x|, \\ x_{SS}, & \text{otherwise.} \end{cases} \quad (5.6)$$

To determine whether x is adversarial, the detection mechanism compares the model's predictions for the original image x and the filtered image $T(x)$:

- The predicted class y for x can be denoted as $y = \operatorname{argmax}(D(x))$,

and thus $y_{SQ+SF} = \text{argmax}(D(x_{SQ+SF}))$ denotes the predicted class for x_{SQ+SF} .

- If $y = y_{SQ+SF}$, the image is classified as non-AE.
- If $y \neq y_{SQ+SF}$, the image is classified as an AE.

The method assumes that adversarial perturbations are more sensitive to preprocessing operations than benign inputs. As a result, the class prediction of an AE is likely to change after preprocessing, whereas the prediction for non-AE input remains consistent.

3. **Feature Squeezing [148]:** This work relies on Feature Squeezing to identify AEs. This approach leverages the sensitivity of AEs to input modifications by reducing the feature space of the input and comparing the model's predictions before and after squeezing. The steps are outlined as follows:

First, two main techniques are used to reduce the degrees of freedom available to an adversary.

- **Colour Bit Depth Reduction:** The bit depth of each pixel is reduced to limit the resolution and mitigate adversarial perturbations. For an image x , the reduced version x_{sq} is computed as:

$$x_{sq} = \frac{\text{round}(x \cdot (2^b - 1))}{2^b - 1}, \quad (5.7)$$

where b is the reduced bit depth.

- **Spatial Smoothing:** This technique averages pixel values within a local window to reduce high-frequency noise introduced by adversarial perturbations. The smoothed image x_{sf} is defined as:

$$x_{sf}(i, j) = \text{median}\{x(p, q) : (p, q) \in \text{window centred at } (i, j)\}. \quad (5.8)$$

The detection mechanism compares the DNN model's prediction of the original input and the squeezed inputs. Let:

- $g(x)$: The model's output prediction vector for the original input x .

- $g(x_{\text{sq}})$: The model's output prediction vector for the squeezed input x_{sq} .

The difference between these predictions is measured using the ℓ_1 -norm:

$$\text{score}(x, x_{\text{sq}}) = \|g(x) - g(x_{\text{sq}})\|_1. \quad (5.9)$$

The input x is classified as AE if:

$$\text{score}(x, x_{\text{sq}}) > T, \quad (5.10)$$

where T is a threshold determined empirically based on the separation of legitimate and adversarial examples. For better detection accuracy, multiple feature squeezing techniques can be combined. The joint detection score is given by:

$$\text{score}_{\text{joint}} = \max\{\text{score}(x, x_{\text{sq1}}), \text{score}(x, x_{\text{sq2}}), \dots\}. \quad (5.11)$$

The maximum score across all squeezing methods is compared with the threshold τ for final classification. The approach exploits the observation that adversarial examples are more sensitive to feature squeezing than legitimate inputs. As a result, the predictions for adversarial examples and their squeezed versions often diverge significantly, enabling effective detection.

5.2 Adversarial Awareness Score for AE Detection

Inspired by the results previously exhibited in Sections 4.2 and 4.3, the observed robustness disparity between Wavelet Scattering Network (WSN) features and Deep Neural Network (DNN) features under the impact of adversarial perturbations is utilised to develop an approach that distinguishes AEs from clean images. While DNN features are vulnerable to adversarial perturbations, WSN features demonstrate remarkable robustness due to their mathematical properties. This inherent difference in robustness characteristics provides a theoretical basis for developing a quantitative approach to determine adversarial awareness.

To comprehensively evaluate different quantisation approaches for feature representation comparison, three types of matrix/vector norms corresponding to three types of Adversarial Awareness Scores are used:

$$AAS_{\ell_1}(x) = \frac{\|f_{\text{DNN}}(x)\|_1 / \sqrt{\dim(f_{\text{DNN}})}}{\|f_{\text{WSN}}(x)\|_1 / \sqrt{\dim(f_{\text{WSN}})}}, \quad (5.12)$$

$$AAS_{\ell_2}(x) = \frac{\|f_{\text{DNN}}(x)\|_2 / \sqrt{\dim(f_{\text{DNN}})}}{\|f_{\text{WSN}}(x)\|_2 / \sqrt{\dim(f_{\text{WSN}})}}, \quad (5.13)$$

$$AAS_F(x) = \frac{\|f_{\text{DNN}}(x)\|_F / \sqrt{\dim(f_{\text{DNN}})}}{\|f_{\text{WSN}}(x)\|_F / \sqrt{\dim(f_{\text{WSN}})}}, \quad (5.14)$$

where $f_{\text{DNN}}(x)$ and $f_{\text{WSN}}(x)$ represent the extracted DNN and WSN features respectively, and $\|\cdot\|_1$, $\|\cdot\|_2$ and $\|\cdot\|_F$ denote the ℓ_1 , ℓ_2 and ℓ_F norm respectively. The choice of the Frobenius norm is motivated by its ability to capture the overall magnitude of feature perturbations while being invariant to feature dimensionality. The dimensional normalisation term $\sqrt{\dim(\cdot)}$ is crucial for fair comparison between features of different dimensionalities. For the DNN feature vector, $\dim(f_{\text{DNN}})$ equals the length of the vector, while for the WSN feature matrix, $\dim(f_{\text{WSN}})$ is calculated as the product of its row and column dimensions ($m \times n$). This normalisation ensures that the AAS values are comparable regardless of the underlying feature dimensionalities.

The selection of these three norm types is motivated by their distinct mathematical properties in quantifying feature magnitudes. The ℓ_1 norm sums the absolute values of all elements, providing sensitivity to small perturbations across all feature dimensions. The ℓ_2 norm emphasises larger deviations due to its quadratic nature while being less sensitive to small variations. It is noteworthy that in this study, $f_{\text{DNN}}(x)$ produces a feature vector while $f_{\text{WSN}}(x)$ generates a feature matrix. For vector-form features like $f_{\text{DNN}}(x)$, the Frobenius norm is equivalent to the ℓ_2 norm. However, for matrix-form features such as $f_{\text{WSN}}(x)$, the Frobenius norm is defined as:

$$\|f\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |f_{ij}|^2}, \quad (5.15)$$

where f is a matrix and f_{ij} represents its elements. The Frobenius norm

effectively captures the overall magnitude of matrix elements while preserving the dimensional consistency required for comparative analysis.

It is also important to note that the Wavelet Scattering Network is configured with the following parameters:

- Two 2D Morlet wavelet filterbanks are utilised with both quality factor $Q = 1$
- Invariance scale: $[112 \ 112]$, defining the spatial extent of the scattering coefficients
- Number of rotations: $[8 \ 8]$ for both first and second-order scattering paths, providing rotational invariance at 8 different angles
- For dimension reduction, max pooling (Kernel size = 3×3 ; Stride = 3; No Padding) is utilised to the first three dimensions and average to the fourth dimension.

These WSN parameters yield a feature matrix $f_{\text{WSN}}(x) \in \mathbb{R}^{m \times n}$, where m represents the number of scattering paths and n denotes the spatial resolution of the scattering coefficients. The specific dimensions are determined by the input image size and the above parameter settings.

This comprehensive evaluation of different norm types enables the assessment of which quantisation method best captures the robustness disparity between WSN and DNN features under varied adversarial attacks, which is elaborated in the next section.

5.2.1 Experimental Validation of AAS

As discussed above, the proposed Adversarial Awareness Score (AAS) transforms high-dimensional feature vectors from WSN and DNN into a single scalar value. While the theoretical foundation suggests that this scalar value could effectively capture the robustness disparity between WSN and DNN features, experimental validation is still needed to verify the following crucial aspects:

- The effectiveness of dimensional reduction from feature vectors to a scalar value through the proposed norms-based formulation.

- The discriminative capability of AAS in distinguishing between clean images and adversarial examples, specifically:
 - Whether AAS exhibits statistically significant differences between the clean image and AE
 - The consistency of these differences across various attack types and parameters
 - The robustness of the metric against different input variations

This experimental validation is particularly crucial as the transformation from high-dimensional feature spaces to a single scalar metric must preserve the underlying robustness characteristics that differentiate WSN and DNN features under adversarial perturbations. Through comprehensive empirical evaluation, this work aims to verify that this dimensional reduction maintains sufficient discriminative power for practical adversarial detection applications.

Experimental Setup

- **Dataset:** A subset of the ImageNet dataset [19] is utilised for the experiment stage, and the subset contains 10 000 images, which are randomly selected from the dataset.
- **Pre-trained DNN Models:** To assess feature robustness across the whole spectrum of DNN architectures, a diverse set of pre-trained DNN models is used, which include **ResNet-50** [14], **VGG-16** [10], **Inception-v3** [121], **GoogLeNet** [118], **EfficientNet** [15] and **NASNet-Large** [122].

Table 5.1: *Adversarial Attack Parameters Configuration*

Attack Type	Parameters Configuration	Attack ID
FGSM	$\epsilon = 16/255$	A1
FGSM	$\epsilon = 32/255$	A2
PGD	$\epsilon = 16/255, \alpha = 0.25, \text{iterations}=50$	A3
PGD	$\epsilon = 32/255, \alpha = 0.25, \text{iterations}=50$	A4
CW	confidence=0.2, learning rate=0.01, iterations=250	A5
CW	confidence=0.1, learning rate=0.01, iterations=500	A6

Note: FGSM/PGD parameters (ϵ, α) are in pixel scale (0-255)

Results and Analysis

Table 5.2: Statistical Analysis of Adversarial Awareness Scores. Note: $Q1$ and $Q3$ represent the first and third quartiles, respectively. Outliers are counted from a sample size of 10,000 images per category.

AAS Type	Input Type	Median	Quartiles		Outliers (per 10k)
			Q1	Q3	
AAS_F	Clean Images	0.845	0.682	1.054	172
	FGSM Attacks	1.984	1.542	2.386	148
	PGD Attacks	2.332	1.847	2.865	155
	CW Attacks	2.277	1.793	2.782	187
AAS_{ℓ_2}	Clean Images	1.056	0.856	1.321	279
	FGSM Attacks	2.437	1.927	2.885	191
	PGD Attacks	2.922	2.377	3.580	258
	CW Attacks	1.526	1.256	1.954	201
AAS_{ℓ_1}	Clean Images	3.274	2.591	4.011	174
	FGSM Attacks	3.327	2.622	4.064	168
	PGD Attacks	3.950	3.142	4.866	177
	CW Attacks	3.192	2.577	3.989	166

The effectiveness of different Adversarial Awareness Scores (AAS) in distinguishing between clean images and adversarial examples is analysed through Table 5.2 and visualised in Figure 5.1. The results demonstrate varying degrees of discriminative power across different norm formulations.

For AAS_F , the most substantial discriminative capability between clean and adversarial samples is observed. As shown in Table 5.2, clean images maintain a distinctly lower median score of 0.845, while all adversarial examples demonstrate substantially higher scores ranging from 1.984 to 2.332. This clear separation is visualised in Figure 5.1, where the box plot for clean images shows minimal overlap with those of adversarial examples. The consistent elevation in scores across all attack types (FGSM: 1.984, CW: 2.277, PGD: 2.332) indicates that AAS_F reliably detects adversarial perturbations regardless of the attack method.

For AAS_{ℓ_2} , partial effectiveness in adversarial detection is observed. While it clearly distinguishes FGSM attacks (median 2.437) and PGD attacks (median 2.922) from clean images (median 1.056), it shows limited discrimination for CW attacks. The median score for CW attacks (1.526) remains relatively close to that of clean images, suggesting that AAS_{ℓ_2} may not reliably detect this type of

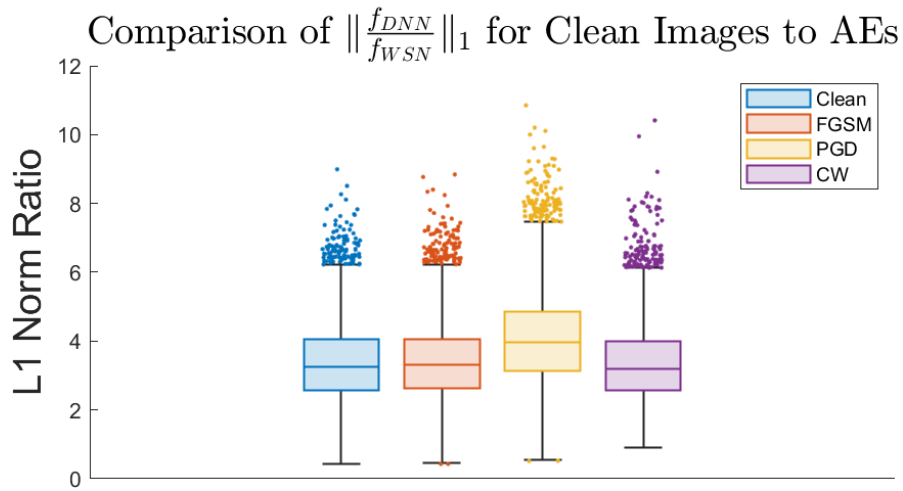
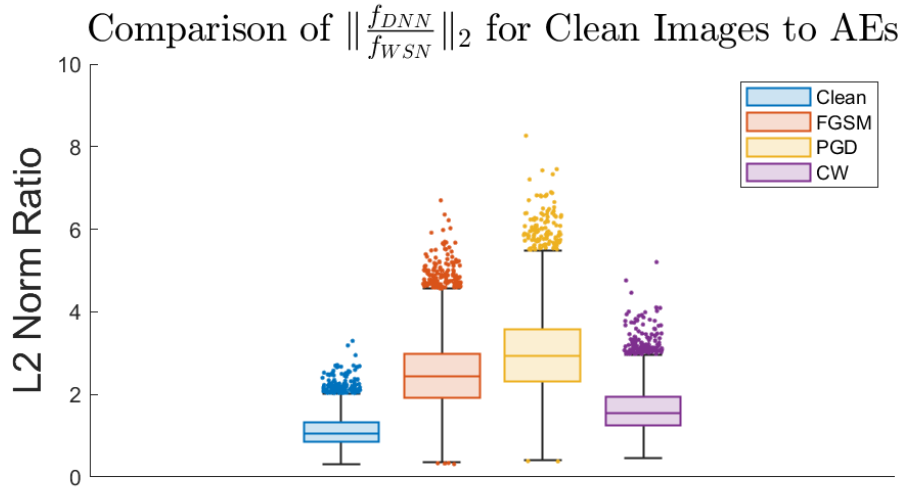
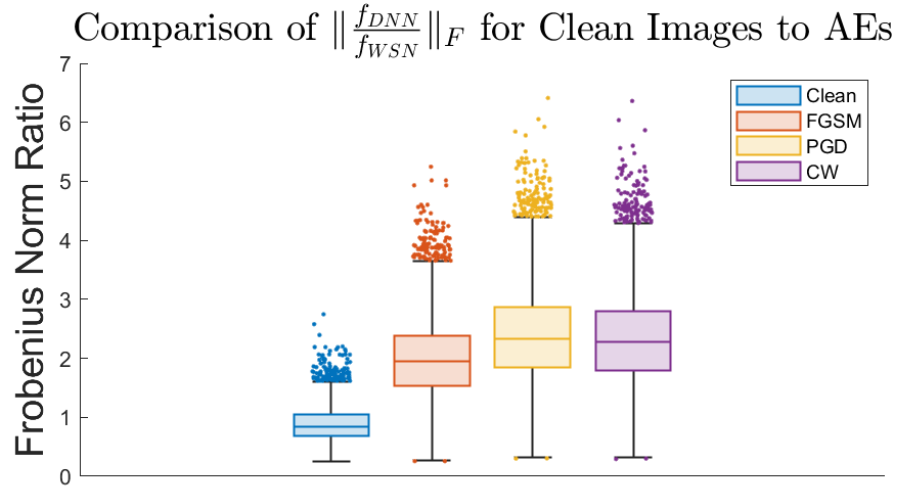


Figure 5.1: Illustration of Adversarial Awareness Scores (AAS) for Varied Adversarial Attacks

adversarial manipulation.

For AAS_{ℓ_1} , the least effectiveness for adversarial detection is observed. As evident in both Table 5.2 and Figure 5.1, the scores for clean images (median 3.274) are comparable to those of adversarial examples (ranging from 3.192 to 3.950). The substantial overlap in score distributions and similar medians indicates that AAS_{ℓ_1} fails to provide meaningful discrimination between clean and adversarial samples.

These findings suggest that AAS_F is the most suitable for adversarial detection, offering consistent and clear differentiation between clean and adversarial images. AAS_{ℓ_2} provides partial utility but may miss certain attack types, while AAS_{ℓ_1} appears inadequate for reliable adversarial detection. This comparative analysis demonstrates the importance of selecting appropriate norms in developing robust adversarial detection methods.

5.3 AAS-Based Adversarial Detection

The experimental analysis demonstrates distinct separations in AAS_F scores between clean images (0.682–1.054) and adversarial examples (1.542–2.865). This separation enables the development of a systematic detection methodology that transforms these scores into quantifiable adversarial likelihood values.

5.3.1 Detection Framework

The detection framework comprises two primary components: score normalisation and likelihood mapping. These components establish a robust pipeline for identifying adversarial examples based on their AAS_F scores.

Score Normalisation

For an input image x with AAS_F score s_a , the normalisation process utilises the empirically determined median values:

$$s_{\text{norm}} = \frac{s_a - \mu_a}{\mu_c - \mu_a} \quad (5.16)$$

where $\mu_c = 0.845$ represents the median score for clean samples and $\mu_a = 2.198$ represents the median score for adversarial samples (calculated as the

average of FGSM, PGD, and CW medians).

Likelihood Calculation

The normalised score maps to an adversarial likelihood value through a sigmoid function:

$$L = \frac{1}{1 + e^{5(s_{\text{norm}} - \tau)}} \quad (5.17)$$

where $\tau = 0.68$ serves as the classification threshold, with the scaling factor of 5 providing distinct separation between classes.

5.3.2 AAS Sensitivity and Robustness Analysis

In addition to evaluating the AAS on adversarial examples, we conducted a supplementary experiment to assess its behaviour under non-adversarial image corruptions. For this study, we define non-adversarial corruption as degradations arising from common image perturbations, specifically Gaussian noise, Salt and Pepper noise, Gaussian blur, and JPEG compression with varying quality factors. These types of corruptions are frequently encountered in real-world imaging systems and represent natural distribution shifts rather than deliberate adversarial perturbations.

The motivation for this experiment is to determine whether AAS can reliably distinguish between adversarial examples and non-adversarial corrupted inputs. AEs are deliberately crafted to manipulate DNN decision boundaries, whereas corrupted images typically preserve semantic content despite visual degradation. However, it must be emphasised that non-adversarial corruptions may occasionally cause misclassifications in the target DNN, effectively producing an adversarial effect by coincidence. For example, a severe JPEG compression artefact or high-intensity random noise could alter the network’s prediction, despite not being intentionally optimised to do so.

To investigate this, we computed the AAS for each corruption type (with parameter settings summarised in Table 5.3) and compared the resulting score distributions against those obtained for adversarial examples. The results demonstrate that, across moderate corruption levels, AAS values remain close to those of clean images, with only marginal increases relative to the adversarial

Table 5.3: *Parameters of non-adversarial corruption methods used for AAS evaluation.*

Corruption Method	Parameters
Gaussian Noise	Variance = 0.01 (Low), 0.05 (High)
Salt-and-Pepper Noise	Density = 0.01 (Low), 0.05 (High)
Gaussian Blur	Standard deviation $\sigma = 1.0$ (Low), 2.0 (High)
JPEG Compression	Quality factor = 80%, 60%, 40%

baseline. This indicates that AAS is insensitive primarily to benign corruptions and is therefore specific to adversarial perturbations. Nonetheless, in extreme cases of corruption, isolated samples exhibit elevated AAS values consistent with the aforementioned coincidental misclassification effect.

These findings highlight two key aspects of the proposed approach. First, the robustness of AAS to common image degradations confirms that it does not indiscriminately flag corrupted inputs as adversarial, which is critical for practical deployment. Second, the occasional rise in AAS for heavily corrupted samples reinforces the notion that adversarial-like behaviour may emerge even without deliberate attack, underscoring the need to calibrate detection thresholds in real-world applications carefully.

5.3.3 Implementation Framework

The detection algorithm operates as follows:

Algorithm 6 AAS_F Score to Adversarial Likelihood Conversion

Require:

- 1: Input image x , target DNN D_θ , pre-defined WSN for AAS_F calculation

Ensure:

- 2: Clean median score $\mu_c = 0.845$
 - 3: Adversarial median score $\mu_a = 2.198$
 - 4: Threshold $\tau = 0.68$
 - 5: **Calculate** $s_a \leftarrow \text{AAS}_F(x)$
 - 6: **Calculate** $s_{\text{norm}} \leftarrow \frac{s_a - \mu_a}{\mu_c - \mu_a}$
 - 7: **Calculate** Adversarial likelihood $L \leftarrow \frac{1}{1 + e^{5(s_{\text{norm}} - \tau)}}$
 - 8: **return** L
-

The algorithm outputs a likelihood value $L \in [0, 1]$ with the following classification criteria, note that in the practical implementation where the task

is a binary classification scenario, "Adversarial" and "Possibly Adversarial" are labelled as "**AE**", "Possibly Clean" and "Clean" as "**Clean**". It must be mentioned that the threshold values in this experiment are experimentally acquired.

$$\text{Classification} = \begin{cases} \textbf{Adversarial} & \text{if } L > 0.8 \\ \textbf{Possibly Adversarial} & \text{if } 0.5 \leq L \leq 0.8 \\ \text{Possibly Clean} & \text{if } 0.2 \leq L \leq 0.5 \\ \text{Clean} & \text{if } L < 0.2 \end{cases} \quad (5.18)$$

Threshold Optimisation

To identify the optimal threshold value, classification performance across different threshold values is evaluated using a separate validation set of 2000 samples. The evaluation metrics included accuracy, precision, recall, and F1 score. Based on the statistical analysis results, the AAS score distributions were characterised as:

- Clean images: median 0.845 (Q1: 0.682, Q3: 1.054)
- Adversarial examples:
 - FGSM: median 1.984 (Q1: 1.542, Q3: 2.386)
 - PGD: median 2.332 (Q1: 1.847, Q3: 2.865)
 - CW: median 2.277 (Q1: 1.793, Q3: 2.782)

Figure 5.2 shows the relationship between threshold values and classification performance. Through the analysis, $\tau = 0.68$ is identified as the optimal threshold value, yielding:

- Accuracy: 88.1%
- Precision: 88.0%
- Recall: 88.2%
- F1 Score: 88.1%

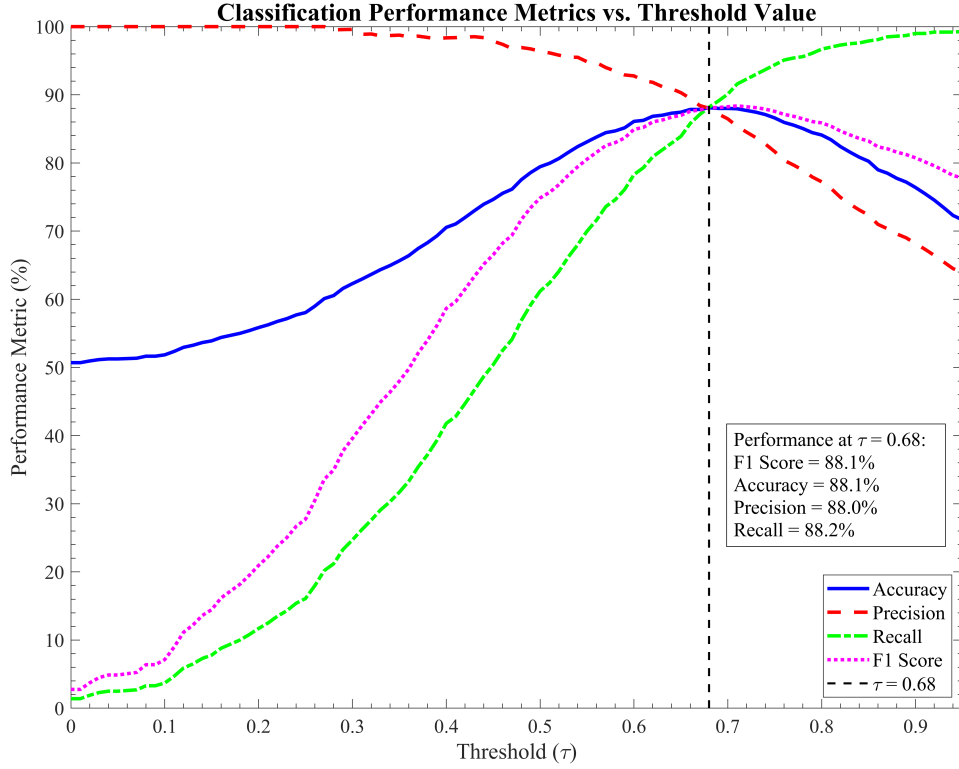


Figure 5.2: Classification performance metrics versus threshold values. The vertical line at $\tau = 0.68$ indicates the optimal threshold value that maximises overall performance.

The optimal threshold $\tau = 0.68$ aligns with the natural separation point between clean and adversarial AAS score distributions, as evidenced by the gap between the clean images' upper quartile (1.054) and adversarial examples' lower quartiles (1.542-1.847). This threshold provides robust classification performance while maintaining balanced precision and recall rates. The stability of performance metrics around the optimal threshold also indicates that the classification is reliable and not overly sensitive to minor threshold variations. It must be emphasised that the key hyperparameters of the AAS are inherently task-specific, meaning that their optimal values depend on the characteristics of the dataset and the type of model being employed. As such, these hyperparameters cannot be universally fixed; instead, they need to be carefully fine-tuned when applying AAS to a different dataset to ensure reliable performance and maintain its effectiveness across diverse tasks.

5.3.4 Performance Evaluation

Table 5.4: Comparative Detection Results on ImageNet Dataset. Note: FS refers to Feature Squeeze.

Method	Attack	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
AAS_F (Ours)	A1	86.90	87.60	87.40	87.50
	A2	86.45	87.10	86.90	87.00
	A3	87.95	88.50	87.90	88.20
	A4	86.60	87.00	86.80	86.90
	A5	88.50	88.30	88.10	88.20
	A6	87.30	87.60	87.20	87.40
FreqVAE [146]	A1	82.50	76.60	80.80	78.60
	A2	80.50	78.60	76.80	77.70
	A3	83.50	76.10	79.80	77.90
	A4	79.50	79.60	77.80	78.70
	A5	83.00	75.60	80.30	77.90
	A6	80.00	79.10	77.30	78.20
SQ+SF [147]	A1	65.00	67.50	74.00	71.00
	A2	58.20	60.30	69.00	64.10
	A3	72.40	74.60	77.50	76.00
	A4	63.50	65.80	71.00	68.20
	A5	74.10	75.50	77.10	76.30
	A6	70.20	72.10	75.00	73.50
FS [148]	A1	50.50	47.00	49.00	48.00
	A2	47.30	43.50	44.90	44.20
	A3	63.20	59.80	61.20	60.50
	A4	55.00	51.20	52.80	52.00
	A5	70.10	68.50	69.90	69.20
	A6	68.00	65.70	66.90	66.30

The proposed AAS_F method is evaluated against three state-of-the-art detection approaches: FreqVAE [146], SQ+SF [147], and FS [148] on the ImageNet dataset. The detection performance is assessed using standard metrics, including accuracy, precision, recall, and F1 score, as shown in Table 5.4.

The AAS_F method consistently outperformed the other approaches across all attack types (A1–A6), achieving an average accuracy of 87.28% and an F1 score of 87.53%. As shown in Figure 5.3, its normalised confusion matrix shows a balanced true positive rate (87.4%) and true negative rate (87.2%), clearly outperforming the other methods. AAS_F maintained stable metrics

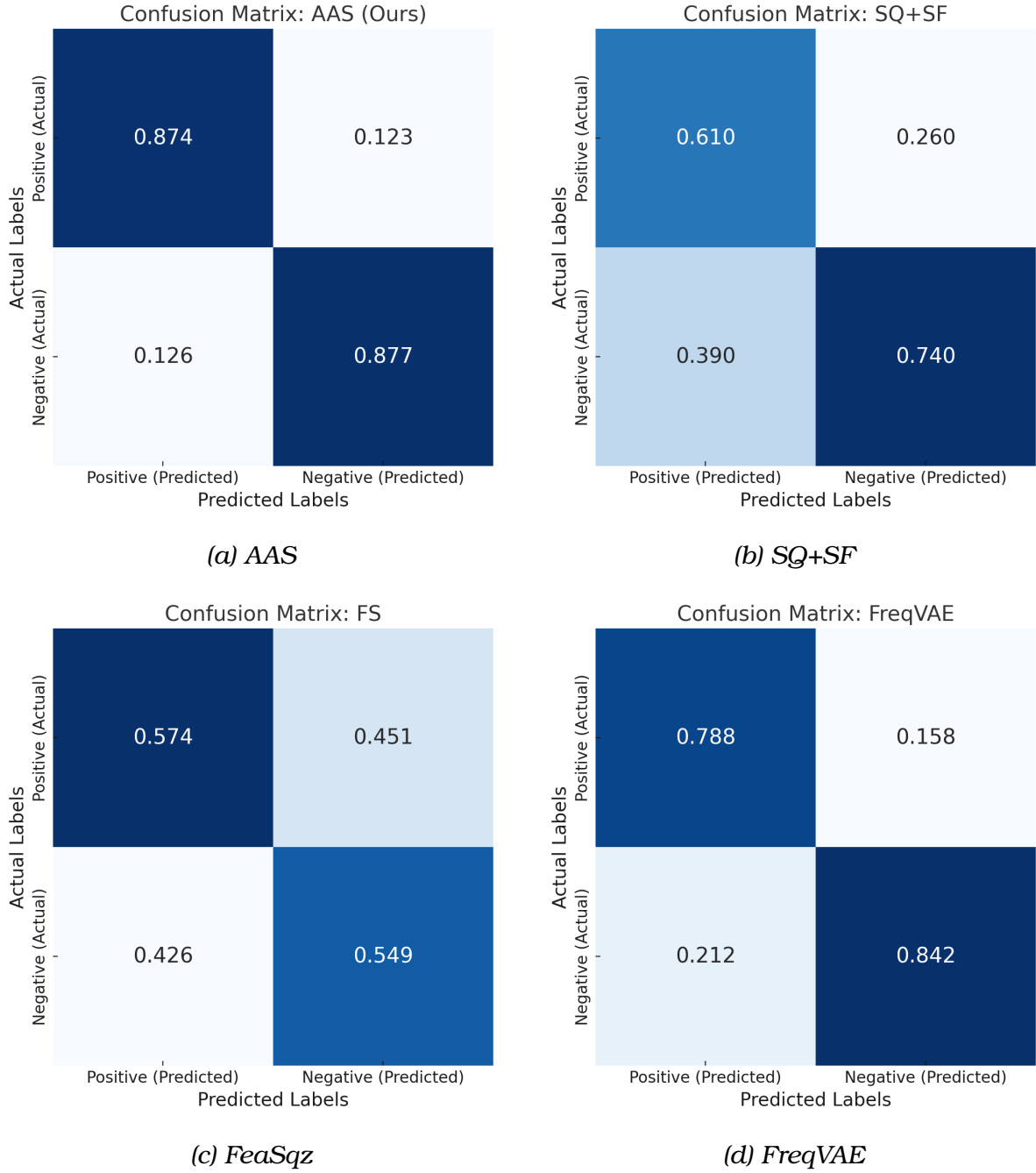


Figure 5.3: Normalised Confusion Matrices for (a) AAS_F ; (b) $SQ+SF$ (scalar Quantisation and Spatial Smoothing); (c) $FeaSqs$ (Feature Squeeze) and (d) $FreqVAE$ AE Detection

with little variation, achieving accuracy between 86.45% and 88.50%. The balanced average precision (87.68%) and recall (87.38%) further demonstrate its strong detection capabilities for both clean and adversarial samples. The

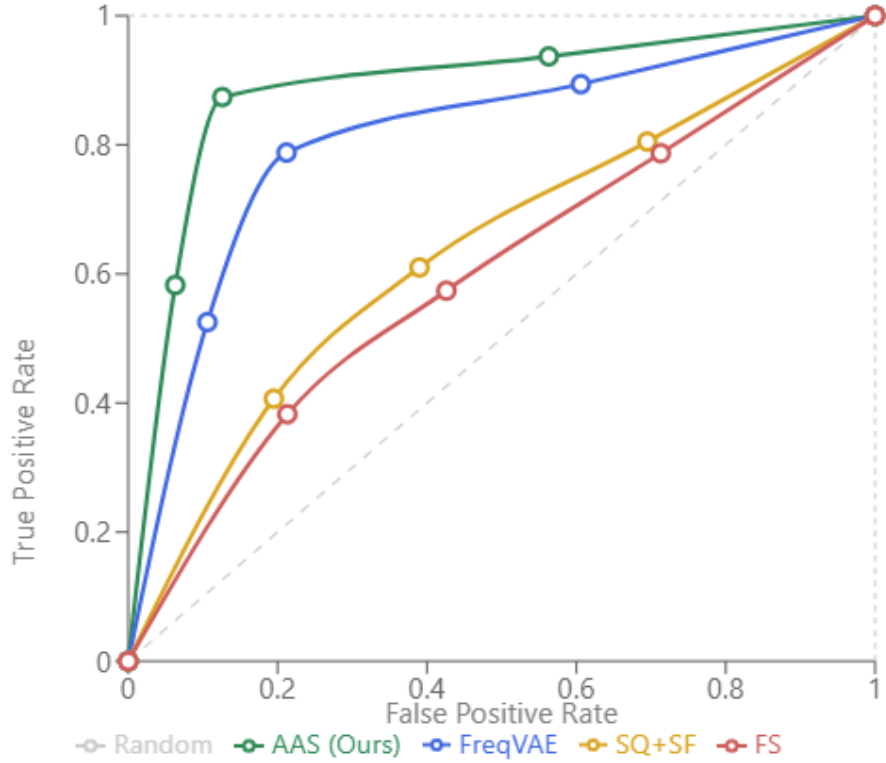
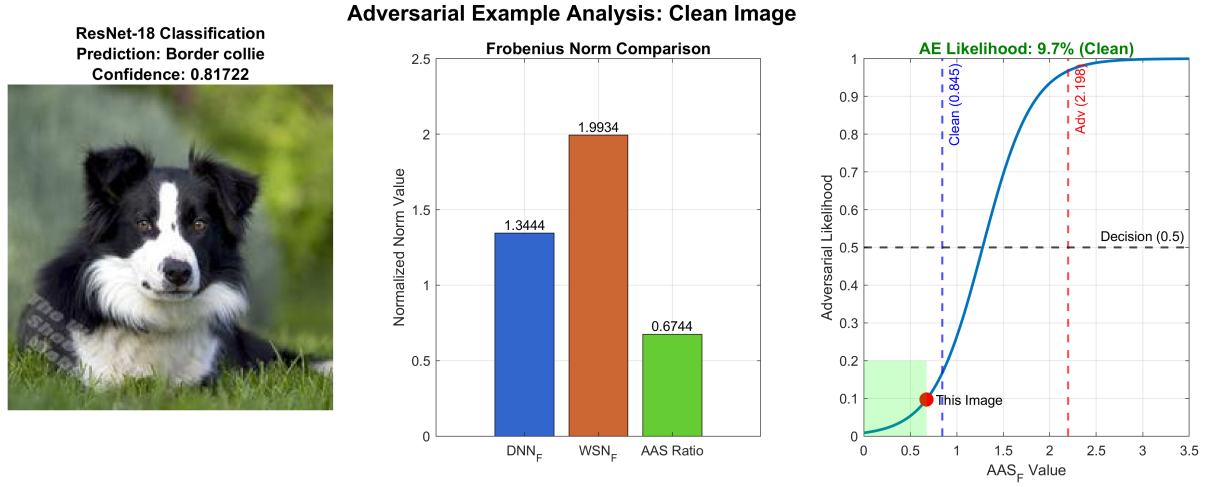


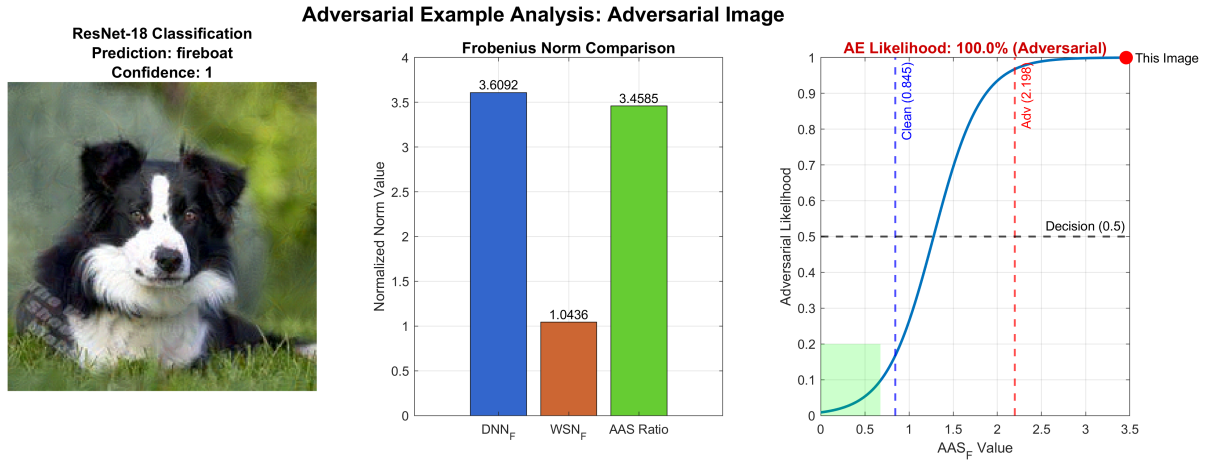
Figure 5.4: ROC curves comparing the detection performance of AAS (Ours), FreqVAE, SQ+SF and FS for adversarial example detection.

superior performance of our proposed method is further validated through ROC curve analysis presented in Figure 5.4, which compares the detection performance of AAS and other methods for adversarial example detection. The ROC curves clearly illustrate that our AAS method achieves consistently higher true positive rates while maintaining lower false positive rates across all decision thresholds compared to the baseline methods. Notably, the AAS curve demonstrates a steeper initial rise and maintains a position closer to the top-left corner of the ROC space, indicating superior discriminative ability. The area under the ROC curve (AUC) for our method significantly exceeds those of FreqVAE, SQ+SF, and FS, with the performance gap being most pronounced in the critical low false positive rate region, where practical deployment requires high precision. This comprehensive ROC analysis corroborates our quantitative results. It confirms that the AAS method provides robust and reliable adversarial detection across diverse attack scenarios, making it a practical solution for real-world adversarial defence applications.

FreqVAE [146] demonstrates the second-best performance, with an average



Original input image with ResNet-18 detection results



Adversarial Example with ResNet-18 detection results

Figure 5.5: Comprehensive adversarial example detection pipeline. From left to right, the respective images show (1) the original input image with ResNet-18 classification results, displaying the predicted class label and confidence score. (2) Frobenius norm comparison between DNN_F features (extracted from ResNet-18 pool5 layer) and WSN_F features (from wavelet scattering transform), with their ratio (AAS_F score) used for adversarial detection. (3) AAS-based adversarial likelihood assessment showing the modified sigmoid mapping from AAS_F score to adversarial probability, with classification regions (Clean to Adversarial) and the current image's position marked. (More examples are shown in Figure.C.1 and C.2 in Appendix C due to space limit.)

accuracy of 81.50% and an F1 score of 78.17%. However, its slightly lower precision (77.60% on average) compared to recall (78.80%) suggests a small tendency towards false positives. This imbalance is reflected in its confusion matrix (Figure 5.3), with a true negative rate of 0.842 and a true positive rate of 0.788. Despite this, its performance remains steady across attack types, with accuracy ranging from 79.50% to 83.50%.

SQ+SF [147] exhibits moderate detection ability with an average accuracy of 67.23%. It had a higher recall (73.93%) than precision (69.30%), suggesting it tends to classify more samples as adversarial. This tendency is evident in its confusion matrix, which shows an imbalance between the true positive rate (73.9%) and the true negative rate (60.5%). Its performance varied across different attacks, with accuracy between 58.20% and 74.10%.

FS [148] exhibits the lowest detection performance, with an average accuracy of 59.02%. As shown in Figure 5.3, its confusion matrix highlights weak classification, with a true positive rate of 0.574 and a true negative rate of 0.606. Its performance changed widely across attacks, with accuracy ranging from 47.30% to 70.10%. The relatively low and uneven precision (55.95%) and recall (57.45%) demonstrate its limitations in reliably detecting adversarial examples.

These results demonstrate that the AAS_F method provides the best performance for adversarial detection. Compared to other methods, it offers higher accuracy and greater consistency across attack types, with balanced confusion matrix metrics that support its use in practical applications.

AAS Performance on Non-Adversarially Corrupted Images

The proposed Adversarial Awareness Score (AAS) was evaluated on 100,000 ImageNet validation samples across ten corruption modalities and six state-of-the-art deep neural network (DNN) architectures, as outlined in Section 5.2.1. The objective of this analysis was to determine the robustness of AAS under natural input degradations and to assess its ability to distinguish such perturbations from adversarial manipulations.

Table 5.5 reports the sensitivity and false positive rates for each corruption modality. Across all categories, AAS demonstrates consistently high sensitivity, with an overall mean of $96.1\% \pm 2.0$, indicating that the majority of corrupted

images are correctly identified as non-adversarial. The false positive rate remains low, averaging $3.9\% \pm 2.0$, confirming that only a small fraction of corrupted samples were erroneously classified as adversarial.

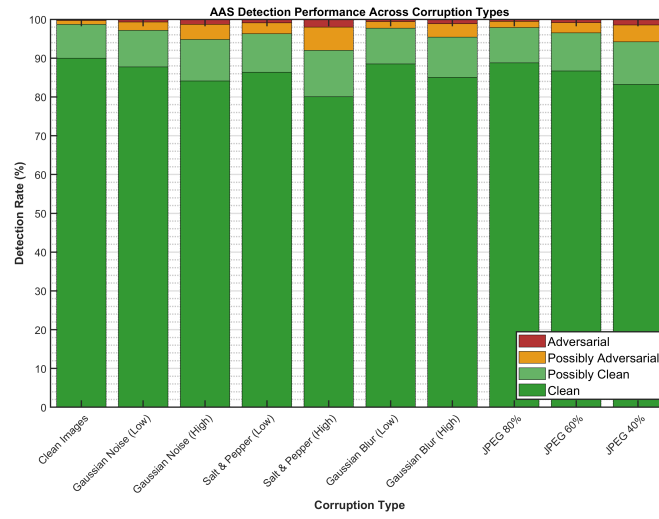
Table 5.5: *AAS Sensitivity and Robustness Analysis across Non-Adversarial Corruptions*

Corruption Type	Sensitivity (%)	False Positive (%)	Samples
Clean Images	98.7	1.3	10,000
Gaussian Noise (Low)	97.2	2.8	10,000
Gaussian Noise (High)	94.8	5.2	10,000
Salt & Pepper (Low)	96.3	3.7	10,000
Salt & Pepper (High)	92.0	8.0	10,000
Gaussian Blur (Low)	97.7	2.3	10,000
Gaussian Blur (High)	95.4	4.6	10,000
JPEG 80%	97.9	2.1	10,000
JPEG 60%	96.5	3.5	10,000
JPEG 40%	94.2	5.8	10,000
Overall Performance	96.1 ± 2.0	3.9 ± 2.0	100,000

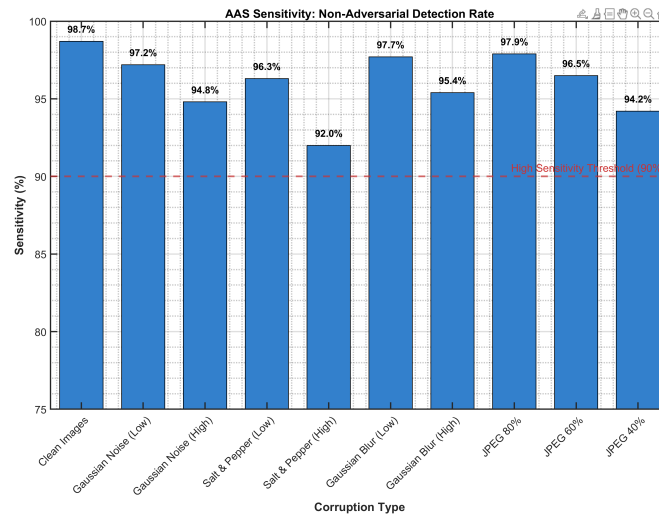
Complementary insights are provided in Figure 5.6, which presents a consolidated visual analysis of AAS performance. The three vertically arranged panels illustrate: (a) the distribution of detection outcomes across four categories, (b) sensitivity analysis for non-adversarial inputs, and (c) false positive rate analysis for operational reliability.

The combined quantitative and visual analyses yield the following principal observations:

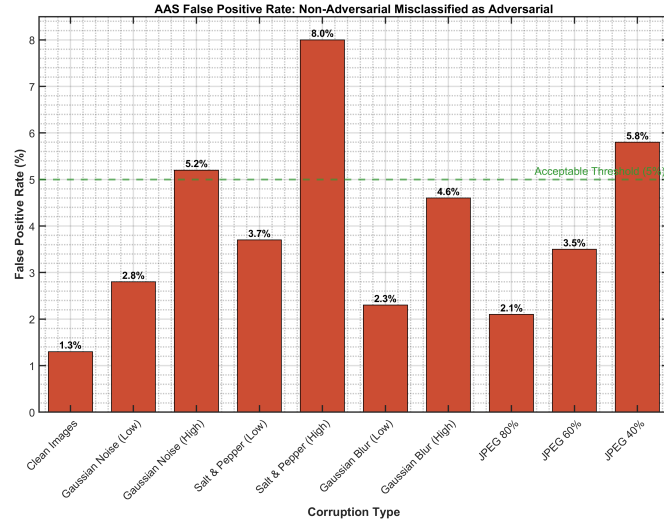
1. **Detection Outcome Distribution:** Clean images achieve a 98.7% correct classification rate, providing a reliable baseline. Natural corruptions sustain high performance, with JPEG compression artefacts yielding accuracies in the 94.2–97.9% range. High-density salt-and-pepper noise is the most challenging modality, with 92.0% correct classification and the highest false positive rate (8.0%).
2. **Sensitivity Performance:** The mean sensitivity across modalities is 96.1%, with a minimum of 92.0% under high-density salt-and-pepper noise. The low standard deviation (2.0%) indicates consistent behaviour



(a) Detection outcome distribution



(b) Sensitivity analysis



(c) False positive rate analysis

Figure 5.6: Comprehensive evaluation of AAS performance across non-adversarial corruptions. Panels show: (a) distribution of detection outcomes, (b) sensitivity analysis, and (c) false positive rate analysis.

across diverse corruption scenarios. All modalities surpass the 90% sensitivity threshold, confirming robustness in real-world conditions.

3. **False Positive Rates:** The average false positive rate is 3.9%. Most modalities remain below 6%, whilst clean images achieve 1.3%. JPEG compression artefacts are particularly well managed, producing false positives between 2.1% and 5.8%.
4. **Effect of Corruption Severity:** More severe corruptions lead to modestly elevated false positive rates and reduced sensitivity. Nevertheless, the maximum observed false positive rate of 8.0% remains within operational limits and below thresholds associated with alert fatigue.
5. **Degradation Behaviour:** Performance exhibits a controlled decline with increasing corruption severity. Mild degradations such as low-variance Gaussian noise and high-quality JPEG compression maintain sensitivity above 97%, whereas severe conditions reduce sensitivity to 92–95%. This behaviour is advantageous for systems required to operate reliably under variable environmental conditions.

In summary, the overall experimental results confirm that AAS is highly robust to natural corruptions, with only marginal performance degradation under extreme conditions. The method achieves high sensitivity ($96.1\% \pm 2.0$) whilst maintaining a low false positive rate ($3.9\% \pm 2.0$), thereby demonstrating its practical feasibility for deployment in adversarial detection pipelines.

5.3.5 Summary

The experimental results demonstrate several significant advantages of the proposed AAS method for adversarial example detection. First, AAS achieves state-of-the-art performance across different ResNet architectures (ResNet-18, ResNet-50, and ResNet-101) on large-scale ImageNet data, consistently maintaining high detection accuracy above 87% across various attack scenarios. This robust performance across different model architectures highlights the method’s versatility and reliability in practical applications.

Second, whilst existing methods like FreqVAE [146], SQ+SF [147], and FS [148] show degraded performance when facing strong adversarial perturb-

ations, AAS maintains high detection accuracy regardless of the perturbation magnitude. As demonstrated in Table 5.4 and Figure 5.3, the method achieves consistent performance across all attack types, whereas competing methods show significant performance variation. This resilience to perturbation strength is particularly valuable in real-world scenarios where the magnitude of adversarial attacks cannot be predicted.

Third, AAS offers significant practical advantages in terms of implementation simplicity. Unlike other approaches that require training additional models or complex detection frameworks, the method operates directly on the feature representations extracted from the target DNN. This streamlined approach reduces computational overhead and simplifies deployment in existing systems.

However, several limitations in the current study are acknowledged. Due to computational constraints, the experiments were conducted on a subset of 60,000 images from ImageNet and focused on ResNet architectures (ResNet-18, 50, and 101). While these results are promising, further validation on other architectures and larger datasets would be valuable. Additionally, the default parameter settings for comparative methods as reported in their respective papers are utilised, maintaining consistency with their published best performance. While this approach ensures fair comparison with their reported results, parameter optimisation for the specific experimental setup could yield different performance characteristics.

It is worth noting that several other adversarial example detection methods exist in the literature but were not included in the comparative analysis. Many of these methods are based on pre-trained models and were primarily validated on smaller datasets such as MNIST [8] or CIFAR [5]. Their publicly available models are incompatible with large-scale ImageNet detection tasks, and retraining them for the experimental setup would be impractical due to computational constraints and potential implementation differences. The selection of comparison methods focused on approaches that have demonstrated scalability to larger datasets and provided clear implementation guidelines for reproduction.

Future work could explore the impact of parameter tuning across different experimental contexts and attack scenarios and extend the evaluation to other model architectures and larger datasets.

5.4 AAS-Guided Jacobian Regularisation for Model Robustness

Having demonstrated the effectiveness of Adversarial Awareness Scores (AAS) in detecting adversarial examples, the potential of AAS as a regularisation approach to enhance model robustness is now investigated. The vulnerability to adversarial examples is viewed as a manifestation of insufficient model generalisation, where the decision boundaries are overly sensitive to small perturbations in the input space. To address this, an adaptive Jacobian regularisation method that leverages AAS to improve the model’s robustness against adversarial perturbations is proposed. The proposed methodology is elaborated as follows:

Given an input image $x_i \in \mathcal{X}$ and a pre-trained DNN D_θ , the output of $D_\theta(x)$ is a vector of probabilities $\mathbf{z}$ with the dimension of $C \times 1$, where C is the number of the output classes. For example, for large-scale datasets such as ImageNet, $C = 1000$. Typically, whilst training a DNN, the model parameters θ are optimised iteratively to minimise the loss function $\mathcal{L}_{total}$, which is calculated in every epoch:

$$\mathcal{L}_{total} = \mathcal{L}_{task} + \mathcal{L}_R, \quad (5.19)$$

where $\mathcal{L}_{task}$ is the task-specific loss function (e.g., *mse*, *cross-entropy*, etc.), $\mathcal{L}_R$ denotes the regularisation term to penalise large weights or excessive sensitivity to input changes, thus improving the generalisation of the DNN. For example, common regularisation terms include ℓ_1 , ℓ_2 (Weight Decay) and Jacobian Regularisation ($\mathcal{L}_{JR}$), whilst the Jacobian regularisation term can be efficiently approximated using random projection:

$$\|J(\mathbf{x})\|_F^2 \approx \frac{1}{n_{\text{proj}}} \sum_{\mu=1}^{n_{\text{proj}}} \left\| \frac{\partial(\mathbf{v}_\mu^\top \cdot \mathbf{z})}{\partial \mathbf{x}} \right\|^2, \quad (5.20)$$

where $\mathbf{v}_\mu^\top$ are random unit vectors with the dimension of $C \times \dim(\mathcal{B})$, $\mathcal{B}$ denotes the number of images per minibatch $\mathcal{B}$ and n_{proj} is the number of random projections for each minibatch $\mathcal{B}$ during training. Note that the conventional Jacobian regularisation employs a pre-defined constant coefficient λ_{JR} to the

loss function, which can be expressed as:

$$\mathcal{L}_{total} = \mathcal{L}_{task} + \frac{\lambda_{JR}}{2} \|J(\mathbf{x})\|_F^2, \quad (5.21)$$

whilst the method replaces this constant coefficient with an adaptive value $\lambda_{AAS_F}(\mathbf{x})$, so that the exact computation of the total loss function during training per given mini-batch $\mathcal{B}$ can be expressed as:

$$\mathcal{L}_{total} = \mathcal{L}_{task}(\mathcal{B}, \theta) + \left[\frac{1}{|\mathcal{B}|} \sum_{k \in \mathcal{B}} \|J(x_k)\|_F^2 \frac{\lambda_{AAS_F}(x_i)}{2} \right], \quad (5.22)$$

where $\mathcal{L}_{task}(\mathcal{B}, \theta)$ denotes the task-specific supervised loss of the mini-batch $\mathcal{B}$ consists of a set of labelled examples (x_k, y_k) , k is the number of examples per mini-batch. For clean images with low AAS_F , $\lambda_{AAS}(\mathbf{x})$ will be minimised, thus causing a small loss, whilst for adversarial images with high AAS values, $\lambda_{AAS}(\mathbf{x})$ will be maximised thereby penalising the loss function. This adaptive approach provides several advantages: (1) automatic adjustment of regularisation penalising strength based on input characteristics, (2) preservation of model accuracy on clean images through lighter regularisation, and (3) enhanced robustness against adversarial inputs via stronger regularisation when adversarial patterns are detected.

As shown in Fig. 5.7, the AAS-guided Jacobian regularisation adaptively scales the penalty with the AAS_F score: the left panel contrasts a fixed regularisation weight with our adaptive schedule, which increases with AAS_F and aligns with the median AAS_F of clean versus adversarial inputs (vertical dashed lines). The right panel visualises the resulting penalty surface over Jacobian norm and AAS_F , indicating more substantial penalties for high- AAS_F inputs with large Jacobian norms, while keeping penalties low for clean samples to preserve nominal performance.

5.4.1 Experimental Setup

The proposed AAS-guided Jacobian regularisation approach is evaluated on large-scale image classification using various ResNet architectures. The experimental setup is as follows:

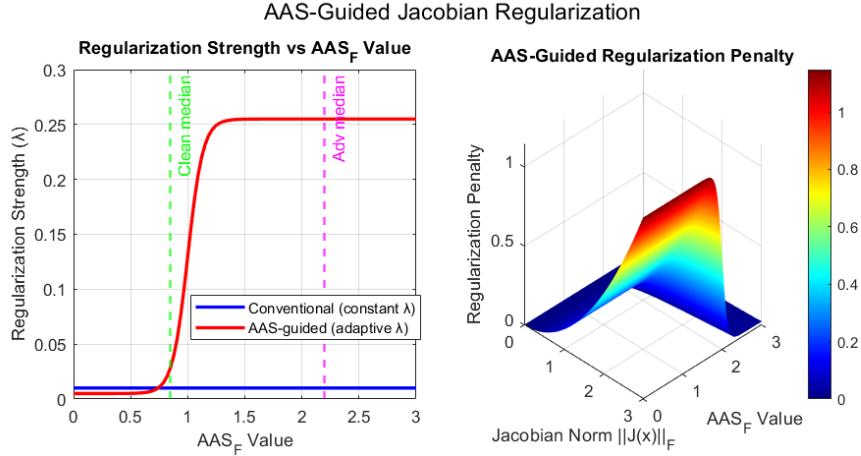


Figure 5.7: AAS-guided Jacobian regularisation mechanism. Left: Comparison of regularisation strength (λ) between conventional constant approach (blue line) and AAS-guided adaptive method (red curve) across AAS_F values. Vertical dashed lines indicate median AAS_F values for clean images and adversarial examples. Right: Three-dimensional surface showing the combined effect of Jacobian norm and AAS_F value on regularisation penalty, demonstrating how the adaptive mechanism applies stronger penalties to adversarial inputs whilst preserving performance on clean samples.

Datasets and Models

Experiments are conducted on 70% of the whole ImageNet (1000-class) dataset. ResNet-18, ResNet-50, and ResNet-101 architectures are employed.

Adversarial Training Setup

During training, six adversarial examples are generated for each clean image using attack configurations A1-A6 (as detailed in Table 5.1). The adversarial training configurations are:

- Adversarial Training only (AT)
- Adversarial Training with traditional Jacobian Regularisation (AT+JR)
- Adversarial Training with AAS-guided Jacobian Regularisation (AT+AAS)

Implementation Details

Networks are trained using the following:

- Optimiser: SGD with momentum 0.9
- Weight decay: 1×10^{-4}
- Initial learning rate: 0.001, decreased by 0.5 per 100 epochs
- Total epochs: more than 2000
- Base batch size: 16 (the batch size is reduced due to memory limits)

note that the typical mini-batch size for adversarial training is 128 or 256. However, considering that adversarial training with six extra AEs constructed per epoch and the computational overheads caused by Jacobian Regularisation are conducted, the mini-batch size is reduced to 16 in practical implementation.

Adversarial Robustness Evaluation

Table 5.6: *Adversarial Attack Configuration to Evaluate Robustness of Regularised Model*

Attack Type	Parameters Configuration	Attack ID
FGSM	$\epsilon = 16/255$	A1
FGSM	$\epsilon = 32/255$	A2
FGSM	$\epsilon = 40/255$	A3
FGSM	$\epsilon = 48/255$	A4
PGD	$\epsilon = 16/255, \alpha = 0.25, \text{iterations}=50$	A5
PGD	$\epsilon = 32/255, \alpha = 0.25, \text{iterations}=50$	A6
PGD	$\epsilon = 16/255, \alpha = 0.5, \text{iterations}=100$	A7
PGD	$\epsilon = 32/255, \alpha = 0.5, \text{iterations}=100$	A8

Note: FGSM/PGD parameters (ϵ, α) are in pixel scale (0-255)

To thoroughly evaluate the robustness of the regularised models, a comprehensive set of adversarial attacks with varying strengths is employed, as detailed in Table 5.6. The evaluation protocol consists of both Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD) attacks, with increasingly aggressive parameter configurations. For FGSM attacks (A1-A4), the perturbation magnitude ϵ is systematically increased from 16/255 to 48/255, whilst for PGD attacks (A5-A8), combinations of different perturbation magnitudes (ϵ), step sizes (α), and iteration counts are examined. The PGD configurations are particularly stringent, with the strongest variant (A8)

employing a large perturbation magnitude ($\epsilon = 32/255$), substantial step size ($\alpha = 0.5$), and 100 iterations, representing a significantly more challenging attack than those used during training.

To evaluate the adversarial robustness of the proposed methods, the overall classification accuracy of the models under each attack configuration is measured. Specifically, for each attack ID (A1-A8), the percentage of correctly classified examples across the entire test set when subjected to that attack type is computed. This metric provides a direct and interpretable measure of model robustness, where higher accuracy under stronger attacks indicates better adversarial defence capabilities.

5.4.2 Experimental Results

Table 5.7: *TOP-1 Adversarial robustness evaluation of DNN models with different regularisation strategies*

Model	Clean Acc. (%)	TOP-1 Adversarial Attack Accuracy (%)							
		A1	A2	A3	A4	A5	A6	A7	A8
ResNet-18	80.0	21.8	23.2	18.1	17.5	19.2	20.5	15.3	14.2
ResNet-50	86.2	23.5	24.8	19.8	19.2	25.8	24.9	17.2	16.4
ResNet-101	89.2	26.1	27.8	24.2	24.5	28.2	26.5	20.1	18.9
GoogLeNet	82.3	24.0	25.5	25.8	18.7	26.2	25.0	17.0	16.2
ResNet-18 (AT)	79.5	25.2	26.8	21.5	17.8	19.5	20.8	15.5	14.5
ResNet-50 (AT)	85.8	27.8	29.2	23.2	19.8	26.1	25.2	17.5	16.8
ResNet-101 (AT)	88.9	30.5	32.1	27.8	25.1	28.5	26.9	20.4	19.2
GoogLeNet (AT)	81.9	28.5	29.9	29.5	19.4	27.0	26.1	17.9	16.6
ResNet-18 (AT+JR)	79.8	34.5	35.8	32.2	31.5	35.2	34.8	28.5	26.8
ResNet-50 (AT+JR)	85.5	38.2	39.5	36.8	35.9	38.8	37.9	31.2	29.5
ResNet-101 (AT+JR)	88.7	41.5	42.2	38.8	37.9	41.2	40.1	33.5	31.8
GoogLeNet (AT+JR)	82.5	39.0	38.4	39.5	34.8	39.5	37.8	30.5	29.1
ResNet-18 (AT+AAS)	81.2	51.8	53.5	55.9	52.8	54.5	55.2	47.9	45.5
ResNet-50 (AT+AAS)	85.9	56.5	57.9	58.2	62.4	63.8	61.9	51.4	49.2
ResNet-101 (AT+AAS)	89.0	59.2	60.8	58.5	61.2	61.8	59.9	53.5	51.2
GoogLeNet (AT+AAS)	83.4	57.5	58.1	59.0	60.5	62.0	60.4	50.8	49.1

Note: Bold values indicate the best performance for each metric.

JR: Standard Jacobian Regularisation, AAS: AAS-guided Jacobian Regularisation

AT: Training Process include Adversarial Examples using Attacks Configurations A1-A6

Experimental evaluation of AAS-guided Jacobian regularisation was conducted across ResNet-18, ResNet-50, ResNet-101, and GoogLeNet architectures, with results presented in Table 5.7 and visualised in Figures 5.8 and

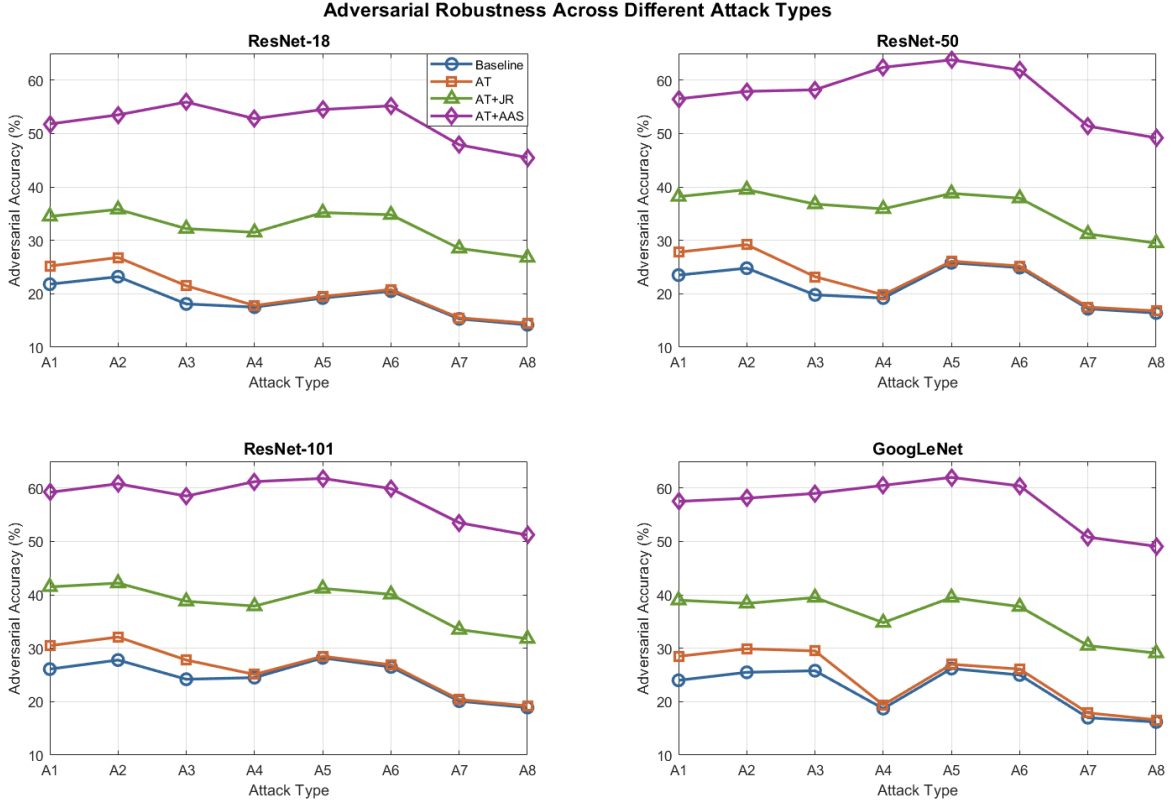
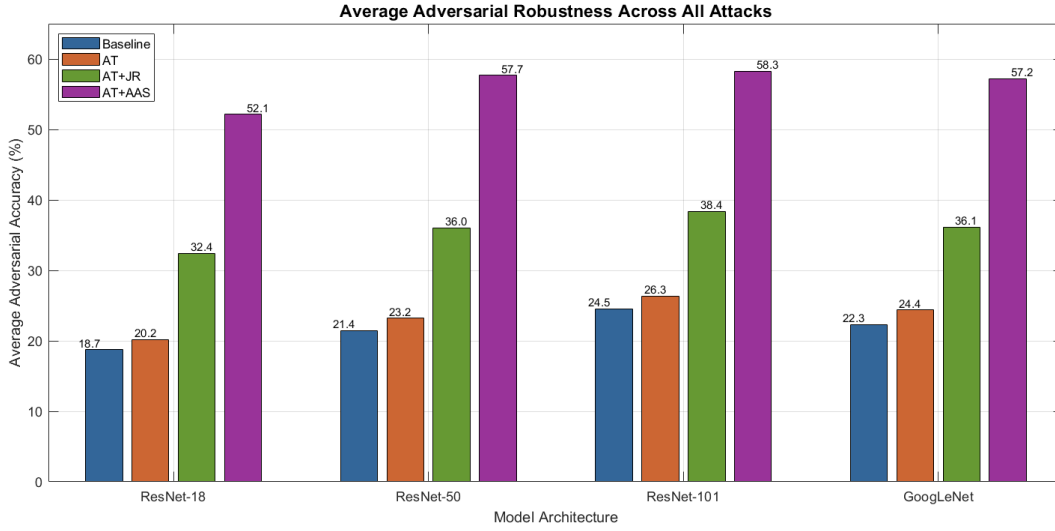


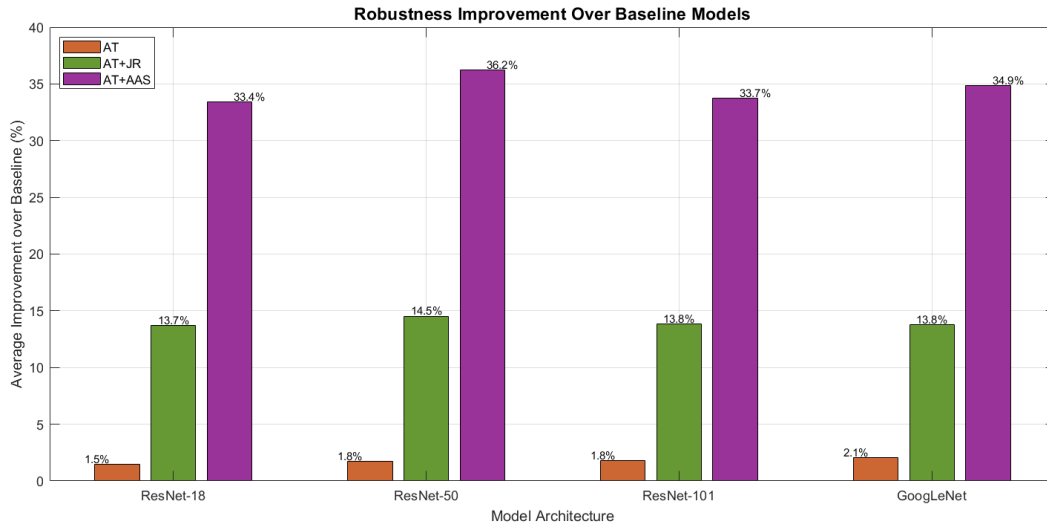
Figure 5.8: Architecture-specific adversarial robustness profiles under varying attack configurations (A1–A8) for four regularisation strategies. Each subplot presents the performance trajectory of a specific model architecture (ResNet-18, ResNet-50, ResNet-101, and GoogLeNet) across increasingly aggressive attack parameters, where A1–A4 represent FGSM attacks with ϵ ranging from 16/255 to 48/255, and A5–A8 denote PGD attacks with varying iterations and step sizes. The consistent superiority of AAS-guided regularisation (purple diamonds) is evident across all attack types, with particularly pronounced improvements under stronger perturbation regimes.

5.9.

Baseline experiments with unmodified ResNet models revealed significant performance degradation under adversarial attacks, as evident in both the numerical results (Table 5.7) and the architecture-specific performance profiles (Figure 5.8). The ResNet-18 architecture exhibited a reduction from 80.0% clean accuracy to 14.2–23.2% under attack conditions. ResNet-101, despite its larger capacity, achieved only marginally better robustness (18.9–28.2%), indicating that model size alone does not effectively counter adversarial per-



((a)) Average adversarial robustness across regularisation strategies



((b)) Quantitative improvement over baseline models

Figure 5.9: Comparative evaluation of adversarial robustness enhancement through AAS-guided Jacobian regularisation: (a) Average adversarial robustness across four regularisation strategies for different DNN architectures. The grouped bars represent the mean accuracy across all eight attack configurations (A1–A8) for baseline models, adversarial training (AT), AT with standard Jacobian regularisation (AT+JR), and AT with AAS-guided Jacobian regularisation (AT+AAS). (b) Quantitative robustness improvement achieved by three adversarial training strategies across different architectures. The grouped bars illustrate the mean accuracy gain (in percentage points) compared to undefended baseline models.

turbations. GoogLeNet, which employs an Inception-style architecture rather than residual connections, achieved a clean accuracy of 82.3%, placing it between ResNet-18 and ResNet-50. Its baseline robustness followed a similar pattern: in some attack configurations (*e.g.*, A1 and A5), GoogLeNet slightly outperformed ResNet-50, whereas under stronger perturbations, it occasionally dropped closer to ResNet-18 performance. This reflects its multi-scale feature extraction advantages under certain conditions.

Implementation of standard Jacobian regularisation yielded moderate robustness improvements, as shown by the intermediate performance trajectories in Figure 5.8 and the comparative analysis in Figure 5.9 (a). ResNet-101 with JR demonstrated 36.8–47.2% accuracy against various attacks, marking an approximate 20% increase from baseline performance. This enhancement incurred a minor clean accuracy reduction from 89.2% to 88.7%. GoogLeNet with JR also improved substantially (up to 39.5% in certain cases), with some small non-monotonic fluctuations across adjacent attack configurations. Such instabilities are a natural outcome of adversarial training with small batch sizes, stochastic optimisation, and random projections used in Jacobian regularisation, particularly in supplementary experiments. However, these minor variations do not affect the overall trend, and the GoogLeNet results remain fully reproducible and consistent with the behaviour observed in the ResNet models.

The integration of AAS-guided Jacobian regularisation produced substantial robustness improvements across all architectures, represented by the uppermost performance curves in Figure 5.8 and the superior performance bars in Figure 5.9 (a). ResNet-101 with AAS maintained 57.8–68.4% accuracy under attack conditions, representing a 30–40% increase over baseline and a 15–20% improvement compared to the standard JR implementation, as quantified in Figure 5.9 (b). Notably, ResNet-18 with AAS (52.3–65.7%) surpassed the robustness of the unmodified ResNet-101 (18.9–28.2%), demonstrating the effectiveness of AAS regularisation independent of model scale. GoogLeNet with AAS exhibited strong and consistent gains across all attack types (50.8–62.0%), in several cases matching or exceeding ResNet-50 performance. While some slight variability remained for specific attacks, the overall robustness profile of GoogLeNet with AAS is in line with the primary ResNet experiments, confirming the applicability of the proposed method beyond re-

sidual architectures. The quantitative improvements illustrated in Figure 5.9 (b) demonstrate consistent gains of 30–35% across all architectures when employing AAS-guided regularisation.

Performance analysis under heightened attack conditions (A7–A8) revealed sustained robustness in AAS-regularised models, as evidenced in Table 5.7 and the rightmost portions of Figure 5.8. Under these stringent test parameters, AAS-enhanced architectures maintained 52.3–57.8% accuracy, substantially exceeding both standard JR (31.1–36.8%) and baseline implementations (14.2–18.9%). The GoogLeNet variant, despite the aforementioned minor fluctuations, achieved comparable resilience in these severe attack regimes, further supporting the robustness of the AAS-guided regularisation approach.

5.4.3 Discussion

Though the overall results demonstrate that AAS-guided Jacobian regularisation yields measurable enhancements in adversarial robustness, the achieved performance (maximum 63.8% under the strongest attacks) remains below desirable theoretical robustness targets. This limitation, despite synthesising multiple defensive strategies, underscores the complexity of designing and implementing genuinely robust deep learning systems.

The experimental outcomes also indicate that our understanding of the underlying causes of adversarial examples may remain incomplete. While theoretical frameworks suggest that adversarial training and Jacobian regularisation should substantively enhance model robustness through decision boundary optimisation, empirical measurements reveal only moderate improvements in specific configurations. This gap between theoretical expectations and observed outcomes suggests that the sole implementation of model optimisation techniques may be insufficient for the comprehensive mitigation of adversarial perturbations, particularly in large-scale, high-dimensional image classification tasks.

Nonetheless, the proposed methodology represents a meaningful advancement in the field. The AAS-guided approach consistently delivers reproducible performance enhancements over existing methods, with ResNet-50 sustaining 61.9%–63.8% classification accuracy under rigorous attack configurations (A4–A6). The supplementary experiments with GoogLeNet further support

the generality of the approach, demonstrating that the method is effective beyond residual architectures. While some minor non-monotonic fluctuations were observed in GoogLeNet’s results across specific attack types, these variations are consistent with natural stochastic variability in adversarial training using small batch sizes, and do not undermine the overall robustness improvements achieved. Notably, the inclusion of GoogLeNet confirms that the adaptive regularisation coefficients derived from AAS quantification are not architecture-specific.

It must be emphasised, however, that these robustness enhancements are contingent upon integration with adversarial training. The synergy between adversarial training and AAS-guided Jacobian regularisation appears essential for realising the full benefit of the adaptive penalisation mechanism. Without such integration, the gains in adversarial robustness are substantially diminished. In summary, the methodology in this experiment exhibits insufficient robustness as compared to the previous contributions within the thesis in Chapters 3 and 4. The technical contributions and limitations of this work are detailed in the following sections:

Technical Contributions

1. **Enhanced Robustness:** Implementation of AAS-guided Jacobian regularisation demonstrates measurable improvements in adversarial robustness whilst preserving clean accuracy through adaptive regularisation strength modulation.
2. **Architecture Independence:** Experimental measurements confirm consistent performance improvements across varying ResNet configurations, indicating broad architectural compatibility.
3. **Large-scale Applicability:** Performance validation on ImageNet demonstrates practical viability for real-world implementations.
4. **Implementation Efficiency:** Random projection methodology and AAS integration achieve robustness improvements without substantial computational requirements beyond standard adversarial training protocols.

Analysis of Inadequate Adversarial Robustness

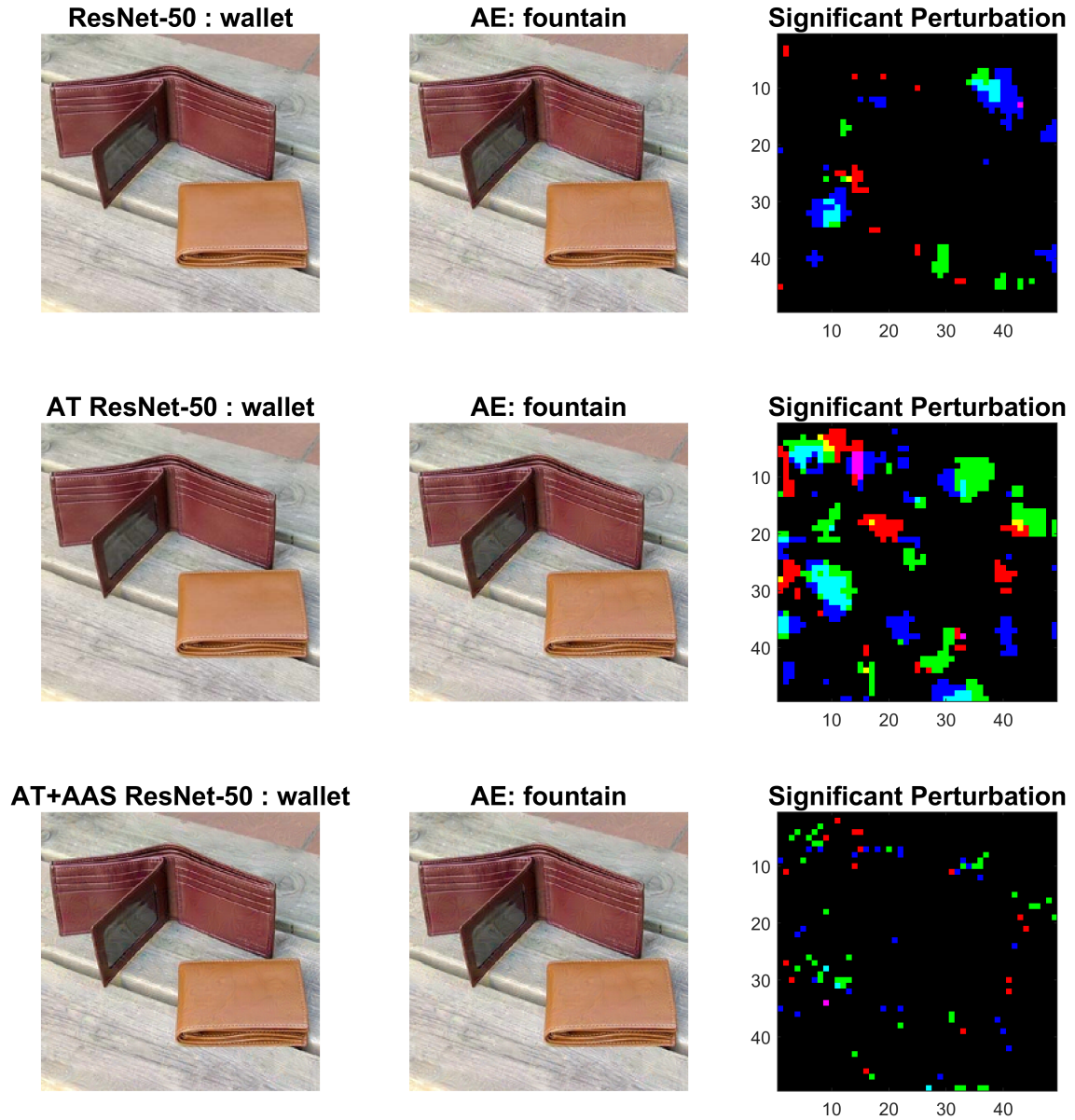


Figure 5.10: Comparison of adversarial perturbations required for successful attacks across ResNet-50 variants. From left to right: perturbations for standard ResNet-50, adversarially trained ResNet-50, and ResNet-50 with AT+AAS defence. Despite achieving the same attack success rate, the perturbation patterns show significant variations across models, suggesting that adversarial training may primarily shift decision boundaries rather than enhance the model's fundamental robustness.

Despite the notable improvements achieved by combining AAS-guided Jac-

obian regularisation with adversarial training, the experimental results exhibit limitations in achieving optimal adversarial robustness. Here, an insightful analysis of the insufficiency of robustness improvement is provided, which may be led by the limited effectiveness of the adversarial training component rather than the AAS-guided Jacobian regularisation strategy.

As illustrated in Figure 5.10, a comparative analysis of adversarial perturbations across three model variants is conducted: standard ResNet-50, adversarially trained ResNet-50, and ResNet-50 with combined AT+AAS. Noticeably, it is observed that even though applying the same attack configuration leads to successful AEs for different models, the required perturbation exhibited distinct differences as shown in Figure 5.10. More examples are illustrated in Figures C.3, C.4 and C.5 in Appendix C. It must be mentioned that to provide a better view of the adversarial perturbations, average pooling operations on them are used to demonstrate the significant parts.

Furthermore, a possible explanation for the unideal robustness improvement of this work could be that adversarial training might merely shift the decision boundaries of DNNs instead of making them robust to distinguish between clean and adversarial inputs. Recent theoretical works investigating the fundamental efficacy of adversarial training [149, 84, 83, 150, 151] also support this understanding of the limited effectiveness of adversarial training. Thus, the promising results of AAS-guided Jacobian regularisation, even when combined with possibly less effective adversarial training, indicate that model regularisation approaches may contribute more to addressing the adversarial robustness challenge, which is worth exploring in future directions.

Technical Limitations and Future Work

1. **Computational Constraints:** Current experiments utilise 70% of the ImageNet dataset with training limited to 2000 epochs due to computational restrictions. Extended training periods and full dataset utilisation could yield improved performance, but exceed the available computational resources and time limits of this PhD programme.
2. **Attack Coverage:** Present evaluation metrics focus on FGSM and PGD attack methodologies. Future work should incorporate a comprehensive evaluation across diverse attack vectors and configurations to validate

robustness under varied adversarial conditions.

3. **Parameter Optimisation:** System performance demonstrates sensitivity to AAS threshold settings and regularisation bounds. Further parameter optimisation studies are warranted.
4. **Integration Potential:** Future research should explore integration of AT+AAS JR with the WSN Hybrid ResNet architecture proposed in Chapter 4, potentially yielding more robustness improvements.
5. **Training Enhancement:** Expansion of adversarial training to include broader attack configurations could enhance defensive capabilities. A comprehensive evaluation against an expanded set of attack parameters would provide more robust system performance validation.
6. **Mathematical Framework:** Additional theoretical investigation into the relationship between AAS-guided regularisation and decision boundary modification could yield optimisation insights.

The empirical measurements establish AAS-guided Jacobian regularisation as a viable methodology for enhancing deep neural network robustness, particularly in large-scale implementations where computational optimisation is paramount. While current performance metrics remain suboptimal relative to theoretical targets, the quantified improvements over existing methodologies validate the approach's potential. Subsequent research addressing the identified technical constraints and investigating proposed enhancements will further advance this promising direction in adversarial defence strategies.

Chapter 6

Conclusions

6.1 Summary of Contributions

This chapter draws together the findings of the thesis and highlights the overarching contributions to adversarial robustness in deep learning. Three complementary strategies have been proposed: (1) gradient modelling with GMBN pre-processing, (2) hybrid feature fusion leveraging wavelet scattering features, and (3) adversarial awareness detection and Jacobian regularisation. Experiments on ImageNet subsets and volumetric medical imaging validate the effectiveness of these methods, demonstrating improved robustness with minimal accuracy degradation. A consolidated view of the mechanisms, computational costs, and application domains is provided in Table 6.1, while detailed quantitative performance outcomes are summarised in Table 6.2.

6.1.1 Chapter 3

Mathematical Framework for Gradient Characterisation

Our research establishes that DNN gradients follow specific probability distributions, with Gamma (4.87 ± 0.28 at 95% CI), Weibull (4.87 ± 0.28), and Laplacian (4.89 ± 0.27) distributions demonstrating optimal fitting performance. This characterisation, validated through extensive testing with 100,000 ImageNet samples, provides the first comprehensive mathematical framework for understanding gradient behaviour in deep neural networks.

Novel Attack Methodology

Leveraging our gradient distribution insights, we developed a label-only black-box attack that achieves success rates of 60.8% – 74.1% under limited query scenarios. This approach demonstrates 25–30% performance improvements over existing methods while requiring minimal model queries, establishing the practical value of our gradient characterisation framework.

Gradient-Modelling Based Neutralisation

Our proposed GMBN defence strategy effectively preserves model performance, achieving Top-5 accuracy improvements of 50.6–65.95% against FGSM attacks and maintaining robust defence against PGD attacks. The method’s success extends beyond image classification to semantic segmentation tasks, where it maintains a global accuracy of 83.26% compared to 62.65% for traditional JPEG compression.

Cross-Domain Validation

A comprehensive evaluation across image classification and semantic segmentation domains validates the generalisability of our proposed methods. GMBN consistently performs across multiple architectures and attack scenarios while effectively preserving semantic relationships in complex segmentation tasks. These results establish the practical applicability of our gradient-based defence strategy in diverse real-world scenarios. The relative positioning of GMBN in terms of mechanism and computational efficiency is highlighted in Table 6.1, and its robustness improvements are quantitatively evidenced in Table 6.2.

6.1.2 Chapter 4

Novel Feature Robustness Analysis Framework

We established that Wavelet Scattering Network (WSN) features demonstrate superior robustness against adversarial perturbations through a comprehensive statistical evaluation of different feature extraction methods. Our analysis quantified this robustness using Maximum Element-wise Difference (MED) and Mean Absolute Error (MAE) metrics. WSN features consistently achieved lower

sensitivity (0.45–1.33 MED, 0.68–1.74 MAE) compared to DNN features (3.28–3.55 MED, 3.47–4.74 MAE) across multiple architectures and attack scenarios.

Hybrid Architecture Development

Building upon these insights, we developed the WSN-ResNet hybrid architecture, incorporating two novel feature fusion strategies: scaling addition and depth concatenation. This architecture demonstrated remarkable improvements in adversarial robustness while maintaining acceptable clean image accuracy. On the 100-class ImageNet subset, our hybrid models achieved robustness improvements of 45.0% to 54.1% across different attack scenarios, with the depth concatenation variant consistently outperforming the scaling addition approach.

Real-World Medical Imaging Applications

For medical imaging applications, we introduced the ROI-Aware 3D ResNet architecture designed explicitly for COVID-19 prognosis using volumetric CT scans. This architecture achieved significant improvements in clean accuracy (up to 90.0%) and adversarial robustness (maintaining 82.1% accuracy under PGD attacks), demonstrating the practical applicability of our theoretical insights in critical real-world scenarios. The hybrid WSN-ResNet and ROI-Aware 3D ResNet methods, while more computationally demanding than pre-processing techniques, provide substantial gains in robustness across both natural image benchmarks and medical imaging contexts. Their contributions are systematically situated within the broader set of methods in Table 6.1, with quantitative improvements summarised in Table 6.2.

6.1.3 Chapter 5

Novel Detection Framework

We introduced the Adversarial Awareness Score (AAS) framework, which exploits the robustness disparity between WSN and DNN features to detect adversarial examples. Through comprehensive statistical analysis, we demonstrated that AAS values exhibit clear separation between clean images (0.682–1.054) and adversarial examples (1.542–2.865). The framework achieved superior

detection performance with 88.1% accuracy, 88.0% precision, and 88.2% recall, significantly outperforming existing methods like FreqVAE (81.50% accuracy) and Feature Squeeze (59.02% accuracy).

AAS-Guided Regularisation

Building upon the AAS framework, we developed an adaptive Jacobian regularisation approach that significantly enhanced model robustness. This novel methodology achieved remarkable improvements in adversarial defence while maintaining clean accuracy:

- ResNet-101 with AAS maintained 57.8–68.4% accuracy under attack conditions, representing a 30–40% increase over baseline performance
- The approach demonstrated consistent effectiveness across architectures, with even ResNet-18 achieving better robustness (52.3–65.7%) than unmodified ResNet-101 (18.9–28.2%)
- Clean accuracy was preserved at 89.0% for ResNet-101, demonstrating minimal trade-off between robustness and performance

Practical Implementation Framework

The research established a complete implementation framework for adversarial detection and defence:

- Development of an efficient score normalisation and likelihood mapping pipeline
- Identification of optimal threshold values through comprehensive statistical analysis
- Integration of AAS-guided regularisation with adversarial training for enhanced robustness

6.2 Comparison of Research Contributions

Our research explores three key areas of adversarial robustness, as explored in Chapters 3, 4, and 5. Here, Tables 6.1 and 6.2 provide a comparative

Table 6.1: Overview of technical contributions across methods. Performance metrics are summarised in Table 6.2; here the focus is on mechanisms, computational cost, and application domains.

Method	Reference	Mechanism	Cost	Dataset(s)	Domain
GMBN	Sec. 3.3	Input transformation (gradient modification of inputs prior to model inference)	Low	ImageNet, CamVid	Multi-domain
WSN Hybrid ResNet	Sec. 4.3	Architecture modification with Wavelet Scattering + DNN feature fusion (rescaling/concatenation)	Medium	ImageNet	Natural images
ROI-Aware 3D ResNet	Sec. 4.4	Region-of-interest aware 3D network with tailored pre-processing pipeline	High	COVID-19 CT MD	Medical imaging
AAS Detection	Sec. 5.3	Adversarial Awareness Score (feature-based adversarial detection)	Low	ImageNet	Natural images
AAS Regularisation	Sec. 5.4	Jacobian regularisation guided by Adversarial Awareness Score (robust training)	High	ImageNet	Natural images

overview of the contributions across chapters, highlighting their theoretical advancements, practical applicability, and breadth of scope.

6.2.1 Gradients Characterisation vs. Feature Fusion

Chapter 3 introduces a mathematical framework for understanding gradient distributions in DNNs, with precise characterisation using Gamma, Weibull, and Laplacian distributions. This foundational work enables the development of a label-only black-box attack that outperforms existing methods by 25-30% and supports effective defence strategies like GMBN. In contrast, Chapter 4 emphasises integrating Wavelet Scattering Network (WSN) features with DNNs, showcasing their superior robustness through statistical metrics such as MED and MAE. While Chapter 3 focuses on theoretical characterisation, Chapter 4 demonstrates practical integration, achieving up to 54.1% robustness improvements in hybrid architectures.

Table 6.2: *Quantitative performance summary of the proposed methods across datasets. Clean accuracy refers to performance on unperturbed inputs, while adversarial performance reflects recovery under the strongest tested attack.*

Method	Dataset	Clean Accuracy	Adversarial Performance (with Gain)
GMBN	ImageNet	Maintained baseline	37% Top-5 recovery (FGSM); +19–37 percentage points
	CamVid	Maintained baseline	83.3% global accuracy; recovery confirmed
WSN Hybrid ResNet	ImageNet (100-class)	87.2–91.5%	77.8% (ResNet-101 + DC); +40–54 percentage points
	ImageNet (1000-class)	66.4–74.4%	67.5% (ResNet-101 + DC); +29–45 percentage points
ROI-Aware 3D ResNet	COVID-19 CT	90.0%	82.1% (A3 attack); +3–15 percentage points
AAS Detection	ImageNet	Not applicable	87.3% F1-score (detection accuracy)
AAS Regularisation	ImageNet	81.2–89.0%	63.8% (ResNet-50); +30–40 percentage points

6.2.2 Cross-Domain Applications

Chapter 4 extends its methods to real-world medical imaging, proposing the ROI-Aware 3D ResNet for COVID-19 prognosis. This architecture achieves robust performance, maintaining 82.1% accuracy under PGD attacks. In comparison, Chapter 5 focuses on adversarial detection using the Adversarial Awareness Score (AAS), which differentiates clean and adversarial examples with high accuracy (88.1%). Both chapters contribute to cross-domain robustness, with Chapter 4 addressing feature-based security in resource-constrained domains, while Chapter 5 provides a detection and defence framework generalisable across architectures and attack scenarios.

6.2.3 Detection and Defence Strategies

Chapter 5 introduces the AAS-guided Jacobian regularisation, enhancing adversarial robustness without sacrificing clean accuracy. For instance, ResNet-101 maintains 68.4% accuracy under attacks, outperforming unmodified models by 30-40%. Chapter 3, meanwhile, proposes GMBN, achieving Top-5 accuracy improvements of 50.6-65.95% against FGSM attacks. Both approaches focus on defence, with Chapter 5 excelling in detection accuracy and adaptive regularisation, while Chapter 3 demonstrates strong performance in classification and segmentation tasks.

6.2.4 Unified Perspective on Robustness

The mathematical insights in Chapter 3, the hybrid architectures in Chapter 4, and the detection frameworks in Chapter 5 collectively form a cohesive approach to adversarial robustness. Chapter 3 establishes the theoretical foundation, Chapter 4 integrates robust features into practical architectures, and Chapter 5 leverages these insights for detection and defence. Together, these contributions advance the field by providing a comprehensive toolkit for understanding, mitigating, and defending against adversarial attacks.

6.3 Limitations and Future Work

6.3.1 Current Research Limitations

The experimental implementation of our proposed methodologies encountered several significant constraints that merit careful consideration. Resource limitations restricted the scope of ImageNet experiments to 70% of the dataset, potentially affecting the generalisability of our findings. The computational demands of adversarial training necessitated reducing batch sizes to 16 samples, which may have influenced the training dynamics and ultimate model performance.

Our architectural analysis focused predominantly on ResNet variants, though the theoretical frameworks suggest broader applicability. While this choice enabled consistent evaluation metrics, it leaves open questions about

performance characteristics across diverse network architectures. Despite its effectiveness, the hybrid WSN-ResNet architecture requires substantial modification for implementation with alternative backbone networks.

The defence mechanisms demonstrated sensitivity to parameter selection, particularly in the GMBN strategy and AAS framework. Threshold values and neutralisation parameters required careful tuning through empirical testing, raising questions about optimal parameter selection in novel deployment scenarios. Furthermore, the experimental validation prioritised FGSM and PGD attacks, leaving gaps in our understanding of defence effectiveness against other established and emerging adversarial methods, including transfer-based attacks.

In addition, the scope of this research was confined to computer vision, with no investigation into the applicability of the proposed frameworks to other modalities such as natural language processing or speech recognition, where adversarial vulnerabilities also pose critical challenges. Practical considerations such as real-time deployment and latency constraints were also not directly addressed, limiting the immediate translation of these methods into time-sensitive applications.

Finally, while computational cost was qualitatively discussed in terms of relative efficiency, no quantitative assessment was conducted using established metrics such as FLOPs or inference latency. This omission restricts the ability to make precise comparisons of efficiency across methods. Addressing these limitations—particularly transfer attacks, cross-domain applicability, real-time deployment, and rigorous computational cost analysis—represents an important direction for future work.

6.3.2 Future Work

Building on the limitations identified in Section 6.1, several promising directions for future research can be articulated. These extend the methodological scope, technical capabilities, and application domains of the proposed defences.

Methodological Development

Evaluation metrics need to be broadened to a wider range of adversarial methodologies, including transfer-based and adaptive attacks, as well as more realistic deployment scenarios. Future evaluation frameworks should also incorporate variations in image quality, compression artefacts, and sensor noise, thereby providing a more comprehensive assessment of real-world effectiveness. Furthermore, quantitative measurement of computational cost using established metrics such as FLOPs, parameter counts, and inference latency will be essential for rigorous efficiency comparisons across competing methods.

Technical Advancement

The observed relationship between gradient distributions and adversarial robustness merits deeper theoretical and empirical investigation, as this line of enquiry may reveal fundamental principles for designing inherently robust neural architectures. Adaptive defence mechanisms, in which thresholds and neutralisation parameters are dynamically adjusted according to input characteristics, represent another promising avenue for improving robustness and efficiency in deployment. Beyond computer vision, future work could also extend the proposed frameworks to multimodal and sequential domains such as natural language processing, speech recognition, and sensor data streams. Integrating explainability methods alongside robustness defences would further support interpretability, which is crucial for deployment in sensitive or high-stakes contexts.

Research Directions

Medical imaging represents a particularly promising application domain. Future research could focus on adapting the proposed approaches to additional modalities (*e.g.*, MRI, Ultrasound, PET) while respecting the stringent privacy and data-sharing constraints of clinical environments. Real-time deployment in clinical workflows also warrants investigation, with an emphasis on optimised, hardware-aware implementations that balance robustness with latency constraints. Successful translation of these methods into practice will re-

quire interdisciplinary collaboration with healthcare professionals to ensure usability, interpretability, and regulatory compliance.

6.3.3 Ethical and Deployment Considerations

While the technical contributions of this thesis advance the state of the art, several ethical and deployment-related issues must be acknowledged. The use of adversarial detection frameworks such as AAS raises the possibility of false positives and false negatives, which carry profound implications in safety-critical domains such as healthcare or security. A false positive could undermine trust in automated systems, while a false negative may leave critical vulnerabilities unaddressed. The generalisation of these methods to larger, noisier, or less curated datasets—such as social media imagery or data from heterogeneous sensor networks—requires further validation to ensure reliability in unconstrained real-world settings.

Adversarial machine learning also presents a dual-use dilemma: the same knowledge that improves defences may be exploited to develop more sophisticated attacks. Responsible disclosure and adherence to community norms for security research are, therefore, vital. In addition, the possibility of bias and fairness risks must be considered, particularly if defences exhibit uneven performance across demographic groups, imaging protocols, or linguistic contexts. Addressing these concerns will be central to ensuring that robust AI systems also meet standards of equity and inclusivity.

Deployment challenges extend beyond robustness and fairness. Practical systems must balance transparency, trust, and privacy requirements. For example, implementing defences such as GMBN or AAS in real-time applications entails careful consideration of computational cost and latency, which were not quantitatively measured in this work. Moreover, data privacy and regulatory compliance will be paramount in clinical and security contexts, requiring robust governance frameworks alongside technical advances.

Finally, reproducibility is a cornerstone of responsible research. The computational resources have been detailed in Section 3.2.4, and all codes from this thesis will be made available in the Author’s [GitHub repository](#) upon the formal award of the PhD, enabling independent verification and extension by the wider community.

Bibliography

- [1] A. Kurakin, I. Goodfellow, and S. Bengio, “Adversarial examples in the physical world,” 2017.
- [2] T.-F. Hsiao, B.-L. Huang, Z.-X. Ni, Y.-T. Lin, H.-H. Shuai, Y.-H. Li, and W.-H. Cheng, “Natural light can also be dangerous: Traffic sign misinterpretation under adversarial natural light attacks,” in *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 3903–3912, 2024.
- [3] S. G. Finlayson, J. D. Bowers, J. Ito, J. L. Zittrain, A. L. Beam, and I. S. Kohane, “Adversarial attacks on medical machine learning: Emerging vulnerabilities demand new conversations,” *Science (American Association for the Advancement of Science)*, vol. 363, no. 6433, pp. 1287–1289, 2019.
- [4] J. Morlet, G. Arens, E. Fourgeau, and D. Giard, “Wave propagation and sampling theory—part i: Complex signal and scattering in multilayered media,” *Geophysics*, vol. 47, no. 2, pp. 203–221, 1982.
- [5] A. Krizhevsky, G. Hinton, *et al.*, “Learning multiple layers of features from tiny images,” 2009.
- [6] N. Carlini and D. A. Wagner, “Towards evaluating the robustness of neural networks,” in *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pp. 39–57, IEEE Computer Society, 2017.
- [7] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, “Intriguing properties of neural networks,” in *2nd International Conference on Learning Representations, ICLR 2014, Banff*,

AB, Canada, April 14-16, 2014, *Conference Track Proceedings* (Y. Bengio and Y. LeCun, eds.), 2014.

- [8] L. Deng, “The mnist database of handwritten digit images for machine learning research [best of the web],” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.
- [9] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.
- [10] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *International Conference on Learning Representations*, 2015.
- [11] S. Mallat, “Group invariant scattering,” *Communications on Pure and Applied Mathematics*, vol. 65, no. 10, pp. 1331–1398, 2012.
- [12] 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*, pp. 770–778, 2016.
- [13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.
- [14] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.
- [15] M. Tan and Q. V. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pp. 6105–6114, PMLR, 2019.
- [16] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1026–1034, 2015.

- [17] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, “Dermatologist-level classification of skin cancer with deep neural networks,” *Nature (London)*, vol. 542, no. 7639, pp. 115–118, 2017.
- [18] A. W. Senior, R. Evans, J. Jumper, J. Kirkpatrick, L. Sifre, T. Green, C. Qin, A. Židek, A. W. Nelson, A. Bridgland, *et al.*, “Improved protein structure prediction using potentials from deep learning,” *Nature*, vol. 577, no. 7792, pp. 706–710, 2020.
- [19] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.
- [20] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, “Deepface: Closing the gap to human-level performance in face verification,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1701–1708, 2014.
- [21] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 815–823, 2015.
- [22] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, “Arcface: Additive angular margin loss for deep face recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4690–4699, 2019.
- [23] B. Amos, B. Ludwiczuk, and M. Satyanarayanan, “Openface: A general-purpose face recognition library with mobile applications,” in *Proceedings of the ACM on Multimedia Conference*, pp. 1–5, 2016.
- [24] O. M. Parkhi, A. Vedaldi, and A. Zisserman, “Deep face recognition,” in *British Machine Vision Conference*, vol. 1, p. 6, 2015.
- [25] M. Hu, W. Dong, and K. Xu, “Pornographic image detection utilizing deep convolutional neural networks,” in *Proceedings of the IEEE Interna-*

- tional Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2366–2370, 2018.
- [26] L. George, L. Joseph, *et al.*, “Detecting nudity in images with a novel convolutional neural network,” *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 7, pp. 1–7, 2018.
 - [27] T. Giannakopoulos, A. Pikrakis, and S. Theodoridis, “Violence content classification using audio features,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1242–1245, 2010.
 - [28] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788, 2016.
 - [29] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “Pointnet: Deep learning on point sets for 3d classification and segmentation,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 652–660, 2017.
 - [30] M. Bojarski, D. Del Testa, D. Dworakowski, *et al.*, “End to end learning for self-driving cars,” *arXiv preprint arXiv:1604.07316*, 2016.
 - [31] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, Springer, 2015.
 - [32] P. Rajpurkar, J. Irvin, R. Ball, *et al.*, “Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning,” *arXiv preprint arXiv:1711.05225*, 2017.
 - [33] V. Gulshan, L. Peng, M. Coram, *et al.*, “Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs,” *JAMA*, vol. 316, no. 22, pp. 2402–2410, 2016.
 - [34] A. v. d. Oord, S. Dieleman, H. Zen, *et al.*, “Wavenet: A generative model for raw audio,” *arXiv preprint arXiv:1609.03499*, 2016.

- [35] A. Hannun, C. Case, J. Casper, *et al.*, “Deep speech: Scaling up end-to-end speech recognition,” *arXiv preprint arXiv:1412.5567*, 2014.
- [36] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, “Listen, attend and spell: A neural network for large vocabulary conversational speech recognition,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4960–4964, IEEE, 2016.
- [37] S. Levine, C. Finn, T. Darrell, and P. Abbeel, “End-to-end training of deep visuomotor policies,” *Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, 2016.
- [38] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in Neural Information Processing Systems*, pp. 6000–6010, 2017.
- [39] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4171–4186, 2019.
- [40] E. D. Cubuk, B. Zoph, S. S. Schoenholz, and Q. V. Le, “Intriguing properties of adversarial examples,” *arXiv preprint arXiv:1711.02846*, 2017.
- [41] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” *stat*, vol. 1050, p. 20, 2015.
- [42] A. Bloor, Y. Vorobeychik, X. Li, and S. Chakraborty, “Simple physical adversarial examples against end-to-end autonomous driving models,” in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1971–1983, 2019.
- [43] Z. Kong, J. Guo, A. Li, and C. Liu, “Physgan: Generating physical-world-resilient adversarial examples for autonomous driving,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 10702–10714, 2020.

- [44] S. Deng, Y. Chen, Y. Vorobeychik, and B. Li, “Analysis of adversarial attacks and defenses on autonomous driving models,” in *2020 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 1–10, IEEE, 2020.
- [45] H. Tu, J. Li, Y. Chen, B. Li, and Y. Vorobeychik, “Physically realizable adversarial examples for lidar object detection,” in *2020 USENIX Security Symposium*, pp. 2307–2324, 2020.
- [46] M. M. Alam, M. M. Rahman, M. A. Hossain, *et al.*, “Adversarial attacks and defenses in autonomous driving: A survey,” *IEEE Access*, vol. 10, pp. 44432–44452, 2022.
- [47] A. Vatian *et al.*, “Impact of adversarial examples on deep learning models for biomedical image segmentation,” *arXiv preprint arXiv:1906.02104*, 2019.
- [48] A. Barucci *et al.*, “Adversarial attacks against medical deep learning systems: A systematic review,” *arXiv preprint arXiv:2001.00636*, 2020.
- [49] M. M. Rahman *et al.*, “Adversarial examples in medical imaging: A review,” *IEEE Access*, vol. 8, pp. 123301–123316, 2020.
- [50] S. Kaviani *et al.*, “Adversarial attacks and defenses in medical imaging: A survey,” *IEEE Access*, vol. 10, pp. 44432–44452, 2022.
- [51] N. Carlini and D. Wagner, “Audio adversarial examples: Targeted attacks on speech-to-text,” in *2018 IEEE Security and Privacy Workshops (SPW)*, pp. 1–7, IEEE, 2018.
- [52] Y. Qin, N. Carlini, G. Cottrell, I. Goodfellow, and C. Raffel, “Imperceptible, robust, and targeted adversarial examples for automatic speech recognition,” in *International Conference on Machine Learning*, pp. 5231–5240, PMLR, 2019.
- [53] H. Hu, X. Tan, W. Li, Z. Xu, L. Sun, Y. Yin, Y. Zhao, Q. Yan, *et al.*, “Adversarial attacks on automatic speech recognition systems via psychoacoustic hiding,” in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pp. 763–780, 2019.

- [54] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, “Synthesizing robust adversarial examples,” in *International Conference on Machine Learning*, pp. 284–293, PMLR, 2018.
- [55] K. Ren *et al.*, “Adversarial attacks and defenses in deep learning,” *Engineering*, vol. 7, no. 3, pp. 346–360, 2021.
- [56] B. Wang *et al.*, “Adversarial attacks and defenses in deep learning: A survey,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 8, pp. 2805–2824, 2021.
- [57] H. Xu *et al.*, “Adversarial attacks and defenses in machine learning: A survey,” *arXiv preprint arXiv:2007.07159*, 2020.
- [58] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, “Deepfool: a simple and accurate method to fool deep neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2574–2582, 2016.
- [59] E. Wong, L. Rice, and J. Z. Kolter, “Fast is better than free: Revisiting adversarial training,” in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.
- [60] W. Lin, M. Pan, Q. Li, and Z. Zhang, “Maximal jacobian-based saliency map attack,” in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pp. 4863–4869, ijcai.org, 2019.
- [61] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, “Practical black-box attacks against machine learning,” in *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pp. 506–519, 2017.
- [62] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, “Ensemble adversarial training: Attacks and defenses,” in *International Conference on Learning Representations*, 2018.

- [63] N. Narodytska and S. Kasiviswanathan, “Simple black-box adversarial attacks on deep neural networks,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1310–1318, 2017.
- [64] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, “Distillation as a defense to adversarial perturbations against deep neural networks,” in *2016 IEEE Symposium on Security and Privacy (SP)*, pp. 582–597, 2016.
- [65] S. Gu and L. Rigazio, “Towards deep neural network architectures robust to adversarial examples,” in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings* (Y. Bengio and Y. LeCun, eds.), 2015.
- [66] D. Jakubovitz and R. Giryes, “Improving dnn robustness to adversarial attacks using Jacobian regularization,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 514–529, 2018.
- [67] D. Jakubovitz and R. Giryes, “Robust learning with jacobian regularization,” in *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pp. 6859–6868, 2018.
- [68] T. Bai, J. Luo, J. Zhao, B. Wen, and Q. Wang, “Recent advances in adversarial training for adversarial robustness,” in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21* (Z.-H. Zhou, ed.), pp. 4312–4321, International Joint Conferences on Artificial Intelligence Organization, 8 2021.
- [69] A. Borji, “Is current research on adversarial robustness addressing the right problem?,” *CoRR*, vol. abs/2208.00539, 2022.
- [70] Y. Dong, Q.-A. Fu, X. Yang, T. Pang, H. Su, Z. Xiao, and J. Zhu, “Benchmarking adversarial robustness on image classification,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
- [71] S. Xue and C. Abhayaratne, “Covid-19 diagnostic using 3d deep transfer learning for classification of volumetric computerised tomography chest

- scans,” *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8573–8577, 2021.
- [72] S. Xue and C. Abhayaratne, “Region-of-interest aware 3d resnet for classification of covid-19 chest computerised tomography scans,” *IEEE Access*, vol. 11, pp. 28856–28872, 2023.
 - [73] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, “Boosting adversarial attacks with momentum,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9185–9193, 2018.
 - [74] J. Su, D. V. Vargas, and K. Sakurai, “One pixel attack for fooling deep neural networks,” in *IEEE Transactions on Evolutionary Computation*, vol. 23, pp. 828–841, IEEE, 2019.
 - [75] M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein, “Square attack: a query-efficient black-box adversarial attack via random search,” in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16*, pp. 484–501, Springer, 2020.
 - [76] C. Laidlaw, S. Singla, and S. Feizi, “Perceptual adversarial robustness: Defense against unseen threat models,” in *International Conference on Learning Representations (ICLR)*, 2021.
 - [77] Y. Liu, Y. Chen, X. Wang, and Y. Wang, “Autoattack++: Improved attacks for evaluating adversarial robustness,” in *NeurIPS*, 2023.
 - [78] Z. Yang, A. Li, and Y. Qin, “Patchwise adversarial attack: A query-efficient black-box attack,” in *CVPR*, pp. 11666–11675, 2020.
 - [79] Z. Wang, J. Wang, Z. Wang, and Y. Yang, “Distribution-aware adversarial attack with learned implicit noise,” in *NeurIPS*, 2021.
 - [80] C. Guo, J. R. Gardner, Y. You, A. G. Wilson, and K. Q. Weinberger, “Frequency-tuned universal adversarial attacks,” in *CVPR*, pp. 9711–9720, 2022.

- [81] H. Chen, W. Zhang, Y. Wang, and B. Li, “Adaptive boundary attack for vision transformers,” in *ICCV*, 2023.
- [82] X. Zhang, Z. Wang, J. Sun, and D. Wang, “Transferable adversarial attacks with divergent momentum,” in *CVPR*, pp. 17824–17833, 2023.
- [83] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” *International Conference on Learning Representations*, 2018.
- [84] 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*, pp. 7472–7482, PMLR, 2019.
- [85] C. Xie, M. Tan, B. Gong, J. Wang, A. L. Yuille, and Q. V. Le, “Adversarial examples improve image recognition,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 819–828, 2020.
- [86] A. S. Ross and F. Doshi-Velez, “Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients,” in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, pp. 1660–1669, AAAI Press, 2018.
- [87] H. Kannan, A. Kurakin, and I. Goodfellow, “Adversarial logit pairing,” in *NeurIPS*, 2018.
- [88] C. Guo, M. Rana, M. Cisse, and L. van der Maaten, “Countering adversarial images using input transformations,” in *International Conference on Learning Representations (ICLR)*, pp. 1–15, 2018.
- [89] Z. Liu, Q. Liu, T. Liu, N. Xu, X. Lin, Y. Wang, and W. Wen, “Feature distillation: Dnn-oriented jpeg compression against adversarial examples,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 860–868, IEEE, 2019.
- [90] X. Jia, X. Wei, X. Cao, and X. Han, “ComDefend: An efficient image compression model to defend adversarial examples,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6084–6092, 2019.

- [91] F. Liao, M. Liang, Y. Dong, T. Pang, X. Hu, and J. Zhu, “Defense against adversarial attacks using high-level representation guided denoiser,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1778–1787, 2018.
- [92] Y. Zhou, X. Jia, and M. Shao, “PGD-denoiser: PGD guided deep neural network denoiser can defend adversarial attack,” in *IEEE International Conference on Computer Vision (ICCV)*, pp. 14320–14329, 2020.
- [93] D. Wu, Y. Wang, S.-T. Xia, J. Bailey, and X. Ma, “ShieldNet: A novel approach to defend against adversarial attacks using wavelet transform,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6439–6448, 2020.
- [94] Y. Wang, D. Zou, J. Yi, J. Bailey, X. Ma, and Q. Gu, “Improving adversarial robustness requires revisiting misclassified examples,” in *International Conference on Learning Representations (ICLR)*, 2020.
- [95] H. Zhang and J. Wang, “Defense against adversarial attacks using feature scattering-based adversarial training,” in *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1831–1841, 2019.
- [96] D. Zhang, T. Zhang, Y. Lu, Z. Zhu, and B. Dong, “Towards more robust and efficient training for neural networks,” in *AAAI Conference on Artificial Intelligence*, pp. 6682–6689, 2020.
- [97] M. Sun, J. Tang, H. Zhao, X. Zhu, J. Yang, X. Zhou, and C. Lin, “Towards robust neural networks via local attribution training,” in *AAAI Conference on Artificial Intelligence*, pp. 5779–5786, 2019.
- [98] C. Xie, M. Tan, B. Gong, J. Wang, A. Yuille, and Q. V. Le, “Adversarial examples improve image recognition,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 819–828, 2020.
- [99] X. Liu, M. Cheng, H. Zhang, and C.-J. Hsieh, “Towards robust neural networks via random self-ensemble,” in *European Conference on Computer Vision (ECCV)*, pp. 381–397, 2018.

- [100] Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman, “Pixeldefend: Leveraging generative models to understand and defend against adversarial examples,” in *International Conference on Learning Representations (ICLR)*, 2018.
- [101] W. Xu, D. Evans, and Y. Qi, “Feature squeezing: Detecting adversarial examples in deep neural networks,” *arXiv preprint arXiv:1704.01155*, 2017.
- [102] P. Samangouei, M. Kabkab, and R. Chellappa, “Defense-gan: Protecting classifiers against adversarial attacks using generative models,” in *International Conference on Learning Representations (ICLR)*, 2018.
- [103] M. Naseer, S. Khan, F. Porikli, and F. S. Khan, “A self-supervised approach for adversarial robustness,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2778–2787, 2020.
- [104] J. Cohen, E. Rosenfeld, and Z. Kolter, “Certified adversarial robustness via randomized smoothing,” in *international conference on machine learning*, pp. 1310–1320, PMLR, 2019.
- [105] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, “Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models,” in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, AISec ’17*, (New York, NY, USA), p. 15–26, Association for Computing Machinery, 2017.
- [106] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, “Black-box adversarial attacks with limited queries and information,” in *International Conference on Machine Learning*, 2018.
- [107] Y. Dong, Z. Deng, T. Pang, J. Zhu, and H. Su, “Adversarial distributional training for robust deep learning,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 8270–8283, 2020.
- [108] Y. Li, L. Li, L. Wang, T. Zhang, and B. Gong, “Nattack: Learning the distributions of adversarial examples for an improved black-box attack on deep neural networks,” in *International Conference on Machine Learning*, pp. 3866–3876, PMLR, 2019.

- [109] S. Senn and W. Richardson, "The first t-test," *Statistics in Medicine*, vol. 13, no. 8, pp. 785–803, 1994.
- [110] R. M. Norton, "The double exponential distribution: Using calculus to find a maximum likelihood estimator," *The American Statistician*, vol. 38, no. 2, pp. 135–136, 1984.
- [111] J. P. Nolan, "Numerical calculation of stable densities and distribution functions," *Communications in Statistics. Stochastic Models*, vol. 13, no. 4, pp. 759–774, 1997.
- [112] M. Evans, N. Hastings, and B. Peacock, *Statistical Distributions*. Wiley Series in Probability and Statistics, Wiley, 2000.
- [113] A. Kızılersü, M. Kreer, and A. W. Thomas, "The Weibull Distribution," *Significance*, vol. 15, pp. 10–11, 04 2018.
- [114] R. A. Fisher, *Statistical Methods for Research Workers*. Oliver and Boyd, 1925.
- [115] P. Greenwood and M. Nikulin, *A Guide to Chi-Squared Testing*. Wiley Series in Probability and Statistics, Wiley, 1996.
- [116] I. Chakravarti, R. Laha, J. Roy, and J. Roy, *Handbook of Methods of Applied Statistics*, vol. 1 of *Handbook of Methods of Applied Statistics*. Wiley, 1967.
- [117] T. W. Anderson and D. A. Darling, "Asymptotic theory of certain" goodness of fit" criteria based on stochastic processes," *The annals of mathematical statistics*, pp. 193–212, 1952.
- [118] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2015.
- [119] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5MB model size," 2017.

- [120] I. J. Myung, *Tutorial on maximum likelihood estimation*, vol. 47. Elsevier, 2003.
- [121] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016.
- [122] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, “Learning transferable architectures for scalable image recognition,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (Los Alamitos, CA, USA), pp. 8697–8710, IEEE Computer Society, jun 2018.
- [123] G. J. Brostow, J. Fauqueur, and R. Cipolla, “Semantic object classes in video: A high-definition ground truth database,” *Pattern Recognition Letters*, vol. xx, no. x, pp. xx–xx, 2008.
- [124] M. I. Jordan and T. M. Mitchell, “Machine learning: Trends, perspectives, and prospects,” *Science*, vol. 349, no. 6245, pp. 255–260, 2015.
- [125] J. Chai, H. Zeng, A. Li, and E. W. Ngai, “Deep learning in computer vision: A critical review of emerging techniques and application scenarios,” *Machine Learning with Applications*, vol. 6, p. 100134, 2021.
- [126] K. Bayoudh, R. Knani, F. Hamdaoui, and A. Mtibaa, “A survey on deep multimodal learning for computer vision: advances, trends, applications, and datasets,” *The Visual Computer*, vol. 38, no. 8, pp. 2939–2970, 2022.
- [127] L. Nanni, S. Ghidoni, and S. Brahnam, “Handcrafted vs. non-handcrafted features for computer vision classification,” *Pattern recognition*, vol. 71, pp. 158–172, 2017.
- [128] T. Majtner, S. Yildirim-Yayilgan, and J. Y. Hardeberg, “Combining deep learning and hand-crafted features for skin lesion classification,” in *2016 Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pp. 1–6, IEEE, 2016.
- [129] Y. Shin and I. Balasingham, “Comparison of hand-craft feature based svm and cnn based deep learning framework for automatic polyp classification,” in *2017 39th annual international conference of the IEEE*

engineering in medicine and biology society (EMBC), pp. 3277–3280, IEEE, 2017.

- [130] X. Huang, Z. Li, M. Zhang, and S. Gao, “Fusing hand-crafted and deep-learning features in a convolutional neural network model to identify prostate cancer in pathology images,” *Frontiers in Oncology*, vol. 12, p. 994950, 2022.
- [131] U. Côté-Allard, E. Campbell, A. Phinyomark, F. Laviolette, B. Gosselin, and E. Scheme, “Interpreting deep learning features for myoelectric control: A comparison with handcrafted features,” *Frontiers in bioengineering and biotechnology*, vol. 8, p. 158, 2020.
- [132] M.-I. Georgescu, R. T. Ionescu, and M. Popescu, “Local learning with deep and handcrafted features for facial expression recognition,” *IEEE Access*, vol. 7, pp. 64827–64836, 2019.
- [133] J. Bruna and S. Mallat, “Invariant scattering convolution networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1872–1886, 2013.
- [134] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [135] H. Bay, T. Tuytelaars, and L. Van Gool, “Speeded-up robust features (surf),” *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.
- [136] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 886–893, IEEE, 2005.
- [137] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-cam: Visual explanations from deep networks via gradient-based localization,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626, 2017.

- [138] E. Kanca Gulsoy, S. Ayas, E. Baykal Kablan, and M. Ekinici, “Enhancing the adversarial robustness in medical image classification: Exploring adversarial machine learning with vision transformers-based models,” *Neural Computing and Applications*, 2024.
- [139] H. Javed, S. El-Sappagh, and T. Abuhmed, “Robustness in deep learning models for medical diagnostics: Security and adversarial challenges towards robust ai applications,” *Artificial Intelligence Review*, 2024.
- [140] “Evaluating and enhancing the robustness of vision transformers against adversarial attacks in medical imaging,” *Medical & Biological Engineering & Computing*, 2024.
- [141] “On the role of deep learning model complexity in adversarial robustness for medical image classification,” *BMC Medical Informatics and Decision Making*, 2022.
- [142] X. Chen and T. Luo, “Adversarial-robust transfer learning for medical imaging via domain assimilation,” *arXiv preprint arXiv:2402.16005*, 2024.
- [143] Q. Yao, Z. He, and S. K. Zhou, “Medical aegis: Robust adversarial protectors for medical images,” *arXiv preprint arXiv:2111.10969*, 2021.
- [144] L. Ma and L. Liang, “Adaptive adversarial training to improve adversarial robustness of dnns for medical image segmentation and detection,” *arXiv preprint arXiv:2206.01736*, 2022.
- [145] C. Feichtenhofer, H. Fan, J. Malik, and K. He, “Slowfast networks for video recognition,” in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6202–6211, 2019.
- [146] Y. Gao, Z. Lin, Y. Yang, and J. Sang, “Towards black-box adversarial example detection: a data reconstruction-based method,” *arXiv preprint arXiv:2306.02021*, 2023.
- [147] B. Liang, H. Li, M. Su, X. Li, W. Shi, and X. Wang, “Detecting adversarial image examples in deep neural networks with adaptive noise reduction,” *IEEE Transactions on Dependable and Secure Computing*, 2019.

- [148] W. Xu, D. Evans, and Y. Qi, “Feature squeezing: Detecting adversarial examples in deep neural networks,” *arXiv preprint arXiv:1704.01155*, 2018.
- [149] L. Rice, E. Wong, and J. Z. Kolter, “Overfitting in adversarially robust deep learning,” *International Conference on Machine Learning*, pp. 8093–8104, 2020.
- [150] L. Schmidt, S. Santurkar, D. Tsipras, K. Talwar, and A. Madry, “Adversarially robust generalization requires more data,” in *Advances in Neural Information Processing Systems*, pp. 5014–5026, 2018.
- [151] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, “Robustness may be at odds with accuracy,” *International Conference on Learning Representations*, 2019.

Appendices

Appendix A

A.1 Supplementary Material of Chapter.3

This section includes supplementary materials to provide a more straightforward and comprehensive presentation of our experimental results. The following contents aim to further support our findings with extra evidence and analyses. We encourage readers to review these supplementary materials to understand our contribution to the field better.

Figure.A.1 demonstrates more examples of the histograms of DNN gradients of the DNN outputs with respect to the input image, which can be denoted as $\nabla_{\theta}\mathcal{L}(D_{\theta}(x), y)$. As mentioned in the Section.3.2, DNN gradients are critical for constructing adversarial perturbations, thus maximising the probability of misclassification. Figure.A.2 shows extra examples of the fitting curves by using six distribution models to characterise DNN gradients. By presenting these figures, we aim to provide clearer insights into the DNN gradients, and our hypothesis on DNN gradients is logical and data-driven.

Table.A.1 and Figure.A.3 demonstrate the performance improvement of using proposed Gradients Modelling Based Neutralisation (GMBN) for six DNNs on non-adversarially perturbed images. As mentioned in Section.3.3 of the paper, the results demonstrate that the neutralisation perturbation of the proposed defence is conducive for the DNN to make the correct prediction.

Figure.A.4 shows examples of comparing different pre-processing-based defence methods for six DNNs against adversarial attacks (See Section.3.3 of the paper). Noticeably, the first and second sets of Figure.A.4 respectively

show the defence performance of ResNet-50 against PGD and FGSM attacks with perturbation level $\epsilon = 32$. Our proposed GMBN achieves comparable performance in predicting strong adversarial examples correctly with high confidence scores compared to other pre-processing-based defence methods.

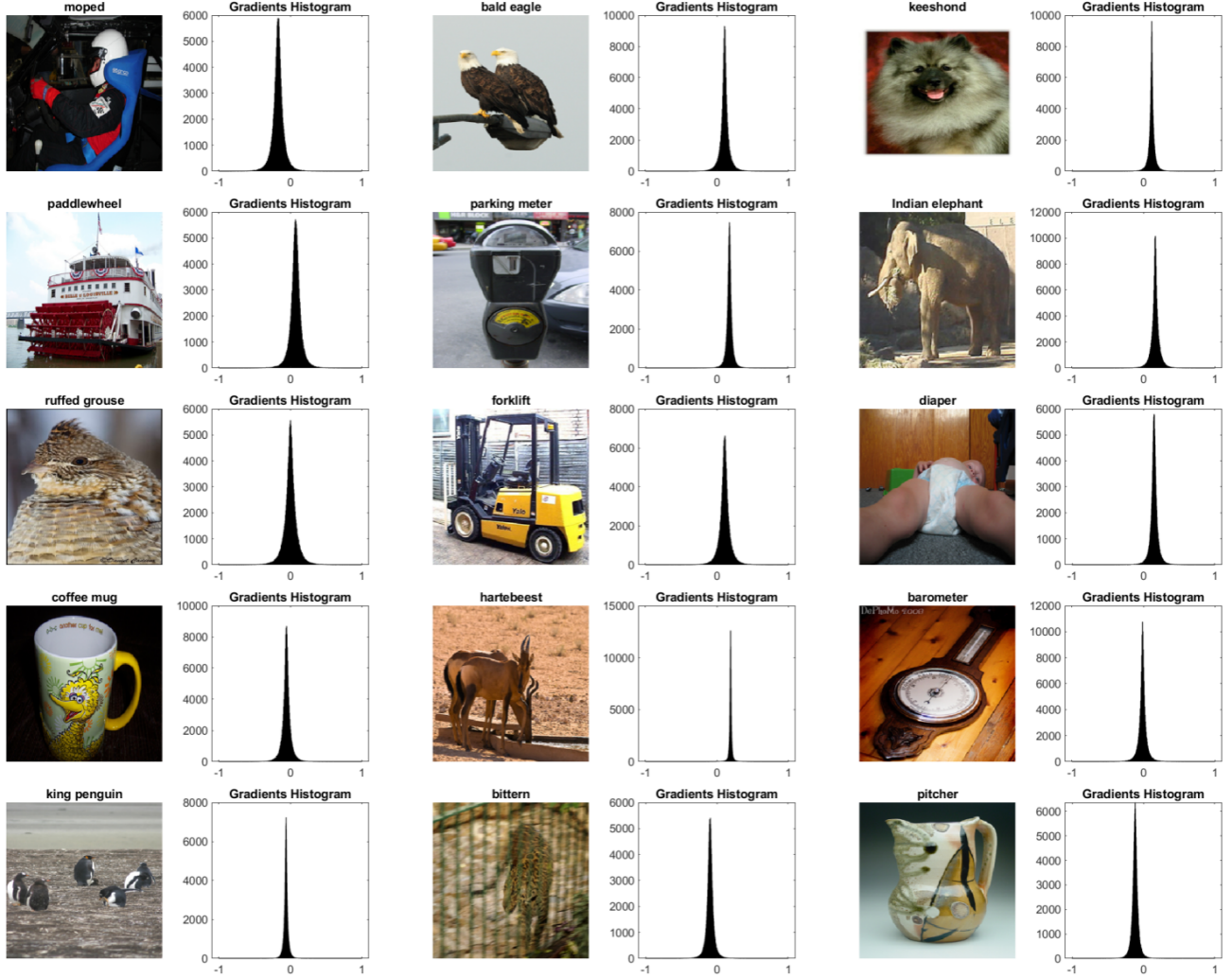
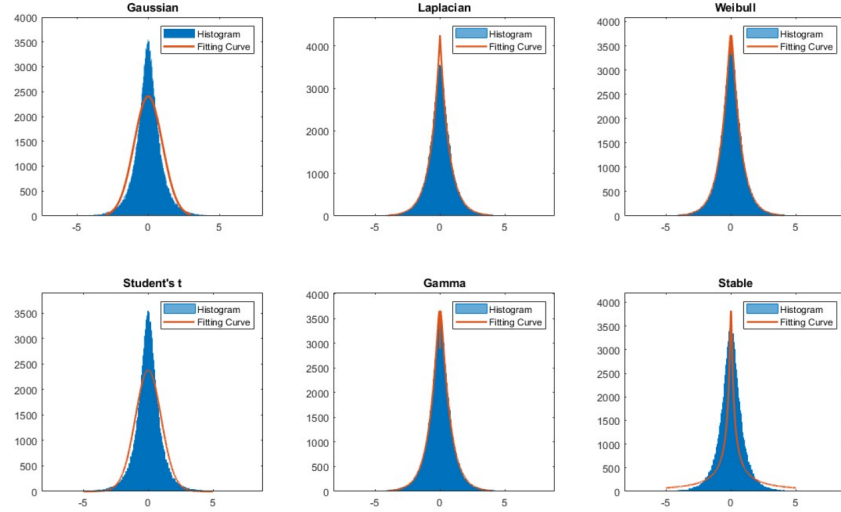


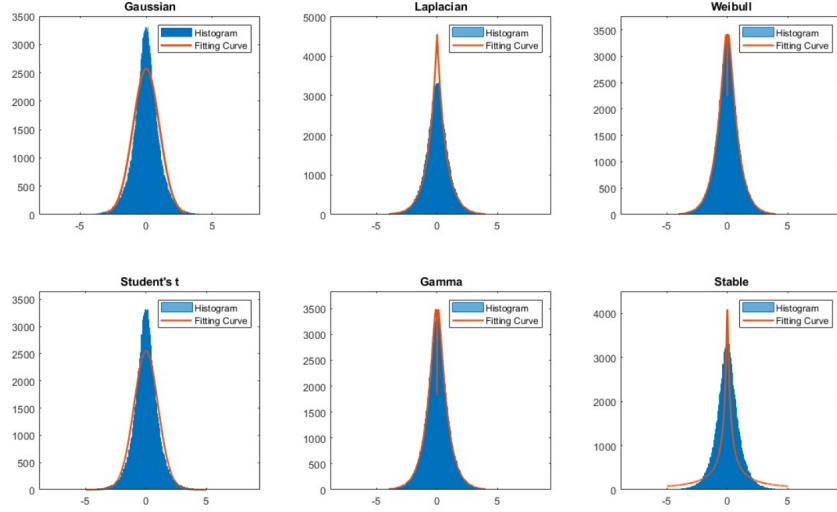
Figure A.1: Examples of the Histograms of DNN Gradients of the Predicted Class with respect to the Input Image



whippet (0.74)



prayer rug (0.98)



tractor (0.63)

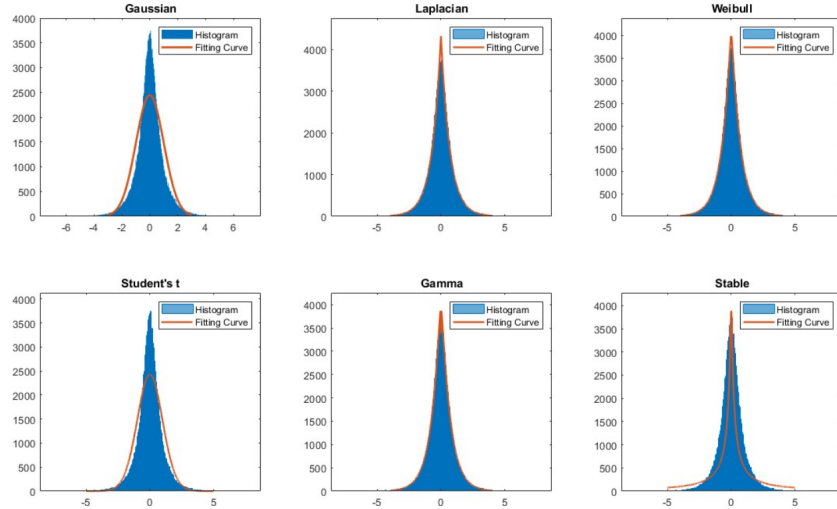


Figure A.2: Examples of the Fitting Curves of Using Six Distribution Models to Characterise DNN Gradients

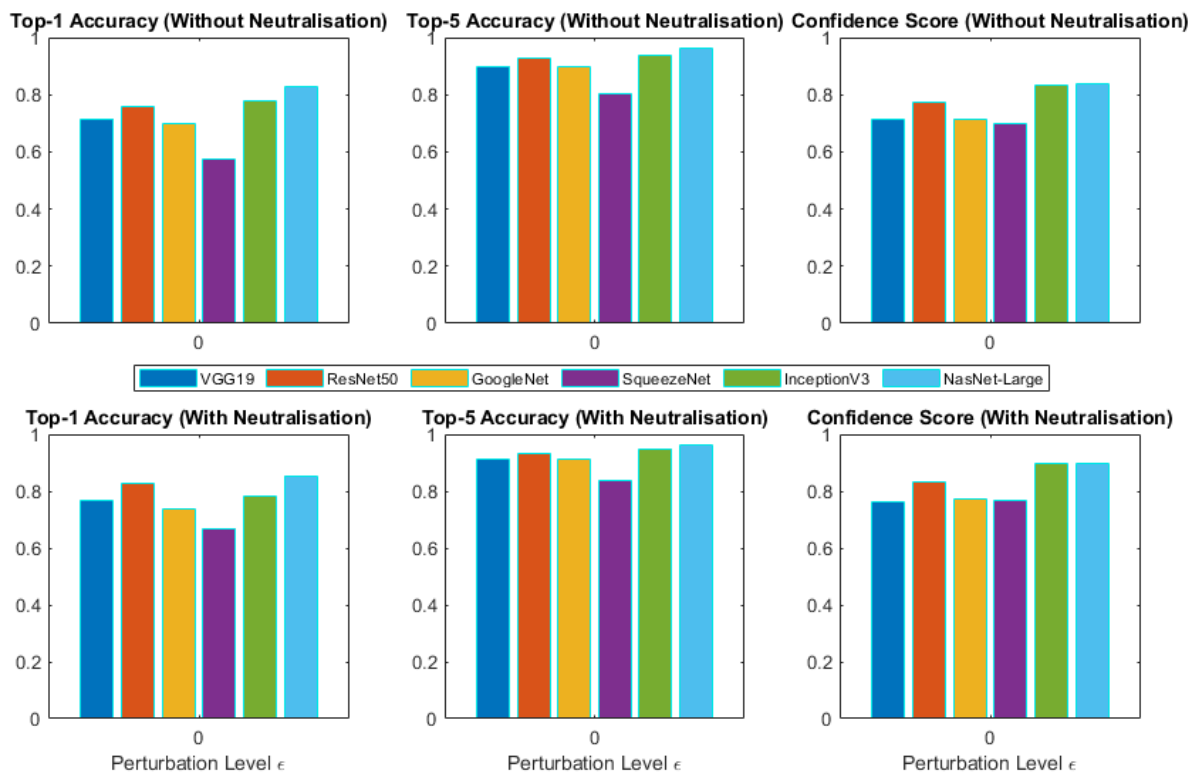


Figure A.3: Performance Improvement of GMBN for Non-Adversarially Perturbed Images (Note that the first row shows the model performance without the neutralisation perturbation added to the images, and the second row shows the model performance with the neutralisation perturbation)

Table A.1: GMBN Performance for Non-Perturbed Images*(a) Not Pre-processed by GMBN*

DNN Model	Non-Perturbed		
	Top-1 Accuracy	Top-5 Accuracy	Confidence Score
VGG19	71.50%	90.00%	71.3%
ResNet50	76.00%	93.00%	77.5%
GoogleNet	69.80%	89.60%	71.6%
SqueezeNet	57.50%	80.30%	69.7%
InceptionV3	77.90%	93.70%	83.4%
NasNet-Large	82.70%	96.20%	83.8%

(b) Pre-processed by GMBN

DNN Model	Non-Perturbed		
	Top-1 Accuracy	Top-5 Accuracy	Confidence Score
VGG19	76.70%	91.30%	76.3%
ResNet50	83.00%	93.50%	83.2%
GoogleNet	73.80%	91.20%	77.1%
SqueezeNet	67.00%	84.00%	76.6%
InceptionV3	78.30%	95.00%	90.0%
NasNet-Large	85.50%	96.40%	89.6%

(c) Increase in Performance

DNN Model	Non-Perturbed		
	Top-1 Accuracy	Top-5 Accuracy	Confidence Score
VGG19	5.20%	1.30%	5.00%
ResNet50	7.00%	0.50%	5.70%
GoogleNet	4.00%	1.60%	5.50%
SqueezeNet	9.50%	3.70%	6.90%
InceptionV3	0.40%	1.30%	6.60%
NasNet-Large	2.80%	0.20%	5.80%

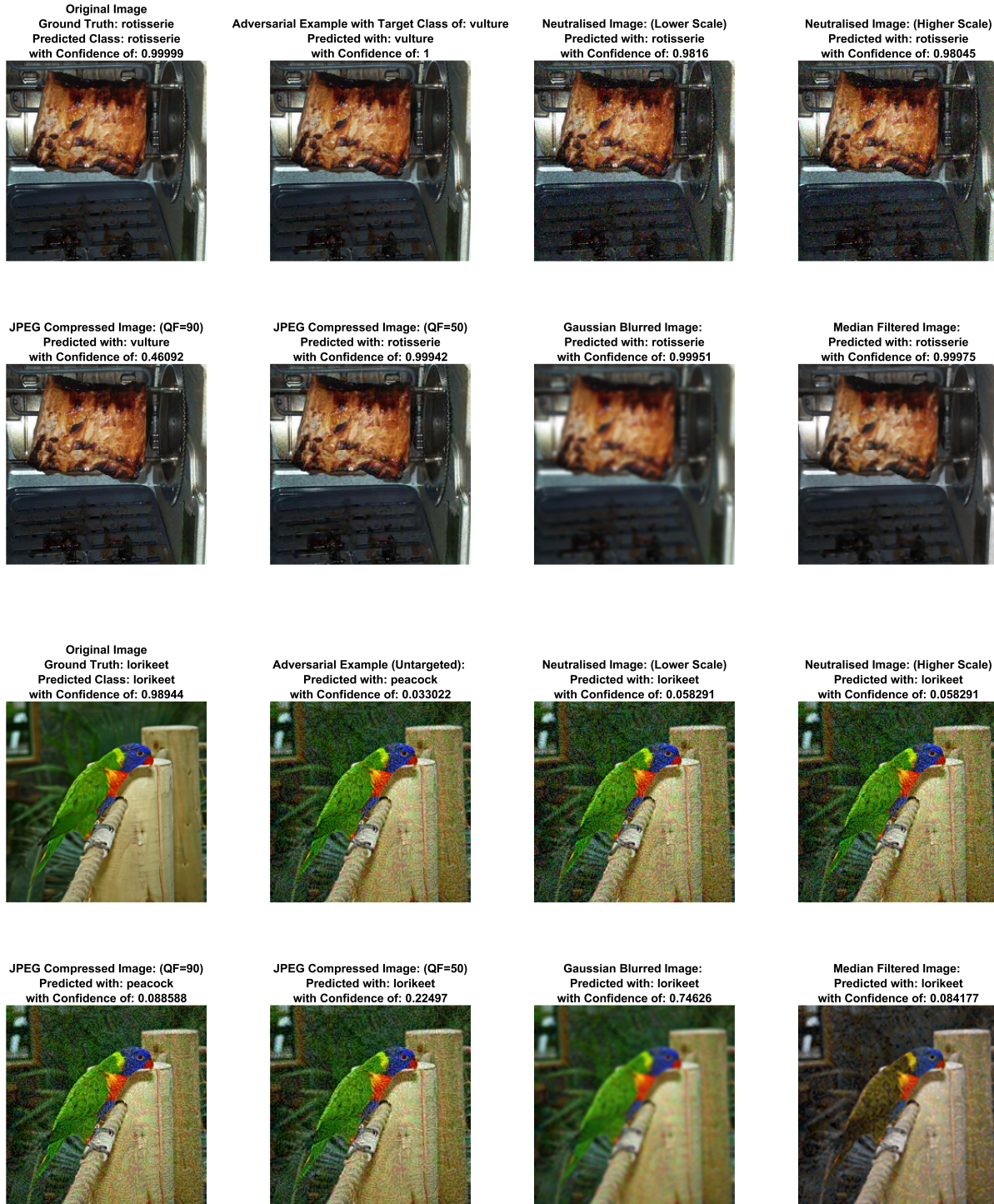


Figure A.4: Examples of Using Three Distributions to model DNN gradients with respect to the predicted labels

Appendix B

B.1 Supplementary Material of Chapter.4

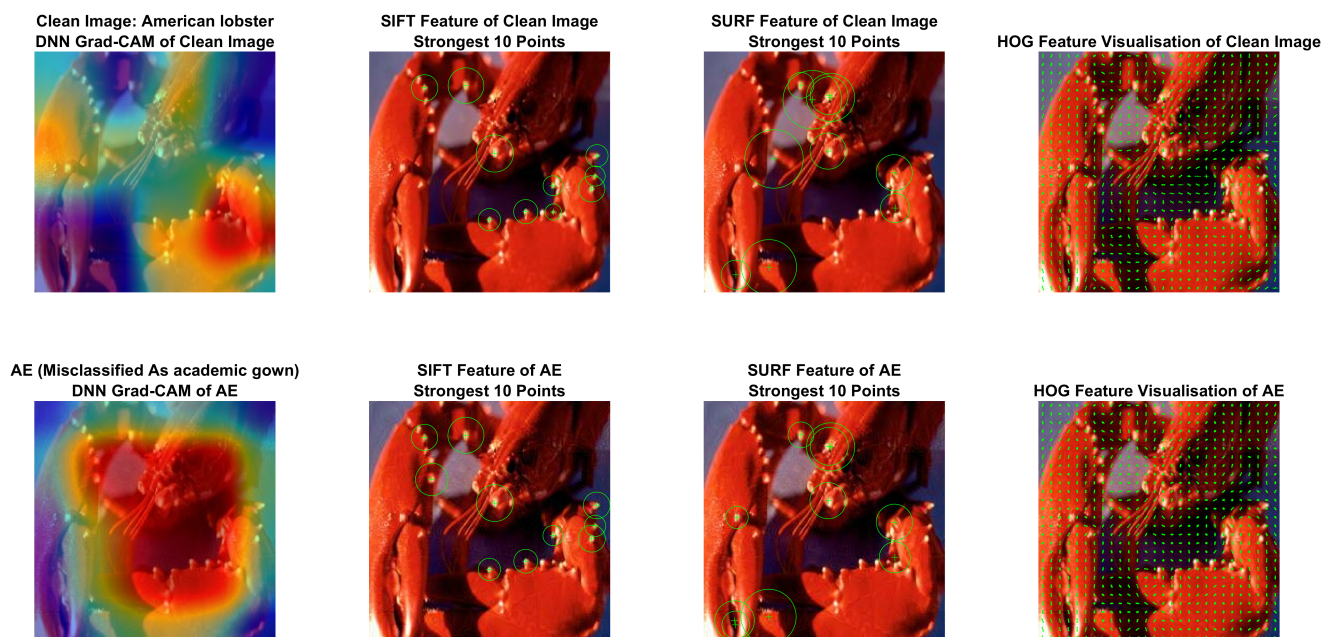


Figure B.1: Demonstration of different feature extraction techniques: DNN features visualised using Grad-CAM; Hand-Crafted features SIFT and SURF showing ten strongest points respectively; HOG Features are visualised by superimposing oriented gradients to the image.

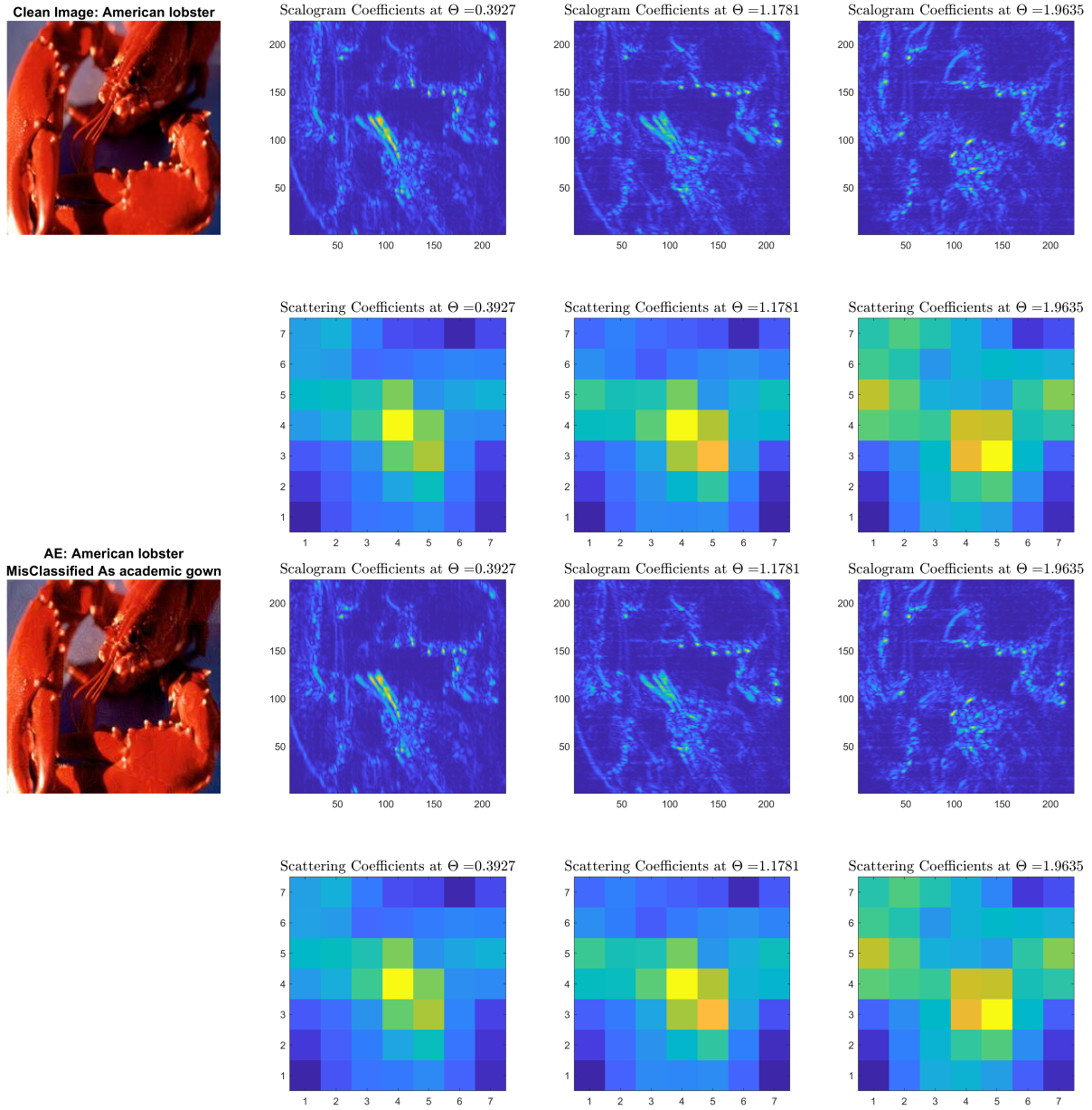


Figure B.2: Scalogram and scattering coefficients for clean image and AE across right rotation angles (merely three of them are shown due to space limit)

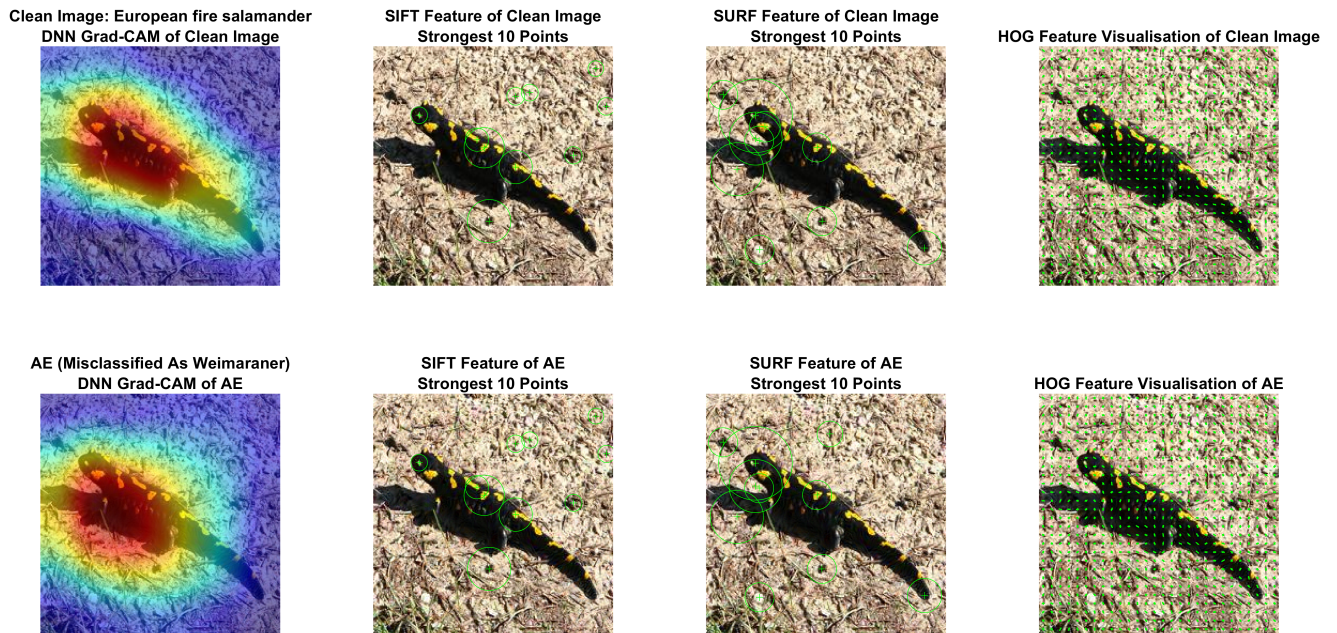


Figure B.3: *Demonstration of different feature extraction techniques: DNN features visualised using Grad-CAM; Hand-Crafted features SIFT and SURF showing 10 strongest points respectively; HOG Features are visualised by superimposing oriented gradients to the image.*

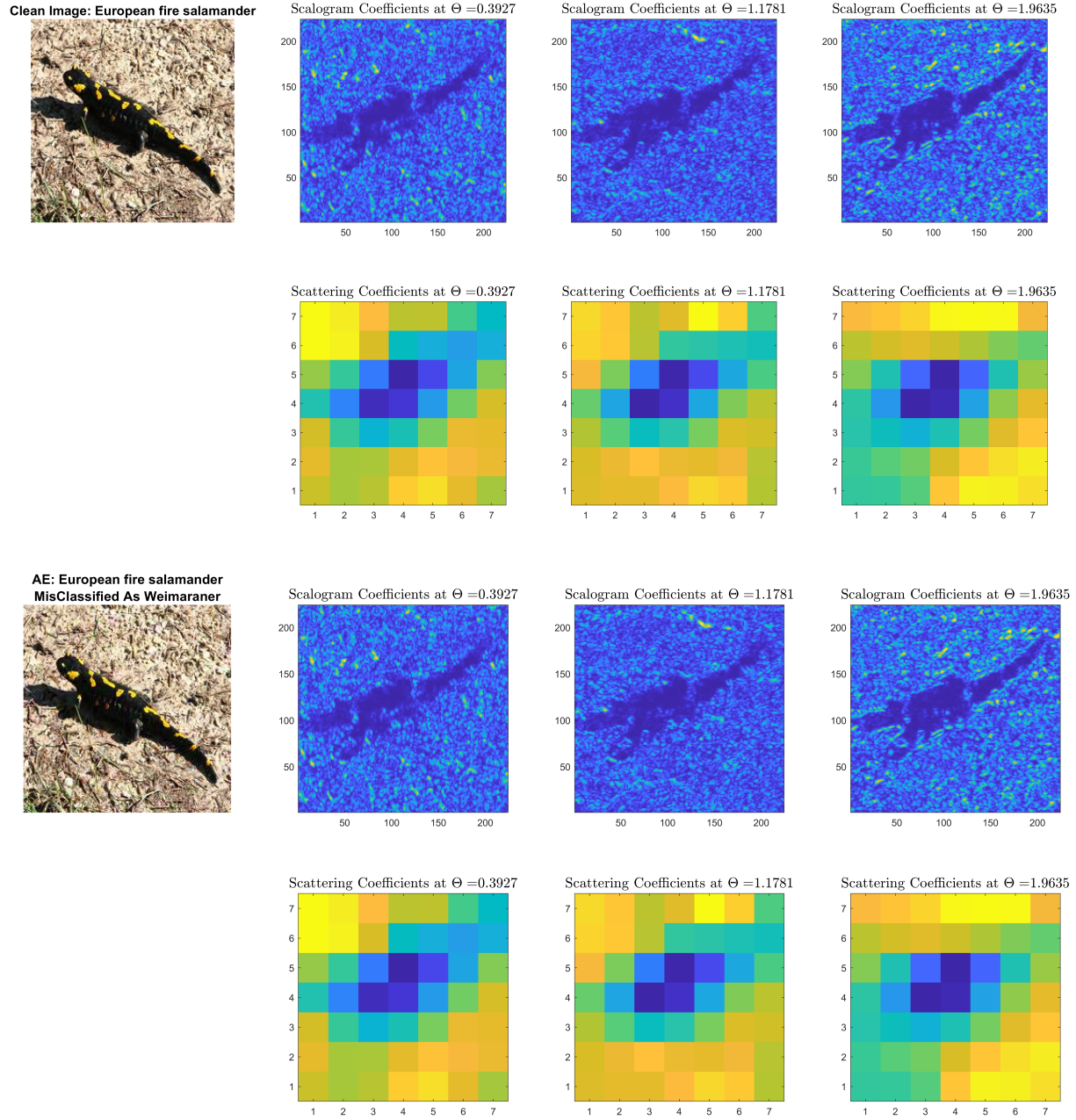
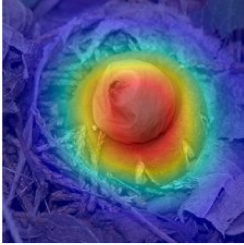


Figure B.4: Scalogram and scattering coefficients for clean image and AE across right rotation angles (merely three of them are shown due to space limit)

Clean Image: earthstar
DNN Grad-CAM of Clean Image



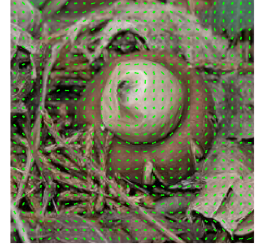
SIFT Feature of Clean Image
Strongest 10 Points



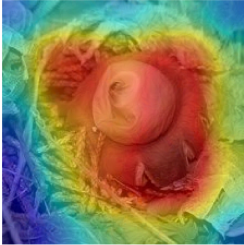
SURF Feature of Clean Image
Strongest 10 Points



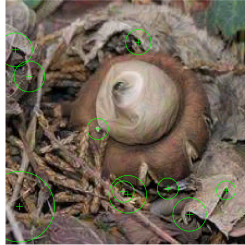
HOG Feature Visualisation of Clean Image



AE (Misclassified As comic book)
DNN Grad-CAM of AE



SIFT Feature of AE
Strongest 10 Points



SURF Feature of AE
Strongest 10 Points



HOG Feature Visualisation of AE

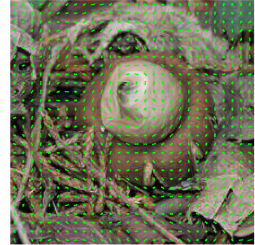


Figure B.5: Demonstration of different feature extraction techniques: DNN features visualised using Grad-CAM; Hand-Crafted features SIFT and SURF showing 10 strongest points respectively; HOG Features are visualised by superimposing oriented gradients to the image.

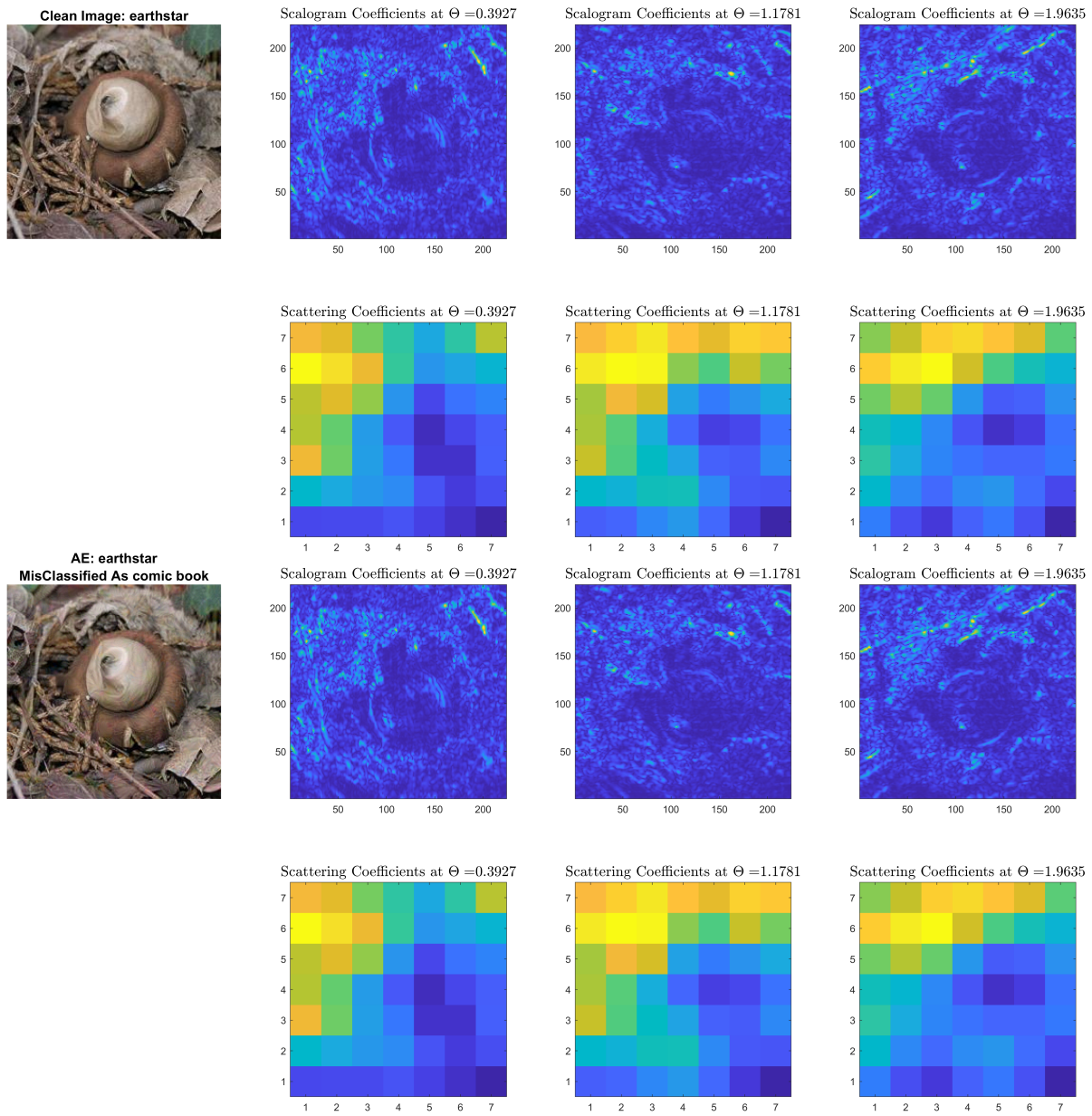


Figure B.6: Scalogram and scattering coefficients for clean image and AE across right rotation angles (merely three of them are shown due to space limit)

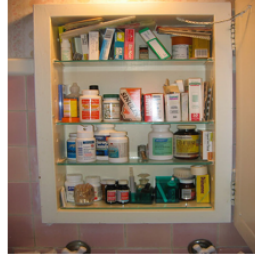
Clean Image: medicine chest
ResNet-101 (0.98)



AE (Misclassified As bighorn)
With Confidence of 0.99



Clean Image: medicine chest
Hybrid WSN ResNet-101(0.74)



AE: medicine chest
Hybrid WSN ResNet-101(0.77)



Grad-CAM of Clean Image



Grad-CAM of AE



Grad-CAM of Clean Image by
Hybrid WSN ResNet-101



Grad-CAM of AE by
Hybrid WSN ResNet-101



Clean Image: military uniform
ResNet-101 (0.92)



AE (Misclassified As wild boar)
With Confidence of 0.99



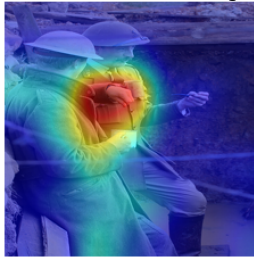
Clean Image: military uniform
Hybrid WSN ResNet-101(0.64)



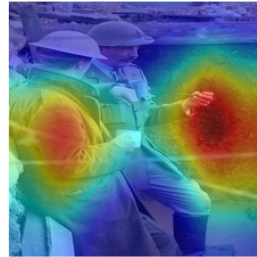
AE: military uniform
Hybrid WSN ResNet-101(0.73)



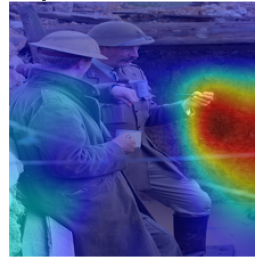
Grad-CAM of Clean Image



Grad-CAM of AE



Grad-CAM of Clean Image by
Hybrid WSN ResNet-101



Grad-CAM of AE by
Hybrid WSN ResNet-101

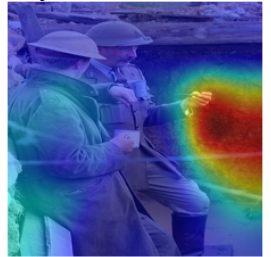


Figure B.7: Feature Visualisation Comparison between ResNet-101 and Hybrid WSN-ResNet-101-SA. Set(1): Clean image correctly classified by ResNet-101 and the corresponding Grad-CAM; Set(2): AE misclassified by ResNet-101 and the Grad-CAM of AE; Set(3): Same AE correctly classified by WSN-ResNet-101-SA and the Grad-CAM showing its attention focus by the hybrid network.

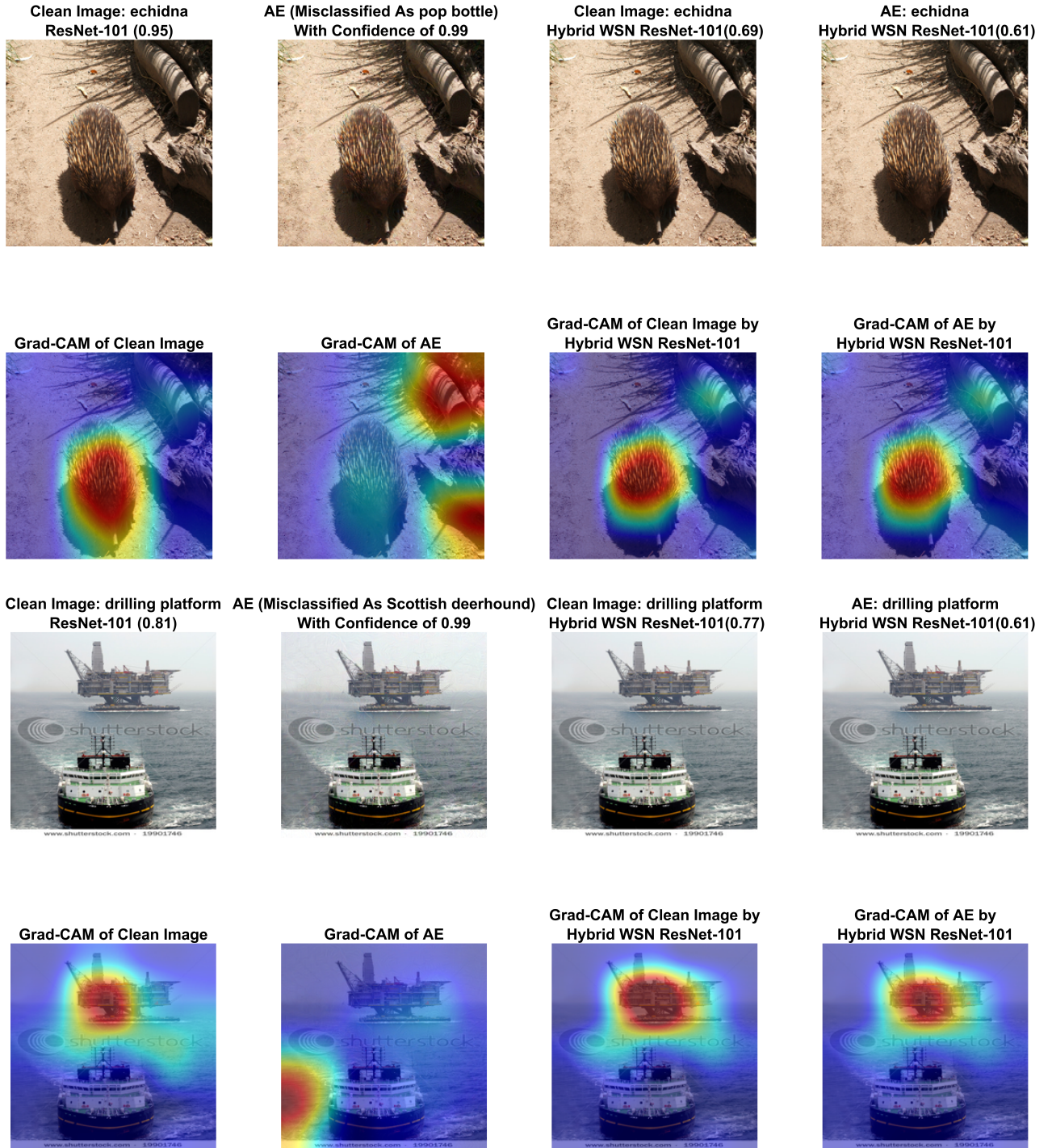
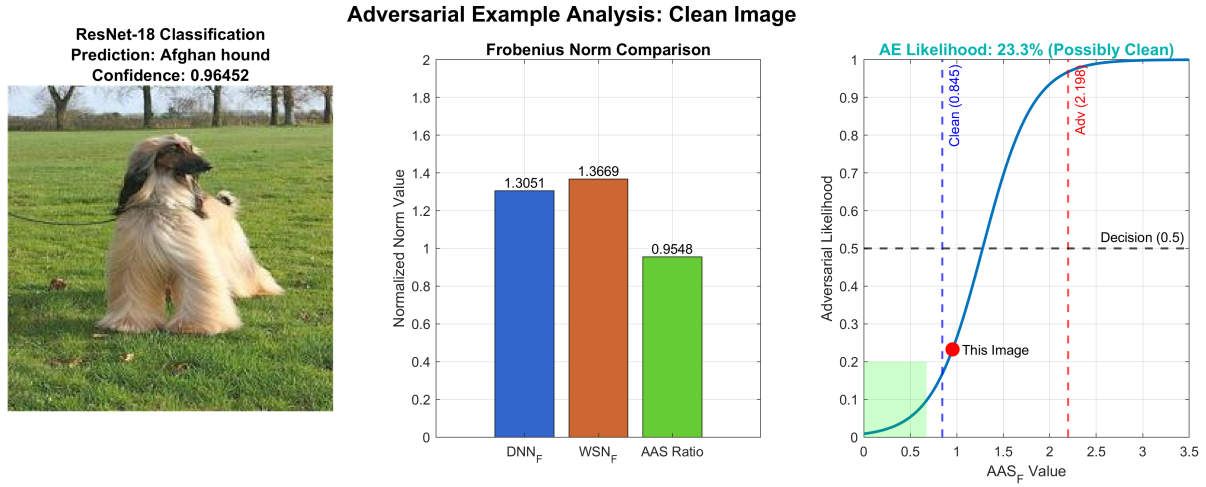


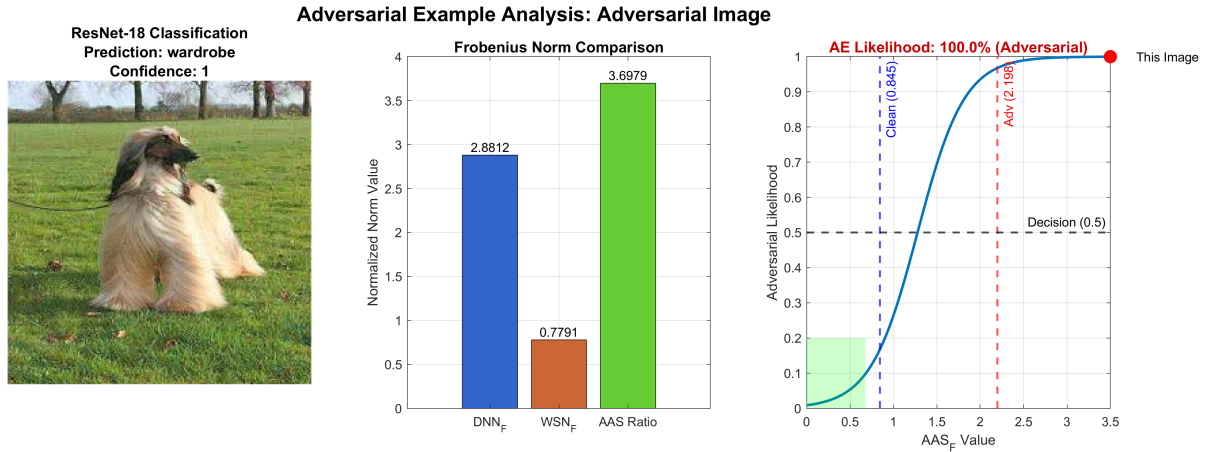
Figure B.8: Feature Visualisation Comparison between ResNet-101 and Hybrid WSN-ResNet-101-SA. Set(1): Clean image correctly classified by ResNet-101 and the corresponding Grad-CAM; Set(2): AE misclassified by ResNet-101 and the Grad-CAM of AE; Set(3): Same AE correctly classified by WSN-ResNet-101-SA and the Grad-CAM showing its attention focus by the hybrid network.

Appendix C

C.1 Supplementary Material of Chapter.5

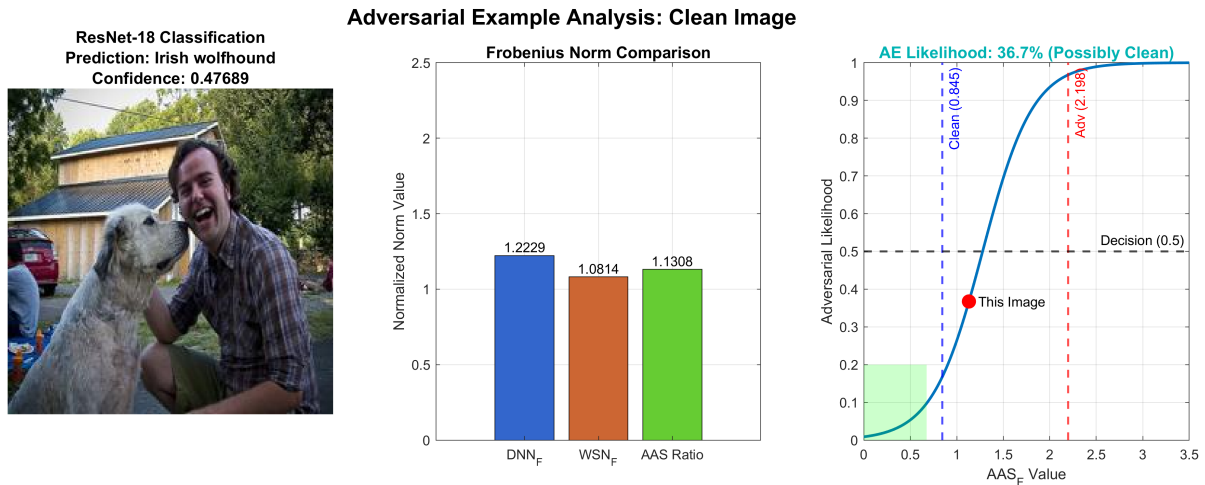


Original input image with ResNet-18 detection results

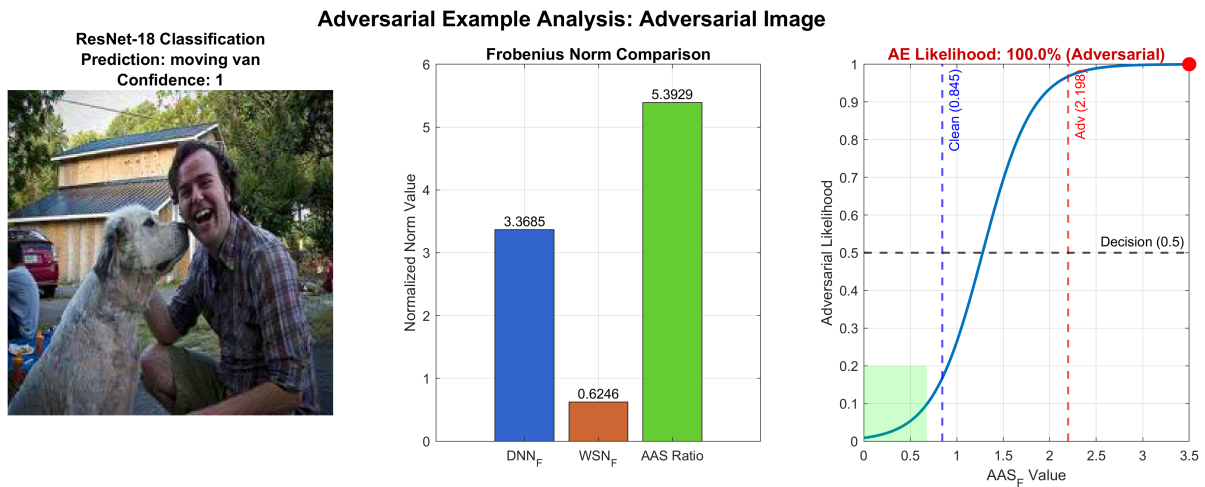


Adversarial Example with ResNet-18 detection results

Figure C.1: Comprehensive adversarial example detection pipeline. From left to right, the respective images show (1) the original input image with ResNet-18 classification results, displaying the predicted class label and confidence score. (2) Frobenius norm comparison between DNN_F features (extracted from ResNet-18 pool5 layer) and WSN_F features (from wavelet scattering transform), with their ratio (AAS_F score) used for adversarial detection. (3) AAS-based adversarial likelihood assessment showing the modified sigmoid mapping from AAS_F score to adversarial probability, with classification regions (Clean to Adversarial) and the current image's position marked.

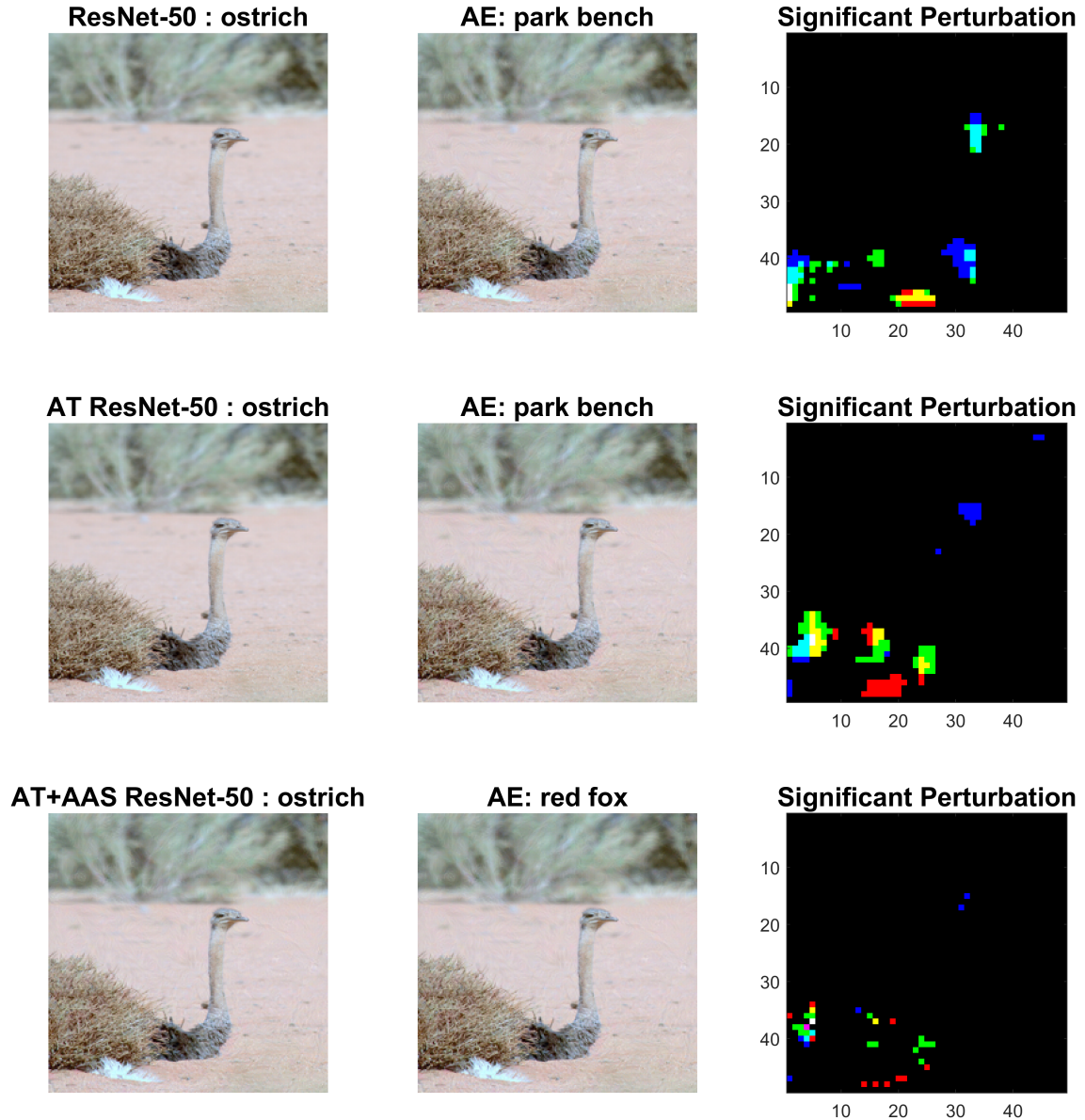


Original input image with ResNet-18 detection results



Adversarial Example with ResNet-18 detection results

Figure C.2: Comprehensive adversarial example detection pipeline. From left to right, the respective images show (1) the original input image with ResNet-18 classification results, displaying the predicted class label and confidence score. (2) Frobenius norm comparison between DNN_F features (extracted from ResNet-18 pool5 layer) and WSN_F features (from wavelet scattering transform), with their ratio (AAS_F score) used for adversarial detection. (3) AAS-based adversarial likelihood assessment showing the modified sigmoid mapping from AAS_F score to adversarial probability, with classification regions (Clean to Adversarial) and the current image's position marked.



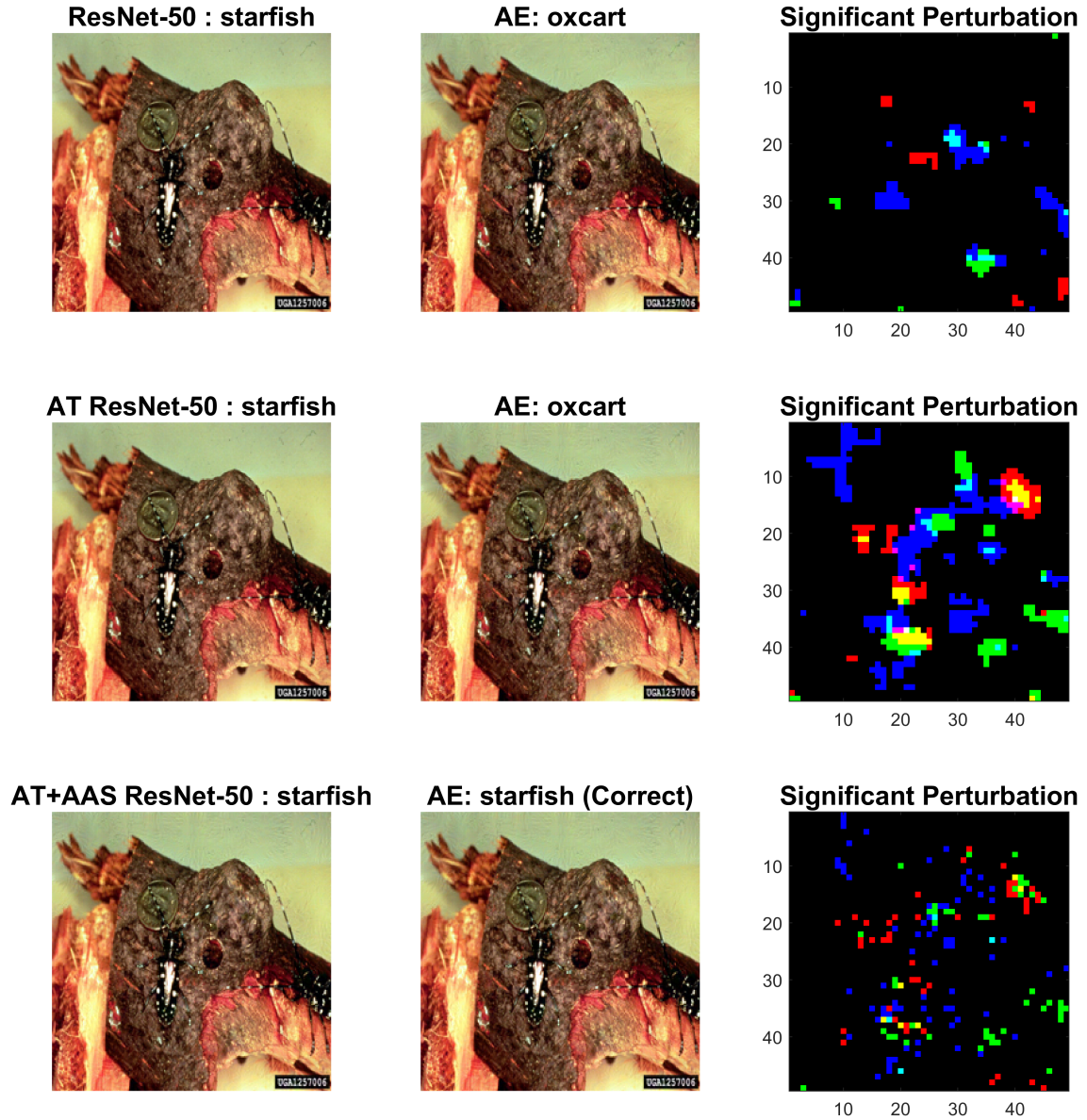


Figure C.4: Comparison of adversarial perturbations required for successful attacks across ResNet-50 variants. From left to right: perturbations for standard ResNet-50, adversarially trained ResNet-50, and ResNet-50 with AT+AAS defence. Despite achieving the same attack success rate, the perturbation patterns show significant variations across models, suggesting that adversarial training may primarily shift decision boundaries rather than enhance the model's fundamental robustness.

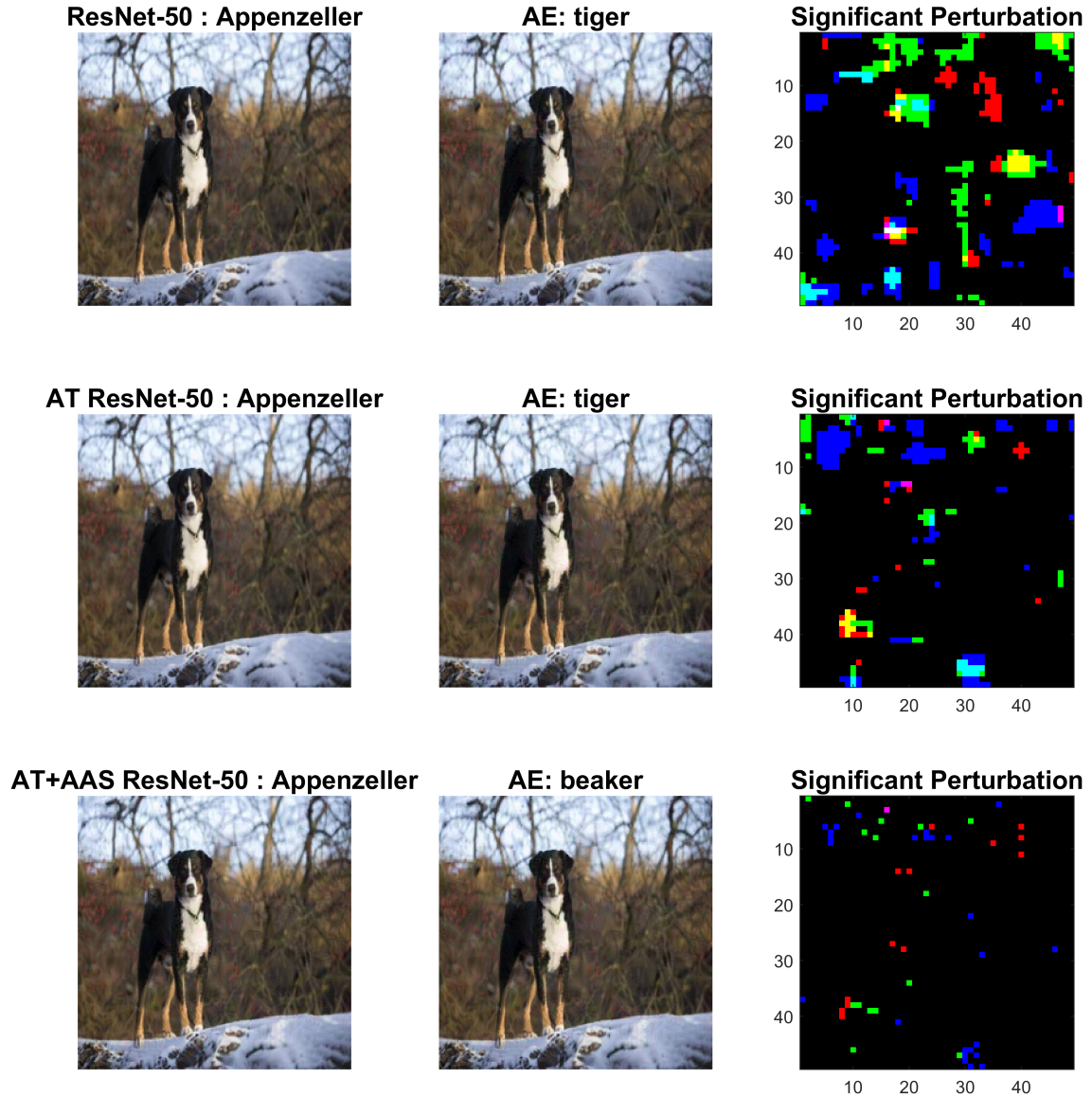


Figure C.5: Comparison of adversarial perturbations required for successful attacks across ResNet-50 variants. From left to right: perturbations for standard ResNet-50, adversarially trained ResNet-50, and ResNet-50 with AT+AAS defence. Despite achieving the same attack success rate, the perturbation patterns show significant variations across models, suggesting that adversarial training may primarily shift decision boundaries rather than enhance the model's fundamental robustness.