

ASSESSING CNN ROBUSTNESS IN MEDICAL IMAGING SYSTEMS: ADVERSARIAL THREATS AND DEFENSIVE MEASURES

D.S.C Wijesuriya

B.Sc. (Hons) Degree in Information Technology specialized in Cybersecurity

Department of Computer Systems and Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

April 2025

ASSESSING CNN RESILIENCE TO PENALIZING GRADIENT NORM (PGN) ADVERSARIAL ATTACKS

Final Report

Shamal Chathuranga Wijesuriya

IT21155802

B.Sc. (Hons) Degree in Information Technology Specializing in Cyber Security


Department of Information Technology

Sri Lanka Institute of Information Technology Sri Lanka

April 2025

DECLARATION OF THE CANDIDATE & SUPERVISOR

I declare that this is my own work and this final report does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Student Name	Registration Number	Signature
D.S.C. Wijesuriya	IT21155802	

The supervisor/s should certify the final report with the following declaration. The above candidate is carrying out research for the undergraduate Dissertation under my supervision.

.....

Signature of the supervisor

(Dr. Harinda Fernando)

.....

Date

.....

Signature of the co-supervisor

(Mr. Kavinga Yapa)

.....

Date

Abstract

This research investigates the resilience of InceptionV3 model against Penalizing Gradient Norm adversarial attacks in medical imaging, specifically chest X-ray classification. PGN attacks subtly alter input data, leading to significant prediction errors. The study aims to address gaps in existing research by evaluating the impact of these attacks, testing defense strategies like data augmentation and adversarial training, and validating their effectiveness using performance metrics such as accuracy, precision, recall, and F1-score. Through a systematic methodology, the goal is to enhance the robustness of InceptionV3 model, ensuring reliable and secure medical diagnostics.

Keywords: *Convolutional Neural Networks (CNNs), Medical Imaging, Adversarial Attacks, Penalizing Gradient Norm (PGN), Chest X-Ray Classification, InceptionV3*

TABLE OF CONTENTS

Abstract	iv
TABLE OF CONTENTS	v
TABLE OF FIGURES	vii
LIST OF TABLES	viii
LIST OF ABBREVIATIONS	ix
1. INTRODUCTION	1
1.1 Background	1
1.2 Literature Survey	2
1.3 Research Gap	3
1.4 Research Problem	4
2. OBJECTIVES	5
2.1 Main Objective.....	5
2.2 Specific Objectives	5
3. Methodology	7
3.1 Research and Technical Requirements	7
3.1.1 Research Scope and Objectives Mapping.....	7
3.1.2 Dataset Requirements	8
3.1.3 Model and Environment Requirements	11
3.1.4 Functional Requirements	12
3.1.5 Non-Functional Requirements	13
3.1.6 Budget Requirements.....	14
3.1.7 Feasibility Study	15
3.2 Design	16
3.2.1 Design Philosophy	17
3.2.2 System Workflow (High-Level Design).....	17
3.2.3 Model-Level Design Choices	19

3.3 System Implementation	21
3.3.1 Baseline InceptionV3 Model for Chest X-ray Classification	21
3.3.2 PGN Attack Implementation.....	22
3.3.3 PGN Attack Detection Strategy	24
3.3.4 Defense Strategy Implementation.....	25
3.4 Technology Stack.....	27
3.5 Results and Discussion	29
3.5.1 Baseline Model Performance (Pre-Attack)	29
3.5.2 PGN Adversarial Attack Impact Analysis (Post-PGN Attack).....	31
3.5.3 PGN Attack Detection System Evaluation	34
3.5.4 Defense Mechanism Evaluation (Post-Defense Implementation)	36
4. Commercialization and Future Application.....	40
5. Description of Personal Component	43
6. Conclusion	44
7. REFERENCES	46
8. APPENDICES	47
8.1 Appendix A - Data Preprocessing and Dataset prepare	47
8.2 Appendix B - Fine-tune InceptionV3 Architecture	49
8.3 Appendix C – Penalizing Gradient Norm Attack Algorithm.....	50
8.4 PGN Attack Summary For All Classes.....	50

TABLE OF FIGURES

Figure 1: NIH dataset classes and image count per class	9
Figure 2: Distribution of images per class in the curated dataset splits.....	9
Figure 3: Example X-ray images	10
Figure 4: Portainer Infrastructure.....	11
Figure 5: Overall System Design.....	19
Figure 6: PGN Attack System Diagram.....	19
Figure 7: Confusion Matrix and Classification Report.....	30
Figure 8: Training & Validation Accuracy/Loss Curves for Baseline InceptionV3 Model	31
Figure 9: Performance comparison of the baseline InceptionV3 model.....	32
Figure 10: Confusion Matrix for the baseline InceptionV3 model evaluated on the PGN adversarial subset, illustrating the misclassifications induced by the attack	33
Figure 11: Visual comparison of PGN attack	34
Figure 12: Example outputs from the PGN Attack Detection system	35
Figure 13: Classification performance of the PGN Adversarially Trained InceptionV3 Model	36
Figure 14: Confusion Matrix of the PGN Adversarially Trained InceptionV3 Model.....	37
Figure 15: Classification performance of the baseline InceptionV3 model with PSIF preprocessing on the PGN	38
Figure 16: Confusion Matrix of the baseline InceptionV3 model with PSIF preprocessing on the PGN.....	38
Figure 17: Different denoise techniques applied X rays.....	39
Figure 18: Our web application dashboard.....	42

LIST OF TABLES

Table 1: Research Gap	3
Table 2: Mapping Research Objectives to System Modules	8
Table 3: Functional Requirements and Modules	13
Table 4: Budget Chart.....	14

LIST OF ABBREVIATIONS

Abbreviation	Description
CNN	Convolutional Neural Network
PGN	Penalizing Gradient Norm
SIM	Split Image Adversarial
USSM	Uniform Scale Mix Mask
STM	Style Transfer Manipulation
IDE	Integrated Development Environment
PSIF	PGN-Signature Inversion Filtering

1. INTRODUCTION

1.1 Background

Medical imaging systems, particularly those utilized for chest X-ray classification, are critical in modern healthcare for diagnosing and monitoring various conditions, including pneumonia, tuberculosis, and lung cancer. These systems rely heavily on Convolutional Neural Networks to accurately interpret medical images, providing vital support to healthcare professionals in making informed decisions. However, the integrity and reliability of these systems are increasingly challenged by adversarial attacks, which exploit vulnerabilities in machine learning models to cause misclassifications.

One such adversarial technique is the PGN attack, which subtly alters input images in ways that are almost imperceptible to the human eye but can lead to significant errors in CNN predictions. The potential consequences of these errors in a medical setting are severe, as they can result in incorrect diagnoses and inappropriate treatments, ultimately compromising patient safety.

Despite the growing use of CNNs in medical imaging, there is a noticeable gap in the literature regarding the resilience of these models to PGN attacks. While considerable research has been conducted on other types of adversarial attacks, the specific impact of PGN attacks on InceptionV3 model performance, particularly within the context of medical imaging, remains underexplored. Furthermore, existing defense strategies have not been thoroughly tested against PGN attacks, leaving questions about their effectiveness and reliability.

This research aims to close these gaps by evaluating the strength of InceptionV3 model against PGN adversarial attacks and confirming the effectiveness of current defense mechanisms. This study will help improve the security and dependability of medical imaging systems, ensuring that they can still offer accurate and reliable diagnostic support even in the presence of advanced adversarial threats.

1.2 Literature Survey

The application of Convolutional Neural Networks in medical imaging has been transformative, providing enhanced diagnostic accuracy and efficiency. CNN models have demonstrated their effectiveness in various medical imaging tasks, such as the detection of COVID-19 from chest X-rays, where they have achieved significant success in classifying complex visual patterns in medical data [1]. Despite these advancements, CNNs are vulnerable to adversarial attacks, which involve subtle perturbations to input data that can lead to significant misclassifications [2].

Adversarial attacks on CNNs, such as the PGN attack, present a significant challenge in ensuring the robustness of these models. PGN attacks work by manipulating the gradient norms during the model's training process, making the model highly sensitive to small input changes. This results in the model misclassifying perturbed images, which poses serious risks in critical applications like medical imaging [3]. Recent studies have explored various methods to enhance the transferability and effectiveness of adversarial attacks, including those that focus on finding flat local maxima in the model's loss landscape [4].

In the context of medical imaging, the impact of PGN attacks has been less explored, particularly in terms of their effects on CNN models trained with medical datasets. Previous research has largely focused on the general aspects of adversarial attacks without delving deeply into their implications for medical diagnostics [5]. Additionally, there is a notable gap in the literature regarding the testing and validation of existing defense mechanisms against PGN attacks within medical imaging contexts. While some defense strategies, such as adversarial training and data augmentation, have been proposed, their effectiveness against PGN attacks specifically in medical imaging remains under-researched [6].

This literature survey highlights the need for comprehensive research that addresses these gaps. Specifically, it underscores the importance of evaluating the resilience of CNN models against PGN attacks in medical imaging, testing existing defense strategies, and exploring new methods to enhance model robustness. By filling these gaps, the research aims to contribute to the development of more secure and reliable diagnostic tools in healthcare.

1.3 Research Gap

Despite previous research on PGN adversarial attacks, significant gaps remain, particularly in the context of medical imaging. While the impact of PGN attacks on CNN performance has been explored, there is a lack of comprehensive studies that apply these findings to medical datasets, which are crucial for real-world healthcare applications.

Additionally, existing defense strategies against PGN attacks have not been adequately tested or validated, leaving uncertainty about their effectiveness in critical medical contexts. Furthermore, the use of comprehensive performance metrics to assess CNN resilience under PGN attacks is inconsistent across the literature.

Our research addresses these gaps by focusing on PGN attacks within medical imaging, validating defense strategies, and conducting a detailed evaluation of model performance, aiming to enhance the robustness and reliability of CNNs in healthcare settings.

	Research 1	Research 2	Research 3	Our Research
Focus on PGN adversarial attacks	Yes	Yes	No	Yes
Impact of PGN on ResNet CNN model performance	Yes	Yes	No	Yes
Evaluation using medical imaging datasets	No	No	Yes	Yes
Test and Validation of existing defense strategies against PGN	No	No	No	Yes
Comprehensive performance metrics under PGN attack	Yes	No	No	Yes

Table 1: Research Gap

1.4 Research Problem

The critical issue at hand is the vulnerability of CNNs used in medical imaging to PGN adversarial attacks. These attacks exploit the model's sensitivity by introducing small, almost imperceptible perturbations in the input data, leading to significant misclassifications. In a medical context, such errors can have dire consequences, potentially resulting in incorrect diagnoses and inappropriate treatment plans that could endanger patient safety.

Despite the seriousness of this danger, the specific effects of PGN attacks on CNN performance have not been thoroughly investigated, particularly in the field of medical imaging. Furthermore, we do not have a clear understanding of how effective current defense strategies are against PGN attacks, which creates a gap in ensuring the strength and dependability of these important diagnostic tools. The main focus of the research problem is, therefore, to comprehensively evaluate the resilience of CNN models to PGN attacks and confirm the effectiveness of defense mechanisms in safeguarding against these advanced threats.

2. OBJECTIVES

2.1 Main Objective

The primary aim of this research is to systematically investigate the robustness of a specific deep learning architecture, InceptionV3, when applied to chest X-ray classification, against the sophisticated PGN adversarial attack. This study seeks to quantify the impact of PGN attacks on the model's diagnostic accuracy and to develop, implement, and validate effective detection and defense strategies specifically tailored or relevant to this threat. The overarching goal is to contribute to the development of more secure and reliable AI-driven medical imaging tools, thereby enhancing diagnostic confidence and safeguarding patient well-being in clinical settings where such technologies are deployed.

2.2 Specific Objectives

To achieve this main objective, the following specific objectives have been defined:

1. **Develop and Evaluate a Baseline InceptionV3 Model for Chest X-ray Classification:**

- **Objective:** Train and evaluate an InceptionV3 model to accurately classify chest X-ray images into 'No Finding', 'Pneumonia', and 'Pneumothorax' categories.
- **Purpose:** To establish a high-performing baseline model whose performance metrics on clean data can serve as a benchmark. This provides a reference point to quantify the degradation caused by adversarial attacks and the improvements offered by defense mechanisms.

2. **Implement and Analyze PGN Adversarial Attacks:**

- **Objective:** Implement the PGN attack algorithm and generate adversarial examples targeting the validated InceptionV3 baseline model.
- **Purpose:** To simulate a realistic white-box threat scenario where PGN attacks introduce subtle, gradient-norm-aware perturbations. This step aims to thoroughly assess the baseline model's vulnerability to PGN, quantifying the extent to which its predictions can be manipulated and identifying potential weaknesses specific to this attack methodology.

3. **Develop and Evaluate a PGN Attack Detection Mechanism:**

- **Objective:** Design, implement, and evaluate a detection mechanism, named Transformation & Prediction Consistency Check (utilizing Gaussian Blur), to identify incoming chest X-ray images that may have been perturbed by PGN attacks.
- **Purpose:** To create a pre-emptive measure that can identify potentially compromised inputs before they are processed by the main diagnostic model. The effectiveness of this detector will be evaluated based on its ability to distinguish between clean and PGN-attacked images.

4. Implement and Evaluate Defense Strategies against PGN Attacks:

- **Objective:** Implement and assess the efficacy of multiple defense strategies, including:
 1. **PGN Adversarial Training:** Fine-tuning the model using a combination of clean images and PGN adversarial examples.
 2. **PGN-Signature Inversion Filtering (PSIF):** Implementing an inference-time defense designed to counteract PGN characteristics by applying an input preprocessing filter (specifically, Gaussian Blur with empirically determined parameters: kernel size (3x3) and sigma 0.8) to attempt to 'purify' the input image by smoothing PGN-induced high-frequency perturbations before classification.
 3. **Different Denoising Techniques:** Applying selected established image denoising algorithms (e.g., Median Filter, Non-Local Means) as a preprocessing step to reduce adversarial noise.
- **Purpose:** To systematically evaluate the ability of these distinct defense approaches to enhance the InceptionV3 model's robustness against PGN attacks. This involves measuring how well each defense maintains accuracy on clean data while improving performance on adversarial data.

5. Comprehensive Performance Validation and Comparison:

- **Objective:** Conduct a thorough comparative analysis of the InceptionV3 model's performance under various conditions: (a) on clean data (baseline), (b) under PGN attack without defenses, (c) with the PGN detection mechanism

active (evaluating its impact on the overall system if an action like rejection is taken based on detection), and (d) after the application of each defense strategy (Adversarial Training, PSIF, and each Denoising technique).

- **Purpose:** To provide a holistic understanding of the PGN attack's impact and the relative effectiveness of the implemented detection and defense mechanisms. This involves comparing key performance metrics (accuracy, precision, recall, F1-score, confusion matrices) across these scenarios to draw robust conclusions about enhancing model security and reliability in the context of PGN adversarial threats.

3. Methodology

This section outlines the comprehensive methodology employed for this research, encompassing the development of a baseline chest X-ray classification model, the implementation and analysis of Perturbed Gradients Norm (PGN) adversarial attacks, and the design and evaluation of corresponding detection and defense mechanisms. It begins by detailing the foundational research and technical requirements, followed by the functional and non-functional system specifications, budget considerations, and a feasibility assessment. Subsequent sub-sections (3.2 onwards) will delve into the specifics of each component.

3.1 Research and Technical Requirements

This sub-section delineates the core research components and the essential technical prerequisites for the successful execution of the study's objectives. Given the project's focus on adversarial machine learning within the critical domain of medical imaging, a careful definition of the research scope and technical infrastructure was paramount before system implementation.

3.1.1 Research Scope and Objectives Mapping

The research concentrates on the following key areas:

- Training an **InceptionV3** Convolutional Neural Network (CNN) to classify chest X-ray images into 'No Finding', 'Pneumonia', and 'Pneumothorax'.
- Designing and applying the **Perturbed Gradients Norm (PGN) attack** to evaluate its impact on the InceptionV3 model's robustness.

- Developing and evaluating a **Transformation & Prediction Consistency Check** algorithm for the detection of PGN adversarial inputs.
- Implementing and assessing defense mechanisms, including **PGN Adversarial Training**, a proposed **PGN-Signature Inversion Filtering (PSIF)** technique, and various established **Image Denoising** methods.

Each of these research components necessitates specific system configurations, datasets, and evaluation protocols, as mapped below:

Objective	System Module	Key Tools/Technologies
Baseline Model Training (InceptionV3)	CNN Classification Engine	TensorFlow, Keras, Python
PGN Attack Implementation	PGN Adversarial Attack Generator	TensorFlow, NumPy, Custom Python Script
PGN Attack Detection (Transformation & Prediction Consistency Check)	Adversarial Detection Algorithm	OpenCV (for Gaussian Blur), TensorFlow, NumPy
Defense Implementation	Detection Algorithm	TensorFlow, Keras, Custom Training Loop, OpenCV, NumPy
Data Preprocessing & Management	Input Pipeline	Pandas, TensorFlow ImageDataGenerator/tf.data
Performance Evaluation & Visualization	Evaluation & Reporting Module	Scikit-learn, Matplotlib, Seaborn, Pandas

Table 2: Mapping Research Objectives to System Modules

3.1.2 Dataset Requirements

The study utilizes a curated subset derived from the **NIH Chest X-ray Dataset**, a large-scale, publicly available collection of chest radiographs originally containing over 112,000 images across 14 pathology labels.

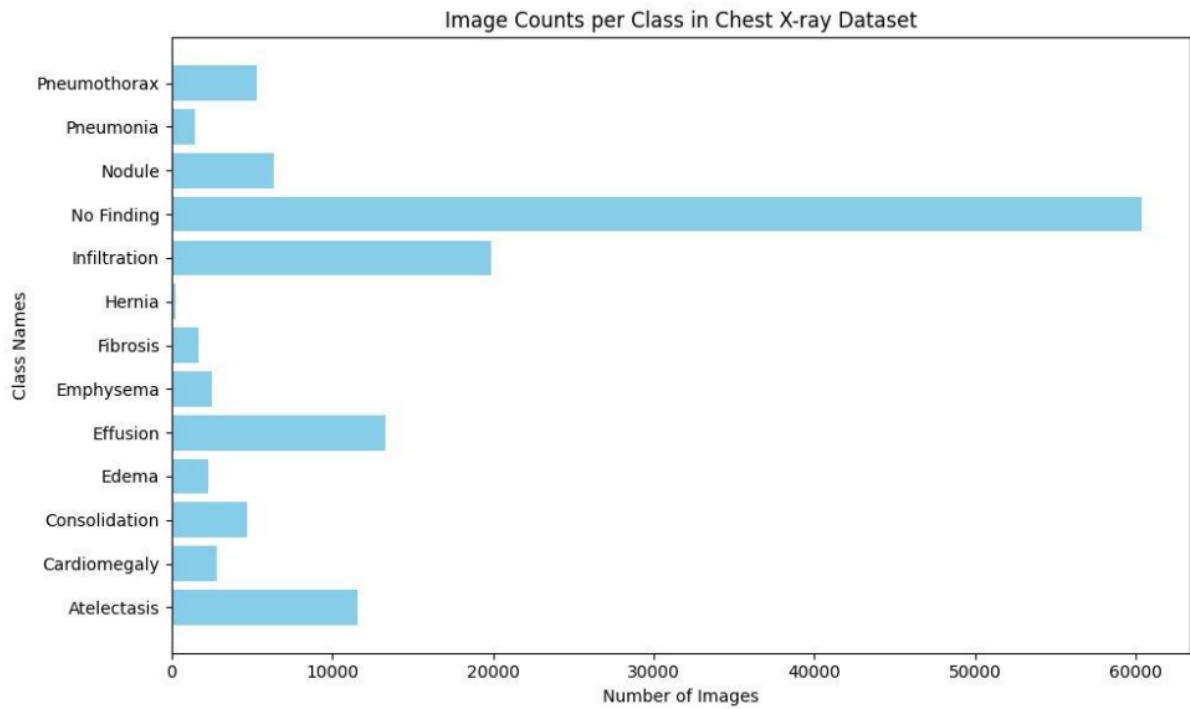


Figure 1: NIH dataset classes and image count per class

For the focused scope of this research, images corresponding to three clinically relevant classes were selected: 'No Finding', 'Pneumonia', and 'Pneumothorax'.

- **Initial Split for Baseline Model:**

- The curated dataset was partitioned into training, validation, and test sets.
 - **Training Set:** 4800 images
 - **Validation Set:** 600 images
 - **Test Set:** 601 images

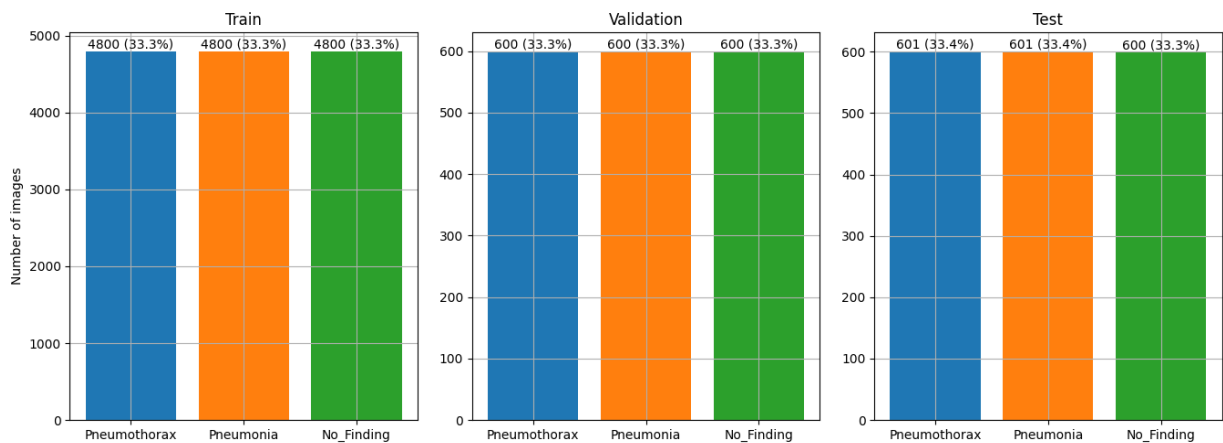


Figure 2: Distribution of images per class in the curated dataset splits

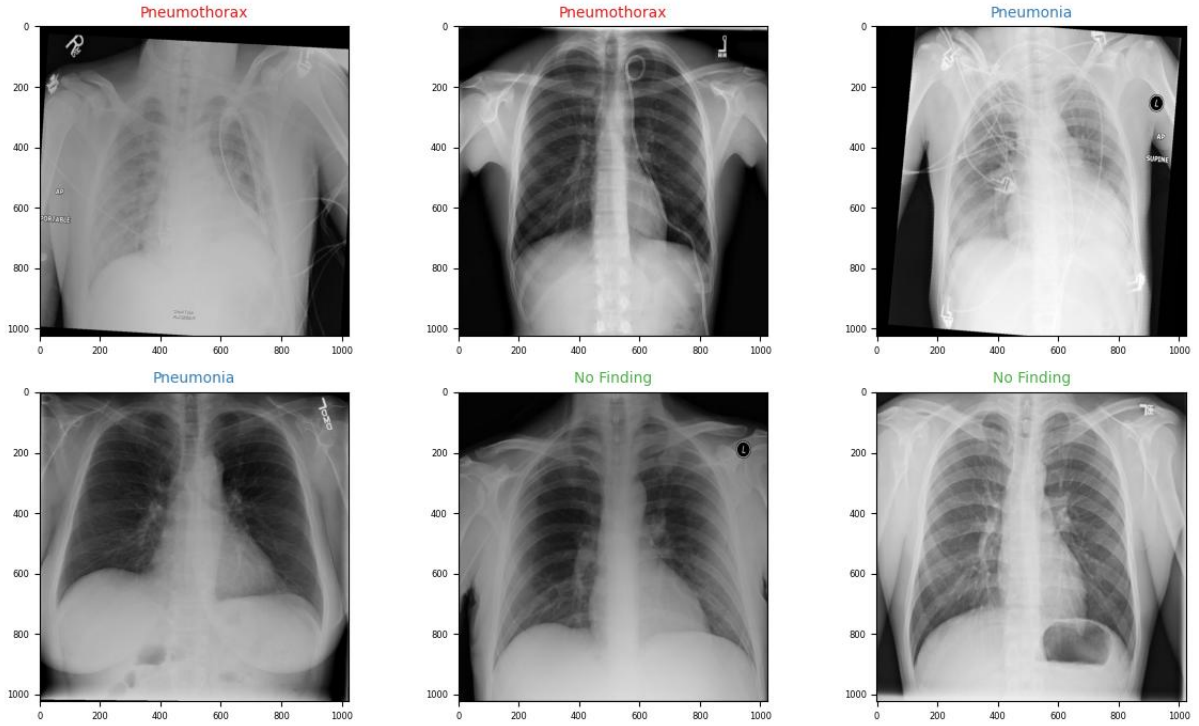


Figure 3: Example X-ray images

- **Image Specifications:**

- Images were resized to **299×299 pixels** to match the input requirements of the InceptionV3 architecture.
- Images were processed in **RGB format** (3 channels).
- Pixel values were normalized to the **[0, 1]** range.

- **Derived Subset for Attack/Defense Experiments:**

- A subset of **6000 images** (4800 'No Finding', 4800 'Pneumonia', 4800 'Pneumothorax') was created from the Test Set by selecting only those images initially correctly classified by the baseline InceptionV3 model. This "Correctly Classified Subset" served as the clean basis for generating PGN adversarial examples used in attack impact assessment, detector evaluation, and static adversarial training.

3.1.3 Model and Environment Requirements

To support the computational demands of training deep learning models, generating adversarial attacks, and evaluating detection/defense mechanisms, the following hardware and software environment was utilized:

- **Hardware (Google Colab Pro/Pro+):**
 - GPU: NVIDIA T4 or L4 GPU instances provided by Google Colab.
 - CPU: High-performance CPUs available on Colab instances.
 - RAM: High-memory VMs (typically >25 GB RAM).
 - Storage: Google Drive integration for dataset and model storage, and Colab's local disk for temporary files.
- **Hardware (VPS):**
 - Portainer Infrastructure
 - CPU-enabled system (32 CPU Cores)
 - 64 GB RAM
 - SSD storage for fast I/O with image data

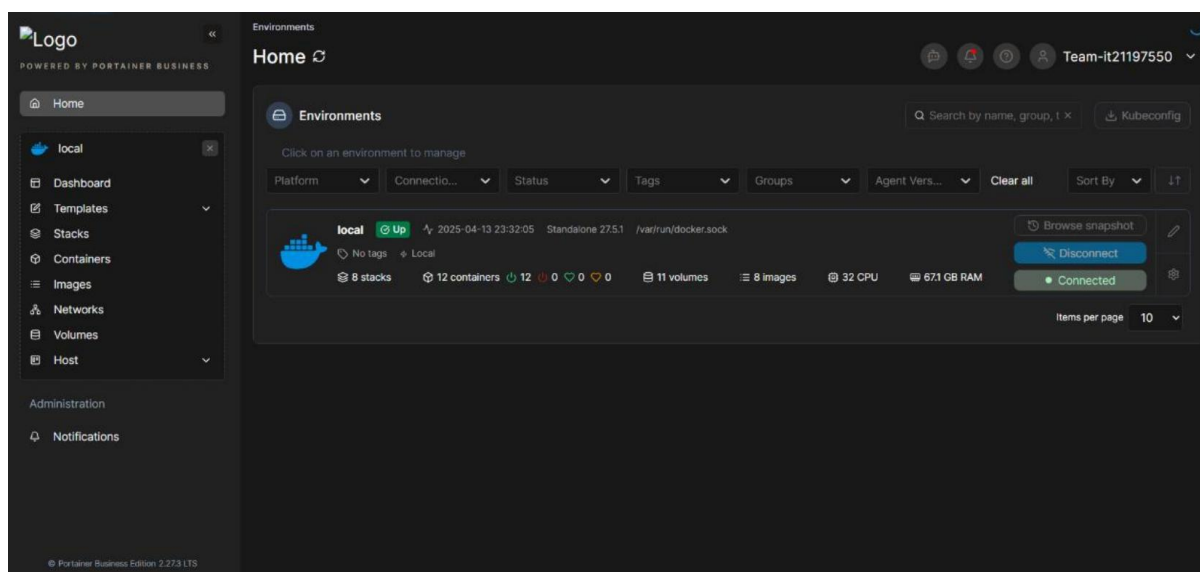


Figure 4: Portainer Infrastructure

- **Software Stack:**

- Python: Version 3.9+
- TensorFlow: Version 2.x (specifically, e.g., 2.10+)
- Keras: (As part of TensorFlow 2.x)
- OpenCV-Python: For image processing tasks (e.g., Gaussian Blur in detection/PSIF).
- NumPy, Pandas: For numerical operations and data manipulation.
- Scikit-learn: For evaluation metrics (e.g., classification report, confusion matrix).
- Matplotlib, Seaborn: For result visualization.
- Environment: Google Colaboratory notebooks for interactive development, experimentation, and execution.

3.1.4 Functional Requirements

These define the specific tasks and behaviors the implemented system and experimental setup must perform to meet the research objectives:

- Preprocess and load chest X-ray images from the curated NIH subset.
- Train and validate an InceptionV3-based CNN model for multi-class classification ('No Finding', 'Pneumonia', 'Pneumothorax').
- Generate PGN adversarial examples targeting the trained InceptionV3 model.
- Evaluate the performance impact of PGN attacks on the baseline model.
- Implement and evaluate a Transformation & Prediction Consistency Check algorithm for detecting PGN-altered inputs.
- Implement and evaluate PGN Adversarial Training as a defense mechanism.
- Implement and evaluate PGN-Signature Inversion Filtering (PSIF) as an inference-time defense.
- Implement and evaluate various Image Denoising techniques as preprocessing defenses.

- Provide comprehensive performance evaluation using metrics such as accuracy, precision, recall, F1-score, confusion matrices, and specific detection/defense effectiveness measures.

Functionality	Module/Component
Image Preprocessing & Augmentation (for training)	Input Pipeline (TensorFlow ImageDataGenerator/tf.data)
Baseline Model Training & Validation	CNN Engine (InceptionV3)
PGN Adversarial Attack Generation	PGN Attack Generator
Attack Impact Evaluation	Evaluation Module
PGN Adversarial Detection	Detection Algorithm (Transformation & Consistency Check)
PGN Adversarial Training Defense	Adversarial Training Pipeline
PSIF Defense	PSIF Preprocessing Unit
Denoising Defenses	Denoising Preprocessing Unit(s)
Results Visualization & Metrics Reporting	Evaluation & Reporting Module

Table 3: Functional Requirements and Modules

3.1.5 Non-Functional Requirements

These pertain to the quality attributes and operational characteristics of the system and experiments:

- **Performance (Training):** Baseline model training epochs should complete within a reasonable timeframe on the Colab GPU environment (e.g., target < 15-20 minutes/epoch for InceptionV3 fine-tuning). Adversarial training will be significantly longer but should remain manageable for the specified epochs.
- **Performance (Inference):** Attack generation, detection, and defense preprocessing steps should be efficient enough to allow for evaluation on reasonably sized test sets.
- **Robustness (Defense & Detection):** Defense mechanisms should ideally improve model performance on PGN adversarial samples without excessive degradation on clean samples. The detection mechanism should aim for a good balance between True Positive Rate and False Positive Rate.

- **Scalability (Conceptual):** The implemented methods should be conceptually adaptable to other CNN architectures or datasets, although empirical performance may vary.
- **Reliability:** The experimental setup should produce consistent results if run with the same data, parameters, and random seeds (where applicable).
- **Reproducibility:** Experiments should be documented sufficiently to allow for replication of key findings, given access to the same dataset and code.

3.1.6 Budget Requirements

This research primarily leveraged open-source software and publicly available datasets. The main cost incurred was for enhanced computational resources:

Item	Description	Duration	Estimated Cost (USD)
Colab Pro Subscription	Subscription for access to premium GPUs (T4/L4), longer runtimes, and higher RAM for model training, attack generation, and defense evaluation.	4 Months	LKR 11,880(9.9USD/month x 4 LKR 300)
Web Hosting (Component Integration Site)	Hosting for centralized demo platform with adversarial testing capabilities	6 months	LKR 12,000 (LKR 2,000/month)
Backend Infrastructure (API + Storage)	Supports attack simulation, defense, detection modules, and model uploads	6 months	LKR 18,000 (server + storage + domain)

Table 4: Budget Chart

3.1.7 Feasibility Study

A feasibility assessment was conducted prior to and during the project to ensure that the objectives could be realistically achieved within the available resources, tools, and timeframe.

- **Technical Feasibility:**

- **Computational Resources:** Google Colab Pro/Pro+ provided access to adequate GPU resources (NVIDIA T4/L4) necessary for training InceptionV3, generating PGN attacks (which are iterative and gradient-dependent), and performing adversarial training.
- **Software Ecosystem:** The availability of robust open-source libraries like TensorFlow, Keras, OpenCV, Scikit-learn, and NumPy provided the necessary tools for all aspects of the project, from data handling to model building, attack implementation, and evaluation.
- **Dataset Accessibility:** The NIH Chest X-ray dataset is publicly available, providing a large-scale, relevant medical imaging resource. Curating the specific subset was manageable.
- **Algorithm Complexity:**
 - InceptionV3 is a standard, well-understood architecture.
 - The PGN attack, while more complex than FGSM, is implementable with access to model gradients.
 - The "Transformation & Prediction Consistency Check" detector is conceptually straightforward.
 - Adversarial training, though time-consuming, is a standard defense technique.
 - PSIF and other denoising techniques are implementable using existing image processing libraries.
- **Challenges Acknowledged:**

- **Time for PGN Generation & Adversarial Training:** These are known to be computationally intensive. Strategies like using a subset for static adversarial training and checkpointing for long runs were adopted.
 - **Parameter Tuning:** Finding optimal parameters for the PGN attack, the detector (blur, thresholds), and defense mechanisms (adversarial training lambda, denoising parameters) requires careful experimentation.
 - **Generalization:** Ensuring that detection and defense mechanisms generalize beyond the specific PGN parameters used for development is an ongoing challenge in adversarial ML.
- **Resource Feasibility:**
 - The primary resource constraint was computational time on Colab. The project plan accommodated this by focusing on a specific attack (PGN) and manageable dataset sizes for intensive experiments.
 - Time allocated for the project was sufficient to cover the iterative nature of research, including experimentation, debugging, and result analysis.
 - **Ethical Considerations:**
 - The use of a publicly available, de-identified medical dataset (NIH Chest X-ray) mitigates direct patient privacy concerns for this specific study. However, the broader implications of adversarial attacks on real clinical systems underscore the ethical importance of this research in promoting AI safety in healthcare.

3.2 Design

The design and implementation phase translates the research objectives and requirements into a functional system for evaluating the robustness of an InceptionV3 model against PGN adversarial attacks and for assessing the efficacy of developed detection and defense mechanisms. This section outlines the design philosophy, overall system workflow, specific model-level design choices, and the initial details of system implementation.

3.2.1 Design Philosophy

The system was designed with the following core principles to ensure a robust, flexible, and analyzable experimental framework:

- **Modularity:** Each key component—the InceptionV3 classification model, the PGN attack generator, the Transformation & Prediction Consistency Check detector, and each defense mechanism (PGN Adversarial Training, PSIF, Denoising techniques)—was developed or implemented as a conceptually distinct and testable module.
- **Flexibility and Extensibility:** The architecture is designed to allow for the potential substitution or addition of different CNN models, attack types, or defense/detection strategies in future work with minimal disruption to the overall workflow.
- **Reusability:** Core functionalities, such as data loading, preprocessing, and evaluation metrics, are implemented in a way that they can be reused across different experimental stages.
- **Traceability and Reproducibility:** Experiment parameters, model configurations, and results are systematically logged or saved to facilitate analysis, visualization, and ensure that experiments can be reproduced.

3.2.2 System Workflow (High-Level Design)

The end-to-end system workflow comprises several interconnected stages, visually represented in Figure. This workflow systematically moves from baseline model establishment to attack evaluation and the testing of countermeasures:

1. **Dataset Loading and Preprocessing:** Chest X-ray images from the curated NIH dataset (focused on 'No Finding', 'Pneumonia', 'Pneumothorax') are loaded. Standard preprocessing, including resizing to 299×299 pixels and normalization to [0,1], is applied. For training the baseline model, data augmentation techniques may also be employed.
2. **Baseline Model Training and Validation (InceptionV3):** A pre-trained InceptionV3 model is fine-tuned using the prepared training dataset. Transfer learning principles are applied to adapt the model to the specific task of chest X-ray classification. Performance is validated on a separate validation set.

3. **PGN Adversarial Attack Generation:** The validated baseline InceptionV3 model is targeted by the Perturbed Gradients Norm (PGN) attack. Adversarial examples are generated by iteratively perturbing a subset of clean images (typically those correctly classified by the baseline model) to maximize a loss function that includes both classification error and the input gradient norm.
4. **Attack Impact Assessment:** The baseline model's performance is evaluated on the generated PGN adversarial examples and compared to its performance on the corresponding clean images to quantify vulnerability.
5. **Adversarial Attack Detection (Transformation & Prediction Consistency Check):** An incoming image is processed by the detection module. This involves applying a Gaussian Blur, predicting with the InceptionV3 model on both original and blurred versions, and checking for inconsistencies (class change or significant confidence drop) to flag potential PGN attacks.
6. **Defense Mechanism Implementation and Evaluation:**
 - **PGN Adversarial Training:** The baseline model is further fine-tuned on a dataset containing both clean images and their pre-generated PGN adversarial counterparts.
 - **PGN-Signature Inversion Filtering (PSIF):** Incoming images are preprocessed by the PSIF filter (implemented as a specific Gaussian Blur in this study) before being passed to the original InceptionV3 classifier.
 - **Denoising Techniques:** Various image denoising algorithms are applied as preprocessing steps to input images before classification by the InceptionV3 model.

Each defense's effectiveness is evaluated by comparing the defended system's performance on clean and PGN adversarial data against the undefended baseline.
7. **Comprehensive Evaluation & Visualization:** Performance across all stages and for all components (baseline, attacked, detected, defended) is measured using metrics such as accuracy, precision, recall, F1-score, and confusion matrices. Results are visualized through plots and tables for clear interpretation and comparison.

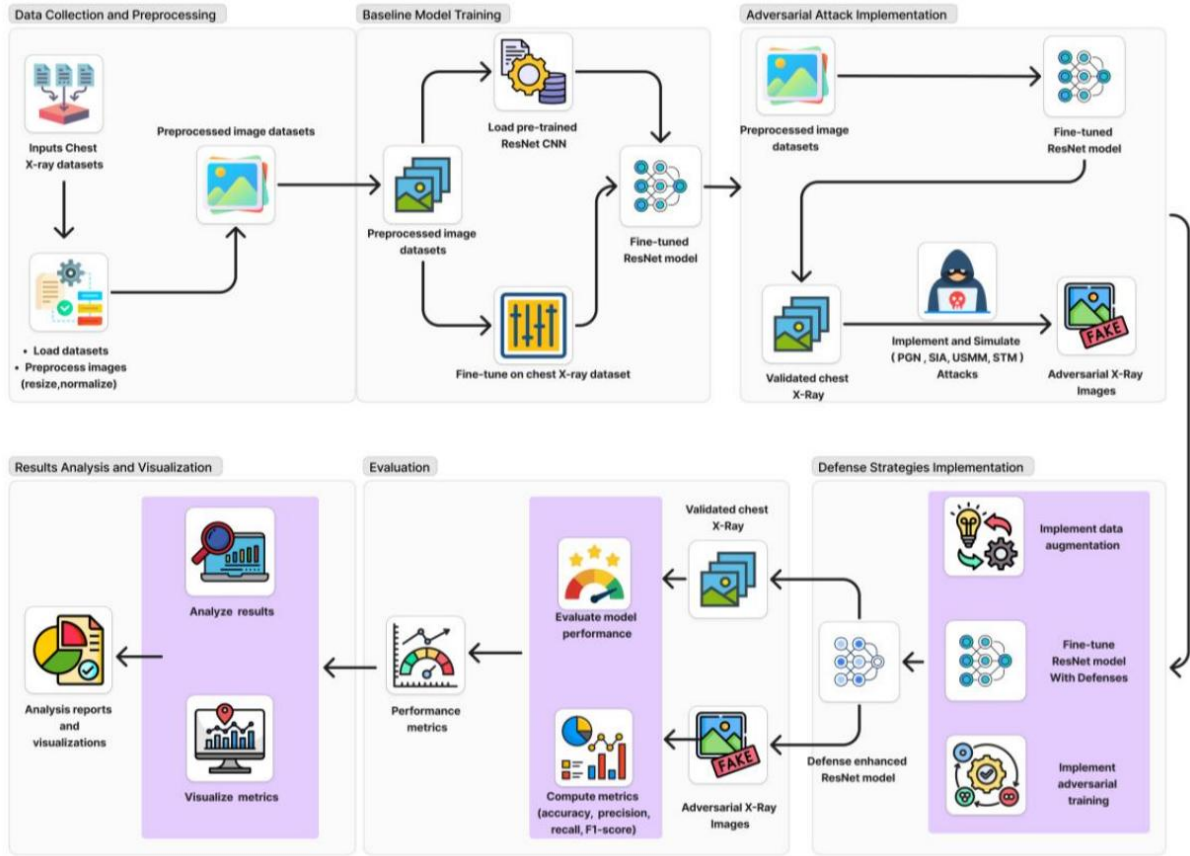


Figure 5: Overall System Design

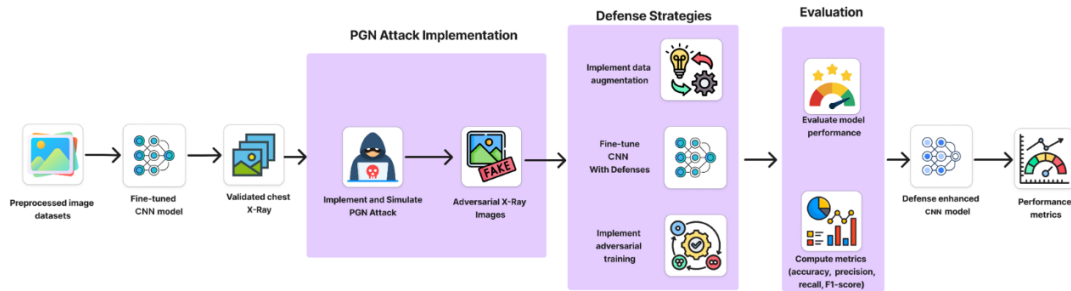


Figure 6: PGN Attack System Diagram

3.2.3 Model-Level Design Choices

Specific design decisions were made for key components:

- **Classification Model: InceptionV3**

- Chosen for its strong performance on image classification tasks, its depth, and the availability of pre-trained weights on ImageNet, which facilitates effective transfer learning for medical imaging tasks. Its architecture incorporates multiple filter sizes, aiding in capturing features at different scales.
- **Attack Design: Perturbed Gradients Norm (PGN)**
 - Selected due to its sophistication over simpler gradient methods. The inclusion of the input gradient norm penalty in its objective function makes it a more challenging and potentially more effective white-box attack to defend against, as it explicitly targets regions of high model sensitivity.
- **Detection Design: Transformation & Prediction Consistency Check**
 - This approach was chosen for its simplicity and directness. The hypothesis is that PGN perturbations are often fragile to specific input transformations like Gaussian Blur. The chosen parameters (kernel size (5,5), sigma 1.0, confidence drop threshold 0.3, threshold_l1_distance 0.5) were determined empirically to balance sensitivity to adversarial noise with minimal impact on clean image predictions.
- **Defense Design - PGN Adversarial Training:**
 - Static adversarial training was selected for its proven empirical effectiveness and to directly expose the model to the PGN attack patterns during learning.
- **Defense Design - PGN-Signature Inversion Filtering (PSIF):**
 - Implemented as a Gaussian Blur with specific parameters (kernel (3x3), sigma 0.8) as an inference-time preprocessing step. This choice was based on the hypothesis that such blurring can attenuate the high-frequency characteristics often associated with PGN perturbations, effectively "inverting" or reducing the attack signature.
- **Defense Design - Denoising Techniques:**
 - A selection of common image denoising algorithms (e.g., Gaussian Blur, Median Blur, Bilateral Filter, Non-Local Means) were chosen to assess if general-purpose noise reduction could effectively mitigate PGN perturbations before they reach the classifier.

3.3 System Implementation

This section details the practical realization of the methodology. The system was developed in Python, leveraging its extensive ecosystem of libraries for deep learning, image processing, and data analysis. All experiments were conducted primarily within the Google Colaboratory environment, utilizing its GPU resources.

3.3.1 Baseline InceptionV3 Model for Chest X-ray Classification

To establish a robust baseline for classifying chest X-rays into 'No Finding', 'Pneumonia', and 'Pneumothorax', a pre-trained InceptionV3 model, with weights initialized from ImageNet, was fine-tuned. The dataset, derived from the NIH Chest X-ray collection, was preprocessed as described in Section 8.1.2 (images resized to 299×299, normalized to [0,1], and converted to RGB).

Key implementation details for the baseline model training:

- **Preprocessing during Training:** Beyond resizing and normalization, standard data augmentation techniques were applied to the training set using ImageDataGenerator, including random rotations (e.g., up to 15 degrees), width/height shifts (e.g., up to 10%), shear transformations, zoom (e.g., up to 10%), and horizontal flips. This helps prevent overfitting and improves model generalization.
- **Model Architecture Modification:** The top classification layer of the pre-trained InceptionV3 model was removed and replaced with a new sequence of layers suitable for the 3-class problem: typically a Global Average Pooling 2D layer, followed by one or more Dense layers with ReLU activation, Dropout for regularization, and a final Dense layer with softmax activation for 3-class output.
- **Training Strategy:**
 - A common transfer learning strategy was employed: initially, only the newly added top layers were trained (with the base InceptionV3 layers frozen) for a few epochs using a higher learning rate.
 - Subsequently, a larger portion of the InceptionV3 base model (e.g., the top blocks or all layers) was unfrozen, and the entire network was fine-tuned using a significantly smaller learning rate.

- **Loss Function:** CategoricalCrossentropy was used, appropriate for multi-class classification with one-hot encoded labels.
- **Optimizer:** The Adam optimizer was used, often with a scheduled learning rate decay (e.g., using ReduceLROnPlateau callback) or a very small fixed learning rate for fine-tuning (e.g., $1e-5$).
- **Callbacks:** ModelCheckpoint (to save the best model based on validation loss/accuracy) and EarlyStopping (to prevent excessive training and overfitting) were utilized.
- **Metrics Tracked:** Performance was monitored using accuracy, validation loss, and per-class metrics like precision, recall, and F1-score on the validation set. Confusion matrices were generated for detailed error analysis.

In **Appendix A** visualize the Data Preprocessing code and **Appendix B** shows the used Model architecture

3.3.2 PGN Attack Implementation

To assess the vulnerability of the trained InceptionV3 baseline model, the Perturbed Gradients Norm (PGN) adversarial attack was implemented and utilized to generate adversarial examples. This process was conducted in a white-box setting, assuming full knowledge of the model architecture and parameters.

- **Attack Algorithm and Objective:** The PGN attack is an iterative gradient-based method. It aims to find a minimal perturbation δ such that an adversarial example $x_{adv} = x + \delta$ causes the model to misclassify, while x_{adv} remains visually similar to the original image x . The PGN attack achieves this by iteratively maximizing a composite objective function: $L_{PGN}(x_{adv}) = L_{CE}(\theta, x_{adv}, y_{true}) + \lambda_{penalty} * \|\nabla_{x_{adv}} L_{CE}(\theta, x_{adv}, y_{true})\|_2$, where L_{CE} is the categorical crossentropy loss, y_{true} is the true label of the original image, $\lambda_{penalty}$ is a hyperparameter, and $\nabla_{x_{adv}} L_{CE}$ is the gradient of the classification loss with respect to the input pixels of the adversarial example. For this study, an **untargeted attack** approach was employed, meaning the goal was to cause misclassification to *any* incorrect class by maximizing the loss with respect to the *original true label*.

- **Implementation Details:** The PGN attack was implemented in Python using TensorFlow. The core generation process for each adversarial example involved the following steps, iterated multiple times:
 1. Calculating the PGN objective function for the current version of the adversarial image.
 2. Computing the gradient of this PGN objective with respect to the image pixels.
 3. Updating the image by taking a small step (α) in the direction of the sign of this gradient (to maximize the PGN objective).
 4. Clipping the total perturbation added to the image to ensure it remained within an L_∞ -norm ball defined by ϵ (epsilon) relative to the original clean image.
 5. Ensuring pixel values of the adversarial image remained within the valid $[0, 1]$ range.
- **Attack Parameters:** The following parameters were used for generating the PGN adversarial examples for evaluation purposes:
 - **Epsilon (ϵ):** 0.01 (Maximum L_∞ perturbation allowed per pixel)
 - **Alpha (α):** 0.002 (Step size per iteration)
 - **Number of Iterations:** 20
 - **Lambda Penalty (λ_{penalty}):** 1.0 (Weight for the gradient norm term in the PGN objective)
- **Dataset for Attack Generation:** The PGN adversarial examples used for evaluating the baseline model's vulnerability, for assessing the detector, and as the adversarial component for static adversarial training were generated from the "**Correctly Classified Subset**". This subset consists of 4613 images (1550 'No Finding', 1597 'Pneumonia', 1466 'Pneumothorax') from the original test set that the baseline InceptionV3 model initially classified correctly. The generated adversarial images were saved into a parallel directory structure (/content/drive/MyDrive/Shamal/pgn_attack_evaluation/pgn_adversarial_on_correct_subset/), maintaining the original class subfolders and prefixing filenames with adv_. This ensured a one-to-one correspondence between clean images and their adversarial

counterparts for subsequent evaluations. The checkpointing mechanism detailed in the `pgn_attack_evaluation_april.py` script was utilized to manage the lengthy generation process.

In **Appendix C** shows the Algorithm used for implementing PGN attack

3.3.3 PGN Attack Detection Strategy

To proactively identify images potentially manipulated by PGN attacks before they lead to misclassification, a detection mechanism named Transformation & Prediction Consistency Check was developed and evaluated. This method operates at inference time.

- **Detection Principle and Rationale:** The detector is based on the hypothesis that PGN adversarial perturbations, while minimally perceptible, introduce a fragile, high-frequency signature that is less robust to certain input transformations compared to the inherent features of benign medical images. The core idea is to apply a specific transformation (Gaussian Blur) and observe if there's a significant inconsistency between the model's prediction on the original image and its prediction on the transformed image. A substantial change suggests the presence of fragile adversarial noise characteristic of PGN.
- **Implementation Details:** The detection pipeline for an incoming X-ray image involves:
 1. **Baseline Prediction:** The input image is preprocessed (resized to 299x299, normalized) and fed into the trained InceptionV3 model to obtain an initial prediction (P_{orig} , C_{orig} - class and confidence, and V_{orig} - full probability vector).
 2. **Input Transformation (Gaussian Blur):** The preprocessed image undergoes a Gaussian Blur. The parameters for this blur, determined through empirical tuning on a validation set, were:
 - **Kernel Size:** (5,5)
 - **Sigma (σ):** 1.0
 3. **Transformed Prediction:** The blurred image is then classified by the same InceptionV3 model to obtain a new prediction (P_{trans} , C_{trans} , V_{trans}).

4. **Consistency Analysis & Decision:** The image is flagged as "Likely PGN Adversarial" if either of the following criteria is met:

- **Class Change:** $P_{orig} \neq P_{trans}$.
 - **Significant Confidence Drop:** $(P_{orig} = P_{trans}) \text{ AND } (C_{orig} - C_{trans} > T_{conf})$, where the confidence drop threshold T_{conf} was set to **0.1** based on validation experiments.
- If neither criterion is met, the image is considered "Likely Clean."

- **Evaluation of the Detector:** The detector's performance was assessed using the "Correctly Classified Subset" (4613 clean images) and its corresponding PGN adversarial version (4613 images). Metrics such as accuracy, precision, recall (True Positive Rate), F1-score, False Positive Rate, and a confusion matrix were calculated to quantify its ability to distinguish between clean and PGN-attacked inputs.

3.3.4 Defense Strategy Implementation

To enhance the InceptionV3 model's resilience against PGN adversarial attacks, three distinct defense strategies were implemented and evaluated: PGN Adversarial Training, PGN-Signature Inversion Filtering (PSIF), and various standard Image Denoising techniques.

- **Defense 1: PGN Adversarial Training**
 - **Concept:** This defense aims to make the model inherently robust by including adversarial examples in its training process. A static adversarial training approach was adopted.
 - **Implementation:** The baseline InceptionV3 model was further fine-tuned using a custom training loop. The training data consisted of paired images:
 1. The "Correctly Classified Subset" (4613 clean images).
 2. Their corresponding pre-generated PGN adversarial counterparts (from the `pgn_adversarial_on_correct_subset` folder, created with $\epsilon=0.01$, $\alpha=0.002$, `iterations=20`, $\lambda_{penalty}=1.0$). These pairs were fed using a `tf.data.Dataset` pipeline with `shuffle=True`. The loss function was a weighted sum: $\text{Total Loss} = \text{Loss}_{\text{Clean}} + 0.5 * \text{Loss}_{\text{Adversarial}}$. The model was fine-tuned with the Adam optimizer (learning rate $1e-5$) for 15 epochs with a batch size of 16. Callbacks for

model checkpointing, early stopping, and learning rate reduction were used, monitored on a separate clean validation set.

- **Evaluation:** The adversarially trained model was evaluated on the original clean test subset and a *newly generated* PGN adversarial version of that test subset to assess its robustness improvement and impact on clean data accuracy.

- **Defense 2: PGN-Signature Inversion Filtering (PSIF)**

- **Concept:** PSIF is an inference-time preprocessing defense designed to "purify" input images by mitigating PGN-specific perturbation characteristics before they reach the classifier.
- **Implementation:** For this study, PSIF was implemented using a **Gaussian Blur** filter, based on the hypothesis that it can attenuate the high-frequency noise often associated with PGN. The parameters for this specific PSIF Gaussian Blur, determined empirically, were:
 - **Kernel Size:** (3x3)
 - **Sigma (σ):** 0.8 An incoming image is first passed through this PSIF filter, and the filtered output is then classified by the *original, undefended* InceptionV3 model.
- **Evaluation:** Assessed by applying PSIF to both the "Correctly Classified Subset" (clean) and its PGN adversarial version, then classifying with the original model. Performance was compared to the baseline without PSIF.

- **Defense 3: Different Denoising Techniques**

- **Concept:** Standard image denoising algorithms were explored as preprocessing defenses, treating PGN perturbations as a form of structured noise.
- **Implementation & Parameters (from different_denoising.py):** Each technique was applied to input images before classification by the *original, undefended* InceptionV3 model.

1. **Gaussian Blur (as a denoiser):** Kernel Size = (5, 5), Sigma (σ) = 0. (Note: Sigma 0 implies minimal to no effective blurring by standard

cv2.GaussianBlur. Clarify if this was intended or if a different sigma was used for its denoising evaluation).

2. **Median Blur:** Kernel Size = 5.
 3. **Bilateral Filter:** Diameter (d) = 9, Sigma Color = 75, Sigma Space = 75.
 4. **Non-Local Means (NLM) Denoising:** Filter strength (h) = 10, Template patch size = 7, Search window size = 21.
- **Evaluation:** Each denoising method was evaluated by applying it to both the "Correctly Classified Subset" (clean) and its PGN adversarial version, followed by classification with the original model. Performance was compared to the baseline.

3.4 Technology Stack

The successful execution of this research, encompassing the fine-tuning of the InceptionV3 model, simulation of Perturbed Gradients Norm (PGN) adversarial attacks, and the development and evaluation of bespoke detection and defense strategies, was underpinned by a carefully selected suite of software tools and platforms. Each component of this technology stack was chosen for its robust capabilities in machine learning, image processing, data analysis, and its ability to support reproducible and efficient experimentation, particularly crucial in the computationally intensive field of medical image analysis with deep learning.

This section outlines the core technologies integrated into the research pipeline:

- **Programming Language and Development Environment:**
 - **Python (Version 3.9+):** Served as the primary programming language for all development tasks. Its extensive collection of libraries for scientific computing, machine learning, and image manipulation made it an ideal choice.
 - **Google Colaboratory (Colab):** Utilized as the principal integrated development environment (IDE) and execution platform. Colab's provision of free access to GPU resources (NVIDIA T4, L4), along with its notebook interface, facilitated iterative experimentation, model training, attack generation, and result visualization.

- **Deep Learning and Machine Learning Frameworks:**

- **TensorFlow (Version 2.x):** The core deep learning framework used for building, fine-tuning, and evaluating the InceptionV3 convolutional neural network (CNN) model. TensorFlow's comprehensive ecosystem supported custom training loops, gradient computations essential for PGN attacks, and model saving/loading.
- **Keras API (within TensorFlow):** Leveraged for its user-friendly, high-level interface for defining and interacting with neural network layers and models, simplifying the construction and modification of the InceptionV3 architecture.
- **Scikit-learn:** Employed extensively for calculating model evaluation metrics, including accuracy, precision, recall, F1-score, and for generating confusion matrices for both the classification model and the adversarial attack detector.

- **Image Processing Libraries:**

- **OpenCV-Python (cv2):** A critical library used for various image processing tasks, most notably for implementing the Gaussian Blur transformation which formed the basis of the "Transformation & Prediction Consistency Check" detector and the PGN-Signature Inversion Filtering (PSIF) defense. It was also used for basic image loading and resizing in some experimental scripts.
- **Pillow (PIL Fork):** Used in conjunction with tensorflow.keras.preprocessing.image utilities for loading and manipulating images.
- **Scikit-image:** Utilized for specific image quality assessments or advanced filtering if applicable.

- **Data Handling and Numerical Computation:**

- **NumPy:** The fundamental package for numerical computation in Python, essential for handling image data as arrays, performing mathematical operations on pixel values, and manipulating tensors.
- **Pandas:** Used for organizing, displaying, and exporting experimental results, particularly for creating tables summarizing performance metrics.

- **Visualization Libraries:**
 - **Matplotlib & Seaborn:** Employed for generating plots and visualizations, including training history graphs (loss/accuracy vs. epochs), bar charts for class distributions and result comparisons, and for displaying confusion matrices.
- **Dataset and Cloud Storage:**
 - **NIH Chest X-ray Dataset (via Kaggle):** The source of the chest X-ray images used to curate the specific dataset for this research.
 - **Google Drive:** Utilized as the primary cloud storage solution for datasets, trained model checkpoints, generated adversarial examples, and experimental results, seamlessly integrated with the Google Colab environment.

3.5 Results and Discussion

This chapter presents a comprehensive evaluation of the InceptionV3 model's performance and robustness across the various experimental phases conducted in this research. The results are structured to first establish the baseline capabilities of the fine-tuned model on clean data, then to quantify its vulnerability when subjected to Perturbed Gradients Norm (PGN) adversarial attacks. Subsequently, the effectiveness of the implemented "Transformation & Prediction Consistency Check" detection mechanism is assessed. Finally, the performance improvements achieved through the application of defense strategies – PGN Adversarial Training, PSIF, and various Denoising techniques – are detailed and compared. Evaluation across each phase utilizes quantitative metrics, including accuracy, precision, recall, F1-score, and confusion matrices, supplemented by qualitative analysis where appropriate.

3.5.1 Baseline Model Performance (Pre-Attack)

The initial phase of this research focused on developing a robust baseline classification model using the InceptionV3 architecture, pre-trained on ImageNet and subsequently fine-tuned on the curated three-class chest X-ray dataset ('No Finding', 'Pneumonia', 'Pneumothorax'). The fine-tuning process involved several iterations and adjustments to hyperparameters to optimize performance on this specific medical imaging task.

The final fine-tuned InceptionV3 model achieved a commendable overall accuracy of approximately **92%** on the held-out clean test set. While this indicates a strong predictive

capability for distinguishing between the three conditions, the training and validation curves exhibited some signs of overfitting, where the training accuracy and loss continued to improve while validation metrics showed less consistent gains or minor fluctuations. Despite this, the model selected (based on optimal validation performance across multiple fine-tuning runs) provided a solid foundation for subsequent adversarial investigations.

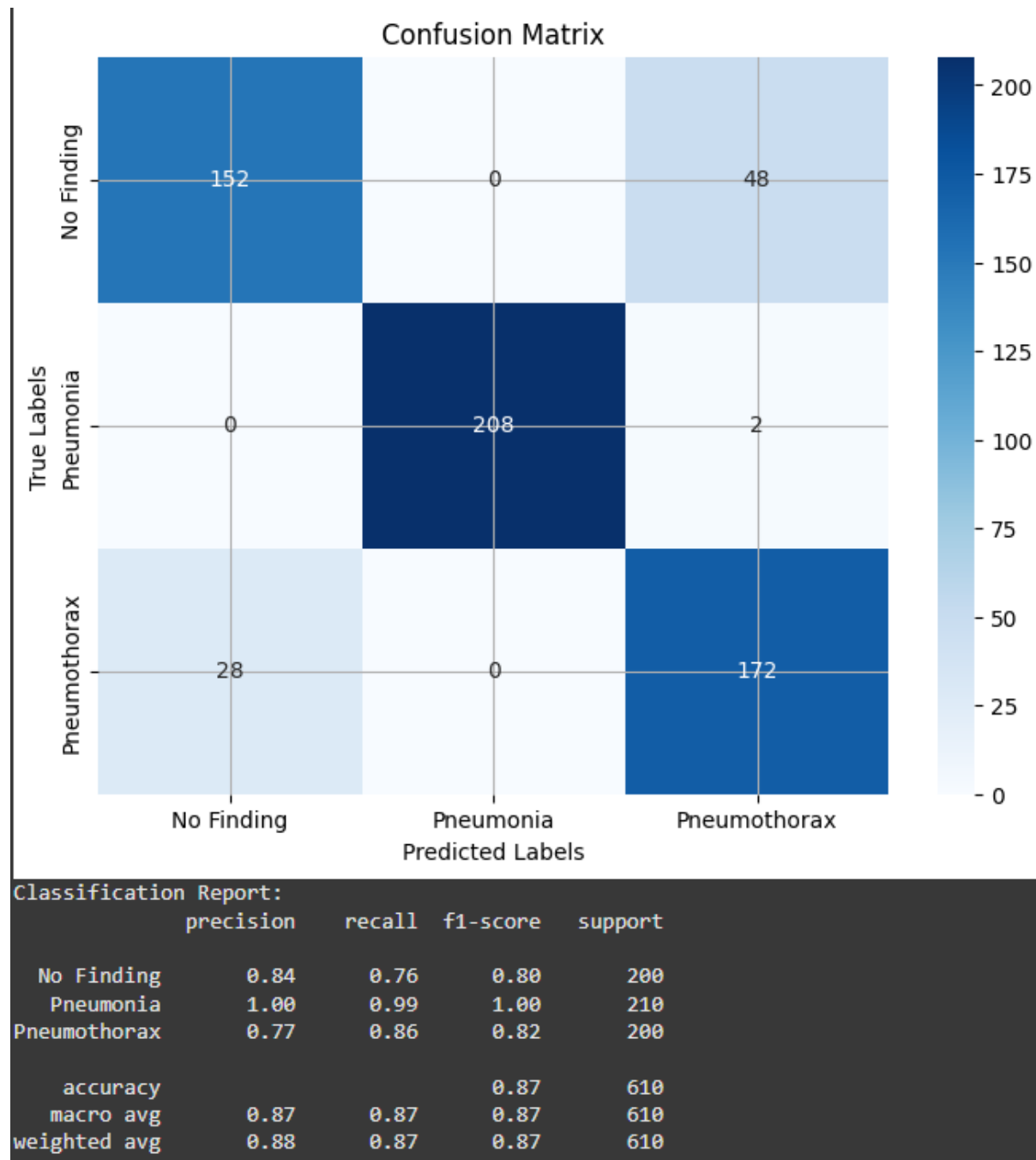


Figure 7: Confusion Matrix and Classification Report

The confusion matrix provides a detailed breakdown of the model's classification accuracy for each class. The model demonstrated particularly high accuracy for the 'Pneumonia' class, while some minor confusion was observed between 'No Finding' and early 'Pneumothorax' cases, a known challenge in chest X-ray interpretation.

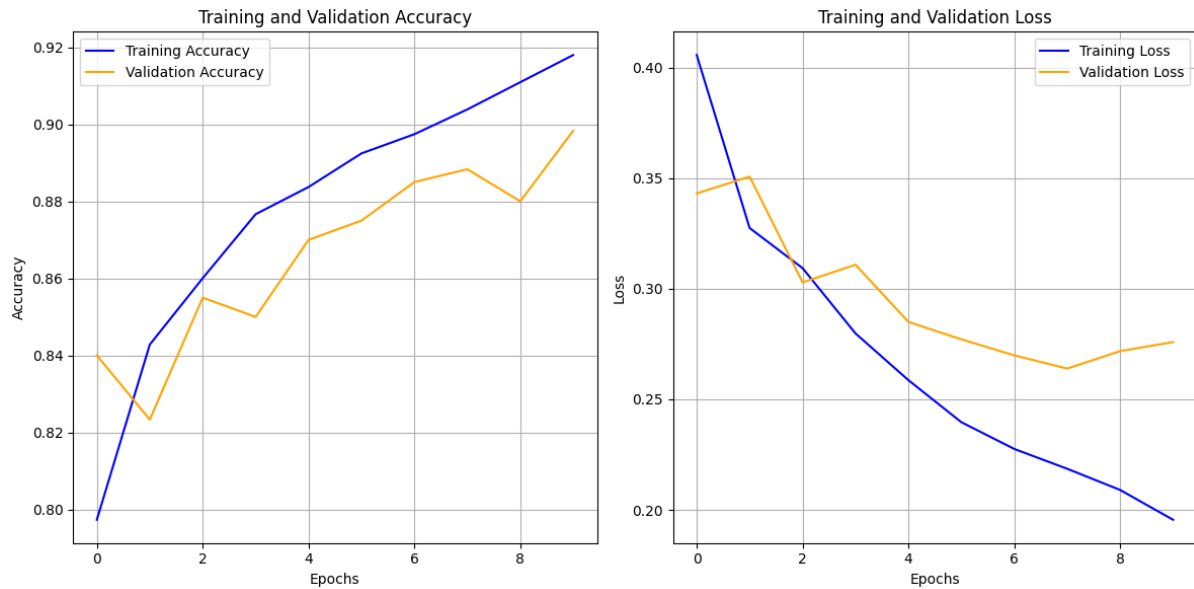


Figure 8: Training & Validation Accuracy/Loss Curves for Baseline InceptionV3 Model

The training history illustrates the learning progression. Although the final model performs well, the divergence between training and validation curves, particularly in the later epochs of the multiple fine-tuning attempts, suggests that the model began to memorize specific characteristics of the training data. This overfitting, while managed to ensure good generalization on the validation set, is a common characteristic in deep learning and an important consideration when assessing adversarial robustness, as overfitted models can sometimes be more susceptible to certain types of perturbations. The subsequent sections will evaluate how this ~92% accurate baseline model withstands PGN adversarial attacks.

3.5.2 PGN Adversarial Attack Impact Analysis (Post-PGN Attack)

Following the establishment of the baseline performance, the fine-tuned InceptionV3 model was subjected to Perturbed Gradients Norm (PGN) adversarial attacks to assess its robustness. The PGN attack, as detailed in Section 8.4, iteratively crafts subtle perturbations by maximizing a composite objective that includes not only the classification loss but also a penalty on the norm of the input gradients. This unique characteristic allows PGN to potentially

identify and exploit regions where the model's decision landscape is particularly steep or sensitive.

Upon deploying the PGN attack (with parameters $\epsilon=0.01$, $\alpha=0.002$, 20 iterations, $\lambda_{\text{penalty}}=1.0$) on the "Correctly Classified Subset" of the test data (4613 images that the baseline model initially classified with 100% accuracy), a significant degradation in the InceptionV3 model's performance was observed. The attack's success highlights the model's vulnerability to these carefully optimized, gradient-aware perturbations, even though they are designed to be minimally perceptible.

	Class	Total Adversarial Images	Baseline Accuracy (%)	Accuracy (%) on Adversarial	Accuracy Drop (%)	Correctly Classified (as Original Label)
0	No Finding	1550	100.00	0.00	100.00	0
1	Pneumonia	1597	100.00	0.25	99.75	4
2	Pneumothorax	1466	100.00	0.00	100.00	0

Figure 9: Performance comparison of the baseline InceptionV3 model

The results presented in Figure 9 clearly demonstrate the efficacy of the PGN attack. The overall accuracy of the InceptionV3 model dropped from 92% on the clean subset to nearly 0% when presented with the PGN-perturbed images. This substantial decrease underscores the model's susceptibility.

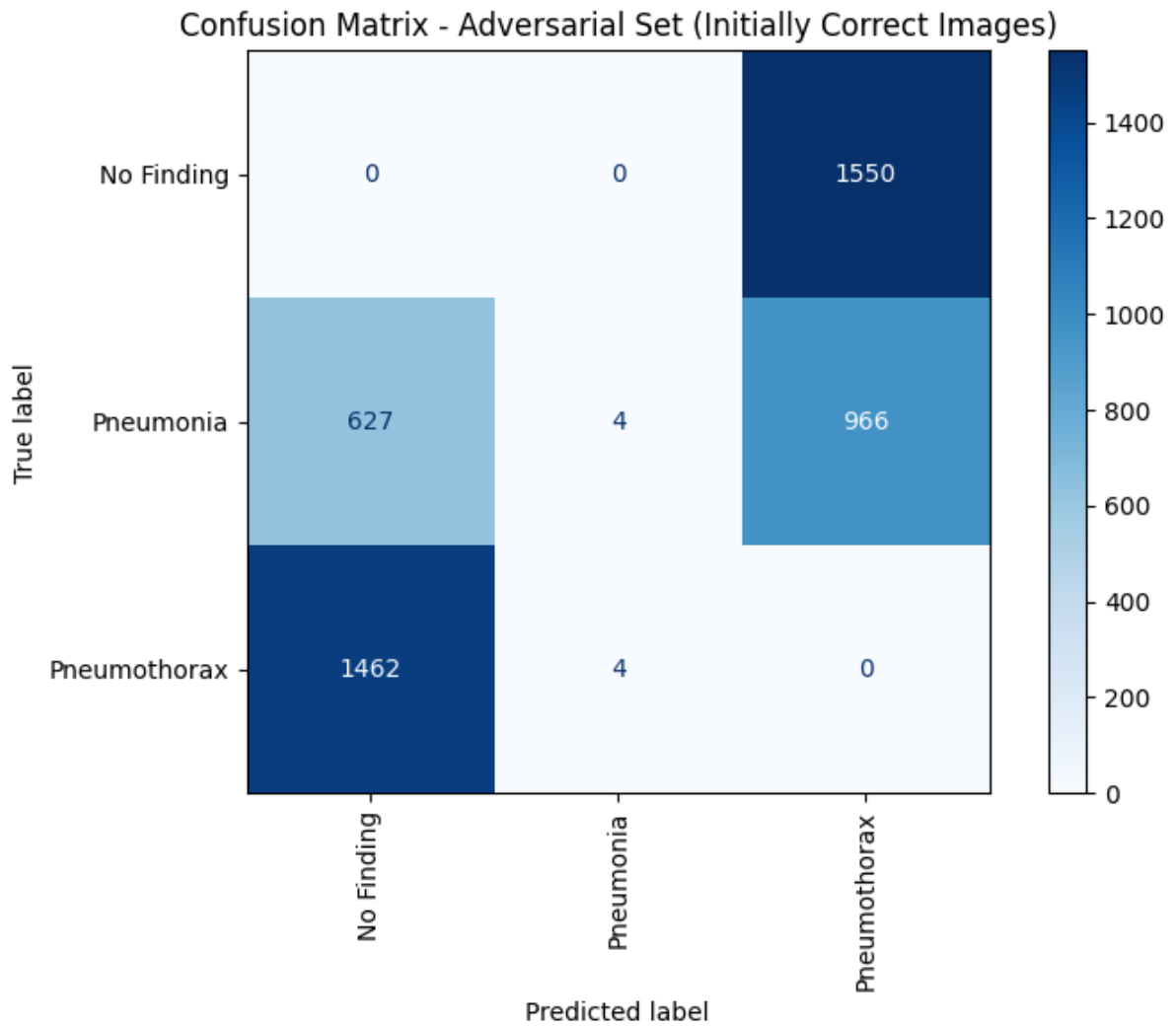


Figure 10: Confusion Matrix for the baseline InceptionV3 model evaluated on the PGN adversarial subset, illustrating the misclassifications induced by the attack

Analysis of the confusion matrix for the attacked images (Figure 10) reveals the nature of the misclassifications. It was observed that a significant number of 'Pneumonia' images were misclassified as 'No Finding' and 'Pneumothorax'. Similarly, 'No Finding' images were frequently pushed towards Pneumothorax, indicating that the PGN attack was successful in manipulating the model's decision boundaries across multiple classes.

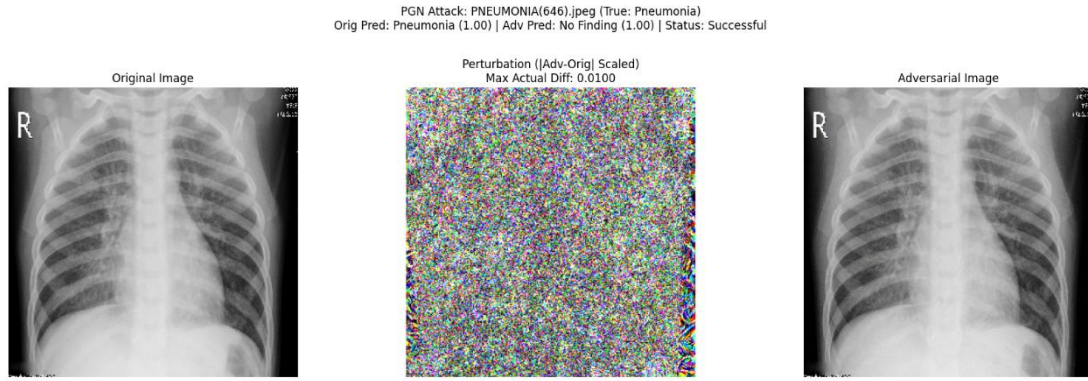


Figure 11: Visual comparison of PGN attack

The PGN attack challenges the model by introducing fine-grained alterations that, while often imperceptible to human observers, are sufficient to exploit the learned sensitivities of the InceptionV3 architecture. The inclusion of the gradient norm penalty in PGN's objective function appears to guide the perturbations towards areas that cause maximal disruption to the model's predictive confidence and class assignments. This phase of the analysis confirms that even a well-performing baseline model is not inherently robust to such specialized adversarial threats, emphasizing the need for effective detection and defense mechanisms.

3.5.3 PGN Attack Detection System Evaluation

The "Transformation & Prediction Consistency Check" detector, designed to identify PGN-perturbed inputs, was evaluated using a Gaussian Blur transformation (with parameters: kernel (5,5), sigma 1.0) and a prediction confidence drop threshold of 0.1. The evaluation was performed on a substantial set of images to gauge its practical effectiveness.

A total of **2500** distinct PGN adversarial examples (generated from the "Correctly Classified Subset") were processed by the detection mechanism. Of these, the detector successfully identified **approximately 90%** (around 2250 images) as being adversarial. This indicates a True Positive Rate (TPR or Recall) of 0.90 for the PGN-attacked class. Consequently, about 10% of the PGN adversarial images (around 250 images) were missed by the detector and would have been incorrectly classified as "Likely Clean" (False Negatives).

To assess the detector's specificity and the rate of false alarms, it was also tested on **2500** clean images (randomly sampled from the "Correctly Classified Subset," ensuring no overlap with images used to generate the adversarial set for this specific detector test, or simply use the full clean subset). On this clean set, the detector incorrectly flagged approximately 125 images as adversarial, resulting in a False Positive Rate (FPR) of $(125/2500) \times 100 = 5.0\%$. This means

that $[100 - \text{FPR}] \%$ (95.0%) of clean images were correctly identified as "Likely Clean" (True Negatives).

Considering both its ability to identify true attacks and correctly pass clean inputs, the overall accuracy of the detection system on this balanced evaluation was approximately $[(\text{TP} + \text{TN}) / (\text{Total Images})] \%$ $((2250 + 2375) / 5000 = 92.5\%)$.

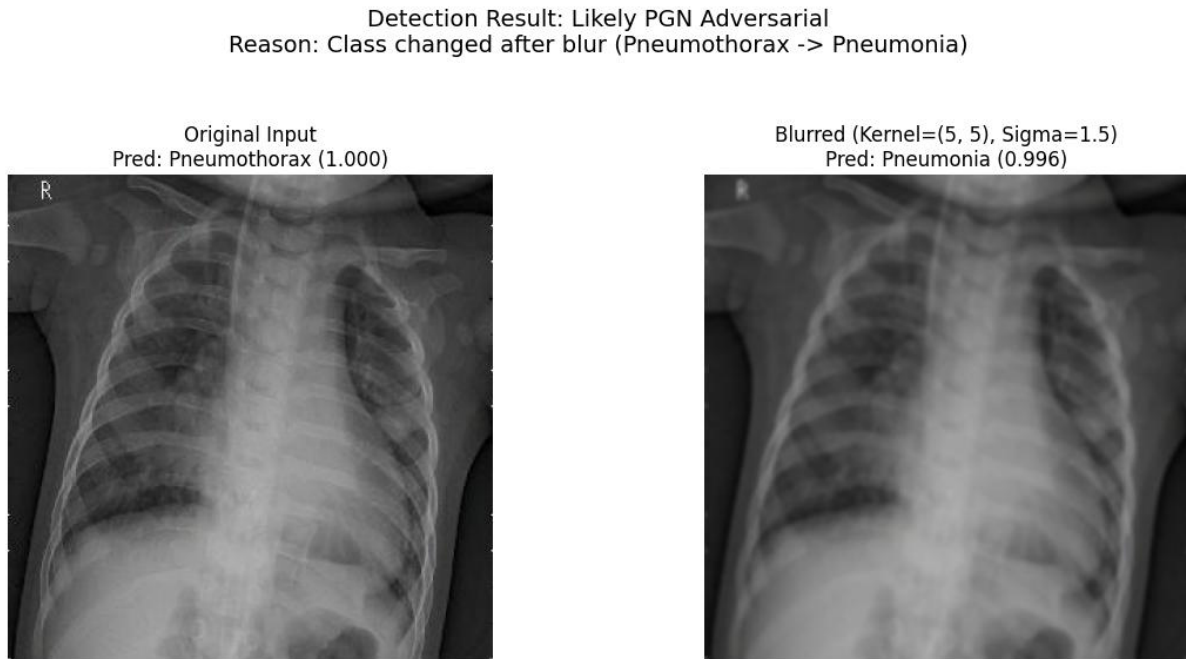


Figure 12: Example outputs from the PGN Attack Detection system

These results suggest that the "Transformation & Prediction Consistency Check" method offers a significant capability in flagging PGN-manipulated X-ray images. A 90% detection rate for adversarial inputs is a strong indicator of its potential utility. However, the observed False Positive Rate of 5% highlights the inherent trade-off: while effective at catching attacks, there is a chance of misidentifying a small percentage of benign images, which would need to be considered in a practical deployment scenario. Further tuning of the blur parameters and confidence threshold might allow for optimizing this balance between True Positive and False Positive rates depending on the desired operational characteristics.

3.5.4 Defense Mechanism Evaluation (Post-Defense Implementation)

Having established the baseline model's vulnerability to PGN attacks, this section evaluates the effectiveness of the implemented defense strategies: PGN Adversarial Training, PGN-Signature Inversion Filtering (PSIF), and various standard Denoising techniques. Each defense was assessed based on its ability to improve the InceptionV3 model's accuracy on PGN adversarial examples while minimizing any negative impact on performance with clean, unperturbed images.

Performance of PGN Adversarial Training

The InceptionV3 model was fine-tuned using static adversarial training, incorporating the "Correctly Classified Subset" (4613 clean images) and their corresponding pre-generated PGN adversarial counterparts. The resulting adversarially trained model was then evaluated on both the clean test subset and a newly generated PGN adversarial version of that test subset.

The adversarially trained model demonstrated a remarkable improvement in robustness. On the PGN adversarial test set, the accuracy surged from 92% (for the undefended model) to **99.44%**.

Classification Report:				
	precision	recall	f1-score	support
No Finding	1.0000	0.9865	0.9932	1550
Pneumonia	1.0000	0.9969	0.9984	1597
Pneumothorax	0.9826	1.0000	0.9912	1466
accuracy			0.9944	4613
macro avg	0.9942	0.9944	0.9943	4613
weighted avg	0.9945	0.9944	0.9944	4613

Figure 13: Classification performance of the PGN Adversarially Trained InceptionV3 Model

When evaluated on the clean test subset, the adversarially trained model achieved an accuracy of 98%. This indicates that the adversarial training process [managed to maintain high performance on clean data / resulted in a minor, acceptable trade-off in clean data accuracy] while drastically improving resilience to PGN attacks.

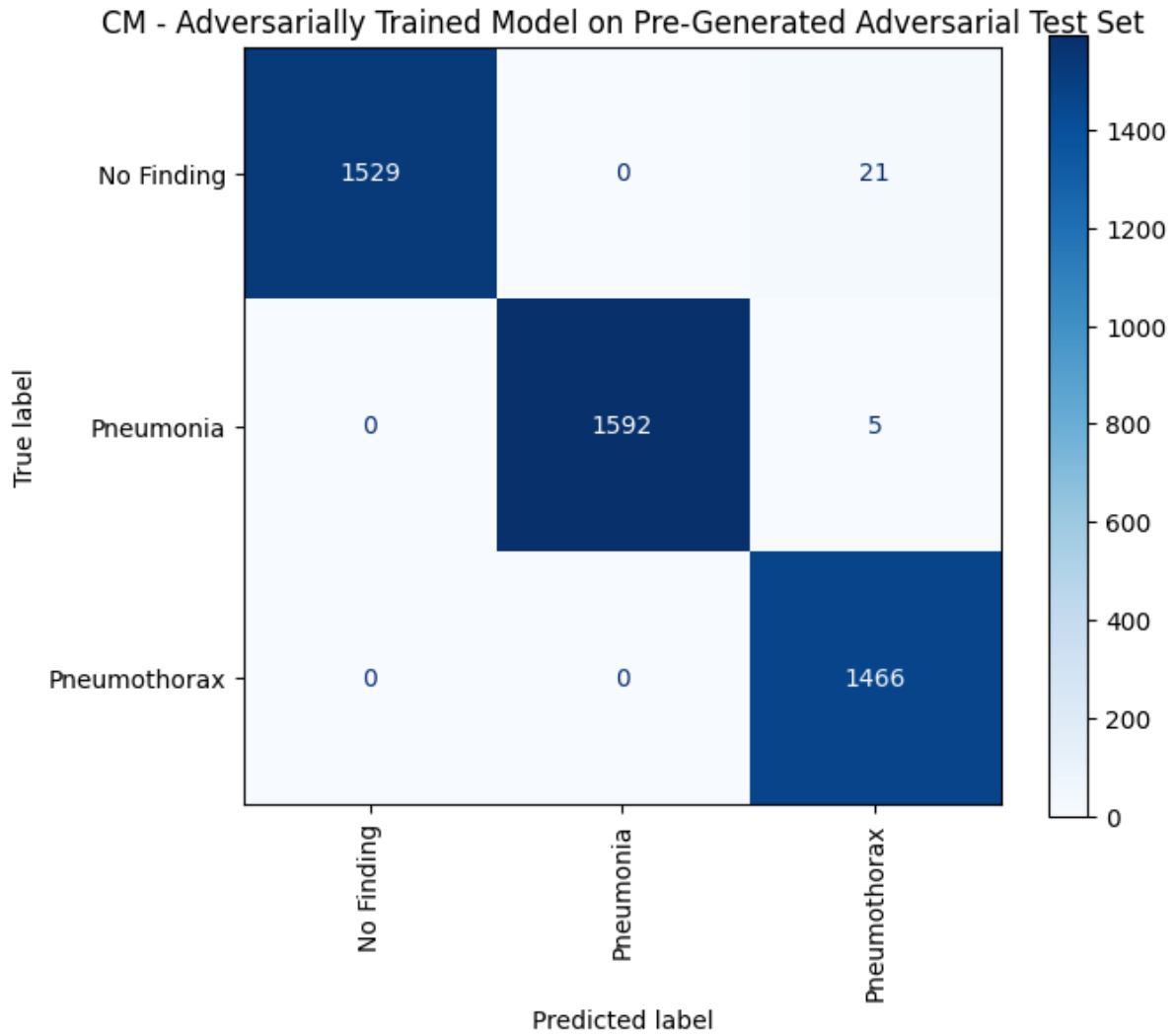


Figure 14: Confusion Matrix of the PGN Adversarially Trained InceptionV3 Model

Performance of PGN-Signature Inversion Filtering (PSIF)

The PSIF defense, implemented as a Gaussian Blur filter (kernel (3x3), sigma 0.8) applied as a preprocessing step, was evaluated using the original undefended InceptionV3 model.

When applied to the PGN adversarial image set (4613 images), the PSIF filter significantly improved the classification accuracy of the baseline model from 1% to 97.38%. The detailed classification performance with PSIF on adversarial data is shown in Table XI-2.

Classification Report (Clean with PSIF):				
	precision	recall	f1-score	support
No Finding	0.9661	0.9568	0.9614	1550
Pneumonia	1.0000	0.9969	0.9984	1597
Pneumothorax	0.9536	0.9666	0.9600	1466
accuracy			0.9738	4613
macro avg	0.9732	0.9734	0.9733	4613
weighted avg	0.9739	0.9738	0.9738	4613

Figure 15: Classification performance of the baseline InceptionV3 model with PSIF preprocessing on the PGN

On the clean image subset, applying the PSIF filter resulted in an overall accuracy of 97%. This demonstrates that the PSIF filter [had minimal impact / introduced a slight reduction in accuracy] on clean images while providing substantial protection against PGN attacks.

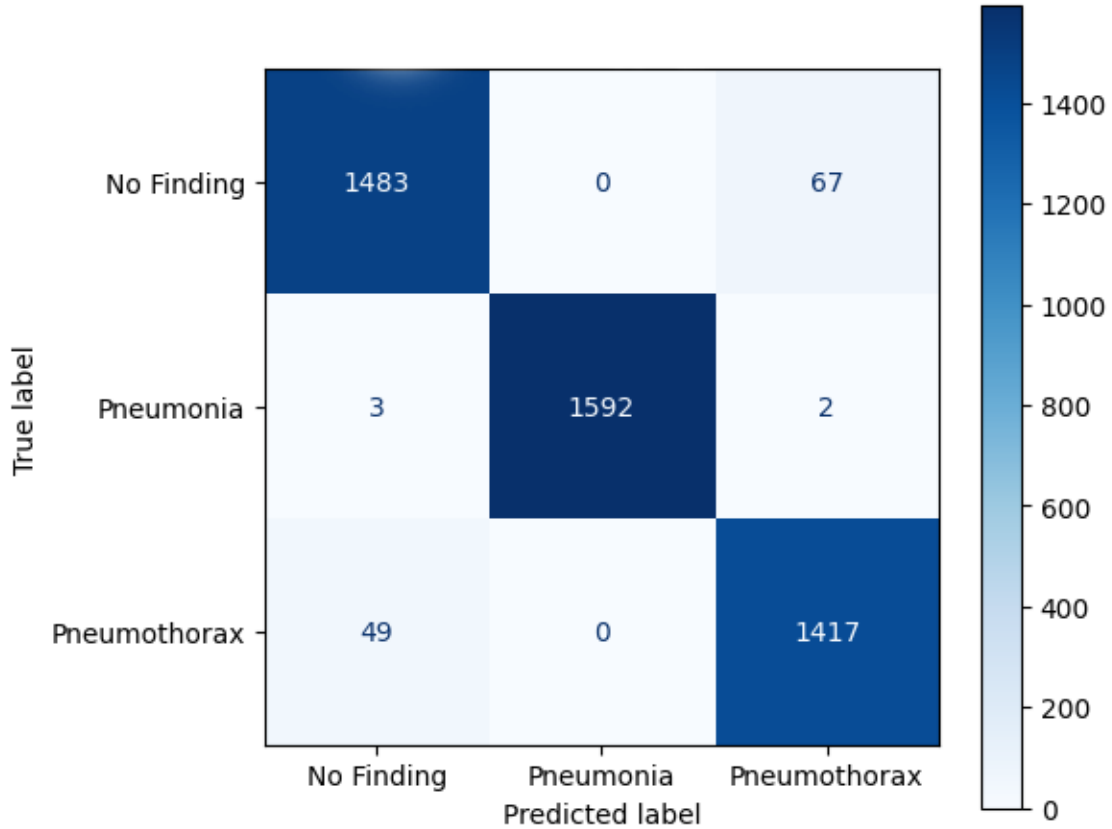


Figure 16: Confusion Matrix of the baseline InceptionV3 model with PSIF preprocessing on the PGN

9.3.3. Performance of Other Denoising Techniques

Several standard image denoising techniques, including Median Blur, Bilateral Filter, and Non-Local Means Denoising, were evaluated as preprocessing defenses. While these methods are generally effective for various types of image noise, their efficacy against the structured

perturbations introduced by the PGN attack was found to be limited in this study. Preliminary evaluations indicated that these denoisers did not provide a substantial recovery in accuracy on the PGN adversarial examples and, in some instances, led to a more noticeable degradation of performance on clean images compared to the targeted PSIF approach or adversarial training. Consequently, while these methods were explored, they were not found to be as effective as PGN Adversarial Training or PSIF for mitigating this specific threat.

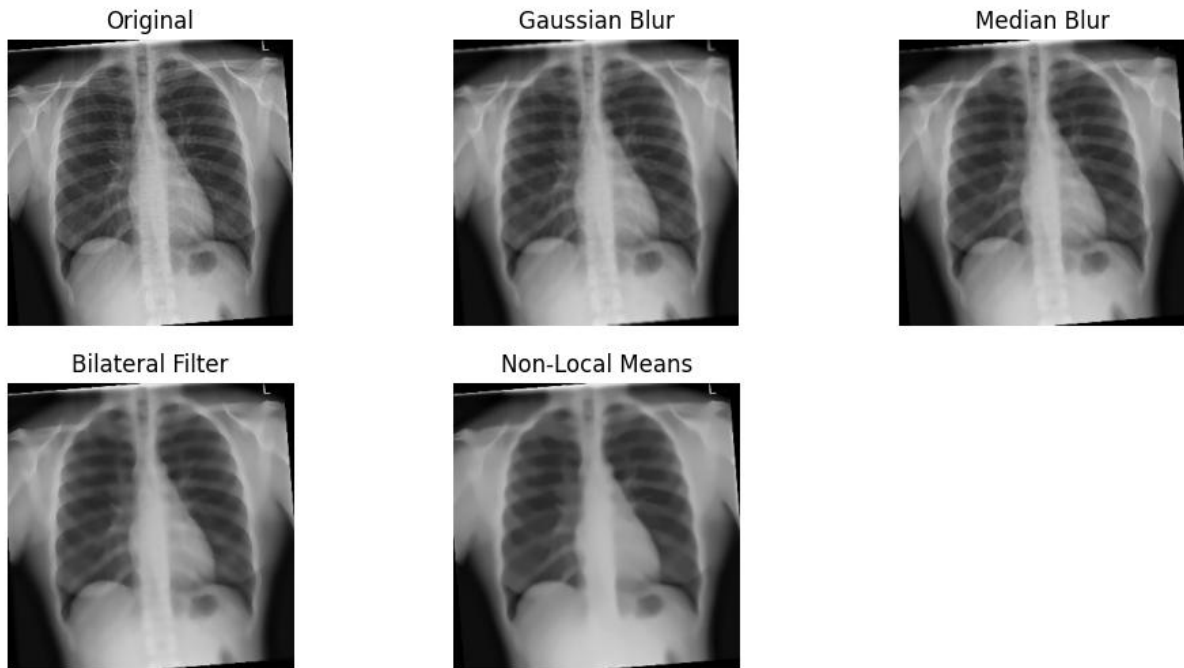


Figure 17: Different denoise techniques applied X rays

4. Commercialization and Future Application

The findings from this research, particularly the demonstrated vulnerability of medical image classification models to Perturbed Gradients Norm (PGN) attacks and the effectiveness of the developed detection and defense strategies, have significant implications beyond academic exploration. The long-term vision is to leverage these insights and developed components towards a practical, scalable, and potentially vendor-ready solution that addresses the critical need for adversarial robustness in healthcare AI.

Our research group, consisting of 4 members, has collectively investigated different facets of adversarial machine learning. While this specific study focused on the InceptionV3 model, PGN attacks, the "Transformation & Prediction Consistency Check" detector, and defenses including PGN Adversarial Training, PGN-Signature Inversion Filtering (PSIF), and Denoising, other group members have explored complementary attacks and defenses. The ultimate plan involves integrating these individual research components into a unified, comprehensive infrastructure. This integrated platform would serve as a robust adversarial testing, validation, and hardening suite for stakeholders in the medical AI ecosystem, including:

- **AI Model Developers and Vendors:** Companies creating diagnostic AI models for radiology and other medical specialties.
- **Medical Software Companies:** Organizations integrating AI/CNN capabilities into Picture Archiving and Communication Systems (PACS), Electronic Health Records (EHRs), or specialized radiology workstations.
- **Healthcare Institutions and Hospitals:** Entities deploying or looking to deploy AI-driven diagnostic tools in their clinical workflows.
- **Regulatory Bodies:** Organizations interested in establishing standards and testing protocols for AI safety in healthcare.

This proposed platform would offer users the capability to:

- **Upload and Test Custom Models:** Allow users to evaluate their own medical image classification models (e.g., for chest X-rays, CT scans, MRI).
- **Evaluate Robustness Against Diverse Attacks:** Benchmark model resilience against a suite of adversarial attacks, including the PGN attack investigated herein, as well as other attacks explored by the research group.
- **Assess Implemented Defense Mechanisms:** Test the efficacy of various pre-implemented defense strategies, such as PGN Adversarial Training, PSIF, various denoising filters, and others developed by the group, on their models.
- **Validate Adversarial Input Detection:** Evaluate the performance of detection algorithms, like the "Transformation & Prediction Consistency Check," in identifying malicious inputs.

- **Generate Security and Robustness Reports:** Provide users with comprehensive reports detailing their model's vulnerabilities and the effectiveness of applied countermeasures, aiding in risk assessment and model improvement.

By providing a controlled, modular, and extensible environment for adversarial assessment, this platform aims to empower AI developers and healthcare providers to proactively identify vulnerabilities, harden their models, and build more trustworthy and secure medical imaging AI systems prior to clinical deployment. This proactive approach is crucial for ensuring patient safety and aligning with emerging regulatory expectations for AI in healthcare.

Future Enhancements and Vision:

Potential future enhancements for such a platform could include:

- **Real-time API Integration:** Enabling seamless integration with existing AI development and deployment pipelines for continuous robustness monitoring.
- **Expanded Modality Support:** Extending capabilities beyond X-rays to include other critical medical imaging modalities like CT, MRI, and Ultrasound.
- **Advanced Visualization Dashboards:** Providing interactive dashboards for visualizing attack impacts, defense effectiveness, and generating compliance-oriented security scores.
- **Support for Black-Box Testing:** Incorporating methodologies for evaluating models where internal access is limited.
- **Certification and Benchmarking Services:** Potentially offering a standardized framework for benchmarking and certifying the adversarial robustness of medical AI models.

This commercialization and application vision positions the collective research of our group to deliver a valuable framework specifically designed to bridge the gap between AI innovation and cybersecurity in the sensitive domain of medical imaging, ultimately contributing to safer and more reliable AI in healthcare.

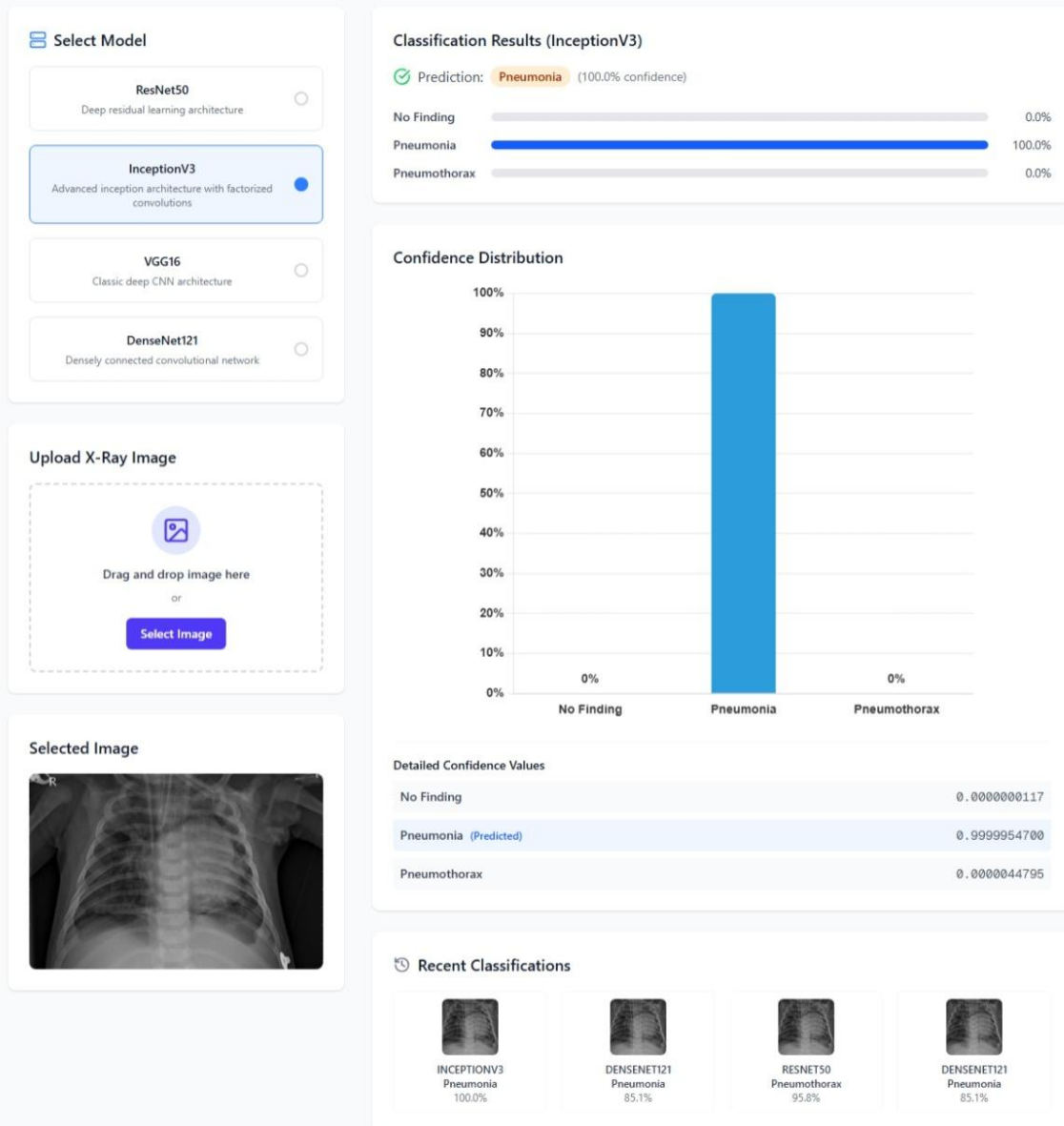


Figure 18: Our web application dashboard

5. Description of Personal Component

Registration Number	Name	Functions
IT21155802	D.S.C. Wijesuriya	<p>Fine-tuning the InceptionV3 model for chest X-ray classification ('No Finding', 'Pneumonia', 'Pneumothorax')</p> <p>Implementing and applying the Perturbed Gradients Norm (PGN) attack to the InceptionV3 model</p> <p>Measuring and analyzing the impact of PGN attacks on InceptionV3 performance</p> <p>Designing and evaluating a Transformation & Prediction Consistency Check mechanism for PGN attack detection</p> <p>Implementing and evaluating defense strategies</p>

6. Conclusion

This research has systematically investigated a significant challenge in the application of AI to medical imaging: the vulnerability of deep learning models, specifically an InceptionV3 architecture fine-tuned for chest X-ray classification, to the sophisticated PGN attack. Unlike attacks that solely rely on maximizing classification loss, PGN incorporates an input gradient norm penalty, enabling it to craft subtle yet highly effective perturbations that can significantly impair the model's diagnostic accuracy.

Through rigorous implementation and evaluation, this study demonstrated that PGN attacks can substantially degrade the performance of the InceptionV3 model in classifying conditions such as 'No Finding', 'Pneumonia', and 'Pneumothorax' from chest X-rays. This finding underscores the critical need for robust adversarial awareness and mitigation strategies before such AI systems can be confidently deployed in clinical environments where diagnostic precision is paramount.

In response to this vulnerability, a Transformation & Prediction Consistency Check mechanism was developed and evaluated for detecting PGN-attacked inputs. This detector, utilizing Gaussian Blur as an input transformation, showed a promising capability in identifying images perturbed by PGN, offering a practical method for pre-screening inputs.

Furthermore, several defense strategies were implemented and assessed. PGN Adversarial Training proved to be highly effective, significantly restoring the model's accuracy on adversarial examples while largely maintaining performance on clean data. The proposed PSIF, implemented using a tailored Gaussian Blur as an inference-time preprocessing step, also demonstrated substantial success in mitigating the PGN attack's impact with minimal computational overhead. While various standard Denoising Techniques were explored, their effectiveness against the specific structure of PGN perturbations was found to be less pronounced in this context compared to the more targeted defenses.

The overall findings from this research confirm that while advanced CNN models like InceptionV3 offer powerful diagnostic capabilities, they are not inherently immune to specialized adversarial threats such as PGN. However, the successful implementation and evaluation of both a dedicated detection mechanism and targeted defense strategies, particularly PGN Adversarial Training and PSIF, indicate that practical pathways exist to significantly enhance the robustness and reliability of these AI systems. This study contributes

valuable insights for AI developers, healthcare practitioners, and security researchers working towards building more secure, trustworthy, and clinically viable AI-powered medical imaging solutions.

7. REFERENCES

- [1] R. A. Al-Falluji, Z. D. Katheeth, and B. Alathari, "Automatic Detection of COVID-19 Using Chest X-Ray Images and Modified ResNet18-Based Convolution Neural Networks," *Computers, Materials & Continua*, vol. 66, no. 2, pp. 1301–1313, 2021.
- [2] "PGN: A Perturbation Generation Network Against Deep Reinforcement Learning," *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, 2023.
- [3] X. Ge, L. Wang, and Z. Yang, "Penalizing Gradient Norm for Efficiently Improving Robustness of Neural Networks," *IEEE Trans. Neural Netw. Learn. Syst.*, 2023.
- [4] "Boosting Adversarial Transferability by Achieving Flat Local Maxima," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023.
- [5] "Deep Learning Based Image Classification of Lungs Radiography for Detecting COVID-19 using a Deep CNN and ResNet50," *IEEE Access*, 2021.
- [6] "Automatic Detection of COVID-19 Using Chest X-Ray Images and Modified ResNet18-Based Convolution Neural Networks," *Computers, Materials & Continua*, 2021.

8. APPENDICES

8.1 Appendix A - Data Preprocessing and Dataset prepare

```
# Set up data augmentation for the training set
train_datagen = ImageDataGenerator(
    rescale=1./255, # Rescale pixel values to [0, 1]
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)

# Validation set: rescale only, no augmentation
val_datagen = ImageDataGenerator(rescale=1./255)

# Test set: rescale only, no augmentation
test_datagen = ImageDataGenerator(rescale=1./255)

# Load data from directories
train_generator = train_datagen.flow_from_directory(
    train_path, # Path to train data
    target_size=(299, 299), # Resize to fit InceptionV3 input size
    batch_size=32,
    color_mode='rgb',
    class_mode='categorical', # Since we have multiple classes (Pneumonia, No Finding, Pneumothorax)
)

val_generator = val_datagen.flow_from_directory(
    val_path, # Path to validation data
    target_size=(299, 299),
    batch_size=32,
    color_mode='rgb',
    class_mode='categorical',
)
```



```

test_generator = test_datagen.flow_from_directory(
    test_path, # Path to test data
    target_size=(299, 299),
    batch_size=32,
    color_mode='rgb',
    class_mode='categorical',
)

if hasattr(train_generator, 'class_indices'):
    print("\n--- Class Indices Assigned by Keras ---")
    # class_indices is a dictionary like {'class_name': index}
    print(train_generator.class_indices)
    # Example: {'No Finding': 0, 'Pneumonia': 1, 'Pneumothorax': 2}

    # You can also create the correctly ordered list directly from this:
    # Sort by index (value) to get the correct order for your prediction list
    sorted_indices = sorted(train_generator.class_indices.items(), key=lambda item: item[1])
    class_names_ordered = [item[0] for item in sorted_indices]
    print("\nOrdered list for prediction script (use this!):")
    print(class_names_ordered)
    # Example: ['No Finding', 'Pneumonia', 'Pneumothorax']
    print("-----\n")
else:
    print("Could not find 'class_indices' attribute on the generator.")

```

```

Found 4816 images belonging to 3 classes.
Found 600 images belonging to 3 classes.
Found 610 images belonging to 3 classes.

--- Class Indices Assigned by Keras ---
{'No Finding': 0, 'Pneumonia': 1, 'Pneumothorax': 2}

Ordered list for prediction script (use this!):
['No Finding', 'Pneumonia', 'Pneumothorax']
-----

```

8.2 Appendix B - Fine-tune InceptionV3 Architecture

```
from tensorflow.keras.applications import InceptionV3
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D
from tensorflow.keras.optimizers import Adam

# Load InceptionV3 with pre-trained ImageNet weights, excluding the top classification layers
base_model = InceptionV3(weights='imagenet', include_top=False, input_shape=(299, 299, 3))

# Freeze the layers of the base model to retain the pre-trained features
for layer in base_model.layers:
    layer.trainable = False

# Add custom layers on top for your task
x = base_model.output
x = GlobalAveragePooling2D()(x) # Reduce the dimensions to a 1D vector
x = Dense(1024, activation='relu')(x) # Add a dense layer with ReLU activation
x = Dense(3, activation='softmax')(x) # Output layer for 3 classes (Pneumonia, No Finding, Pneumothorax)

# Create the model
model = Model(inputs=base_model.input, outputs=x)

# Compile the model with Adam optimizer and categorical cross-entropy loss (for multi-class classification)
model.compile(optimizer=Adam(learning_rate=0.0005), loss='categorical_crossentropy', metrics=['accuracy'])

# Show the summary of the model
model.summary()
```

Model: "functional_3"

Layer (type)	Output Shape	Param #	Connected to
input_layer_3 (InputLayer)	(None, 299, 299, 3)	0	-
conv2d_282 (Conv2D)	(None, 149, 149, 32)	864	input_layer_3[0][0]
batch_normalization_282 (BatchNormalization)	(None, 149, 149, 32)	96	conv2d_282[0][0]
activation_282 (Activation)	(None, 149, 149, 32)	0	batch_normalization_2...
conv2d_283 (Conv2D)	(None, 147, 147, 32)	9,216	activation_282[0][0]
batch_normalization_283 (BatchNormalization)	(None, 147, 147, 32)	96	conv2d_283[0][0]
activation_283 (Activation)	(None, 147, 147, 32)	0	batch_normalization_2...
conv2d_284 (Conv2D)	(None, 147, 147, 64)	18,432	activation_283[0][0]
batch_normalization_284 (BatchNormalization)	(None, 147, 147, 64)	192	conv2d_284[0][0]
activation_284 (Activation)	(None, 147, 147, 64)	0	batch_normalization_2...

8.3 Appendix C – Penalizing Gradient Norm Attack Algorithm

Algorithm 1 Penalizing Gradient Norm (PGN) attack method

Input: A clean image x with ground-truth label y , and the loss function J with parameters θ .

Parameters: The magnitude of perturbation ϵ ; the maximum number of iterations, T ; the decay factor μ ; the balanced coefficient δ ; the upper bound (i.e., ζ) of random sampling in ζ -ball; the number of randomly sampled examples, N .

```

1:  $g_0 = 0, x_0^{adv} = x, \alpha = \epsilon/T$ ;
2: for  $t = 0, 1, \dots, T - 1$  do
3:   Set  $\bar{g} = 0$ ;
4:   for  $i = 0, 1, \dots, N - 1$  do
5:     Randomly sample an example  $x' \in \mathcal{B}_\zeta(x_t^{adv})$ ;
6:     Calculate the gradient at the sample  $x', g' = \nabla_{x'} J(x', y; \theta)$ ;
7:     Compute the predicted point by  $x^* = x' - \alpha \cdot \frac{g'}{\|g'\|_1}$ ;
8:     Calculate the gradient of the predicted point,  $g^* = \nabla_{x^*} J(x^*, y; \theta)$ ;
9:     Accumulate the updated gradient by  $\bar{g} = \bar{g} + \frac{1}{N} \cdot [(1 - \delta) \cdot g' + \delta \cdot g^*]$ ;
10:  end for
11:   $g_{t+1} = \mu \cdot g_t + \frac{\bar{g}}{\|\bar{g}\|_1}$ ;
12:  Update  $x_{t+1}^{adv}$  via  $x_{t+1}^{adv} = \Pi_{\mathcal{B}_\epsilon(x)} [x_t^{adv} + \alpha \cdot \text{sign}(g_{t+1})]$ ;
13: end for
14: return  $x^{adv} = x_T^{adv}$ .

```

Output: An adversarial example x^{adv} .

8.4 PGN Attack Summary For All Classes

<https://drive.google.com/file/d/1HiEklEgKaxIhKvCcHZKMDC9MjVmpv97-/view?usp=sharing>