



Ministry of Higher Education and Scientific Research
Abdelhamid Ibn Badis University
Faculty of Exact Sciences & Informatics
Department of Mathematics and Computer Science
Sector: Computer Science



END OF STUDIES THESIS

For obtaining the Master's Degree in Computer Science

Option : Artificial Intelligence for the Internet of Things

X-PNEUMO: A mobile application for pneumonia detection with meaningful explainability.

By :

ABDELGHANI GANA
MAHMOUD HAMADI

DEFENDED ON **June XX, 2025** IN FRONT OF THE JURY COMPOSED OF:

President:		Senior Lecturer, FSEI, UMAB, Algeria
Examiner:		Senior Lecturer, FSEI, UMAB, Algeria
Supervisor:	Hafsa LATTAR	Senior Lecturer, FSEI, UMAB, Algeria
Co-Supervisor:	Baghdad Atmani	Professeur, FSEI, UMAB, Algeria
Co-Supervisor:	Henda HAJJAMI BEN GHEZALA	Professor Emerita, ENSI, UMA, Tunisia

Promotion : 2024/2025

Dedication

“

*I dedicate this master thesis to my family, my professors
and every one who supported me through out this journey*

”

- Mahmoud Hamadi

“

*I dedicate this master thesis to my family, my professors
and every one who supported me through out this journey*

”

- Abdelghani Gana

Acknowledgements

We would like to express our sincere gratitude to our supervisor, Dr. Hafsa Lattar, for her invaluable guidance, support, and encouragement throughout the course of this thesis. We are also thankful to our co-supervisors, Professor Atmani Baghdad and Professor Emerita Henda Hajjami Ben Ghezala, for their insightful advice and support, which greatly enriched our work.

Abstract

This thesis addresses a critical challenge in the application of deep learning in medical imaging: developing pneumonia detection systems that provide not just accurate diagnoses but also meaningful explanations that align with clinical reasoning. While current explainable artificial intelligence methods can identify pneumonia, they often highlight irrelevant regions in chest X-rays that lack clinical relevance. These meaningless explanations fail to support medical decision-making and diminishing confidence in AI systems. To solve this problem, we developed X-Pneumo, a mobile application that not only detects pneumonia effectively but also provides clear, medically relevant explanations by focusing specifically on lung areas where pneumonia appears, unlike standard XAI explanations that might point to irrelevant areas. X-Pneumo's visual guides highlight only clinically significant regions, matching how doctors examine X-rays, following a complete development process from initial research to final implementation, including careful analysis of existing technologies, design of a specialized deep learning model, and the creation of a practical mobile tool for healthcare settings. Testing showed our approach produces better explanations than conventional methods while maintaining high diagnostic accuracy, representing an important step toward making AI more transparent and useful in clinical practice, with future work potentially expanding this approach to other diseases and further improving the system's reliability across diverse patient cases.

Keywords : Pneumonia detection, Explainable artificial intelligence, Meaningful explanations, Clinical interpretability, Deep learning, Medical imaging.

Résumé

Cette thèse aborde un défi crucial dans l'application de l'apprentissage profond en imagerie médicale : développer des systèmes de détection de pneumonie qui fournissent non seulement des diagnostics précis, mais aussi des explications pertinentes alignées sur le raisonnement clinique. Bien que les méthodes actuelles d'intelligence artificielle explicable (XAI) puissent identifier la pneumonie, elles mettent souvent en évidence des régions non pertinentes dans les radiographies pulmonaires, sans signification clinique. Ces explications inutiles ne soutiennent pas la prise de décision médicale et érodent la confiance dans les systèmes d'IA. Pour résoudre ce problème, nous avons développé X-Pneumo, une application mobile qui non seulement détecte efficacement la pneumonie, mais fournit également des explications claires et médicalement pertinentes en se concentrant spécifiquement sur les zones pulmonaires atteintes, contrairement aux méthodes XAI standard qui peuvent pointer vers des régions sans rapport. Les guides visuels de X-Pneumo mettent en lumière uniquement les zones cliniquement significatives, reflétant ainsi l'analyse des radiologues. Le développement a suivi un processus complet, de la recherche initiale à l'implémentation finale, incluant l'analyse des technologies existantes, la conception d'un modèle d'apprentissage profond spécialisé et la création d'un outil mobile adapté aux environnements médicaux. Les tests ont montré que notre approche génère des explications plus pertinentes que les méthodes conventionnelles tout en maintenant une haute précision diagnostique, représentant ainsi une avancée importante vers une IA plus transparente et utile en pratique clinique. Les travaux futurs pourraient étendre cette approche à d'autres maladies et améliorer la fiabilité du système pour divers cas patients.

Mots clés : Détection de pneumonie, Intelligence artificielle explicable, Explications significatives, Interprétabilité clinique, Apprentissage profond, Imagerie médicale.

ملخص

هذه الرسالة العلمية تتناول تحديًا بالغ الأهمية في مجال تطبيق التعلم العميق في التصوير الطبي، وهو تطوير أنظمة كشف الالتهاب الرئوي التي لا تقتصر على تقديم تشخيصات دقيقة فحسب، بل توفر أيضًا تفسيرات منطقية تتماشى مع الفهم الطبي للمرض. ورغم قدرة أساليب الذكاء الاصطناعي التفسيري الحالية على تشخيص الالتهاب الرئوي، إلا أنها غالبًا ما تركز على مناطق غير مهمة في صور الأشعة السينية للصدر، مما ينتج عنه تفسيرات غير ذات قيمة طبية. لحل هذه الإشكالية، قمنا بتطوير تطبيق X-Pneumo الذي لا يكتفي بالكشف الدقيق عن الالتهاب الرئوي، بل يقدم أيضًا تفسيرات واضحة وذات صلة بالمجال الطبي من خلال التركيز تحديدًا على المناطق الرئوية المصابة، بعكس أنظمة الذكاء الاصطناعي التفسيري التقليدية التي قد تشير إلى مناطق غير مرتبطة بالمرض. يعتمد التطبيق على أدلة بصرية توضح فقط المناطق ذات الأهمية الطبية، بما يتوافق مع آلية فحص الأطباء للأشعة. مر المشروع بمراحل تطوير متكاملة بدءًا من البحث الأولي ووصولاً إلى التنفيذ النهائي، شملت تحليلًا دقيقًا للتقنيات المتاحة، وتصميم نموذج تعلم عميق متخصص، وبناء أداة عملية قابلة للاستخدام في البيئات الطبية. وأثبتت النتائج أن منهجيتنا تقدم تفسيرات أفضل من الأساليب التقليدية مع الحفاظ على مستوى عالٍ من الدقة التشخيصية، مما يمثل تقدمًا مهمًا نحو جعل الذكاء الاصطناعي أكثر وضوحًا وفائدة في التطبيقات الطبية. ويمكن توسيع نطاق هذا العمل مستقبلاً ليشمل أمراضًا أخرى، مع تحسين موثوقية النظام لتشمل حالات مرضية متنوعة.

كلمات مفتاحية : الكشف عن الالتهاب الرئوي، الذكاء الاصطناعي القابل للتفسير، تفسيرات ذات دلالة طبية، قابلية التفسير الطبي، التعلم العميق، التصوير الطبي.

Contents

Dedication	1
Acknowledgements	3
Abstract	4
Résumé	5
6 ملخص	
General Introduction	1
1 Software Requirement Specification (SRS)	3
1.1 Introduction	4
1.1.1 Purpose	4
1.1.2 Intended Audience	4
1.1.3 Scope	4
1.1.4 Definitions, Acronyms, and Abbreviations	4
1.1.5 Overview	5
1.2 Overall Description	6
1.2.1 Product Perspective	6
1.2.2 Product Functions	6
1.2.3 User Characteristics	7
1.2.4 Constraints	7
1.2.5 Assumptions and Dependencies	8
1.2.6 Apportioning of Requirements	9
1.3 Specific Requirements	9
1.3.1 External Interface Requirements	9
1.3.2 Functional Requirements	10
1.3.3 Performance Requirements	14
1.3.4 Logical Database Requirements	14
1.3.5 Design Constraints	15
1.3.6 Software System Attributes	16
1.4 Conclusion	16
2 State-of-the-art	17
2.1 Introduction	18
2.2 Pneumonia Detection	18
2.2.1 What is pneumonia?	18
2.2.2 Deep Learning in Medical Imaging	19
2.2.3 Deep learning based Pneumonia Detection: Related Works	23
2.2.4 Synthesis	23

2.3	Evaluation Metrics for Deep Learning Models	23
2.3.1	Confusion Matrix	24
2.3.2	Accuracy	24
2.3.3	Precision	25
2.3.4	Recall	25
2.4	Beyond Detection: Interpreting Pneumonia Detection Results	25
2.4.1	What is eXplainable Artificial Intelligence?	26
2.4.2	Explainable Artificial Intelligence Methods	27
2.4.3	XAI Methods Applied to Pneumonia Detection: Related Work . .	31
2.4.4	Synthesis	32
2.5	Evaluation Techniques	32
2.5.1	Robustness	33
2.5.2	Fidelity (Faithfulness)	33
2.5.3	Complexity	34
2.5.4	Randomization	34
2.6	Conclusion	35
3	XPneumo: A meaningful explainability-based solution	36
3.1	Introduction	37
3.2	Meaningfull explanation for pneumonia detection	37
3.2.1	Pneumonia detection	38
3.2.2	Initial explanation	39
3.2.3	Meaningful explanation	39
3.3	Experiments	39
3.3.1	Experiments setup	40
3.3.2	Experiments	42
3.3.3	Experiments results and discussion	43
3.4	Conclusion	47
4	Modeling of XPneumo	48
4.1	Introduction	49
4.2	UML presentation	49
4.3	UML diagrams (categories)	49
4.4	Scrum methodology	52
4.5	Our modeling approach	53
4.5.1	Use case diagram	53
4.5.2	Class diagram	55
4.5.3	Component diagram	55
4.5.4	Deployment diagram	56
4.5.5	Activity diagram	56
4.5.6	Sequence diagram	59
4.5.7	State diagram	59
4.6	Conclusion	60
5	Implementation of XPneumo	61
5.1	Introduction	62
5.2	Technologies used	62
5.2.1	Visual Studio Code	62
5.2.2	Kaggle	62
5.2.3	Python	63

5.2.4	Flutter/Dart	63
5.2.5	Modelio	63
5.2.6	Drawio	64
5.2.7	Flask	64
5.2.8	SQLite	65
5.2.9	Tensorflow	65
5.2.10	JWT auth	66
5.2.11	ngrok	66
5.2.12	TLS	66
5.2.13	HTTPS	67
5.3	How to use XPneumo	67
5.3.1	The splash screen	67
5.3.2	The main login page	68
5.3.3	The main screen	73
5.3.4	The analysis screen	74
5.3.5	The database screen	76
5.4	Conclusion	77
	Conclusion et perspectives	78

List of Figures

1.1	Logo	10
2.1	Pneumonia disease	18
2.2	Perceptron and multilayer perceptron	19
2.3	Description of a two-layer convolutional neural network with 2 and 3 filters, respectively.	20
2.4	Description of RNN architecture	21
2.5	Calculation of the output of the attention model based on the weighted average of values	22
2.6	The Transformer - model architecture.	22
2.7	eXplainable AI Benefits in Healthcare	26
2.8	Explaining predictions of an AI system.	27
2.9	Fireboat Recognition: IG Attribution Highlights Water spray Over Vessel Structure	28
2.10	Example of occluded part of the original image	29
2.11	Approximating the complex model to a an interpretable model	29
2.12	Highlight Discriminative Regions by Linearly Combining Convolutional Feature Maps (W1, W2, ..., Wn) to Generate a Heatmap for 'Australian Terrier' Classification	30
2.13	Processing flow diagram of Score CAM including Phase 1: extracting original image features; Phase 2: masking the input image using the feature map after upsampling, re-extracting features through CNN forward propagation and then estimating the contribution of the feature map	30
2.14	Effect of noise level on the heatmap	31
2.15	The Shapley Values determine the contribution of each feature.	31
2.16	Robustness	33
2.17	faithfulness	33
2.18	Complexity	34
2.19	Randomization	35
3.1	Pneumonia designed Explanation focusing on Shoulders	37
3.2	Solution Workflow	38
3.3	Pneumonia Detection Model Architecture	38
3.4	Meaningful process	39
3.5	Pneumonia samples	40
3.6	Normal samples	40
3.7	Labels distribution	41
3.8	CNN Classification report	43
3.9	CNN accuracy and loss plot	44
3.10	CNN confusion matrix	44
3.11	ViT Classification Report	44
3.12	ViT Accuracy and Loss plot	45

3.13	ViT Confusion Matrix	45
3.14	Comparative Analysis of XAI methods	46
3.15	Improvement of Pneumonia Interpretability	47
4.1	UML taxonomy	50
4.2	Scrum process	52
4.3	Use case diagram	53
4.4	Class diagram	55
4.5	Component diagram	55
4.6	Deployment diagram	56
4.7	Activity diagram (the upper half)	57
4.8	Activity diagram (the Bottom half)	58
4.9	Sequence diagram	59
4.10	State diagram	60
5.1	Visual Studio Code logo.	62
5.2	Kaggle	62
5.3	Python	63
5.4	Flutter and Dart	63
5.5	Modelio	64
5.6	Drawio	64
5.7	Flask	64
5.8	SQLite	65
5.9	TensorFlow	65
5.10	JWT auth	66
5.11	ngrok	66
5.12	TLS	66
5.13	HTTPS	67
5.14	Splash Screen	68
5.15	The Main Login	69
5.16	The forgot the password screen	70
5.17	The reset the password screen	71
5.18	The confirmation pop up	72
5.19	The signing-in screen	73
5.20	Main screen	74
5.21	Analysis screen (1)	75
5.22	Analysis screen (2)	76
5.23	Database	77

List of Tables

2.1	Confusion Matrix	24
-----	----------------------------	----

List of abbreviations

XAI	<i>Explainable Artificial Intelligence</i>
DL	<i>Deep Learning</i>
CNNs	<i>Convolutional Neural Networks</i>
CXR	<i>Chest X-Ray</i>
ViT	<i>Vision Transformer</i>
Grad-CAM	<i>Gradient-weighted Class Activation Mapping</i>
LIME	<i>Local Interpretable Model-agnostic Explanations</i>
SHAP	<i>SHapley Additive exPlanations</i>
UML	<i>Unified Modeling Language</i>
Scrum	<i>Scrum (Agile Framework)</i>
RNN	<i>Recurrent Neural Network</i>
TP	<i>True Positive</i>
FP	<i>False Positive</i>
FN	<i>False Negative</i>
Score-CAM	<i>Score-weighted Class Activation Mapping</i>
SVM	<i>Support Vector Machine</i>
KNN	<i>K-Nearest Neighbors</i>
GPU	<i>Graphics Processing Unit</i>
CPU	<i>Central Processing Unit</i>
Adam	<i>Adaptive Moment Estimation</i>
OMG	<i>Object Management Group</i>
XPneumonia	<i>Application Name</i>
UI	<i>User Interface</i>
API	<i>Application Programming Interface</i>
JWT	<i>JSON Web Token</i>

TLS	<i>Transport Layer Security</i>
HTTPS	<i>Hypertext Transfer Protocol Secure</i>
HTTP	<i>Hypertext Transfer Protocol</i>

General Introduction

Pneumonia stands as one of the most significant infectious diseases worldwide, claiming millions of lives annually and representing a major burden on global healthcare systems, particularly affecting vulnerable populations including older adults, young children, and those with underlying chronic conditions such as asthma, diabetes, and other immunocompromising diseases [9]. Accurate and early diagnosis of pneumonia is crucial, as delays or errors can lead to serious complications and higher risk of death, making chest radiography the cornerstone of pneumonia diagnosis by providing non-invasive and accessible means of visualizing pulmonary abnormalities and evaluating infection severity [34]. Nevertheless, this diagnostic modality poses several inherent challenges largely tied to human expertise such as variability among experts, subjective interpretation, limited sensitivity in detecting early stage disease, and ambiguity of overlapping anatomical structures, which can critically affect diagnostic accuracy, delay treatment initiation, and ultimately compromise patient outcomes.

In response to the mentioned challenges, the integration of deep learning technologies into medical imaging has emerged as a promising solution, demonstrating remarkable capabilities and often achieving diagnostic accuracy comparable to trained radiologists [4]. However, while deep learning models can process vast amounts of radiological data and identify subtle patterns invisible to the human eye, their successful deployment in clinical practice requires more than just high accuracy rates, as the "black box" nature of these systems creates a critical barrier to clinical adoption since healthcare professionals require not only accurate diagnostic predictions but also clear explanations of how these conclusions are reached.

The requirement for transparency has led to the application of explainable artificial intelligence in medical imaging [4], which aims to bridge the gap between deep learning performance and clinical interpretability. XAI methods generate an explainable version of chest X-ray that highlight regions of interest. However, a critical examination of current XAI applications reveals a fundamental flaw that undermines their clinical utility. Despite the promise of explainable AI, existing XAI methods often produce explanations that are meaningless in clinical contexts. These systems frequently highlight irrelevant image regions such as equipment edges, the shoulders, heart, text annotations, or random noise patterns that bear no relationship to the underlying pathology instead of focusing the actual lung tissue where pneumonia would manifest.

The disconnect between AI explanations and clinical reasoning creates a paradox: while we have developed accurate techniques for pneumonia detection, the explanations they provide fail to support clinical decision-making. The lack of clinical relevance in these explanations severely limits their practical value and potentially undermines clinician confidence in such technologies. In medical imaging unclear or irrelevant explanations might mislead clinicians or fail to offer the clear guidance needed for making reliable diagnoses recognizing this critical gap between meaningless and meaningful explanation.

This thesis proposes an approach that provides meaningful explanations. Rather than treating explanation generation as an afterthought or post-processing step, our methodology places lung segmentation at the center of the explainability process, ensuring that model attention mechanisms focus exclusively on anatomically relevant regions. By incorporating lung segmentation as a key part of the explanation pipeline, we guide the model to generate explanations that focus solely on the lung tissue, filtering out irrelevant background information. This approach helps ensure that the model attention is directed to clinically relevant areas, which are crucial for accurate pneumonia diagnosis. By using anatomy-based reasoning, we directly address the issue of meaningless explanations, aligning model’s explanations with medical logic and providing clear insights that support clinicians in making confident decisions.

Our methodology marks a shift from generic AI explanations to more anatomy-centered interpretations. It demonstrates how integrating domain knowledge like lung segmentation into the model’s explanations can turn vague visualizations into valuable diagnostic tools. Through a comprehensive evaluation based on factors such as accuracy, clarity, and clinical relevance, this research sets a new benchmark for meaningful AI explainability in medical applications. Ultimately, it bridges the gap between DL’s performance and the trust clinicians place in AI generated explanations, helping improve pneumonia diagnosis.

Reading guide

This thesis will have 5 chapters:

- **Chapter 1: Software Requirement Specification**
In this chapter, we present the software requirements specification to precisely define the solution, which will be in the form of a mobile application, and the technical aspect related to it.
- **Chapter 2: State of the art**
In this chapter, we present a state-of-the-art review on pneumonia detection using deep learning, followed by an overview of related work involving explainable artificial intelligence (XAI) methods applied to pneumonia diagnosis.
- **Chapter 3: XPneumonia: A meaningful explainability-based solution**
This chapter introduces a solution for pneumonia detection with a focus on generating meaningful explanations. It outlines the detection process, the initial explanation approach, and its enhancement, followed by experiments and analysis to evaluate the method’s performance and interpretability.
- **Chapter 4: Modeling of XPneumonia**
This chapter delves into the modeling of our application. We unveil its internal structure through detailed diagrams, providing a clear understanding of how the various components work together.
- **Chapter 5: Implementation of XPneumonia**
In this chapter, we present all the methods, the tools used to implement our solution, and the different user interfaces.

Chapter 1

Software Requirement Specification (SRS)

1.1 Introduction

1.1.1 Purpose

This document describes the Software Requirements Specification (SRS) for the X-PNEUMO mobile application, designed for pneumonia detection with meaningful explainability. It outlines the functionalities, interfaces, and constraints of the system.

1.1.2 Intended Audience

- Hafsa LATTAR
- Baghdad Atmani
- Henda HAJJAMI BEN GHEZALA
- Abdelghani Gana
- Mahmoud Hamadi

1.1.3 Scope

The X-PNEUMO: mobile application is designed for detecting pneumonia disease and providing meaningful explanations. It includes functionalities such as pneumonia diagnosis and pneumonia detection explanations.

1.1.4 Definitions, Acronyms, and Abbreviations

Definitions

Pneumonia An acute respiratory infection caused by viruses or bacteria, leading to inflammation of the lungs' alveoli, which fill with pus and fluid, making breathing painful and limiting oxygen intake.

Deep Learning (DL) A subset of machine learning that uses neural networks with multiple layers to model complex patterns in data, particularly effective in tasks like image classification.

Explainable Artificial Intelligence (XAI) Techniques and methods designed to make the decisions of AI models transparent and interpretable, especially for “black-box” models like deep neural networks.

Heatmap A visual representation of data where values are depicted as colors, often used in XAI to highlight regions of an image that influence a model's prediction.

Lung Segmentation The process of isolating lung regions from medical images to focus analysis on clinically relevant areas.

Acronyms and Abbreviations

XAI Explainable Artificial Intelligence

DL Deep Learning

CNNs Convolutional Neural Networks

CXR Chest X-Ray

ViT Vision Transformer

Grad-CAM Gradient-weighted Class Activation Mapping

LIME Local Interpretable Model-agnostic Explanations

SHAP SHapley Additive exPlanations

UML Unified Modeling Language

Scrum An Agile framework for project management

RNN Recurrent Neural Network

TP True Positive

FP False Positive

FN False Negative

Score-CAM Score-weighted Class Activation Mapping

SVM Support Vector Machine

KNN K-Nearest Neighbors

GPU Graphics Processing Unit

CPU Central Processing Unit

Adam Adaptive Moment Estimation

OMG Object Management Group

XPneumonia Application Name

UI User Interface

API Application Programming Interface

JWT JSON Web Token

TLS Transport Layer Security

HTTPS Hypertext Transfer Protocol Secure

HTTP Hypertext Transfer Protocol

1.1.5 Overview

This document consists of an overview of the product's functionalities and informal requirements, followed by detailed technical specifications in the subsequent sections.

1.2 Overall Description

This section of the SRS will describe the general factors that affect the product and its requirements. However it doesn't state specific requirements. But it provides a background for those requirements, which will be discussed in detail later.

1.2.1 Product Perspective

As perspectives for our Xpneumo application, we aim to improve the robustness of segmentation by enriching the dataset with more rare pathological cases. We also plan to extend the application to cover additional diseases such as brain tumors, hydrocephalus, bone tumors, kidney stones, and more. Furthermore, we envision a more secure authentication process by integrating with trusted entities like medical universities, licensing boards, and physician registries, ensuring that only verified medical professionals can access and use the application.

1.2.2 Product Functions

The main functions include:

User Authentication

- Login
- Registration
- Password reset

Pneumonia Detection

- Upload or capture chest X-ray images.
- Analyze images using the trained deep learning model.
- Display diagnosis results (Normal/Pneumonia).

Explainability

- Provide meaningful explanations for the model's predictions.

Patient Management

- Enter and save patient names for record-keeping.
- View historical diagnosis results.

Account Management

- Delete user account and associated data.
- Logout securely.

Security

- Secure data transmission using HTTPS and JWT authentication.
- Encrypted storage of user credentials and patient data.

1.2.3 User Characteristics

The X-Pneumo mobile application is designed for healthcare professionals, including:

- Medical Practitioners (Doctors, Radiologists):
 - Ability to upload chest X-ray images for pneumonia detection and to generate explanation heatmaps.
 - Ability to access, review, and manage saved patient data linked to their account (e.g., past diagnoses, heatmaps).
 - Familiarity with mobile applications

1.2.4 Constraints

1. **Regulatory Policies** All users have access to all diagnostic functionalities including image analysis, result interpretation, and patient record management. Administrative privileges are not implemented in the current version.
2. **Interfaces to Other Applications** No integrations with external medical systems (PACS, EHR) are implemented in the current version. Future versions may include DICOM-compliant interfaces
3. **Parallel Operation** No parallel operation with legacy systems is required
4. **Audit Functions** All input fields undergo strict validation including:
 - **Credential verification:** Implementation of secure login using username and strong password policies to ensure authentication integrity.
 - **Patient name sanitization:** Input sanitization mechanisms are applied to patient names to prevent injection attacks and ensure proper formatting.
 - **Image format validation:** Only images in JPEG and PNG formats are accepted, ensuring compatibility and reducing security risks associated with unsupported file types.
 - **Diagnostic result logging:** All diagnostic outcomes are securely logged for auditing and traceability, complying with medical data management standards.
5. **Safety and Security Considerations**
 - **End-to-End Encryption:** All data transmissions are secured using HTTPS/TLS protocols to ensure confidentiality and integrity.

- **Token-Based Authentication:** JSON Web Tokens (JWT) are used to manage user sessions securely, enabling stateless and tamper-proof authentication.
- **Secure API Communication:** RESTful APIs developed with Flask enforce strict access control mechanisms, limiting unauthorized interactions.
- **Data Integrity:** A lightweight SQLite database is used with appropriate access restrictions to maintain data consistency and prevent unauthorized modifications.
- **Development Security:** Ngrok is utilized to create encrypted tunnels for secure testing and development environments.

1.2.5 Assumptions and Dependencies

Assumptions

- Developed and tested on Android device.
- **Backend Data Storage:** Data is stored on data base.
- **Permissions:**
 - Camera/Gallery permissions for media access.
 - Internet permissions for online communication.
- **Host Machine Setup:** Assumes Flask and SQLite are installed on the host machine.

Dependencies

Mobile Application

- Flutter SDK (with Dart) for cross-platform development.
- Libraries:
 - `http` – for REST API communication.
 - `flutter_secure_storage` – for secure storage of JWT tokens.

Backend Server

- Flask (Python) for serving REST API endpoints.
- SQLite for lightweight, local database storage.
- Ngrok for creating secure tunnels during external testing phases.

Machine Learning

- TensorFlow/Keras for implementing and running the pneumonia detection model.
- Python 3.x with supporting libraries such as:
 - `numpy` – for numerical operations.
 - `Pillow` – for image loading and preprocessing.

Security

- HTTPS/TLS protocols enforced via Flask and Ngrok for encrypted data transmission.
- JWT (JSON Web Tokens) used for secure user authentication and session management.

/

1.2.6 Apportioning of Requirements

Future updates may include:

- Integration with medical devices (DICOM viewers, digital X-ray machines, PACS).
- Expanded disease detection (tuberculosis, COVID 19, tumors, pneumothorax)
- Hospital system integration
- Cloud-connected artificial intelligence with offline backup Full analysis when online, basic diagnosis when offline.

1.3 Specific Requirements

/

1.3.1 External Interface Requirements

User Interfaces

Hardware Interfaces

This application runs on devices that have android.

Software Interfaces

/

Communication Interfaces

This application requires an internet connection to connect to the database and retrieve/send required data.

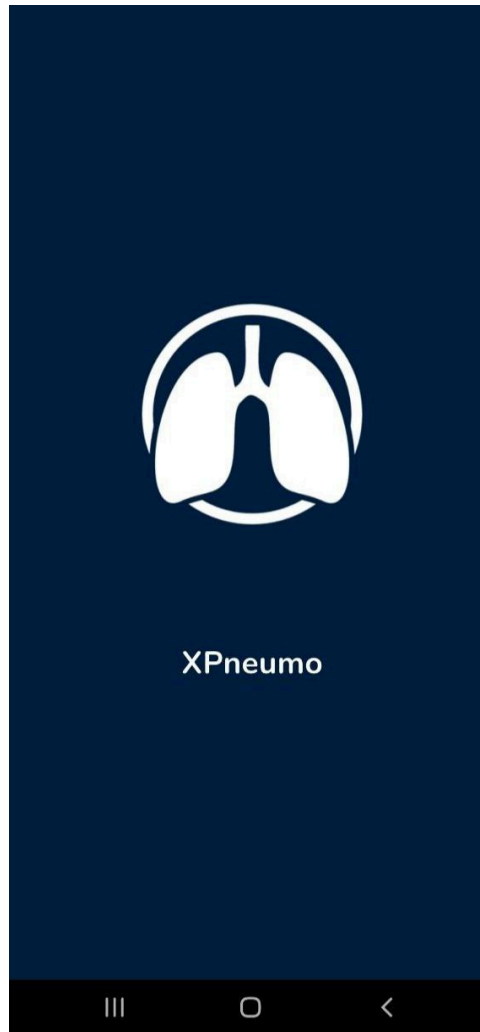


Figure 1.1: Logo

1.3.2 Functional Requirements

Register:

Input: username, password;

Processing:

- **Validate the input data:**
 1. Ensure the username is not empty:
 2. Ensure the password meets the minimum length requirement:
 3. Verify that the password and confirmation password match
- Create a new client account in the database with the provided username and password

Output:

- Successful Registration:
- Failed Registration:

Login:

Input: username, password;

Processing:

- Validate the input data:
 1. Ensure the username and password are not empty.
- Authenticate the user by verifying the credentials against the database.
- Generate and store a JWT token upon successful authentication.

Output:

- Redirect to the home screen (`LoadImg`) upon successful login.
- Error message if authentication fails (e.g., invalid credentials or connection error).

Forgot Password:

Input: Username

Processing:

- Verify username exists.
- Generate 1-hour valid JWT reset token.

Output:

- Reset token.

Reset Password:

Input: Reset token, New password

Processing:

- Validate token expiration.
- Update password hash in database.

Output:

- Success confirmation or token error.

Reset password:

Input: New password, Confirmation password;

Processing:

- Validate the input data:
 1. Ensure the new password and confirmation password match.
 2. Ensure the new password meets the minimum length requirement (6 characters).
- Update the user's password in the database using the reset token.

Output:

- Confirmation message indicating successful password reset.
- Error message if the reset fails (e.g., invalid token or connection error).

Load Image:

Input:

- Patient name
- Image (either from gallery or camera)

Processing:

- Validate the input data:
 1. Ensure the patient name is not empty.
 2. Ensure an image is selected.
- Preprocess the image:
 1. Center-crop the image.
 2. Resize the image.
 3. Normalize the image.
- Classify the image to determine if it is an X-ray.

Output:

- Redirect to the analysis screen if the image is valid (X-ray).
- Error message if the image is not an X-ray or if there is an issue with the input.

Analyse:

Input:

- X-ray image
- Patient name

Processing:

- Preprocess the image:
 1. Grayscale the image.
 2. Resize the image.
 3. Equalize the histogram.
 4. Apply segmentation.
- Run inference using the classifier and segmenter models.
- Generate a heatmap overlay for the X-ray image.
- Save the analysis results to the database.

Output:

- Display the diagnosis result (Normal or Pneumonia).
- Provide an explanation for the diagnosis.
- Option to view the heatmap overlay and detailed explanation.
- Option to start a new analysis.

Explanation Screen:**Input:**

- Heatmap overlay image
- Explanation text

Processing:

- Display the heatmap overlay and explanation text.

Output:

- Visual representation of the heatmap.
- Textual explanation of the diagnosis.

Delete Account:**Processing:**

- Validate the user's authentication token.
- Delete the user's account and all associated data from the database.

Output:

- Redirect to the login screen upon successful deletion.

- Error message if the deletion fails (e.g., invalid token or connection error).

Logout:

Processing:

- Delete the stored JWT token.

Output:

- Redirect to the login screen.

1.3.3 Performance Requirements

- **Image Processing:** <5s on mid-range mobile devices
- **Model Inference:** <2s (TensorFlow Lite)
- **API Response:** <1s for non-image endpoints
- **Concurrency:** Support ≥ 50 simultaneous users (Flask)

1.3.4 Logical Database Requirements

1. User

- **Purpose:** Stores login credentials and account details.
- **Fields:**
 - user_id
 - username
 - password_hash
 - created_at

2. Password Reset Token

- **Purpose:** Manages temporary password reset links.
- **Fields:**
 - token_id
 - user_id
 - reset_token
 - expires_at

3. Analysis

- **Purpose:** Stores X-ray analysis results.
- **Fields:**
 - analysis_id
 - user_id
 - patient_name
 - result
 - image_data (or file path)

Relationships

- User → Analysis (One-to-Many)
- User → PasswordResetToken (One-to-Many)

Storage Notes

- **Images:** Store as Base64 (small DB) or external paths (scalability).
- **Indexes:** Optimize queries on username, reset_token, and user_id.

1.3.5 Design Constraints

Edge Inference

- **TensorFlow Lite:** For edge inference
- **Secure Storage:** For JWT
- **Camera/Gallery Integration:** For image capture and selection

Backend

- **SQLite:** For lightweight storage
- **JWT-based:** Stateless authentication
- **Base64 Image Handling:** For image processing and transmission

Security

- **Password Hashing:** PBKDF2 for secure password storage
- **CORS Configuration:** To handle cross-origin requests
- **Token Expiration:** 24 hours for general tokens, 1 hour for sensitive tokens

1.3.6 Software System Attributes

- **Reliability:** This signifies a system's consistently being able to deliver its intended functionalities within specified performance parameters, showing minimal error occurrences and demonstrating the capacity for the recovery from errors.
- **Availability:** This signifies the probability that the application is functional and accessible to users at any point in time. It reflects the percentage of time the software is up and running as opposed to being down for maintenance or due to failures.
- **Security:** Safeguarding the application and its data from unauthorized access, modification, disclosure, disruption, or destruction. This involves measures to prevent cyberattacks, protect user privacy, and ensure the overall integrity of the software's functionalities.
- **Maintainability:** Maintainability characterizes the ease with which an application can be modified, repaired, or enhanced after it's first deployed. It signifies the ability to fix bugs, improve functionalities, adapt to new requirements, and keep the software running smoothly.

1.4 Conclusion

In this chapter, we have presented The Software Requirements Specification (SRS) according to IEEE Std 830-1998. In the following chapter, we will present a state-of-the-art review on pneumonia detection using deep learning, followed by an overview of related work involving explainable artificial intelligence (XAI) methods applied to pneumonia diagnosis

Chapter 2

State-of-the-art

2.1 Introduction

Pneumonia remains a major cause of morbidity and mortality [9] making rapid and reliable detection essential. Although deep learning models have demonstrated strong performance in this field. However, their "black-box" nature limits their adoption in clinical practice. To address this challenge explainable artificial intelligence methods aim to make model decisions more transparent and to strengthen the trust of healthcare professionals. In this chapter, we first review current deep learning-based approaches for pneumonia detection. Then, we explore recent advances in explainable AI techniques specifically designed to improve the interpretability of pneumonia detection models. Finally, we discuss key evaluation metrics used to assess the performance and reliability of these explainability methods in clinical contexts.

2.2 Pneumonia Detection

This section provides an overview of pneumonia as a medical condition and explores how deep learning is revolutionizing its detection through medical imaging. We begin with general definition of pneumonia and highlights its medical importance followed by a discussion on the role of deep learning in medical diagnostics. Finally, we review related research that highlights the effectiveness of deep learning approaches in detecting pneumonia from chest X-ray images.

2.2.1 What is pneumonia?

Pneumonia is a form of acute respiratory infection that is most commonly caused by viruses or bacteria (See Figure 2.1). The lungs are made up of small sacs called alveoli, which fill with air when a healthy person breathes. When an individual has pneumonia, the alveoli are filled with pus and fluid, which makes breathing painful and limits oxygen intake. These infections are generally spread by direct contact with infected people [36].

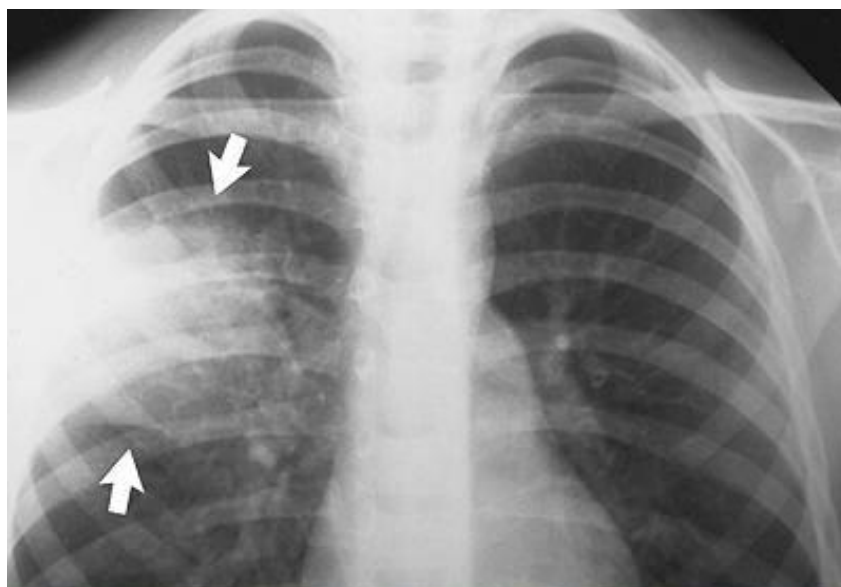


Figure 2.1: Pneumonia disease

Pneumonia can cause mild to life-threatening illness in people of all ages, however it

is the single largest infectious cause of death in children worldwide. It killed more than 808 000 children under the age of 5 in 2017, accounting for 15% of all deaths of children under 5 years. These stark statistics highlight the critical need for early, and accurate pneumonia detection [37].

In the next subsection, we will explore how modern technologies, particularly deep learning, are being leveraged in medical imaging to assist in the early detection of pneumonia.

2.2.2 Deep Learning in Medical Imaging

Medical imaging has revolutionized modern healthcare by enabling non-invasive visualization of the human anatomy. As one of the most rapidly advancing areas in medicine [21], it includes technologies such as ultrasound, radiography, computed tomography, magnetic resonance imaging, positron emission tomography, and endoscopy. These technologies produce visual data that require expert interpretation. Similarly, deep learning methods have been applied to this field to automate interpretation and assist diagnosis [5]. The following section explores deep learning techniques that make this automation possible.

1. Multilayer perceptron or fully connected networks

The multilayer perceptron [44] or fully connected network is the simplest version of artificial networks. The network consists of a layered structure created by an input layer, an output layer, and one or more hidden layers. Each layer comprises several units interconnected with neighboring layers (input and output) but without connections between the units of the same layer. In general, the network layers consist only of several units known as perceptron (See Figure 2.2). The perceptron sums the input values (See Figure 2.2), multiplied by the weights. The weights are known as training parameters and are adjusted later in the training of the network. Finally, the weighted sum is introduced to the activation function (nonlinear function), simulating the activation, or generating the output of each perceptron [5].

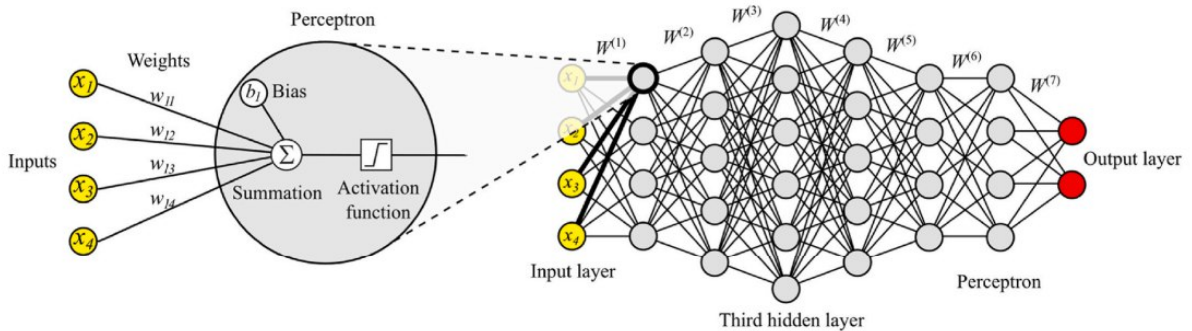


Figure 2.2: Perceptron and multilayer perceptron

2. Convolutional neural networks

Much of the growth of deep learning is mainly due to the advances made in computer vision. In fact, one of the most widely used algorithms in this field is convolution [5], from which the convolutional neural network [26] is derived. The network could

decipher or learn the most complex patterns existing in a set of images, and it does so by employing convolution. Fundamentally, convolution consists of a two-function operator, the image and the filter. The function takes part of the image and highlights patterns by multiplying each point of the image fragment with the filter elements (See Figure 2.3). The result is weighted, and the generated values are placed in the position corresponding to the image fragment. The process is repeated by moving the filter across the entire image, creating an image with highlighted features that depend on the filter structure. In the case of Convolutional neural networks, The images generated by the convolution are known as feature maps. In addition, the bias to each map element is included here, and each map is subjected to the activation function, fulfilling the description of Equation (1). In this context, the training parameters of the network are the weights associated with all filters, i.e., the network learns the optimal filters to highlight the high-level features that converge to the desired task. The process is repeated layer after layer creating increasingly abstract features [5].

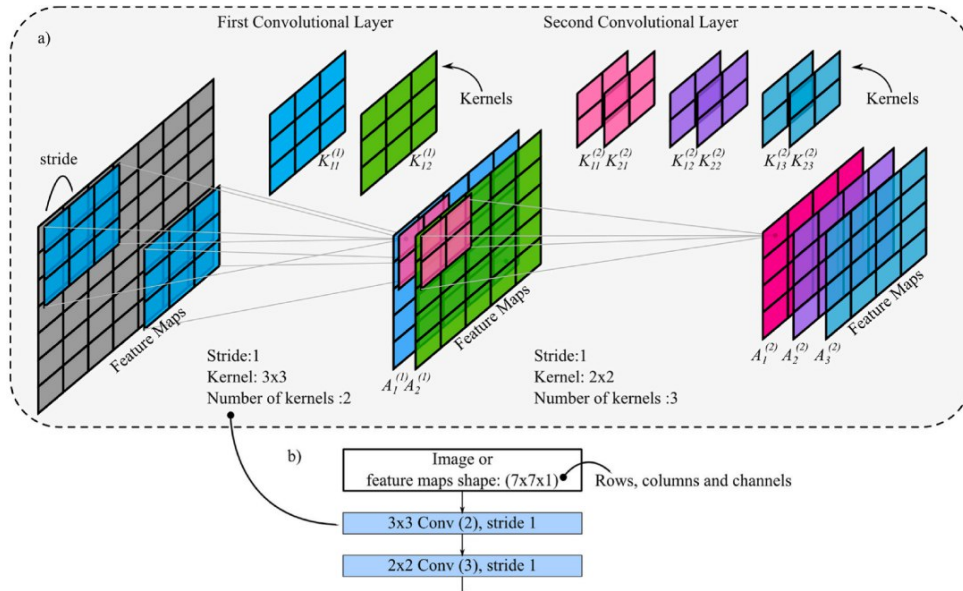


Figure 2.3: Description of a two-layer convolutional neural network with 2 and 3 filters, respectively.

3. Recurrent neural network

Recurrent neural networks [44] are networks with feed-forward loops capable of preserving sequential characteristics, i.e., they can deal with problems with time dependence. The main applications are focused on translations (text sequence), audios, and videos, among the most frequent problems. In medicine, they can be temporal signals associated with the study of a pathology, such as electromyographic signals or longitudinal records to study the evolution of the pathology. Recurrent neural networks can store information from previous data through hidden states and, together with the current input, the output associated with that sequence is calculated. Recurrent neural networks could be considered to ‘memorize’ the preliminary information to arrive at the desired prediction in practice. The information is driven through the different layers, converging on the output associated with the task’s performance, such as prediction or classification. Recurrent neural networks follow the same principles as Multilayer perceptron and Convolutional neural networks, while the only difference is that the network’s

output will depend on initial values [5] (See Figure 2.4).

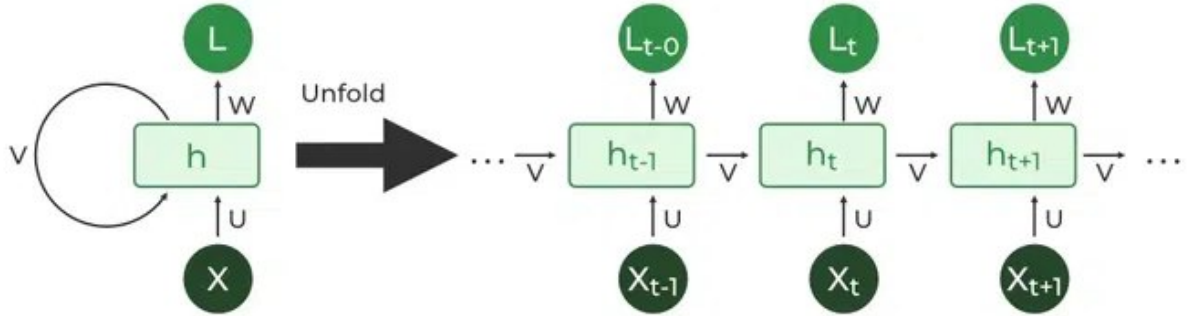


Figure 2.4: Description of RNN architecture

4. Attention models

The attention mechanism [7] is a bio-inspired systems. The principle of attention models is based on the optic nerve of the visual system. In fact, not all perceived information by the eye's retina has the same degree of importance. The brain focuses its attention on objects of interest. For example, while you are reading this text, your attention is focused on the message conveyed by the text, ignoring the other elements surrounding it. This concept has led to the development of attention mechanisms. In medicine For example, in magnetic resonance imaging, the image is square shaped with the axial slice of the brain in the center, implying a quantity of information associated with the background (irrelevant information). In simple terms, attention helps a model decide which parts of the input are more important when making a prediction. For example, rather than averaging all previous outcomes to predict a new one, attention gives more weight to those past outcomes that are most similar to the current input. From neural networks prospect the attention mechanism involves query, key, and value elements. The network uses a scoring function to determine how relevant each key is to the query. These scores are used to weigh the corresponding values leading to a final output [5](See Figure 2.5).

5. Transformers – attention is all YOU need

The Transformer [57] is a deep learning model introduced in 2017 that relies entirely on attention mechanisms, discarding the use of recurrent and convolutional neural networks. It is composed of an encoder and a decoder that are made up of multiple layers. The input typically a sequence of text, is first tokenized and then converted into vector through embedding. The core of the Transformer is the self-attention mechanism, which allows tokens to weigh the importance of other tokens in the sequence by generating query, key, and value vectors. To enhance learning, the model uses multiple attention heads in parallel, a process known as multi-head attention capturing different aspects of the token relationships. These outputs are then passed through feedforward layers and normalization steps. In the decoder

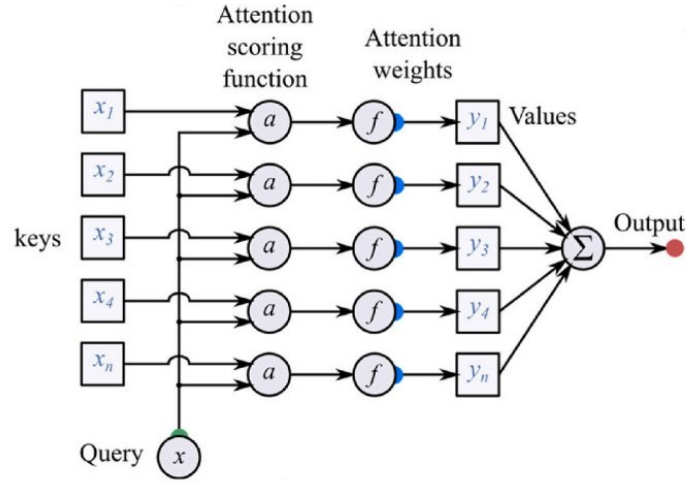


Figure 2.5: Calculation of the output of the attention model based on the weighted average of values

attention is also applied with mechanisms to preserve the sequence's autoregressive nature [5] (See Figure 2.6).

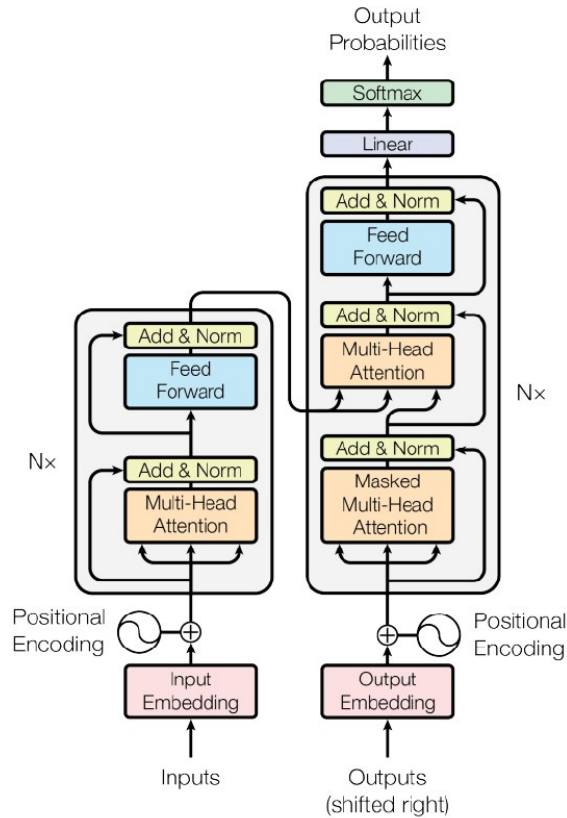


Figure 2.6: The Transformer - model architecture.

In summary, this section has provided a comprehensive overview of deep learning methods commonly used in medical imaging, highlighting their architectures, functioning mechanisms, and clinical relevance. Then, in the following section, we will review various deep learning-based approaches used in pneumonia detection.

2.2.3 Deep learning based Pneumonia Detection: Related Works

Traditionally, pneumonia diagnosis relies on clinical examination, auscultation of respiratory sounds, and medical imaging—primarily chest radiography, which visualizes characteristic pulmonary infiltrates. Laboratory tests, such as blood cultures and cytobacteriological analysis of respiratory secretions, can further identify the causative pathogen [58]. With advancements in artificial intelligence, deep learning approaches have emerged to automate pneumonia detection from radiographic images, demonstrating performance comparable to radiologists.

For instance, Rajpurkar et al. (2017) [41] developed ChexNet, a 121-layer convolutional neural network that achieved an F1-score of 0.435, surpassing average radiologist performance in pneumonia detection. Gabruseva et al. (2020) [15] proposed an algorithm for localizing lung opacities using a RetinaNet-based single-shot detector [27] with Se-ResNext101 encoders [62] pre-trained on ImageNet. Similarly, Kundu et al. (2021) [25] built a computer-aided diagnosis system combining GoogLeNet [54], ResNet-18 [16], and DenseNet-121 [20], achieving 98.81% accuracy on the Kermany and RSNA datasets while demonstrating robustness in statistical tests. Recent studies further optimized deep learning for pneumonia detection. Zhong et al. (2024) [64] compared three approaches: a custom CNN, transfer learning with ResNet152V2 [16], and a fine-tuned ResNet152V2. The fine-tuned model outperformed others in accuracy, precision, recall, and F1 score. Sharma and Guleria (2023) [47] introduced a hybrid VGG-16 [49] and Neural Network model, achieving 92.15% and 95.4% accuracy in binary and multi-class classification, respectively. Their approach surpassed traditional classifiers (e.g., SVM, KNN, Random Forest) paired with VGG-16 for feature extraction.

2.2.4 Synthesis

In the previous section, we have seen how deep learning have become a central tool in both medical imaging field and precisely the detection of pneumonia. We began by understanding pneumonia and its clinical importance. We then highlighted how deep learning methods are transforming medical imaging by automating complex tasks such as disease detection and organ segmentation. Then we reviewed several studies showing their high accuracy in classifying pneumonia and outperforming traditional diagnostic methods and highlighting that CNNs are the most commonly used method. In addition when CNNs combined in ensemble approaches, have shown exceptional potential in improving diagnostic efficiency and assisting healthcare professionals in the timely identification of pneumonia.

Although deep learning models have achieved impressive accuracy in pneumonia detection, their decision making processes often remain opaque ("black-box" nature). In critical fields like healthcare, it is essential not only to predict outcomes but also to understand and interpret how these predictions are derived. In the following section, we address this challenge by exploring methods to improve model transparency and explainability.

2.3 Evaluation Metrics for Deep Learning Models

This section discusses the key evaluation metrics used to assess the performance of deep learning models, focusing on accuracy, precision, and recall [51]. These metrics are essential for analyzing model effectiveness, especially in classification tasks. Before diving

into the evaluation metrics for classification, it is important to understand the confusion matrix.

2.3.1 Confusion Matrix

A confusion matrix [51] is a technique for summarizing the performance of a classification algorithm. It creates an $N \times N$ matrix, where N is the number of classes or categories to be predicted. Each row of the matrix represents the actual class, while each column represents the predicted class. A few terms associated with the confusion matrix are:

- **True positive (TP)**: An instance for which both predicted and actual values are positive.
- **True negative (TN)**: An instance for which both predicted and actual values are negative.
- **False Positive (FP)**: An instance for which predicted value is positive but actual value is negative.
- **False Negative (FN)**: An instance for which predicted value is negative but actual value is positive.

A confusion matrix follows the below format:

Actual\Predicted	Positive	Negative
Positive	True Positives (TP)	False Negatives (FN)
Negative	False Positives (FP)	True Negatives (TN)

Tableau 2.1: Confusion Matrix

2.3.2 Accuracy

Accuracy [51] is a fundamental metric for evaluating the performance of a classification model, providing a quick snapshot of how well the model is performing in terms of correct predictions. It is calculated as the ratio of correct predictions to the total number of input samples.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of input samples}} = \frac{\text{TP} + \text{TN}}{\text{Total number of input samples}}$$

It works great if there are an equal number of samples for each class. For example, we have a 90% sample of class A and a 10% sample of class B in our training set. Then, our model will predict with an accuracy of 90% by predicting all the training samples

belonging to class A. If we test the same model with a test set of 60% from class A and 40% from class B. Then the accuracy will fall, and we will get an accuracy of 60%.

Accuracy is good but it gives a False Positive sense of achieving high accuracy. The problem arises due to the possibility of misclassification of minor class samples being very high.

This situation can be projected onto the pneumonia detection task, where medical datasets often suffer from imbalanced labels [31]. with a higher proportion of any label over the other. As a result, a model might appear to perform well based on high accuracy while in reality it fails to correctly classify the minority label cases. This shows that accuracy alone can be misleading in medical imaging tasks, and should be complemented with other metrics that better capture the model’s performance across both classes.

2.3.3 Precision

Precision [51] indicates out of all positive predictions, how many are actually positive. It is defined as a ratio of correct positive predictions to overall positive predictions. It is calculated as

$$\text{Precision} = \frac{\text{Predictions actually positive}}{\text{Total predicted positive}} = \frac{\text{TP}}{\text{TP} + \text{TN}}$$

Precision is especially useful when the classes in a dataset are imbalanced. In such cases, precision helps to understand how reliable the model’s positive predictions are. If the model frequently predicts the minority class incorrectly, the precision will be low, highlighting that the positive predictions cannot be trusted even if the overall accuracy seems high.

2.3.4 Recall

Recall [51] indicates out of all actually positive values, how many are predicted positive. It is a ratio of correct positive predictions to the overall number of positive instances in the dataset.

$$\text{Recall} = \frac{\text{Predictions actually positive}}{\text{Actual positive values in the dataset}} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Recall becomes important when we care more about identifying all actual positive cases, especially in situations where missing a positive instance can have serious consequences. In such cases, it helps measure how well the model captures all the true positives. A low recall means that many positive instances are going undetected, which can be critical in fields like medical diagnosis.

2.4 Beyond Detection: Interpreting Pneumonia Detection Results

In this section we explore how Explainable Artificial Intelligence [42] bridges the opacity gap by making AI-driven diagnoses more transparent and clinically actionable. We first introduce fundamental XAI concepts, then examine state of the art XAI methods specifically applied to pneumonia detection systems.

2.4.1 What is eXplainable Artificial Intelligence?

Explainable Artificial Intelligence [42] refers to the set of methods and techniques developed to make the decisions made by artificial intelligence models understandable, transparent, and interpretable, especially those considered "black boxes," such as deep neural networks. Unlike simpler models (such as linear regression [30] or decision trees [39]), complex architectures like Convolutional neural networks [26] or transformers [57] do not, by default, offer a direct insight into their internal decision-making mechanisms [45].

The rapid integration of AI in high-stakes domains, particularly healthcare, has amplified the need for clarity and explainability. In such critical applications decisions made by deep learning models can have profound consequences impacting diagnoses, treatment plans, and patient outcomes. Without explainability, clinicians may find it difficult to trust or validate models predictions, which can hinder diagnosis or lead to ethical and legal challenges. Therefore XAI enables practitioners to interpret model outputs, validate them against clinical expertise, and uncover potential biases or errors—supporting faster diagnosis, stronger AI–human collaboration, increased transparency, ethical compliance, and educational value as illustrated in figure 2.7.

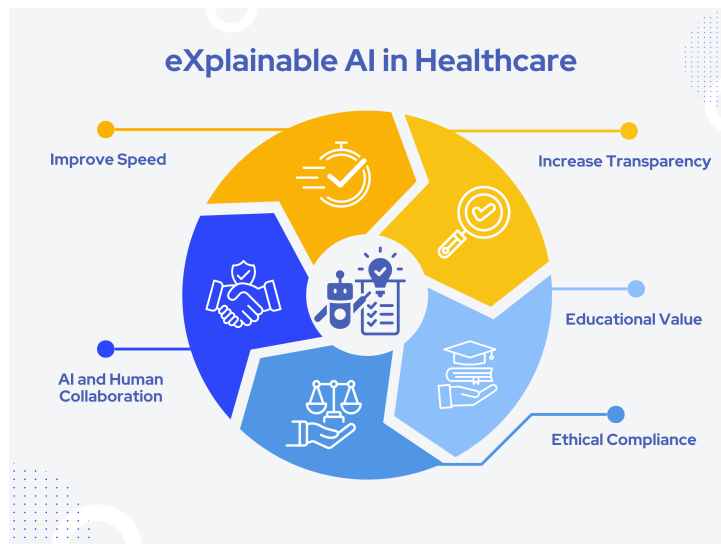


Figure 2.7: eXplainable AI Benefits in Healthcare

In the context of pneumonia result interpretation, the process is illustrated in figure 2.8, the process begins with a chest x-ray image being fed into a black-box model that classifies the image as "PNEUMONIA" indicating a positive diagnosis. To make the prediction more transparent, an explainability method is applied to generate a heatmap that extract the regions of the image most significant to the prediction. Consequently, it reveals the opacity for users by downgrading the black-box process into a visual interpretation.

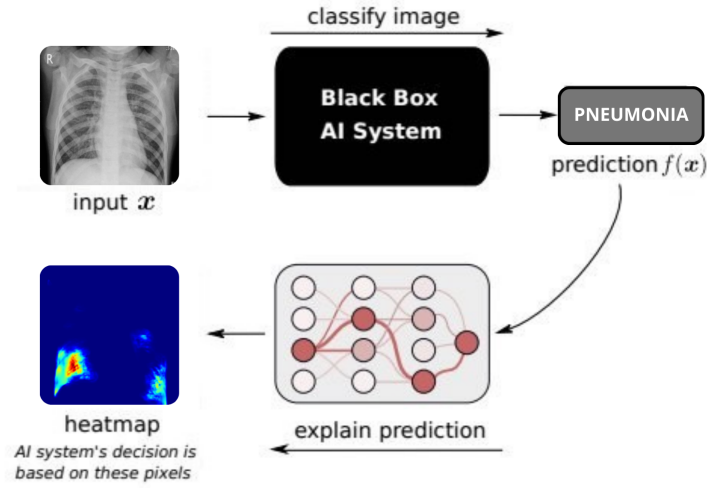


Figure 2.8: Explaining predictions of an AI system.

2.4.2 Explainable Artificial Intelligence Methods

Explainable Artificial Intelligence methods generally produce visual, textual, or numerical explanations. In the context of medical imaging, the most common approach is to generate heatmaps that highlight the regions of an image that most influenced the model's prediction [8]. However, there are many different methods, each with its own approach, yet they share some common criteria such as post-hoc techniques, which explain predictions after the model is trained, and gradient-based methods which use the model's gradients to identify important features and perturbation based methods that alter the input to measure how changes affect predictions.

In this subsection, we will focus on some of the most relevant explainability methods used in this field, including Integrated Gradients [53], Occlusion Sensitivity [63], Local Interpretable Model-agnostic Explanations [42], Grad-CAM [46], Score CAM [60], SmoothGrad [50], and SHAP [28].

1. Integrated Gradients

Integrated Gradients [53] is post-hoc, gradients based method that helps explain predictions by assigning importance scores to input features. It's particularly useful for understanding which parts of the input (like pixels in an image or words in text) most influenced a model's output. The central idea of this method is to examine how the model's prediction changes as we gradually transform from a "baseline" input (typically a zero or black image) to the actual input we want to analyze (See Figure 2.9).

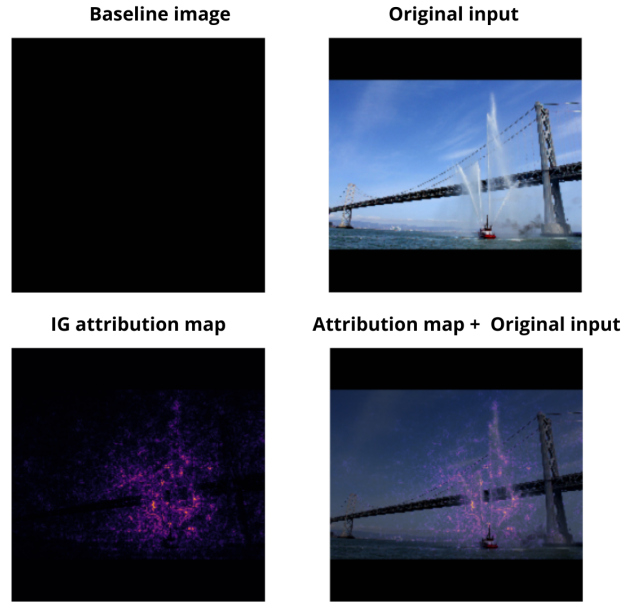


Figure 2.9: Fireboat Recognition: IG Attribution Highlights Water spray Over Vessel Structure

2. Occlusion Sensitivity

Occlusion sensitivity [63] is a post hoc, perturbation-based method used to in explaining image classification tasks. This method perturbs small areas of the input by replacing them with an occluding mask, typically a gray square. This mask is swept across the image temporarily covering different regions (See Figure 2.10). At each position the variation in the model’s prediction score for a target class is measured. Significant drops in confidence indicate that the occluded region is important for the classification. By analyzing these variations, we can deduce which areas of the image contribute to the model’s decision.

3. Local Interpretable Model-agnostic Explanations

Local Interpretable Model-agnostic Explanations [42] is a post hoc, perturbation-type explanation method for black-box model predictions; it approximates the black-box model locally with an interpretable model. Given a test sample and an already trained prediction model, what it does first is generate a surrogate dataset by randomly perturbing the original input. Usually around 5,000 samples are generated by slightly altering feature values that again mostly follow a normal distribution. These perturbed instances are then fed into the original model to predict the outcome. Following this, it weights these instances based on their proximity to the original instance: closer is deemed more relevant. Using the weighted surrogate dataset, This method selects the most relevant features through different methods and fits an interpretable model. In general, this method approximates the behavior of a complex model like a neural network by fitting a interpretable model such as linear regression locally around the prediction (See Figure 2.11). While the neural network may have highly non-linear, complex decision boundaries, it focuses on a small neighborhood around a specific input instance and assumes that in that local region, the model behaves approximately linearly.



Figure 2.10: Example of occluded part of the original image

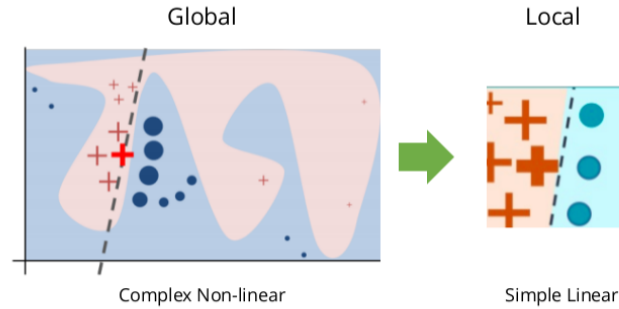


Figure 2.11: Approximating the complex model to a an interpretable model

4. Gradient-weighted Class Activation Mapping

Gradient-weighted Class Activation Mapping [46] is a post-hoc, gradient-based explainability method. It works by computing the gradients of the output class score with respect to the feature maps of the last convolutional layer. These gradients are then globally averaged to obtain a weight for each feature map. The feature maps are multiplied by their corresponding weights and summed to produce an explainability heatmap (See Figure 2.12). This method is especially useful for interpreting convolutional neural networks in image classification tasks.

5. Score CAM

Score-CAM [60] is a post hoc explainability method. Unlike gradient-based methods, Score-CAM does not rely on backpropagated gradients to compute the importance of activation maps. Instead, it works by masking the input image with each activation map and passing the resulting masked images through the model to observe the effect on the target class score (See Figure 2.13). The changes in the output scores are used to weight the contribution of each activation map, leading to a class discriminative heatmap. This strategy makes Score-CAM more robust to

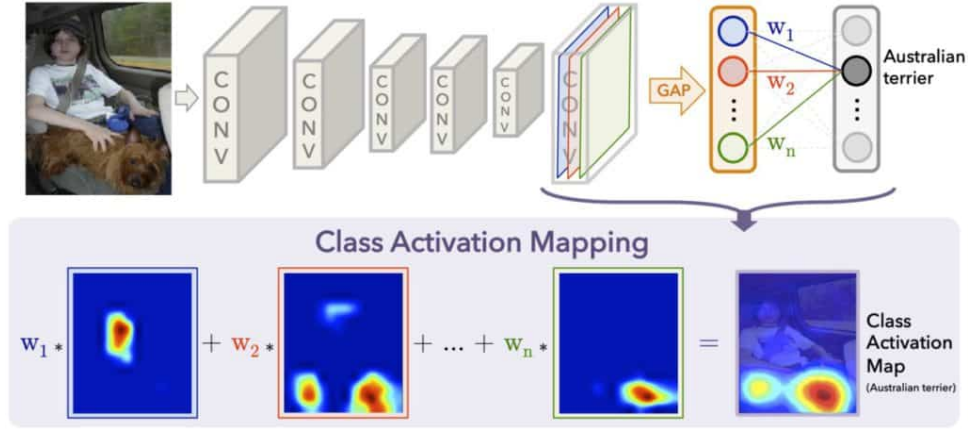


Figure 2.12: Highlight Discriminative Regions by Linearly Combining Convolutional Feature Maps ($w_1, w_2, \dots, w_n$) to Generate a Heatmap for 'Australian Terrier' Classification

gradient noise and better suited for models where gradient computation is unreliable or unavailable.

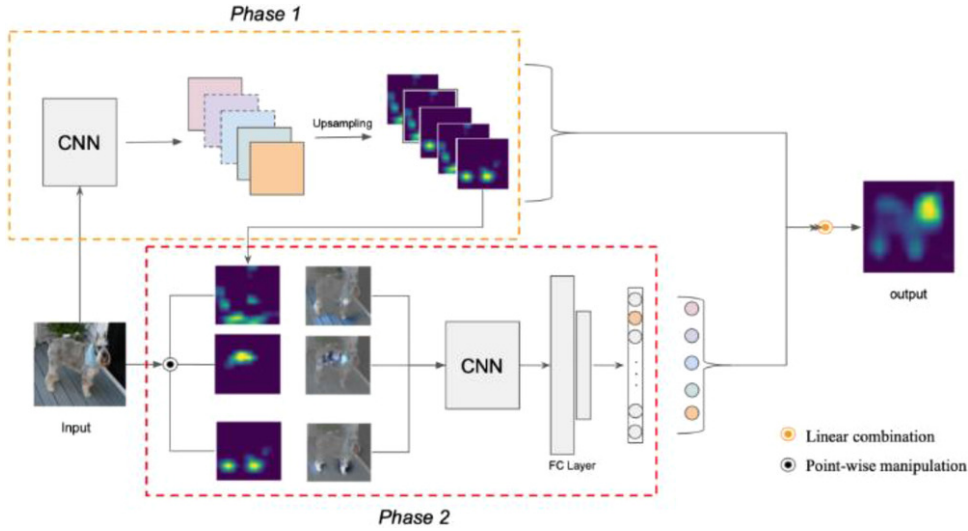


Figure 2.13: Processing flow diagram of Score CAM including Phase 1: extracting original image features; Phase 2: masking the input image using the feature map after upsampling, re-extracting features through CNN forward propagation and then estimating the contribution of the feature map

6. Smooth Grad

SmoothGrad [50] is a post hoc, gradient-based explainability method. The core idea behind this method is to generate multiple slightly perturbed versions of the input by adding random Gaussian noise (See Figure 2.14), compute the gradient of the model's output with respect to each noisy input and then average the resulting gradients. This process helps to mark the most relevant input features more clearly by smoothing out fluctuations caused by noise or model sensitivity to small changes in the input.

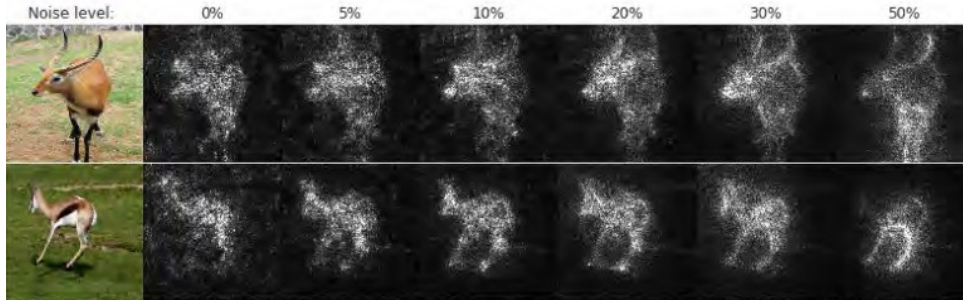


Figure 2.14: Effect of noise level on the heatmap

7. SHapley Additive exPlanations

SHapley Additive exPlanations [28] is a post hoc, model-agnostic explainability method based on Shapley values from game theory. It works by breaking down a model’s prediction into the sum of the contributions of each individual feature. For every prediction, SHAP computes a value that represents how much each feature contributes to the final output (See Figure 2.15). These values can be used to understand the relative importance of each feature and to provide explanations that are understandable to humans—an added value for agencies and teams that need to justify decisions to clients or managers. SHAP also has several notable properties, including its neutrality with respect to models: it can be applied to any machine learning model, generates consistent explanations, and is capable of handling complex model behaviors, such as interactions between features.

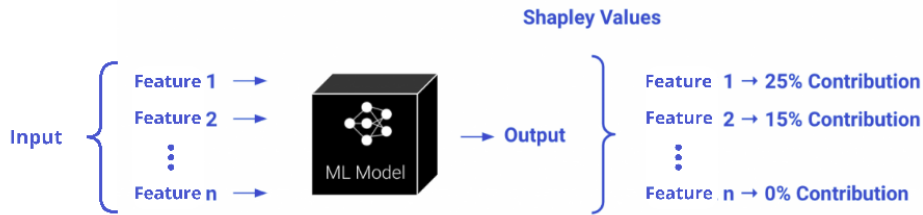


Figure 2.15: The Shapley Values determine the contribution of each feature.

2.4.3 XAI Methods Applied to Pneumonia Detection: Related Work

Pneumonia detection in chest X-rays requires more than a simple classification of “PNEUMONIA” or “NORMAL”; it demands further explanation to support clinical decision making. As discussed, eXplainable Artificial Intelligence methods generate heatmaps that visually highlight the most relevant regions in the input. In this context, the heatmaps represent the areas likely to be infected, thus offering valuable information about the localization of the infection.

Various works have applied eXplainable Artificial Intelligence techniques to pneumonia detection. For instance, Wang et al. (2017) [61] introduced CheXNet, a 121-layer DenseNet model trained on the ChestX-ray14 dataset, which used gradient-weighted class activation mapping (Grad-CAM) to highlight pneumonia-related regions, improving both classification accuracy and interpretability. Rajpurkar et al. (2018) [40] developed CheXGrad, an extension of CheXNet that integrated saliency maps with clinical

reports to provide more context-aware explanations for pneumonia detection. Kermany et al. (2018) [24] employed a transfer learning-based CNN with occlusion sensitivity analysis to identify pneumonia in pediatric chest X-rays, demonstrating how XAI methods can aid in pediatric diagnostics. Rajpurkar et al. (2018) [40] developed CheXGrad, an extension of CheXNet that integrated saliency maps with clinical reports to provide more context-aware explanations for pneumonia detection. Kermany et al. (2018) [24] employed a transfer learning-based CNN with occlusion sensitivity analysis to identify pneumonia in pediatric chest X-rays, demonstrating how XAI methods can aid in pediatric diagnostics. Aggarwal et al. (2020) [1] explored the development of trainable saliency maps to enhance model transparency in medical imaging tasks, offering a novel direction for interpretable AI in clinical applications. The model incorporated Grad-CAM for interpretable visualizations using chest X-ray images. Aminu et al. (2021) [3] utilized occlusion sensitivity maps alongside a deep neural network to localize COVID-19 indicators in chest X-rays. Rousan et al. (2021) [43] proposed a multi-task learning framework combining classification and segmentation, where attention mechanisms and Grad-CAM were used to localize pneumonia lesions while improving diagnostic performance. Rousan et al. (2021) [43] proposed a multi-task learning framework combining classification and segmentation, where attention mechanisms and Grad-CAM were used to localize pneumonia lesions while improving diagnostic performance. Marques et al. (2023) [29] employed convolutional neural networks augmented with Shapley value approximations to improve classification performance and interpretability in pneumonia detection. Ahamed et al. (2023) [2] proposed DTLCx, an enhanced ResNet-based architecture designed to distinguish between normal, conventional pneumonia, and COVID-19 cases. iddiqui et al. (2024) [48] conducted a comprehensive survey outlining recent advancements in deep learning techniques for pneumonia detection, including architecture design, training strategies, and evaluation metrics.

2.4.4 Synthesis

Among the previous works and research in pneumonia detection, gradient-based methods particularly Grad-CAM [46] stands out as the most commonly used explainable AI method. Whether applied to custom convolutional neural networks [26] or to pretrained architectures. Compared to other techniques such as LIME [42] or SHAP [28] it appears more prominently across studies addressing model interpretability in pneumonia imaging.

While previous studies have explored a range of XAI methods for pneumonia detection, selecting the most suitable method for a specific task remains a complex challenge. The appropriateness of an explanation method often depends on factors such as the model architecture, the nature of the dataset, and the user’s needs. However relying only on visual inspection to judge the quality of explanations can be both subjective and insufficient. To overcome this limitation, it becomes essential to establish objective evaluation metrics that can assess the reliability and usefulness of an XAI method. The following section discusses these metrics in detail and examines how they can support a quantitative comparison of explainability techniques.

2.5 Evaluation Techniques

Due to the absence of a ground truth explanation, XAI research has developed alternative metrics to assess the reliability of an explanation method. These evaluation metrics

analyze different properties that an explanation method should satisfy and can thus be used to compare various explainability approaches [18], [6], [33][17]. In what follows, we describe four different evaluation properties.

2.5.1 Robustness

Robustness [10] measures whether an explanation remains stable when the input image is slightly modified. In other words, if a small change is made to the image, it should not significantly affect either the model's prediction or the explanation provided (See Figure 2.16). To assess this stability, two metrics are commonly used: the local Lipschitz estimate and the average sensitivity. Both methods compare the explanation of the original image to that of a slightly perturbed image. If the difference between the two explanations is small, it indicates that the method is robust. The lower the score, the more reliable the explanation is considered to be in the presence of small perturbations.

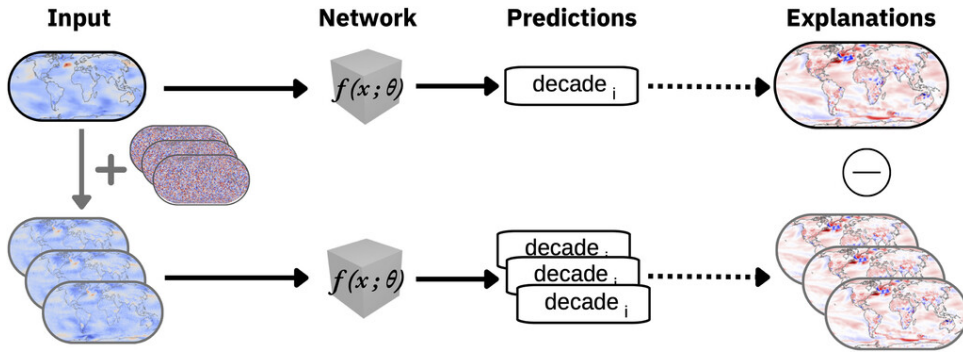


Figure 2.16: Robustness

2.5.2 Fidelity (Faithfulness)

Fidelity [10] measures whether the parts of an image highlighted as important by the explanation actually influence the model's prediction. If these important regions are modified, the model's prediction should change accordingly. To evaluate this, a method is used where the most important pixels, according to the explanation, are perturbed, and the prediction is compared before and after the perturbation (See Figure 2.17). If the prediction changes significantly, it indicates that the explanation is faithful. A higher score indicates better fidelity.

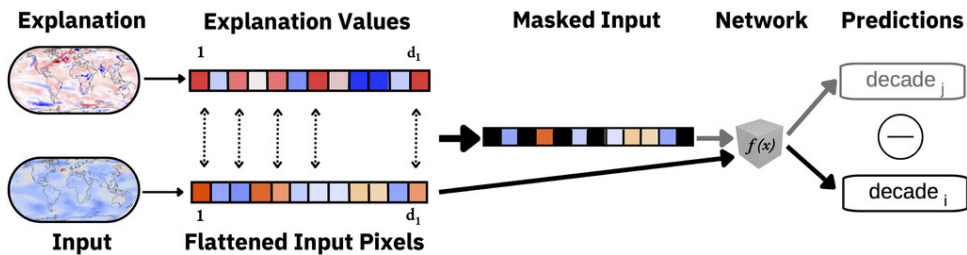


Figure 2.17: faithfulness

2.5.3 Complexity

Complexity [10] measures how clear and concise an explanation is. A good explanation should focus on a few highly important elements rather than spreading information across the entire image (See Figure 2.18). Simple explanations, highlighting only a few regions, are easier for researchers to understand and generally contain less unnecessary noise. To measure this, two indicators are used:

- Complexity (based on entropy), which assesses whether the information is well-concentrated or overly dispersed.
- Sparseness, which evaluates whether the explanation focuses on a few key areas, similar to the Gini index.

An explanation is considered less complex if it highlights only a few highly important pixels. This helps to better interpret the model's behaviour.

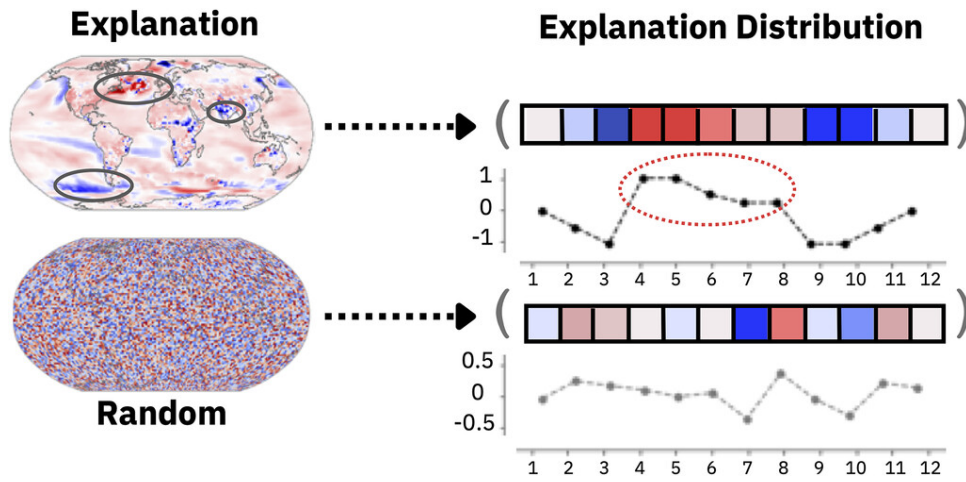


Figure 2.18: Complexity

2.5.4 Randomization

Randomization [10] is used to verify whether an explanation changes when random elements are introduced. Two scenarios are considered:

- Requesting an explanation for a random class that the model did not originally predict.
- Randomly modifying the weights of the neural network (See Figure 2.19).

In both cases, the explanation is expected to change. If it does not, it suggests that the explanation method may not be reliable, as it fails to react to significant changes in the model or the target class.

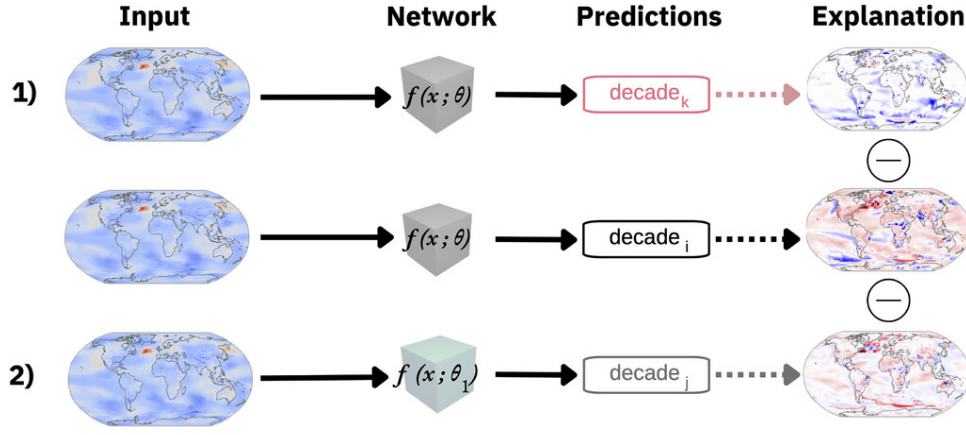


Figure 2.19: Randomization

2.6 Conclusion

This chapter inspect pneumonia detection using deep learning, focusing on recent advances and challenges. We then highlighted the importance of Explainable artificial intelligence in making deep learning models more transparent and trustworthy for clinical use. Finally, we reviewed explainable artificial intelligence methods applied to pneumonia detection and the key metrics used to evaluate their effectiveness setting the stage for building reliable and interpretable medical systems.

Chapter 3

XPneumo: A meaningful explainability-based solution

3.1 Introduction

Although explainable AI methods help us understand how deep learning models make decisions, their explanations often overlook regions that are truly important for diagnosis. Many visualizations highlight areas that are meaningless in the context of pneumonia imaging and do not align with expert interpretations, raising concerns about whether the model is focusing on the right features. As a result, a gap persists between logical explanations and those that make sense in a clinical context. This disconnect undermines trust in the model and limits the practical utility of its explanations. For explainability to be truly meaningful, it must not only open the black box but also produce insights that align with expert knowledge and support reliable decision making. The next sections review current methods, highlight their limitations, and introduce our solution, XPneumo, which aims to generate explanations that are both accurate and clinically meaningful.

3.2 Meaningfull explanation for pneumonia detection

Existing approaches to explainable AI in pneumonia imaging often fail to accurately highlight the infected regions of the lungs. In many cases the generated explanations focus on unrelated areas such as the heart and shoulders (See Figure 3.1) that hold a none diagnostic value in the context of pneumonia infection or a meaningless explanation. Even when the correct infected area within the lungs is partially highlighted, this approaches still assign greater visual importance to irrelevant regions. To address this issue, our proposed solution is designed to guide the explanations specifically toward the lung regions ensuring a meaningfull explanations.

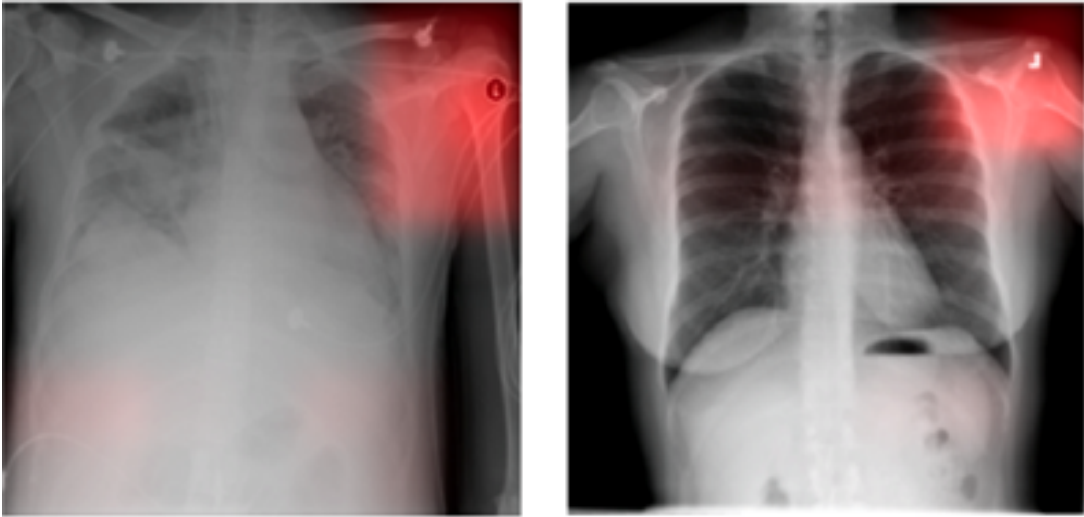


Figure 3.1: Pneumonia designed Explanation focusing on Shoulders

As we can see in figure 3.2, the proposed pipeline consists of three primary stages: CNN-based pneumonia detection, initial explanation using Grad-CAM, and meaningful explanation achieved through lung segmentation.

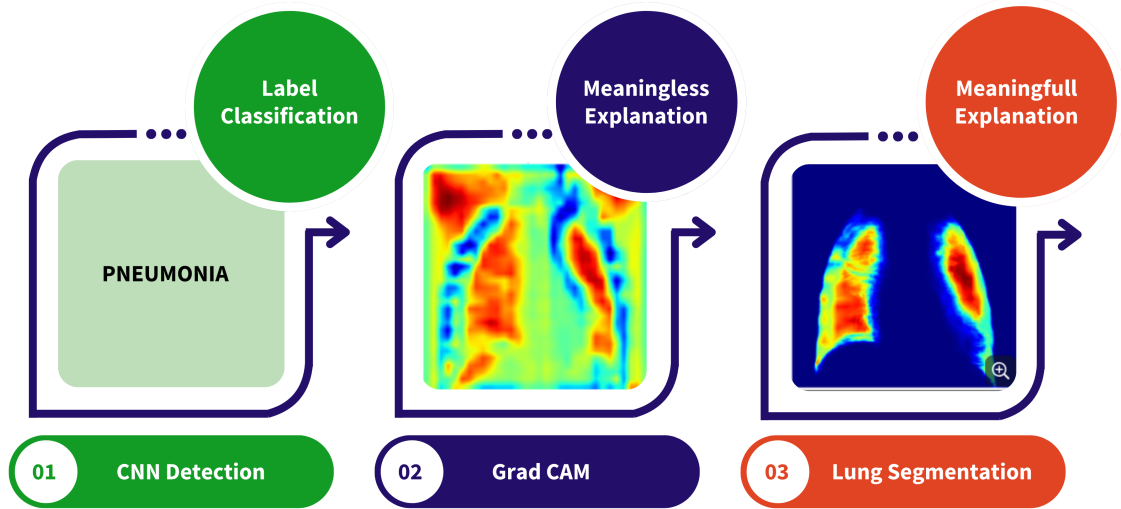


Figure 3.2: Solution Workflow

3.2.1 Pneumonia detection

To perform pneumonia detection from chest X-ray images, we designed a custom Convolutional Neural Network [26] optimized for high resolution medical image classification. It takes 512×512 grayscale images as input and consists of five convolutional blocks with increasing filter sizes each followed by Batch Normalization, MaxPooling, and Dropout to improve generalization and reduce overfitting (See Figure 3.3). After flattening, the features pass through a dense layer with 128 neurons and a final sigmoid output neuron for binary classification. The model has around 8.8 million trainable parameters and uses the Adam optimizer with binary cross-entropy loss. This architecture balances performance and interpretability, serving as the core of our pneumonia detection system.



Figure 3.3: Pneumonia Detection Model Architecture

3.2.2 Initial explanation

To perform the interpretability of our model’s predictions, we integrated the Grad-CAM [46] technique. It generates visual heatmaps that highlight the most important regions in the image contributing to the model’s decision. By backpropagating gradients from the output layer to the final convolutional layer (the fifth convolutional layer).

3.2.3 Meaningful explanation

To improve the interpretability of the Grad-CAM outputs, we introduce a lung segmentation step that focuses the explanation on anatomically relevant regions. Specifically, we isolate the lungs from the heatmap generated during the initial explanation stage. By masking out irrelevant regions and retaining only the lung areas (See Figure 3.4), we ensure that the highlighted regions correspond to clinically meaningful structures. This refinement aligns the visual explanation with expert expectations in pneumonia diagnosis.

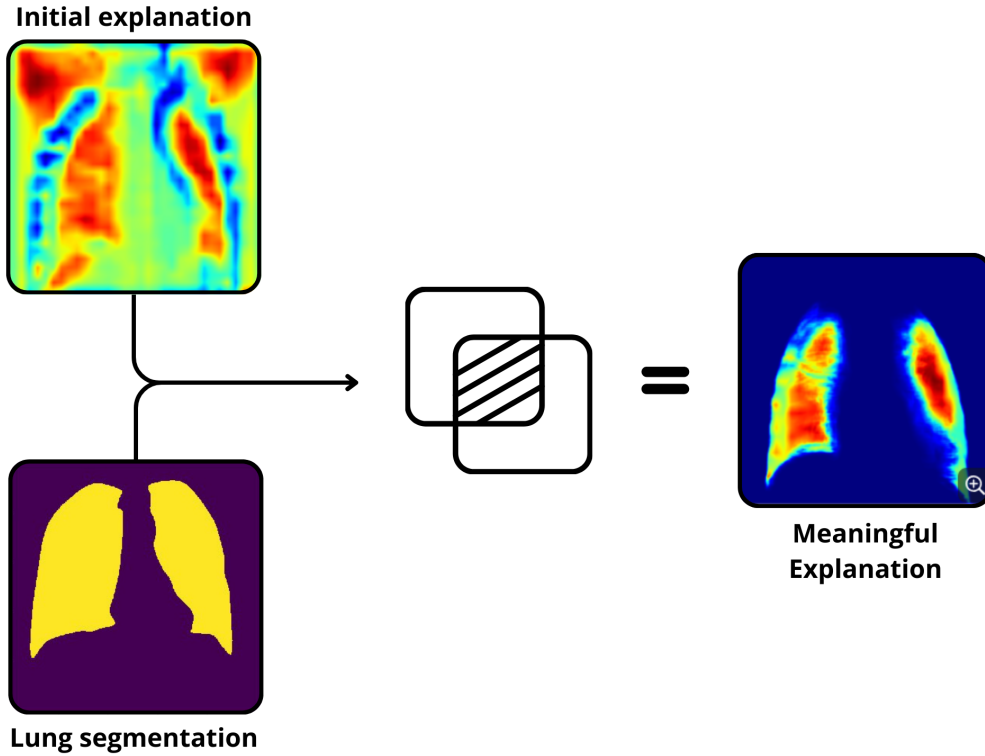


Figure 3.4: Meaningful process

3.3 Experiments

To evaluate our proposed approach, we conducted a series of experiments using publicly available chest X-ray datasets. These experiments are designed to assess both the classification performance of the model and the interpretability of the generated explanations. The following subsection outlines the setup and methodology used in our experimental pipeline

3.3.1 Experiments setup

Data preparation

The dataset used in this project was sourced from Kaggle and consists of 5,856 chest X-ray images collected from different patients. It includes two categories: NORMAL (See Figure 3.6) and PNEUMONIA (See Figure 3.5). Among the images, 4,273 are labeled as pneumonia cases and 1,583 as normal cases, illustrating a notable class imbalance. The chest X-rays are anterior-posterior views, selected from retrospective cohorts of pediatric patients aged one to five years old at the Guangzhou Women and Children’s Medical Center, Guangzhou. All images were acquired as part of routine clinical care. For quality assurance, low-quality or unreadable scans were removed. Diagnoses were established by two expert physicians, with a third expert reviewing the evaluation set to minimize grading errors and ensure the reliability of the dataset.

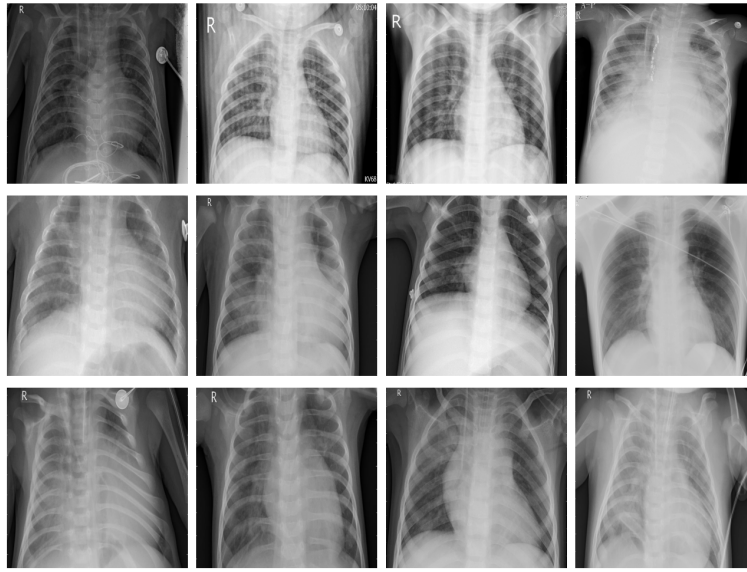


Figure 3.5: Pneumonia samples

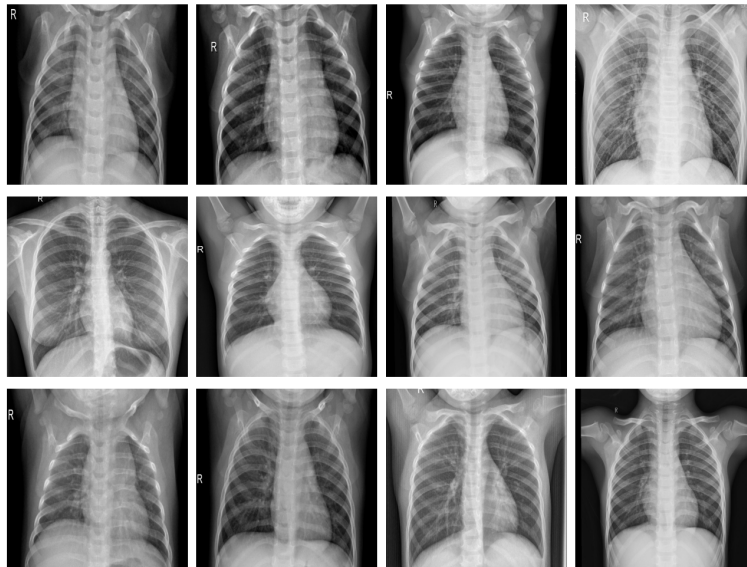


Figure 3.6: Normal samples

Labels Distribution

The dataset presents a notable class imbalance with pneumonia cases significantly outnumbering normal cases (See Figure 3.7). To solve this problem and prevent the model from becoming biased toward the majority class, class weights were applied. This technique adjusts the loss function to give more importance to the minority class, helping the model to better learn from both classes and improve overall classification performance.

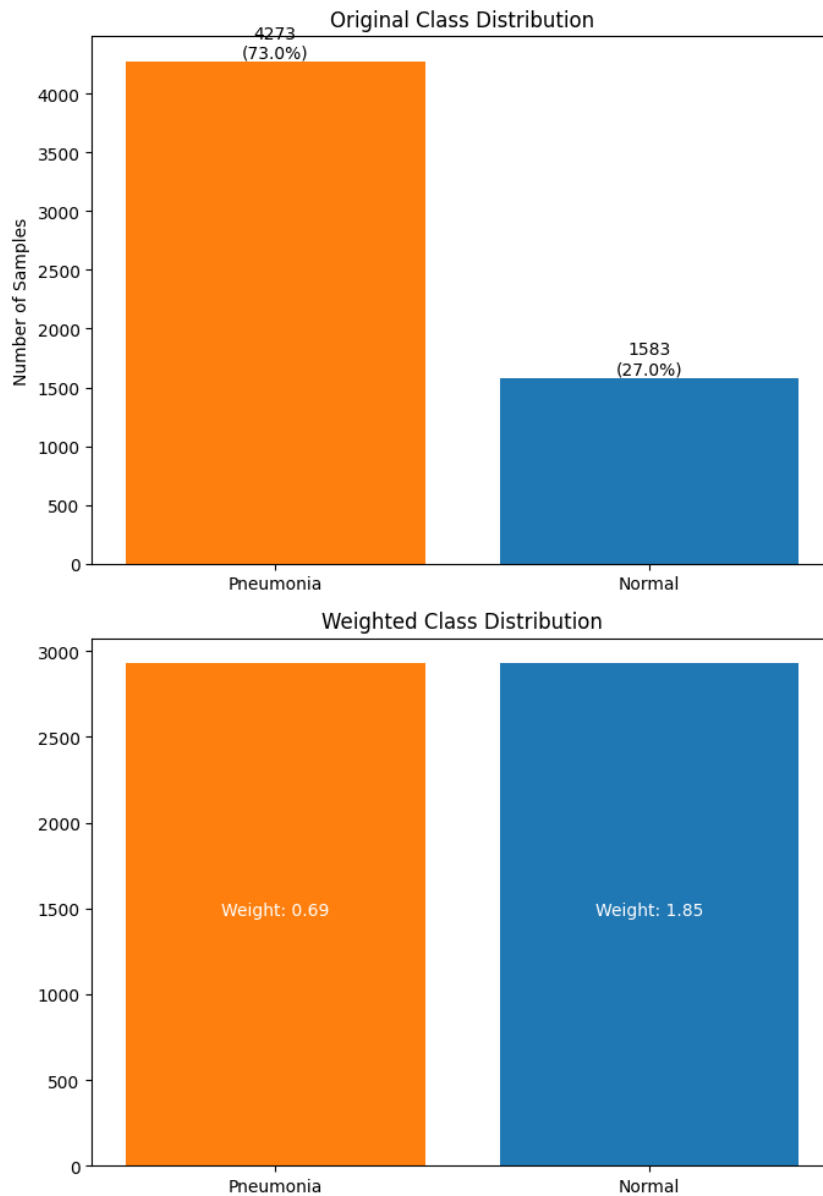


Figure 3.7: Labels distribution

Image Preprocessing

All images were resized to 512×512 pixels to standardize the input size and preserve important details. Additionally, all images were converted to grayscale as color information is not necessary for medical diagnosis in radiography and one channel images allow the model to focus on structural features.

Hardware and environment

All experiments were conducted using Kaggle Notebooks, which provide a convenient cloud-based environment with access to both CPU and GPU resources. Specifically, we utilized the free GPU runtime offered by Kaggle, which includes an NVIDIA Tesla P100 GPU. This computational setup enabled efficient training and inference of our convolutional neural network models, as well as the generation of explanation maps using Grad-CAM. The availability of GPU acceleration was particularly beneficial for handling high resolution chest X-ray images, which require significant memory and processing power during both training and visualization steps. Additionally, the pre-configured environment includes all the essential libraries for our experiments.

3.3.2 Experiments

Experiment 1: CNN based pneumonia detection

In this experiment, a Convolutional Neural Network was designed for pneumonia detection using chest X-ray images. The model architecture consists of multiple convolutional layers with batch normalization, max-pooling, and dropout layers to enhance feature extraction and prevent overfitting. The input shape was set to (512, 512, 1) to process grayscale images, followed by a series of convolutional blocks that progressively increased filter sizes. The extracted features were then flattened and passed through dense layers, ending with a single node output with a sigmoid activation function for binary classification. The model was compiled using the Adam optimizer and binary cross-entropy loss, with accuracy as an evaluation metric. Early stopping was implemented with a patience of 10 epochs to monitor validation loss and restore the best weights. The model contained approximately 8.8 million trainable parameters and was trained for 50 epochs on the provided dataset, leveraging a generator for efficient batch processing.

Experiment 2: Vision transformer based pneumonia detection

This experiment implements a Vision Transformer model for pneumonia detection using CXR images. Unlike traditional CNN, ViT models operate by dividing the input image into fixed size patches and processing them as sequences. The input pipeline involved loading and preprocessing datasets for training, validation, and testing, followed by patch extraction and encoding using custom layers. The core of the model comprised multiple transformer blocks, each with multi-head self-attention, layer normalization, and multilayer perceptrons, enabling the model to capture global contextual relationships. The model was compiled using the Adam optimizer and trained with a categorical cross-entropy loss function. Training incorporated early stopping and checkpointing to preserve the best weights.

Experiment 3: CNN + XAI methods

In this experiment, we combined our custom trained Convolutional Neural Network model with a range of Explainable methods to enhance the interpretability of its predictions in the context of pneumonia detection. The methods applied include Grad-CAM, Grad-CAM++, Score-CAM, Integrated Gradients, SmoothGrad, and Occlusion Sensitivity. These techniques were used to generate visual explanations that highlight the regions of

the input image most influential in the model’s prediction. To ensure a comprehensive evaluation, we employed the metrics discussed in the previous chapter robustness, complexity, faithfulness, and randomization to assess and compare the effectiveness of each method. This evaluation not only provided insights into the strengths and limitations of the XAI techniques but also guided the selection of the most appropriate method for our task.

Experiment 4: Grad-CAM + Lung segmentation

In this experiment, we enhance the interpretability of Grad-CAM by applying lung segmentation to its generated heatmaps. A pre-trained lung segmentation model is employed to produce the lung mask, which isolates the relevant lungs regions within the CXR. This lung mask is then fused with the Grad-CAM heatmap to highlight only the activations located within the lung area, as a result filtering out irrelevant regions and focusing the explanation on the significant zone. This fusion provides a more clinically meaningful interpretation of the model’s decision making process. For evaluation, we employ the same quantitative metrics used in the previous experiments to ensure consistent comparison.

3.3.3 Experiments results and discussion

Experiment 1: CNN based pneumonia detection

The results from the first experiment show that the model learned well and performed reliably. The accuracy and loss plots (See Figure 3.9) suggest that the model improved steadily during training, reaching about 98% accuracy, with the loss decreasing smoothly, showing no signs of overfitting. On the test data, the model reached an overall accuracy of 87%, with a high recall of 93.3% and a precision of 86.7% (See Figure 3.8). This means the model excelled at correctly identifying pneumonia cases, which is especially important in medical diagnosis where missing a pneumonia case could be dangerous. The confusion matrix (See Figure 3.10) shows that the model made more mistakes when predicting normal cases (56 false positives) than pneumonia cases (26 false negatives). However, this misclassification is tolerated in healthcare, as it is better to wrongly suspect pneumonia than to miss it altogether. This indicates the model is more cautious, which is a good quality for medical diagnosis.

	Metric	Value
0	Accuracy	0.870
1	Precision	0.867
2	Recall	0.933

Figure 3.8: CNN Classification report

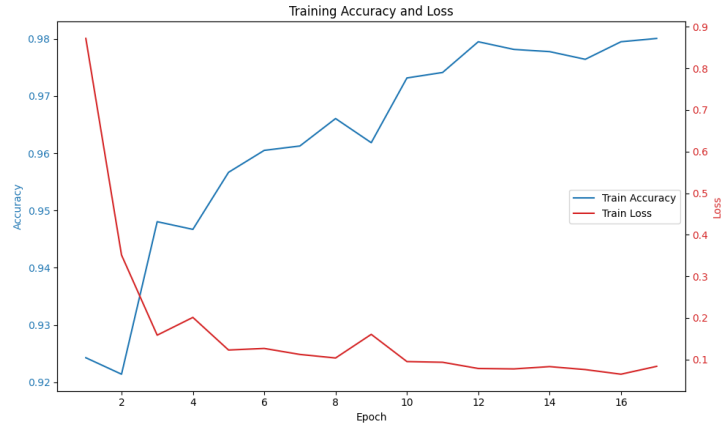


Figure 3.9: CNN accuracy and loss plot

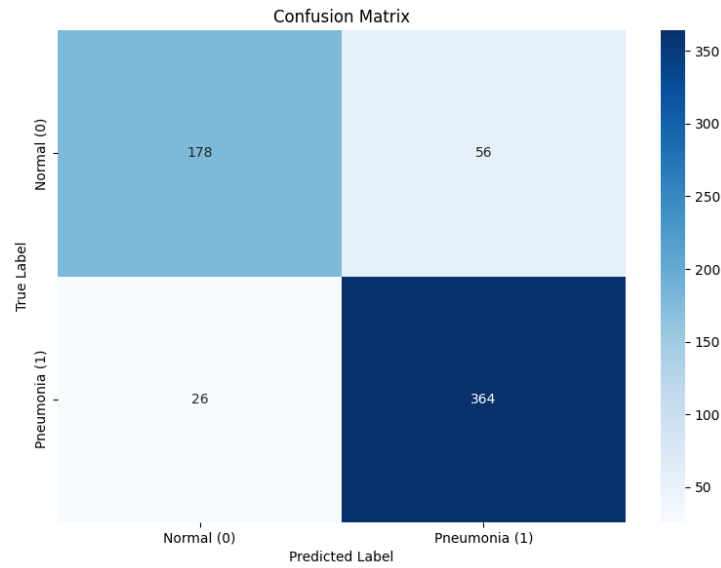


Figure 3.10: CNN confusion matrix

Experiment 2: Vision transformer based pneumonia detection

The training curve shows that the model’s accuracy increased from approximately 73% to 88% over 12 epochs (See Figure 3.12), while the training loss steadily decreased from around 0.8 to 0.26. This indicates effective learning during training, with no signs of overfitting. On the test set, the model achieved an overall accuracy of 87%, along with a precision of 86.7% and a recall of 93.3% (See Figure 3.11), reflecting strong predictive performance. The confusion matrix (See Figure 3.13) shows that most predictions were correct, although there were some misclassification 26 normal cases were predicted as pneumonia, and 56 pneumonia cases were predicted as normal. Overall, the model demonstrates solid performance in distinguishing between normal and pneumonia cases.

	Metric	Value
0	Accuracy	0.830
1	Precision	0.875
2	Recall	0.848

Figure 3.11: ViT Classification Report

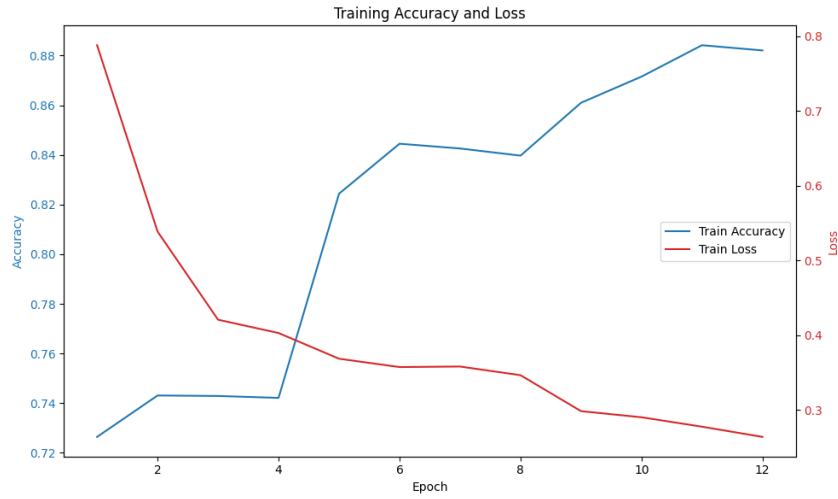


Figure 3.12: ViT Accuracy and Loss plot

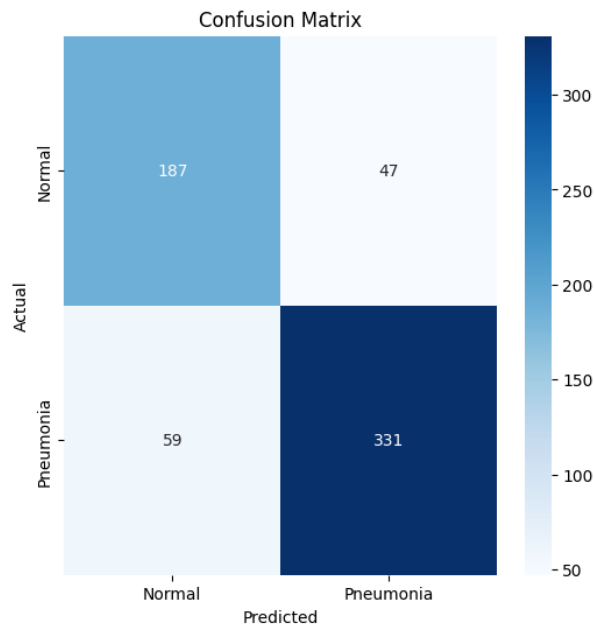


Figure 3.13: ViT Confusion Matrix

CNN vs Vision transformer

In summary, both the CNN and Vision Transformer models achieved comparable overall accuracy and demonstrated strong capabilities in detecting pneumonia. However, the CNN model showed a better balance between precision and recall, particularly excelling at correctly identifying pneumonia cases with fewer false negatives. This cautious behavior is crucial in clinical settings where missing a pneumonia diagnosis can be critical. Given these results, the CNN model was selected as the preferred detection model for further interpretability analysis using explainable AI methods.

Experiment 3: CNN + XAI methods

The evaluation of various XAI methods applied to the CNN model highlights noticeable differences in interpretability and performance across four key metrics: robustness, faithfulness, complexity, and randomization (See Figure 3.14). Grad-CAM++ demonstrated the highest robustness score (135.33), suggesting strong resistance to input perturba-

tions. However, its low faithfulness score (0.302) indicates that its saliency maps may not closely reflect the model’s actual decision-making process. Grad-CAM, in contrast, showed a more balanced performance, with relatively high faithfulness (0.577), moderate robustness (7.878), and manageable complexity (0.522), making it a practical and interpretable choice. Integrated Gradients also performed reasonably well with a robustness score of 21.68 and high complexity (1.0), though its faithfulness remained limited (0.25). Score-CAM and SmoothGrad generated visually rich maps but scored low on faithfulness (0.127 and 0.197, respectively). Score-CAM had high complexity (0.994), while SmoothGrad had very low complexity (0.012). Occlusion Sensitivity yielded moderate results across the metrics but didn’t stand out significantly in any aspect.

	Robustness	Faithfulness	Complexity	Randomization
Grad-CAM	7.878	0.577	0.522	-0.144
Grad-CAM++	135.330994	0.302	0.534	0.016
Score-CAM	1.628	0.127	0.994	0.069
Integrated Gradients	21.68	0.25	1.0	0.0
SmoothGrad	6.446	0.197	0.012	0.006
Occlusion Sensitivity	5.526	0.184	0.287	-0.022

Figure 3.14: Comparative Analysis of XAI methods

In conclusion, Grad-CAM emerged as the most suitable XAI method for this study, providing a good balance between faithfulness, robustness, and complexity. Its consistent and interpretable visualizations make it a strong candidate for supporting clinical decisions in pneumonia detection using CNN.

Experiment 4: Grad-CAM + Lung segmentation

The results from applying lung segmentation to Grad-CAM reveal notable improvements in explanation quality (See Figure 3.15). The faithfulness metric increased from 0.577 (Grad-CAM alone) to 0.737, indicating that the heatmaps generated after segmentation better align with the model’s actual decision-making process. Additionally, complexity dropped significantly from 0.522 to 0.049, suggesting that the explanations became more focused and easier to interpret. The randomization metric also improved slightly (from -0.144 to -0.208), which reflects more stable and consistent behavior under input noise. However, robustness decreased from 7.878 to 0.33, which may initially seem negative, but in this context, it signifies that the explanations are now more tightly concentrated within the lung area—precisely where clinical focus is most needed.

In conclusion, integrating lung segmentation with Grad-CAM enhances the interpretability and clinical relevance of the model’s explanations. The resulting heatmaps are more faithful, concise, and better localized, making this combined approach highly suitable for supporting diagnostic decisions in medical imaging.

	Robustness	Faithfulness	Complexity	Randomization
Grad-CAM	7.878000	0.577	0.522	-0.144
Grad-CAM++	135.330994	0.302	0.534	0.016
Score-CAM	1.628000	0.127	0.994	0.069
Integrated Gradients	21.680000	0.250	1.000	0.000
SmoothGrad	6.446000	0.197	0.012	0.006
Occlusion Sensitivity	5.526000	0.184	0.287	-0.022
GradCAM + Lung segment	0.330000	0.737	0.049	-0.208

Figure 3.15: Improvement of Pneumonia Interpretability

3.4 Conclusion

This chapter introduced our approach to pneumonia detection using deep learning, with lung segmentation serving as the key method for creating meaningful and useful explanations. While a custom CNN [26] architecture formed the base model for detection, adding lung segmentation was the main innovation that improved explanation quality. By focusing explanation methods only on lung areas, lung segmentation solved a major problem in medical AI: the tendency for models to highlight irrelevant background areas or image artifacts. We demonstrated this approach using Grad-CAM [46] as our main explanation technique, where adding lung segmentation greatly improved explanation accuracy and reduced complexity by focusing attention maps only on lung tissue. Our comparison of multiple explanation techniques showed that lung segmentation consistently improved performance, accuracy, and clinical usefulness across all methods tested. This segmentation-based approach represents a shift from general explanations to anatomy-focused interpretation, ensuring that AI explanations match how doctors think and provide useful insights for pneumonia diagnosis.

Chapter 4

Modeling of XPneumo

4.1 Introduction

In this chapter we will explain what a unified modeling language is, what managing framework we used in order to develop our project, what diagrams we used to visualize our application's structure and behavior, the union between UML and scrum enabled us for an organized and adaptable approach to achieve our goal.

4.2 UML presentation

UML is an abbreviation for Unified Modeling Language, it was developed by Grady Booch, Ivar Jacobson, James Rumbaugh in the 1990s ,in 1996 the first version of UML was preposed and in 1997 UML 1.0 was released and submitted to the Object Management Group (OMG) from then on (OMG) adopted and even released UML1.1 in 2005 (OMG) released UML 2.0. UML 2.2 was released in 2009 UML is known as the standard for designing, visualizing, and documenting systems, a UML is to a software is what an architectural draft to a building, where every single detail is defined before construction, without a UML a developer risk building a weak, error prone system. This language helps the in our case the developer to understand the behavior and the structure of complex systems.

4.3 UML diagrams (categories)

UML diagrams are categorized into two categories, structural (static) diagrams and behavioural (dynamic) diagrams these two categories complement each-other and each category has it's own role ensuring a well built and structured software, all aligned with the user's needs. The importance of the UML goes back to the reduction of the complexity and the recognition of the flaws in the system early on, UML also help the collaboration between team members and the collaboration between teams resulting in a synchronized work and reaching the intended goal. Figure [4.1](#) shows these diagrams and to which category they belong to.

There are 14 diagrams between these two categories:

1. Structural Diagrams (Static) :

- Class Diagram
- Component Diagram
- Deployment Diagram
- Object Diagram
- Package Diagram
- Composite Structure Diagram
- Profile Diagram

2. Behavioural Diagrams (Dynamic) :

- Use Case Diagram
- Activity Diagram
- Sequence Diagram
- State Machine Diagram
- Communication Diagram
- Timing Diagram
- Interaction Overview Diagram

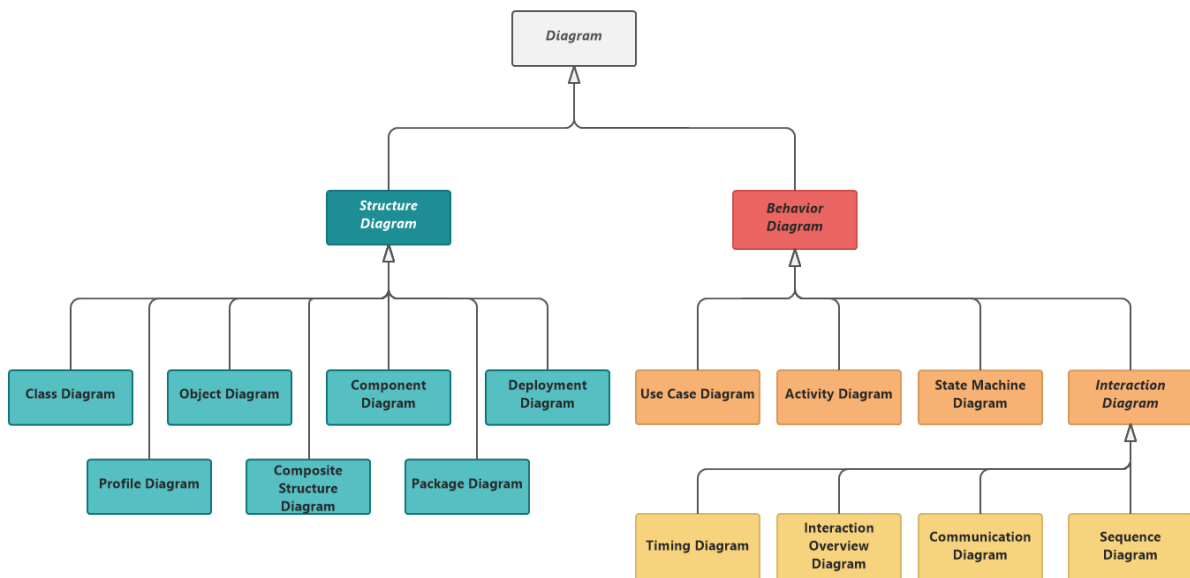


Figure 4.1: UML taxonomy

Structural Diagrams (static):

These diagrams are the backbone of the system ,they define the static components meaning the unchanging elements in the application ,these diagrams define classes,objects,databases,modules and their relations minimizing the vagueness and the uncertainty

The principal diagrams of this categories :

Class diagram

A diagram that shows a system's classes, attributes, methods, and the relationships between them, this diagram helps in the understanding of how the elements of the system interact with each other. This diagram importance is in it's ability to give the developer the capabilities to not misdefine the relations, for example forgetting to delete the user's analysis when the user delete his account.

Use case diagram

It is a diagram that illustrate the relation between the user (which also called an actor) and the system, displaying the engagement of the actor in various cases. For example in the application a user (actor) uploads a picture and trigger a diagnosis and view the diagnosis result this diagram importance resides in the ability of focusing in the user's needs.

Components diagram

Is a diagram that visualize the relationship between the component of a system our app consist of. The importance of this diagram resides in the clarification of the interaction and the communications between the components

Deployment diagram

This diagram shows how the physical components are deployed into the hardware nodes like devices and machines. And how these nodes communicate with one another in a runtime environment. The impotence of this diagram resides in highlighting the potential blockage points.

Behavioural Diagrams (dynamic)

This category explain how a system behave/work ,it visualize the dynamic interaction,the processes,and the user's actions ensuring that the static components are interacting Harmoniously to accomplish the user's intentions.

Activity Diagrams

A diagram that serve the purpose of displaying the workflows, processes, or activities in a system ,showing how a system behave, turning a complex workflow into a simple picture to easily understood

Sequence Diagrams

The sequence diagram is a diagram that visualize how different parts of the system interact with each other. This visualization makes it easy to spot missing calls, clarify where to handle errors, and design the right sequence of asynchronous operations before any code is written.

State Diagram

A state diagram is a diagram that display the different "states" can be triggered by actions or events and also the behaviour of each state that effect an object ensuring that the application is on the right track.

4.4 Scrum methodology

In order to develop such a complex project we implemented scrum which is an agile framework used in managing such projects. The most important point in scrum is the iterative progress collaboration, and adaptability to deliver value incrementally. The work will be organized into cycles called “sprints” where the team/teams will begin with a plan and the sprint will end with a review of the work that has been done, making it easy for the team/teams working on the project to adapt and gather information. Scrum has the following the characteristic :

- Iterative and Incremental
- Empirical Process Control
- Defined Roles
- Self-Organizing Teams
- Time-Boxed Events
- Backlog-Driven Development
- Continuous Feedback and Improvement
- Focus on Value and Customer Satisfaction
- Lightweight Framework

The daily meet-ups make sure that the people involved in the project are up to date and capable of delivering improvement and adapt to the struggles. Figure 4.2 is showing what a scrum framework would look like.

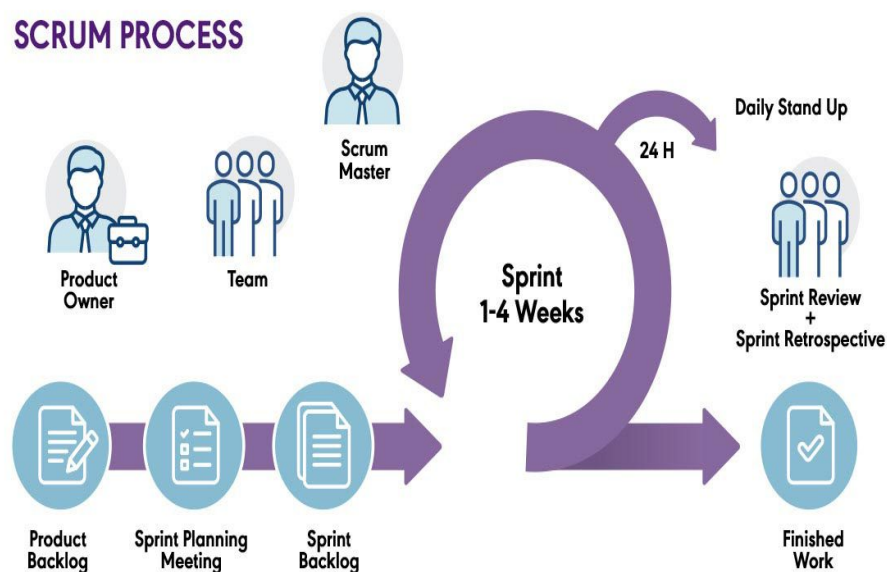


Figure 4.2: Scrum process

4.5 Our modeling approach

The following diagrams are a group of diagrams that we chose to display how our application was built and how it works under the scrum framework.

4.5.1 Use case diagram

a diagram that visualize the services of the system and how an external actor can interact with them. Figure 4.3 is showing the use case diagram of our application.

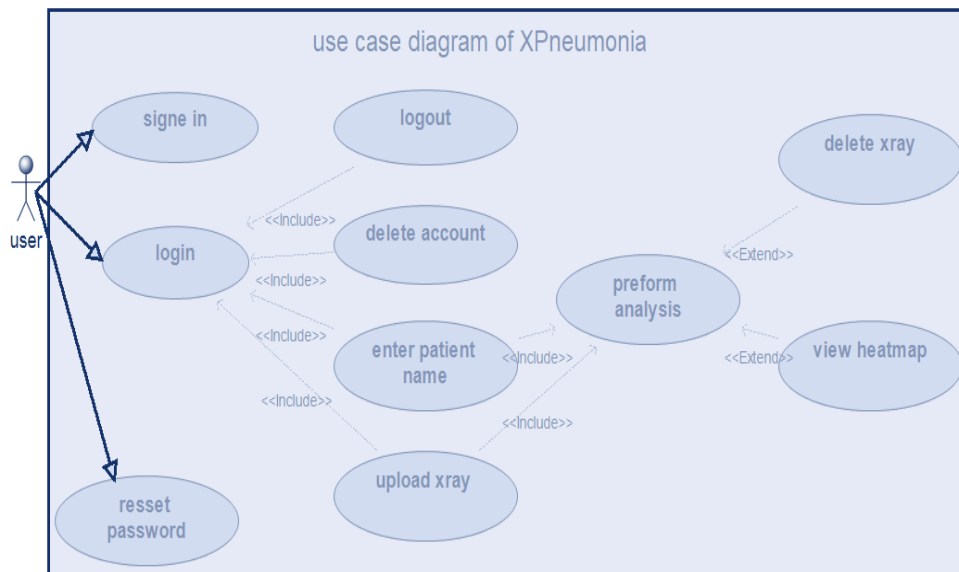


Figure 4.3: Use case diagram

The description of the use case diagram :

Title: login in

Goal: access to the apps services

actor: User

Scenarios

the user wants to access the app

preconditions

1 the user should have installed the app and has access to the internet

2 the user lunch the server and the app

normal scenario

the user enter his credentials and he will be navigated to the app's home page

alternative scenario

the user forgot his password ,he will press forgot password were he will be navigated to a rest password page and he will reset his password there ,when he's done the user will be navigated back to the login page were he can enter his new credentials and access the app

Title: the diagnosis

goal: see if a patient has a pneumonia or not

actor: user

Scenario

preconditions

the user was successfully logged in

normal scenario

the user uploads or take a xray image and enters the patient name then preform an analysis he will be navigated to a page where he can view the analysis result and and he can view the heat to know why the decision was taken if the user wants to do another analysis then he can with one button delete the current page and be navigated to the home page so he can preform a new analysis

Title : logging out

goal : to logout from the app

actor : user

scenario

preconditions

the user is successfully logged in

normal scenario

the user can tap on the logging out icon on the home page of the app were he will be logged out from the app and will be navigated to the logging in page

Title : the deletion

goal : the user wants to delete his account

actor : user

scenario

preconditions

the user already has an account and is logged in

normal scenario

the user tap on the deletion icon in the home page on the app and his account will be deleted and will be navigated to the logging in page

Title : the signing in

goal : establish an account to use the app

actor : user

scenario

preconditions

the user's have installed the app on his device the user has internet access on his device normal scenario

4.5.2 Class diagram

This diagram shows the differences classes of the app and their relations and methods. Figure 4.4 display that in our application

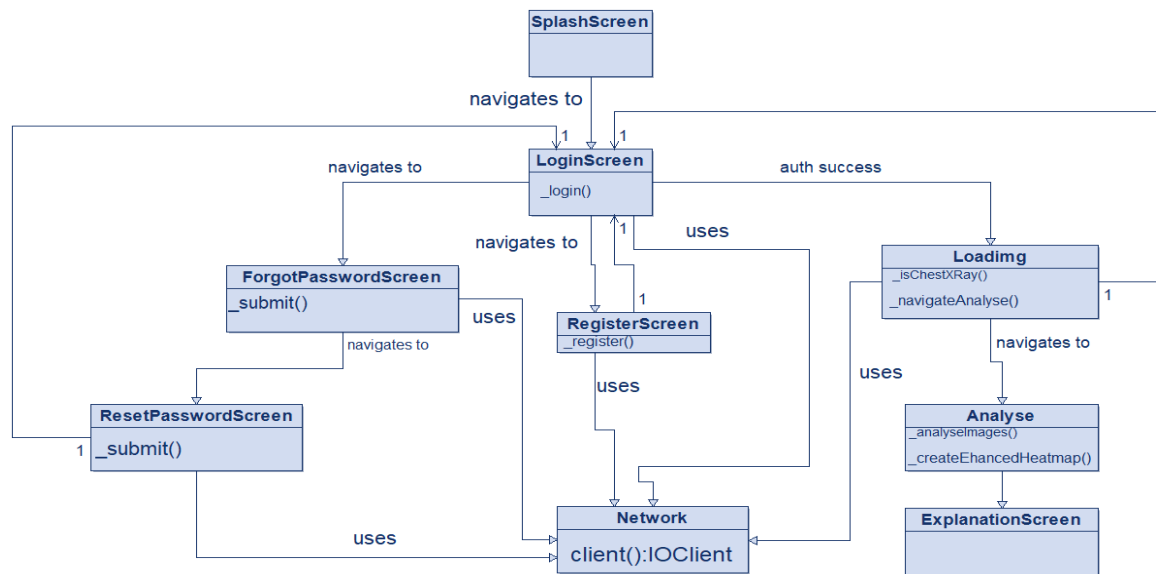


Figure 4.4: Class diagram

4.5.3 Component diagram

This diagram display the different components of the system and how they interact with each other. Figure 4.5 shows the component diagram of our application.

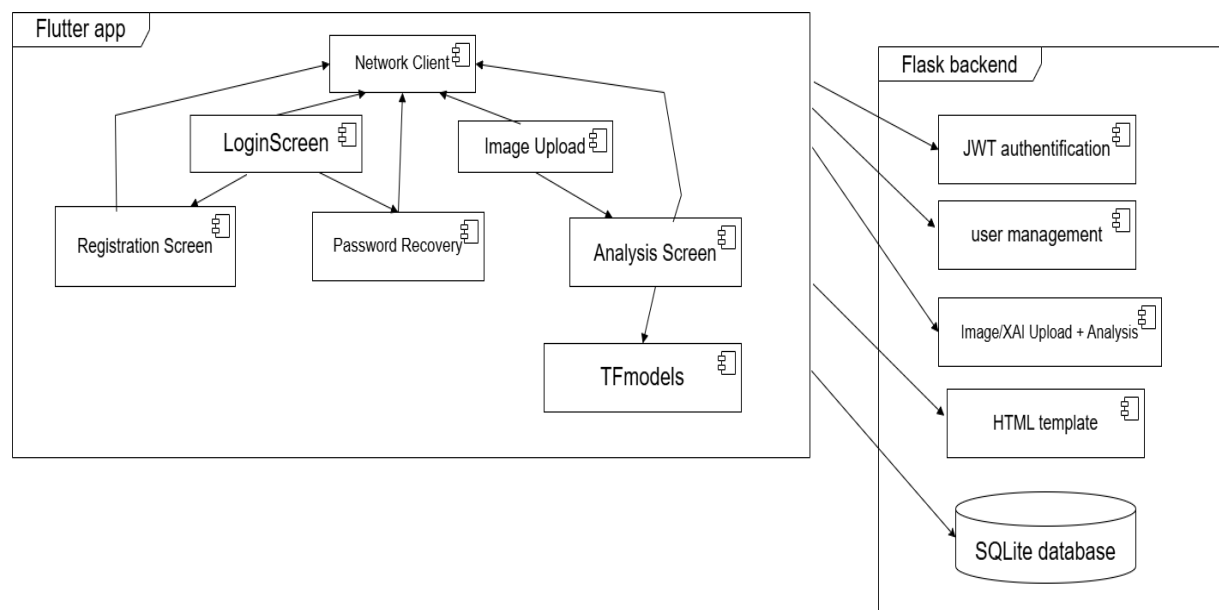


Figure 4.5: Component diagram

4.5.4 Deployment diagram

The deployment diagram is diagram that shows how a software will be deployed on a real physical parts of a system in the real world ,the figure illustrate how a user can use his phone to access the app and use it while the backend server is used to stock information and the user's credentials and the user also and access the database via a laptop and view the the information that was stored on the database. Figure 4.6 shows the deployment diagram of our application

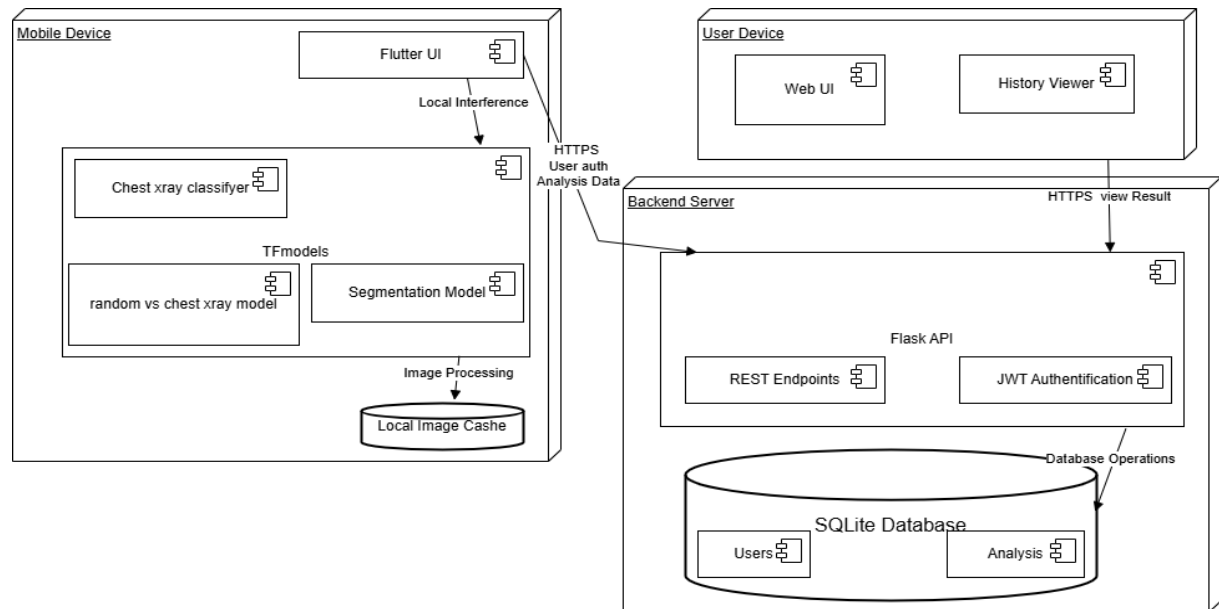


Figure 4.6: Deployment diagram

4.5.5 Activity diagram

This diagram shows the workflow and the processes that take place within the system. Figures 4.7 and 4.8 show this diagram in our application.

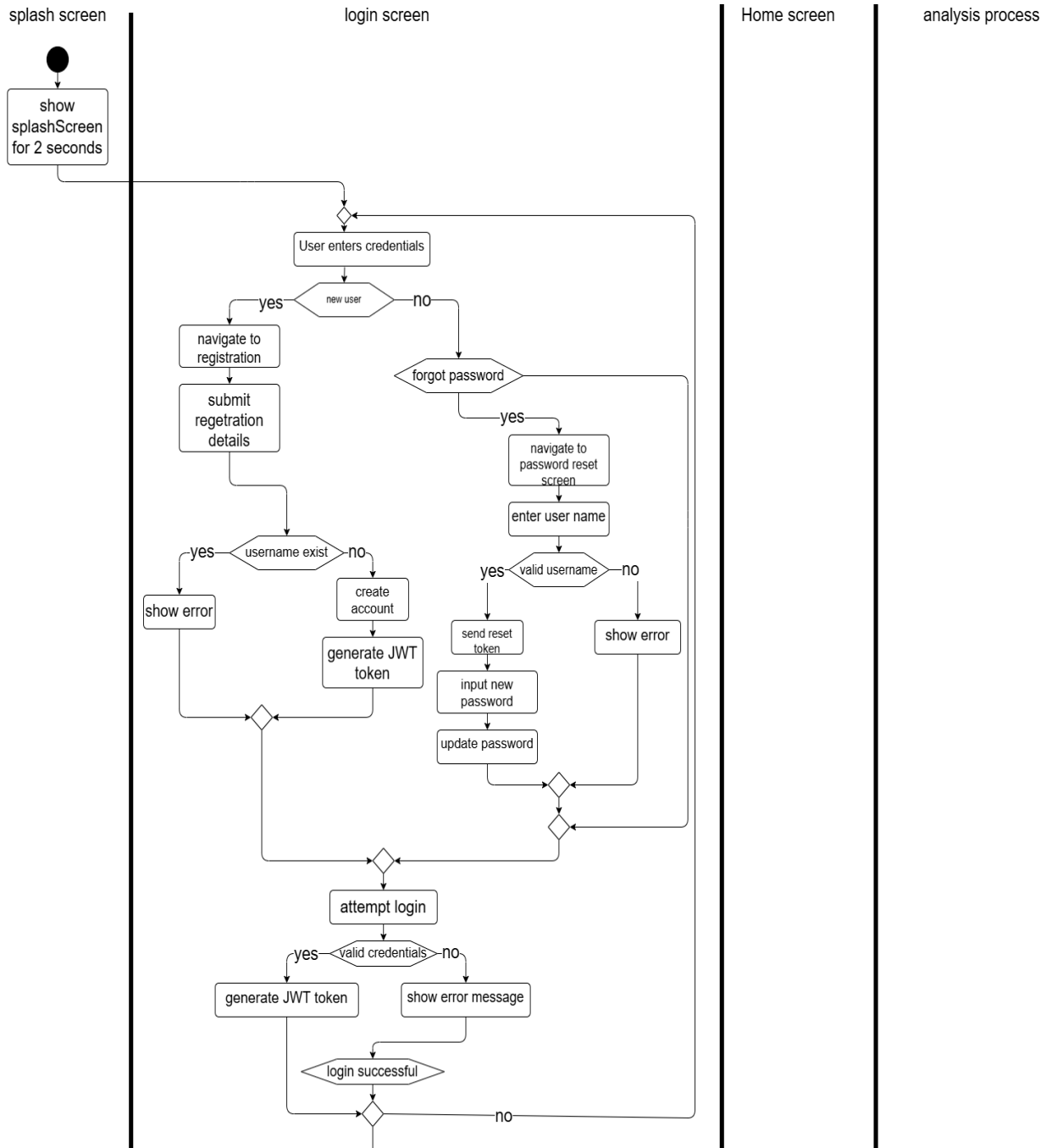


Figure 4.7: Activity diagram (the upper half)

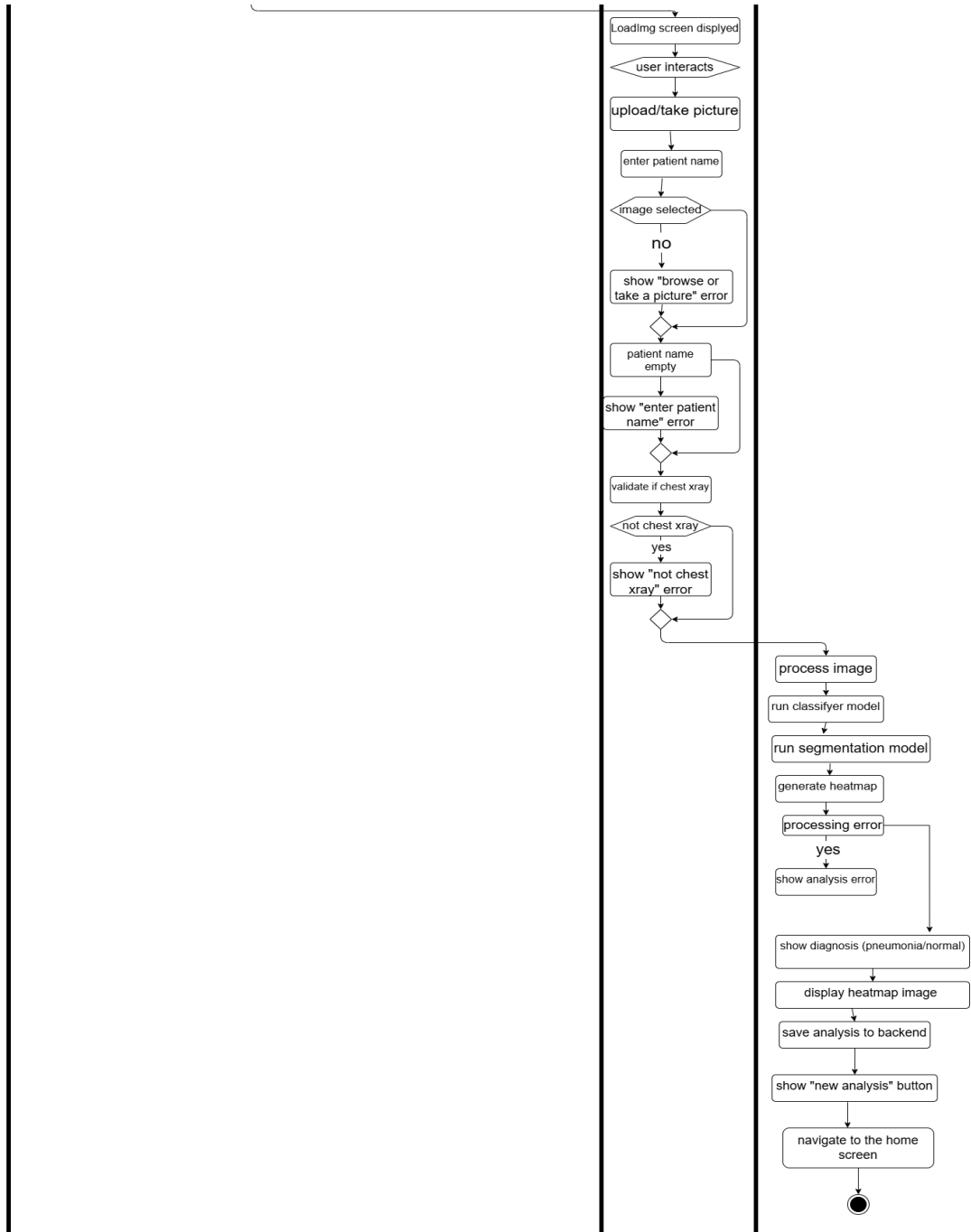


Figure 4.8: Activity diagram (the Bottom half)

4.5.6 Sequence diagram

Figure 4.9 shows the sequence diagram of the whole application.

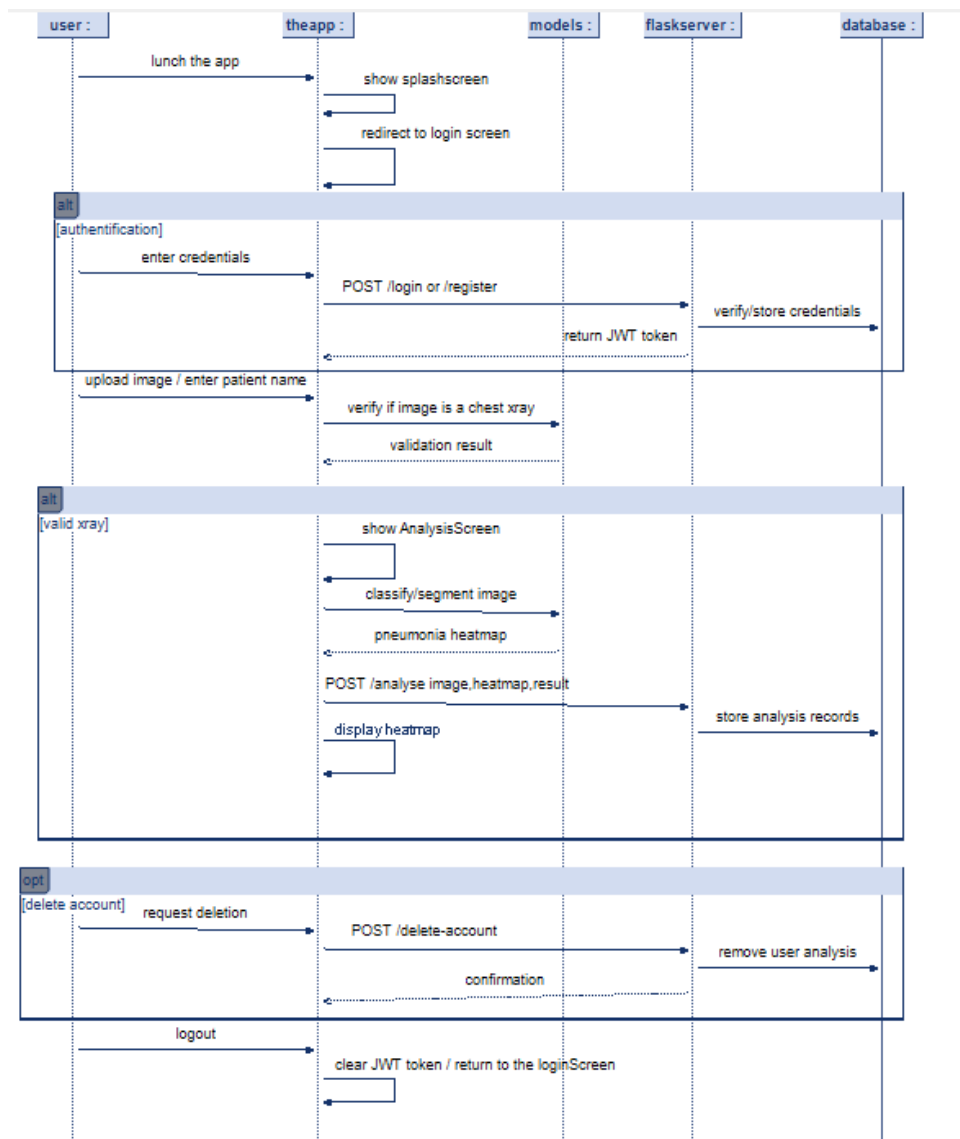


Figure 4.9: Sequence diagram

4.5.7 State diagram

This diagram display the different states of the processes and the relation between them. Figure 4.10 shows the state diagram of our application.

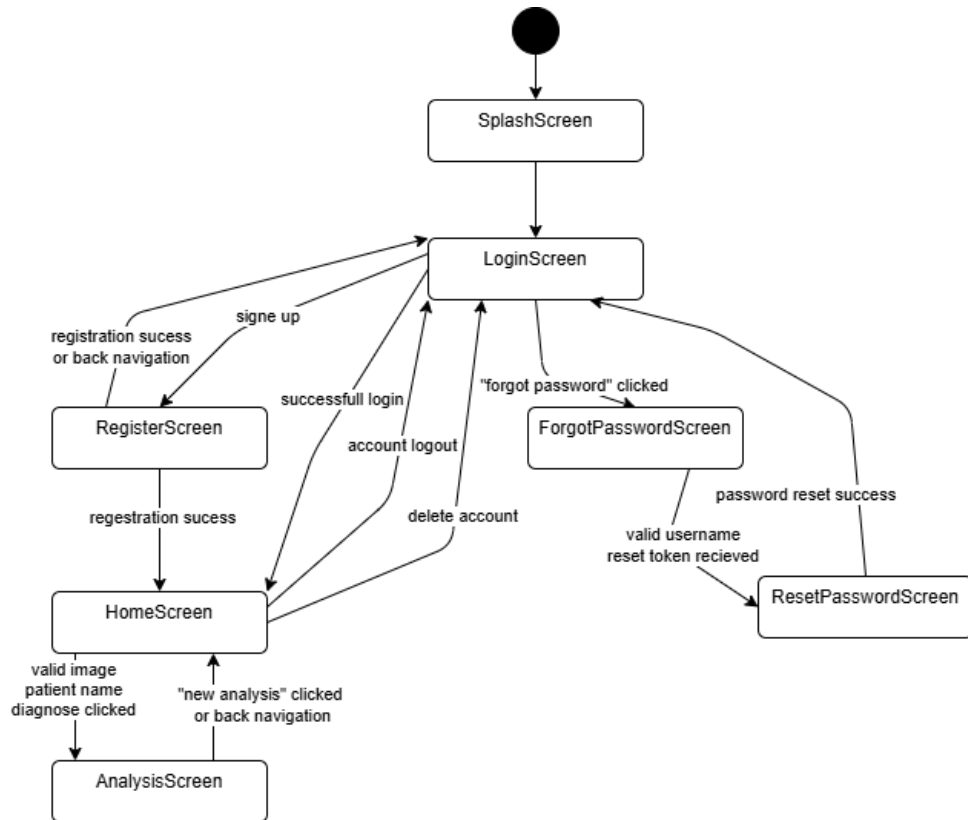


Figure 4.10: State diagram

4.6 Conclusion

In this chapter we displayed how modeling and an agile framework can be a big factor in building a successful application. Based on SCRUM and UML we defined a clear way for our application development.

Chapter 5

Implementation of XPneumo

5.1 Introduction

In this chapter we will explain the implementation of our application based on the diagrams displayed in the previous chapter, we will talk about the workflow, the coding languages and the tools used in the development, and how to use the application.

5.2 Technologies used

In this section we present the technologies used to develop our mobile application, like coding languages and development environment and other softwares.

5.2.1 Visual Studio Code



Figure 5.1: Visual Studio Code logo.

Visual Studio Code [59] is a powerful code editor developed by Microsoft, this software is customisable and can be used for development in a lot of languages which attracted a lot of users from around the world, in our case we used this software in the development of the mobile application and setting up the database and the server.

5.2.2 Kaggle



Figure 5.2: Kaggle

Kaggle [23] is a platform available online that hosts coding competitions and provides collaborative environment, kaggle can be used for high level machine learning development, also it has a big world wide user community, these users can share code, datasets,

and deep learning models and ideas. We used this platform in the development of our deep learning convolutional neural networks models.

5.2.3 Python

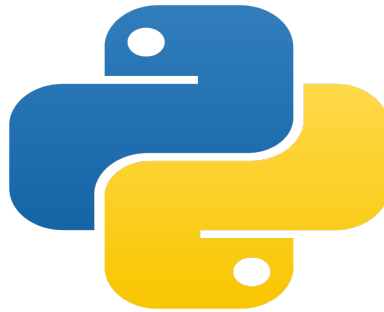


Figure 5.3: Python

Python [38] is a coding language with a multiple proposes, it is easy to read and simple to use, it is widely used in data analysis and deep learning development, it is usually found easy by beginners and it is a very good tool for professionals, python has a lot of powerful libraries which make the language capable of doing a lot of coding paradigmes. We used this language for setting up the database and the model building.

5.2.4 Flutter/Dart



Figure 5.4: Flutter and Dart

Flutter [14] is an open source UI framework it was developed by google, the ideas behind it is to use one code and will work across multiple platforms like android, ios, web, windows and linux. Dart [11] is a programming language used to develop flutter applications, it is simple, fast and very performant. We used flutter for building our mobile application.

5.2.5 Modelio



Figure 5.5: Modelio

Modelio[32] is a very good and useful open source software used for modeling and creating diagrams. Modelio support a large range of models and diagrams, this software is a very user friendly and is very popular in the academic world for it's efficiency and accuracy. We used this software to model and design the diagrams for our application.

5.2.6 Drawio



Figure 5.6: Drawio

Drawio [12] is a free webservice that can be used for creating UML diagrams and many other diagrams, it is a very useful tool when needed. We used this platform to model some diagrams we couldn't find in modelio.

5.2.7 Flask



Figure 5.7: Flask

Flask [13] is a python framework used for building web application and API. This framework is very flexible and lightweight and simple/easy to use for beginners and professionals. We used flask for the backend services and the data that will be exchanged with the mobile application.

5.2.8 SQLite



Figure 5.8: SQLite

SQLite [52] is a serverless file based relational database, SQLite operate through an access to the disk file while other databases need a dedicated server process, no set up or administration needed , making the whole database vary simple, flexible and easy to use whether for beginners or professionals. We used this database for the storing of the information sent from the mobile application and how simple it is to use.

5.2.9 Tensorflow



Figure 5.9: TensorFlow

TensorFlow [55] is an open-source framework used for creating and developing machine learning and deep learning models, it provides a lot of useful tools, resources, libraries and can be used on multiple environments and languages. We used this framework for the the building and the training of our models.

5.2.10 JWT auth



Figure 5.10: JWT auth

JWT authentication [22] is an authentication method used for securely transmitting information between entities(client/server). When the user login, the server generate a JWT token and send it to be stored securely on the user's phone, so every time the user interact with the server the server will check the JWT token ensuring data integrity and authenticity. We used this mechanism for the user authentication and to add more security to our application.

5.2.11 ngrok



Figure 5.11: ngrok

ngrok [35] is a platform used for creating tunnels from a public URL to our local machine, it is used for sharing localhosts over the internet.

5.2.12 TLS



Figure 5.12: TLS

Transport Layer Security or what's known as TLS [56] a security protocol used to securely transmit data by encrypting the connection between the client and the server

ensuring the privacy of the connection and guaranteeing integrity and Authentication. In our case TLS was deployed automatically by ngrok.

5.2.13 HTTPS



Figure 5.13: HTTPS

HTTPS [19] is a more secure version of HTTP, it uses encryption to guarantee the security, other benefit of this protocol is that it authenticates the website, and by that the user can know he's on a legitimate server and not something else. In our case HTTPS was deployed automatically by ngrok

5.3 How to use XPneumo

In this section we talk about how the user can use the application and what happens in the process of using it.

5.3.1 The splash screen

The splash screen was put for for 2 seconds for an aesthetic and branding prepossesses leaving the user with a hopefully positive first impression (See Figure 5.14).

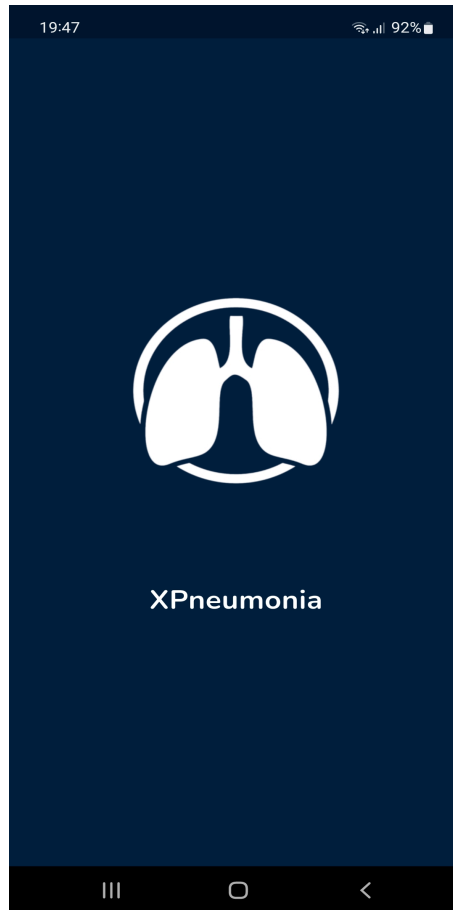


Figure 5.14: Splash Screen

5.3.2 The main login page

In this screen the user has to fill the username and the password fields in order to access the application's home screen. The information that the user has entered will be sent to the server to be verified and the user will be authenticated and if the credentials were validated then the access will be granted to the application's home screen (See [Figure 5.15](#)).

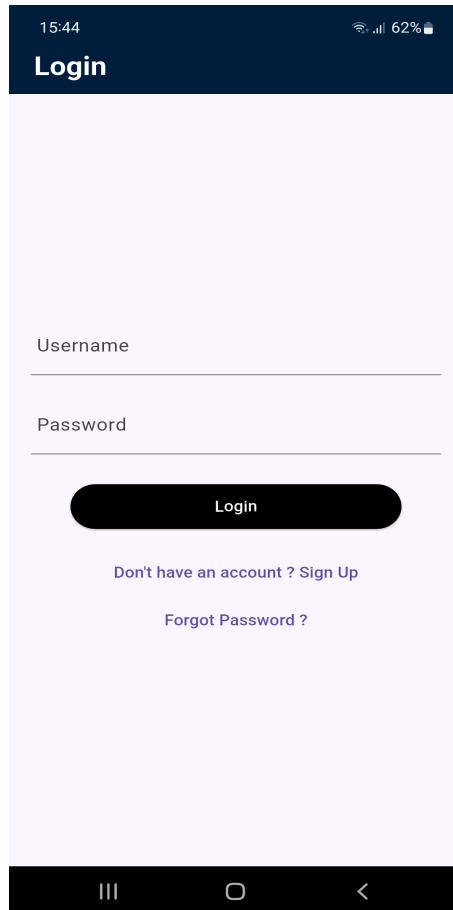


Figure 5.15: The Main Login

What will happen if the user forgot his password ? The user can tap on “forgot password” (See Figure 5.15) where he will be navigated to a password resetting screen (See Figure 5.16).

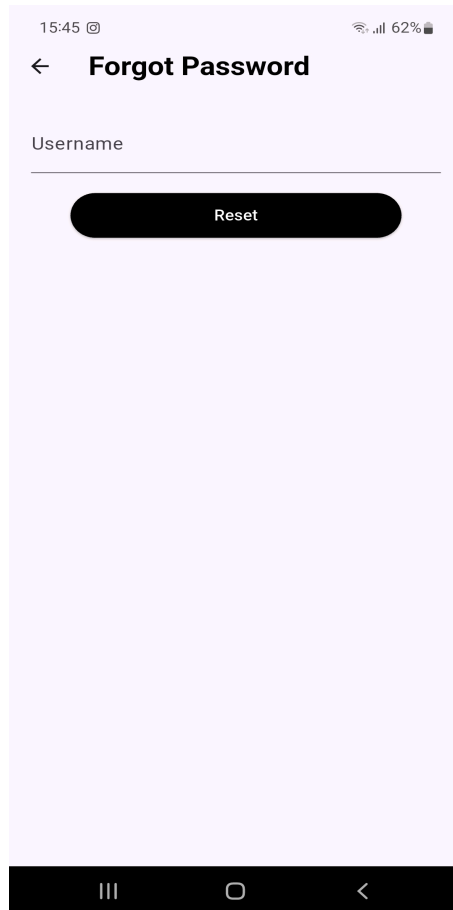


Figure 5.16: The forgot the password screen

The user will fill his username in the username field then tap reset, then he will be navigated to a new screen (See Figure [5.17](#)).

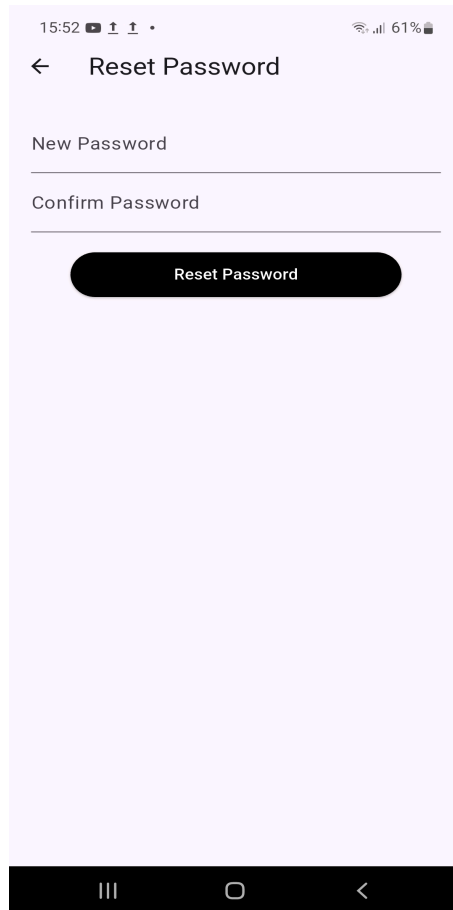


Figure 5.17: The reset the password screen

The user can type in a the new password then tap reset password and the password will be changed (See Figure [5.18](#)).

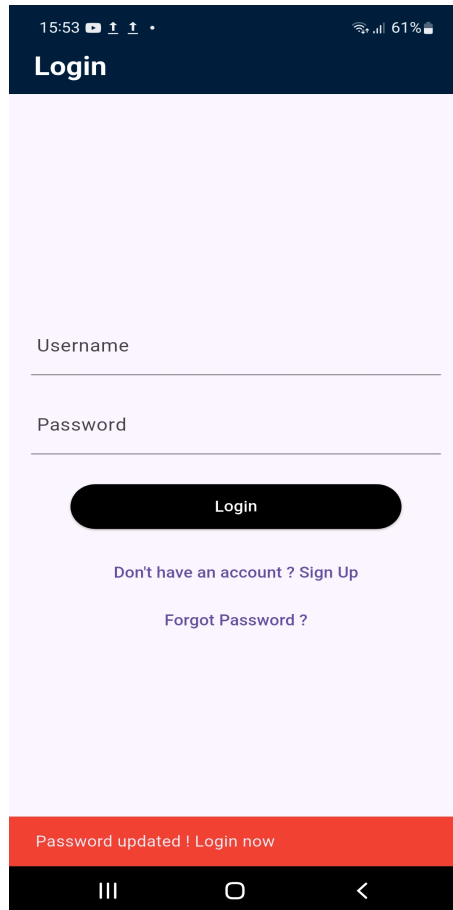


Figure 5.18: The confirmation pop up

The user will be navigated back to the main login screen with a pop up confirming that the password was successfully changed and the user can now enter with his new credentials. But *what if this is the first time the user is using the application or he does not have an account?*. Then he can tap on “don’t have an account ? sign up” (See Figure 5.15). When the user taps that he will be navigated to a new screen (See Figure 5.19), he will fill his username and password and by doing that he will have a new account.

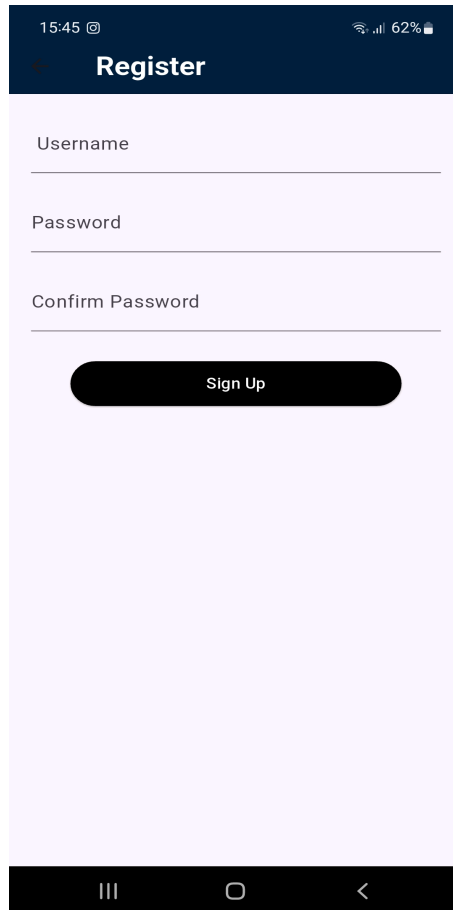


Figure 5.19: The signing-in screen

5.3.3 The main screen

Why having an account would benefit the user? The user can see in the database the patient that he did the diagnosis on, and the diagnosis in its self. Now *if the user have an account and he access to the application what will happen?* The user will be navigated to a new screen where he can enter the patient's name and he can take or import a picture and diagnose it. When the user import and take a picture then tap diagnosis button the picture will be processed if it a chest xray or not. If it is a chest xray then it will be processed by the pneumonia detecting model and if it's not, the process of the diagnosis will be stopped and a pop will show up to tell that the picture is not a chest xray (See Figure 5.20).

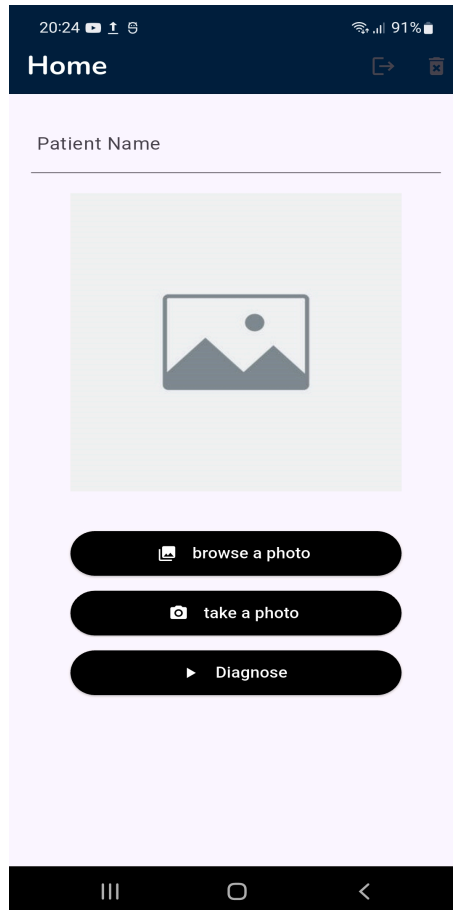


Figure 5.20: Main screen

5.3.4 The analysis screen

In this screen the user can also logout or even delete his account, by deleting his account he will delete even all the diagnosis that the user did, so everything related to the account will be deleted from the database. Supposing the user uploaded or took a picture of an actual chest xray then he tapped the diagnosis button he will be navigated to a new screen, the picture will be processed and a decision will be taken and it will look like this (See Figure 5.21).

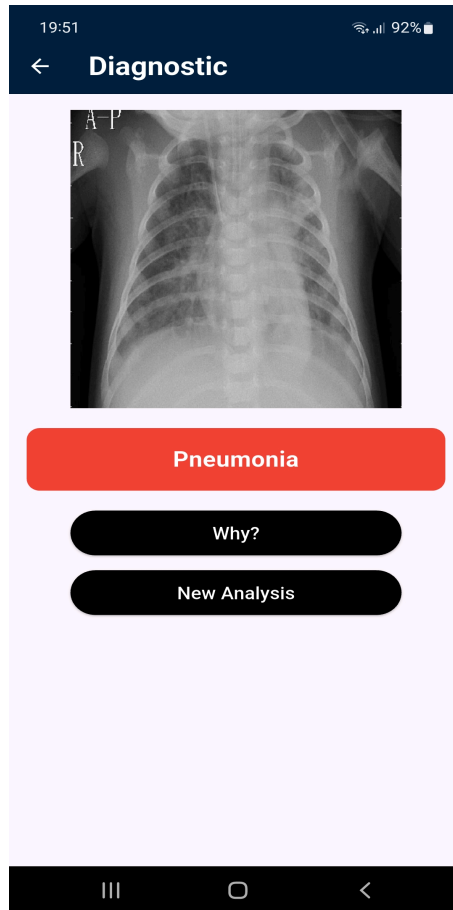


Figure 5.21: Analysis screen (1)

If the user wants to know why this decision was taken then he can press on the button "why ?" where he will be navigated to a new screen and he will see the heat map that display the areas that played the most important part in taking this decision the more red the area is the more important it is to the decision (See Figure [5.22](#)).

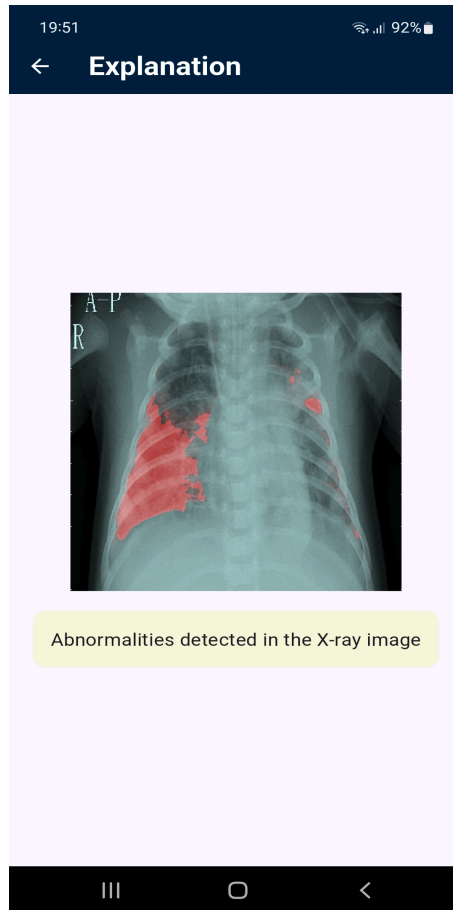


Figure 5.22: Analysis screen (2)

5.3.5 The database screen

What about the diagnosis records ?. The diagnosis can be accessed securely through the computer or the phone and it will look like this (See Figure 5.23).

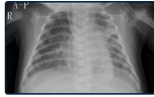
CLINICIAN	PATIENT	RADIOGRAPH	DIAGNOSIS	XAI HEATMAP
abdelghani	jdhdj		Normal	
abdelghani	a		Pneumonia	

Figure 5.23: Database

5.4 Conclusion

In this chapter we discussed the implementation of our application and the tools and technologies we used to do so, the application XPneumo was our AI powered solution to help the medical world in detecting, explaining pneumonia detection and bring new ideas to this field.

Conclusion et perspectives

This thesis addressed a critical challenge in medical AI: developing a pneumonia detection system that not only achieves high accuracy but also provides clinically meaningful explanations that healthcare professionals can trust and understand. Traditional explainable AI methods frequently generate meaningless explanations by highlighting irrelevant areas of chest X-rays, limiting their practical utility in clinical settings. Our research introduced X-Pneumo, an innovative solution that transforms meaningless explanations into meaningful ones by ensuring that AI explanations focus exclusively on clinically relevant regions where pneumonia actually occurs. In the first chapter of this thesis we reviewed the fundamental requirements and specifications for developing a reliable medical AI system, establishing the technical and functional foundations necessary for clinical deployment. In the second chapter we examined the current state of pneumonia detection technologies and explainable AI methods, identifying key limitations in existing approaches and evaluation metrics. The third chapter presented our core contribution the X-Pneumo system detailing its architecture and demonstrating through extensive testing how our approach significantly improves explanation quality and clinical relevance. In the fourth chapter we modeled the application architecture using UML diagrams and outlined our agile development methodology. Finally, in the fifth chapter we implemented and deployed X-Pneumo as a practical mobile application, making advanced AI diagnostics accessible to healthcare providers in various clinical environments. Through comprehensive evaluation on chest X-ray images, we demonstrated that X-Pneumo successfully transforms meaningless explanations into meaningful ones. Traditional XAI methods typically produce meaningless explanations by highlighting irrelevant areas such as background regions, medical equipment, patient positioning markers, or anatomical structures unrelated to pneumonia. In contrast, X-Pneumo generates meaningful explanations that consistently focus on the specific regions where pneumonia pathology is actually present. This fundamental shift from meaningless to meaningful explanations represents a significant advancement in making AI explanations clinically actionable, ensuring that healthcare professionals receive visual insights that align with established medical knowledge rather than misleading or irrelevant highlights. The practical implementation of X-Pneumo as a mobile application further validates the real-world applicability of our approach, providing healthcare professionals with an accessible tool that combines diagnostic accuracy with transparent decision-making. This work contributes to the broader goal of making AI in healthcare not just powerful, but trustworthy and interpretable. Future research directions include extending this approach to other respiratory conditions, enhancing the robustness of our explanation generation across diverse patient populations, and developing comprehensive clinical validation protocols to support widespread adoption in healthcare settings.

Bibliography

- [1] Mehak Aggarwal et al. “Towards trainable saliency maps in medical imaging”. In: *arXiv preprint arXiv:2011.07482* (2020).
- [2] Md Khabir Uddin Ahamed et al. “DTLCx: an improved ResNet architecture to classify normal and conventional pneumonia cases from COVID-19 instances with Grad-CAM-based superimposed visualization utilizing chest X-ray images”. In: *Diagnostics* 13.3 (2023), p. 551.
- [3] Muhammad Aminu, Noor Atinah Ahmad, and Mohd Halim Mohd Noor. “Covid-19 detection via deep neural network and occlusion sensitivity maps”. In: *Alexandria Engineering Journal* 60.5 (2021), pp. 4829–4855.
- [4] Andrés Anaya-Isaza, Leonel Mera-Jiménez, and Martha Zequera-Diaz. “An overview of deep learning in medical imaging”. In: *Informatics in medicine unlocked* 26 (2021), p. 100723.
- [5] Andrés Anaya-Isaza, Leonel Mera-Jiménez, and Martha Zequera-Diaz. “An overview of deep learning in medical imaging”. In: *Informatics in Medicine Unlocked* (2024), pp. 2–8.
- [6] Alejandro Barredo Arrieta et al. “Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI”. In: *Information fusion* 58 (2020), pp. 82–115.
- [7] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. “Neural machine translation by jointly learning to align and translate”. In: *arXiv preprint arXiv:1409.0473* (2014).
- [8] Shahab S Band et al. “Application of explainable artificial intelligence in medical health: A systematic review of interpretability methods”. In: *Informatics in Medicine Unlocked* 40 (2023), p. 101286.
- [9] Rose Grace Bender et al. “Global, regional, and national incidence and mortality burden of non-COVID-19 lower respiratory infections and aetiologies, 1990–2021: a systematic analysis from the Global Burden of Disease Study 2021”. In: *The Lancet Infectious Diseases* 24.9 (2024), pp. 974–1002.
- [10] Philine Lou Bommer et al. “Finding the right XAI method—A guide for the evaluation and ranking of explainable AI methods in climate science”. In: *Artificial Intelligence for the Earth Systems* 3.3 (2024), e230074.
- [11] *Dart*. <https://dart.dev/>.
- [12] *draw.io*. <https://www.drawio.com/>.
- [13] *Flask*. <https://flask.palletsprojects.com/en/stable/>.
- [14] *Flutter*. <https://flutter.dev/>.

- [15] Tatiana Gabruseva, Dmytro Poplavskiy, and Alexandr Kalinin. “Deep Learning for Automatic Pneumonia Detection”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. June 2020.
- [16] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.
- [17] Anna Hedström et al. “Quantus: An explainable ai toolkit for responsible evaluation of neural network explanations and beyond”. In: *Journal of Machine Learning Research* 24.34 (2023), pp. 1–11.
- [18] Robert R Hoffman et al. “Metrics for explainable AI: Challenges and prospects”. In: *arXiv preprint arXiv:1812.04608* (2018).
- [19] *HTTPS Explained*. <https://www.cloudflare.com/fr-fr/learning/ssl/what-is-https/>.
- [20] Gao Huang et al. “Densely connected convolutional networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4700–4708.
- [21] Syed Hussain et al. “Modern Diagnostic Imaging Technique Applications and Risk Factors in the Medical Field: A Review”. In: *BioMed Research International* 2022 (2022), p. 5164970. DOI: [10.1155/2022/5164970](https://doi.org/10.1155/2022/5164970).
- [22] *JWT Authentication*. <https://jwt.io/>.
- [23] *Kaggle*. <https://www.kaggle.com/>.
- [24] Daniel S Kermamy et al. “Identifying medical diagnoses and treatable diseases by image-based deep learning”. In: *cell* 172.5 (2018), pp. 1122–1131.
- [25] Rohit Kundu et al. “Pneumonia detection in chest X-ray images using an ensemble of deep learning models”. In: *PLOS ONE* (2021). DOI: [10.1371/journal.pone.0256630](https://doi.org/10.1371/journal.pone.0256630). URL: <https://doi.org/10.1371/journal.pone.0256630>.
- [26] Yann LeCun et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.
- [27] Tsung-Yi Lin et al. “Focal loss for dense object detection”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2980–2988.
- [28] Scott M Lundberg and Su-In Lee. “A unified approach to interpreting model predictions”. In: *Advances in neural information processing systems* 30 (2017).
- [29] Arthur Gabriel Mathias Marques and Alexei Manso Correa Machado. “Convolutional neural networks with approximation of Shapley values for the classification and interpretation of pneumonia in X-ray images”. In: *Simpósio Brasileiro de Computação Aplicada à Saúde (SBCAS)*. SBC. 2023, pp. 234–243.
- [30] Dastan Maulud and Adnan M Abdulazeez. “A review on linear regression comprehensive in machine learning”. In: *Journal of applied science and technology trends* 1.2 (2020), pp. 140–147.
- [31] Luis J Mena and Jesus A Gonzalez. “Machine Learning for Imbalanced Datasets: Application in Medical Diagnostic.” In: *FLAIRS*. 2006, pp. 574–579.
- [32] *Modelio*. <https://www.modelio.org/index.htm>.
- [33] Sina Mohseni, Niloofar Zarei, and Eric D Ragan. “A multidisciplinary survey and framework for design and evaluation of explainable AI systems”. In: *ACM Transactions on Interactive Intelligent Systems (TiiS)* 11.3-4 (2021), pp. 1–45.

- [34] Gregory J Moran, David A Talan, and Fredrick M Abrahamian. “Diagnosis and management of pneumonia in the emergency department”. In: *Infectious disease clinics of North America* 22.1 (2008), pp. 53–72.
- [35] ngrok. <https://ngrok.com/>.
- [36] World Health Organization. *Pneumonia*. URL: <https://www.who.int/news-room/fact-sheets/detail/pneumonia>.
- [37] World Health Organization. *Pneumonia morbidity*. URL: https://www.who.int/health-topics/pneumonia#tab=tab_1.
- [38] *Python Programming Language*. <https://www.python.org/>.
- [39] J. Ross Quinlan. “Induction of decision trees”. In: *Machine learning* 1 (1986), pp. 81–106.
- [40] Pranav Rajpurkar et al. “Deep learning for chest radiograph diagnosis: A retrospective comparison of the CheXNeXt algorithm to practicing radiologists”. In: *PLoS medicine* 15.11 (2018), e1002686.
- [41] Pranav Rajpurkar et al. *CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning*. 2017. arXiv: [1711.05225 \[cs.CV\]](https://arxiv.org/abs/1711.05225). URL: <https://arxiv.org/abs/1711.05225>.
- [42] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “” Why should i trust you?” Explaining the predictions of any classifier”. In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016, pp. 1135–1144.
- [43] Liqa A Rousan et al. “Chest x-ray findings and temporal lung changes in patients with COVID-19 pneumonia”. In: *BMC Pulmonary Medicine* 20 (2020), pp. 1–9.
- [44] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. “Learning representations by back-propagating errors”. In: *Nature* 323.6088 (1986), pp. 533–536. DOI: [10.1038/323533a0](https://doi.org/10.1038/323533a0). URL: <https://doi.org/10.1038/323533a0>.
- [45] Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. “Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models”. In: *CoRR* abs/1708.08296 (2017). arXiv: [1708.08296](https://arxiv.org/abs/1708.08296). URL: <http://arxiv.org/abs/1708.08296>.
- [46] Ramprasaath R Selvaraju et al. “Grad-cam: Visual explanations from deep networks via gradient-based localization”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 618–626.
- [47] Shagun Sharma and Kalpna Guleria. “A Deep Learning based model for the Detection of Pneumonia from Chest X-Ray Images using VGG-16 and Neural Networks”. In: *Procedia Computer Science* (2023). DOI: [10.1016/j.procs.2023.01.018](https://doi.org/10.1016/j.procs.2023.01.018). URL: <https://www.sciencedirect.com/science/article/pii/S187705092300053X>.
- [48] Raheel Siddiqi and Sameena Javaid. “Deep Learning for Pneumonia Detection in Chest X-ray Images: A Comprehensive Survey”. In: *Journal of Imaging* 10.8 (2024), p. 176. DOI: [10.3390/jimaging10080176](https://doi.org/10.3390/jimaging10080176). URL: <https://doi.org/10.3390/jimaging10080176>.
- [49] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).
- [50] Daniel Smilkov et al. “Smoothgrad: removing noise by adding noise”. In: *arXiv preprint arXiv:1706.03825* (2017).

- [51] Marina Sokolova and Guy Lapalme. “A systematic analysis of performance measures for classification tasks”. In: *Information processing & management* 45.4 (2009), pp. 427–437.
- [52] *SQLite*. <https://sqlite.org/index.html>.
- [53] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. “Axiomatic attribution for deep networks”. In: *International conference on machine learning*. PMLR. 2017, pp. 3319–3328.
- [54] Christian Szegedy et al. “Going deeper with convolutions”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9.
- [55] *TensorFlow*. <https://www.tensorflow.org/?hl=fr>.
- [56] *tls Explained*. <https://ngrok.com/docs/universal-gateway/tls-certificates/>.
- [57] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in neural information processing systems* 30 (2017).
- [58] Vidal. *pneumonia diagnosis*. Vidal.fr. 2021. URL: <https://www.vidal.fr/maladies/voies-respiratoires/pneumonie/symptomes-complications.html>.
- [59] *Visual Studio Code*. <https://code.visualstudio.com/>.
- [60] Haofan Wang et al. “Score-CAM: Score-weighted visual explanations for convolutional neural networks”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. 2020, pp. 24–25.
- [61] Xiaosong Wang et al. “Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2097–2106.
- [62] Saining Xie et al. “Aggregated residual transformations for deep neural networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1492–1500.
- [63] Matthew D Zeiler and Rob Fergus. “Visualizing and understanding convolutional networks”. In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part I 13*. Springer. 2014, pp. 818–833.
- [64] Yihao Zhong et al. “WITHDRAWN: Deep Learning Solutions for Pneumonia Detection: Performance Comparison of Custom and Transfer Learning Models”. In: *medRxiv* (2024). DOI: [10.1101/2024.06.20.24309243](https://doi.org/10.1101/2024.06.20.24309243). eprint: <https://www.medrxiv.org/content/early/2024/10/08/2024.06.20.24309243.full.pdf>. URL: <https://www.medrxiv.org/content/early/2024/10/08/2024.06.20.24309243>.

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي

جامعة عبد الحميد بن باديس-مستغانم-
كلية العلوم الدقيقة والإعلام الآلي

تصريح شرفي خاص بالالتزام بقواعد النزاهة العلمية
لإنجاز البحث

أنا الممضي أدناه،

الطالب(ة): حمادي محمود رقم التسجيل الجامعي 161637041190

الحامل لبطاقة التعريف الوطنية رقم: 121004290 والصادرة بتاريخ: 19-07-2021

عن بلدية مزغران

المسجل بـ: كلية العلوم الدقيقة والإعلام الآلي / قسم : اعلام الي

شعبة : انظمة المعلومات / التخصص : الذكاء الاصطناعي لانتريت الاشياء

والمكلف بإنجاز مذكرة ماستر بعنوان:

**X-PNEUMO: A mobile application for pneumonia detection with meaningful
Explainability**

أصرح بشرفي أنني ألتزم بمراعاة المعايير العلمية والمنهجية ومعايير الأخلاقيات العلمية والنزاهة الأكاديمية
المطلوبة في إنجاز البحث ، وأتحمل المسؤولية الشخصية عن كل المحتوى المتضمن في البحث المذكور أعلاه .

التاريخ: 2025-06-24

إمضاء المعني



* ملحق القرار الوزاري رقم 933 المؤرخ في 28 جويلية 2016 الذي يحدد القواعد المتعلقة بالوقاية من السرقة العلمية ومكافحتها.

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي

جامعة عبد الحميد بن باديس-مستغانم-
كلية العلوم الدقيقة والإعلام الآلي

تصريح شرفي خاص بالالتزام بقواعد النزاهة العلمية
لإنجاز البحث

أنا الممضي أدناه،

الطالب(ة): عبدالغاني قانة رقم التسجيل الجامعي: 181837029604.

الحامل لبطاقة التعريف الوطنية رقم: 404598982 والصادرة بتاريخ: 2023.02.06

عن بلدية مستغانم-مستغانم

المسجل ب: كلية العلوم الدقيقة والإعلام الآلي / قسم : الرياضيات والإعلام الآلي
شعبة : أنظمة المعلومات / التخصص : ذكاء إصطناعي لإنترنت الأشياء

والمكلف بإنجاز مذكرة ماستر بعنوان: X-PNEUMO: A mobile application for
pneumonia detection with meaningful explainability

أصرح بشرفي أنني ألتزم بمراعاة المعايير العلمية والمنهجية ومعايير الأخلاقيات العلمية والنزاهة الأكاديمية
المطلوبة في إنجاز البحث ، وأتحمل المسؤولية الشخصية عن كل المحتوى المتضمن في البحث المذكور أعلاه .

التاريخ: 2025.06.23

إمضاء المعني

