X-RAY CLASSIFICATION USING DEEP LEARNING

Minor project-1 report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Artificial intelligence & Machine learning

By

V.Venkata subhash	(22UEAM0065)	**(VTU 22945)**
S.Ranga Prabhas	(22UEAM0059)	**(VTU 23179)**
Y.Haswanth phani kumar	(22UEAM0067)	**(VTU 23495)**

Under the guidance of
Mrs.P.Girija,B.E.,M.E.,
ASSISTANT PROFESSOR

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

October, 2024

X-RAY CLASSIFICATION USING DEEP LEARNING

Minor project-1 report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Artificial Intelligence & Machine learning

By

V.Venkata subhash	(22UEAM0065)	**(VTU 22945)**
S.Ranga Prabhas	(22UEAM0059)	**(VTU 23179)**
Y.Haswanth phani kumar	(22UEAM0067)	**(VTU 23495)**

Under the guidance of
Mrs.P.Girija.B.E.,M.E.,
ASSISTANT PROFESSOR

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

October, 2024

CERTIFICATE

It is certified that the work contained in the project report titled ”PROJECT-TITLE (X-RAY CLASSIFICATION USING DEEP LEARNING)” by ”V.Venkata subhash (22UEAM0065) ,S.Ranga Prabhas (22UEAM0059),Y.Haswanth phani kumar(22UEAM0067) ” has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Signature of Supervisor
Mrs P. Girija
Assistant Professor
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
October, 2024

Signature of Head of the Department
Dr. S Alex David
Professor & Head
Artificial Intelligence & Machine Learning
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
October, 2024

Signature of the Dean
Prof. Dr. S P. Chokkalingam
Professor & Dean
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
October, 2024

DECLARATION

We declare that this written submission represents my ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)

(V.Venkata subhash)

Date: / /

(Signature)

(S.Ranga Prabhas)

Date: / /

(Signature)

(Y.Haswanth Phani Kumar)

Date: / /

APPROVAL SHEET

This project report entitled (PROJECT TITLE (X-RAY CLASSIFICATION USING DEEP LEARNING) by (V.Venkata Subhash(22UEAM0065) ,S.Ranga Prabhas(22UEAM0059),Y.Haswanth Phani kumar(22UEAM0067)) is approved for the degree of B.Tech in Artificial intelligence & Machine learning.

Examiners **Supervisor**

Mrs P. Girija,B.E., M.E.,
Assistant Professor,

Date: / /

Place:

ACKNOWLEDGEMENT

We express our deepest gratitude to our **Honorable Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (Electrical), B.E. (Mechanical), M.S (Automobile), D.Sc., and Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.** Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, for her blessings.

We express our sincere thanks to our respected Chairperson and Managing Trustee **Mrs. RANGARAJAN MAHALAKSHMI KISHORE,B.E., Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, for her blessings.**

We are very much grateful to our beloved **Vice Chancellor Prof. Dr.RAJAT GUPTA,** for providing us with an environment to complete our project successfully.

We record indebtedness to our **Professor & Dean, Department of Computer Science & Engineering, School of Computing, Dr. S P. CHOKKALINGAM, M.Tech., Ph.D., & Associate Dean, Dr. V. DHILIP KUMAR,M.E.,Ph.D.,** for immense care and encouragement towards us throughout the course of this project.

We are thankful to our **Professor & Head, Department of Artificial Intelligence & Machine Learning, Dr. S.Alex David, M.E., Ph.D.,** for providing immense support in all our endeavors.

We also take this opportunity to express a deep sense of gratitude to our **Internal Supervisor Mrs P. Girija,B.E., M.E., Assistant Professor** for her cordial support, valuable information and guidance, she helped us in completing this project through various stages.

A special thanks to our **Project Coordinator Dr.Prabhu Shankar.B,Ph.D.,** for his valuable guidance and support throughout the course of the project.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

V. Venkata Subhash **(22UEAM0065)**
S.Ranga Prabhas **(22UEAM0059)**
Y.Haswanth Phani kumar **(22UEAM0067)**

ABSTRACT

This project focuses on the deployment of deep learning for the classification of X-ray images, aiming to improve the diagnostic accuracy of medical imaging techniques. With the increasing incidence of chest-related diseases, such as pneumonia and fractures, an automated classification system can assist medical practitioners in earlier detection and treatment. The publicly available dataset was utilized, and the images were preprocessed through normalization and augmentation techniques to enhance model performance. A convolutional neural network (CNN) with transfer learning from ResNet50 was employed to classify images into predefined categories. The model was evaluated using precision, accuracy, and F1-score metrics, which are all relevant for the task. The results demonstrate how deep learning can enable timely and accurate diagnosis. This work paves the way for more advanced applications, including the integration of complex clinical data and improved model architectures, all aimed at enhancing patient outcomes through AI.

Keywords:

- X-ray Classification - Deep Learning - Convolutional Neural Networks (CNN) - Transfer Learning - Medical Imaging - Pneumonia Detection - Image Preprocessing - Data Augmentation - Diagnostic Accuracy - Artificial Intelligence (AI)

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS AND ABBREVIATIONS

ACR	American College of Radiology
ANN	Artificial Neural Network
API	Application Programming Interface
AUC	Area Under the Curve
CT	Computed Tomography
CV	Computer Vision
DICOM	Digital Imaging and Communications in Medicine
DL	Deep Learning
DRI	Magnetic Resonance Imaging
GAN	Generative Adversarial Network
ML	Machine Learning
RN	Residual Network
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
VGG	Visual Geometry Group

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 Introduction

The world is experiencing the revolutionary usage of artificial intelligence and deep learning in medical imaging applications, specifically with regards to the X-ray image analysis. X-rays have been the backbone in diagnosis of diseases and conditions such as pneumonia and fractures. However, it's coupled with complex analyses that contain much human error in the interpretation. Convolutional neural networks are well known for high performance in image classification tasks besides the ability to learn complex patterns and features. The utilization of pre-existing models in transfer learning can enhance classification accuracy without necessarily relying on a large training dataset.

This project aims to develop an automated system for X-ray classification to help doctors in the radiology departments to accomplish their diagnostic activities more efficiently when diseases related to the chest area are involved. Demonstrating how deep learning can improve diagnostic accuracy and alleviate the burden on health-care professionals to achieve better outcomes for patients uses a publicly available dataset.

1.2 Aim of the project

Objective. The project is to develop a highly accurate deep learning model for X-ray images classification with focus on chest conditions, such as pneumonia and fractures. The project is to utilize transfer learning in convolutional neural networks and thus help improve the diagnostic performance to make improved identifications of medical conditions by radiologists. The model, with reduced time taken for the interpretation of images, allows the clinicians to focus more on the need of patients

rather than the computerization of the interpretation. As the datasets used are publicly available, the model will ensure reproducibility and ease of its usage in further research studies. In the broader sense, this project could help in contributing to better patient outcomes through timely diagnosis and intervention. It will represent the unprecedented potential of artificial intelligence that can change the face of medical imaging.

1.3 Project Domain

This project falls under the purview of medical imaging, artificial intelligence, and healthcare informatics wherein advanced computational techniques are combined with clinical applications. Below are some of the salient features of each domain:

1. Medical Imaging Medical imaging forms a significant tool in the diagnosis and follow-up process of a broad spectrum of health conditions. Xrays are one of the most frequently used imaging modalities to provide a lot of useful information of the structural anatomy as well as the physiology of the inside body systems. For this project, the concentration will be on .

Chest X-rays are among the most important images a chest specialist will obtain. These can reveal infections like pneumonia, tuberculosis, lung cancers, and a variety of skeletal abnormalities. Always accurate and timely interpretations of the chest X-rays have a huge impact on the management of the patient's health condition and outcome.

Complexity of Radiological Images Apart from this, variability in image quality and subtle differences in pathologies are the aspects that make it challenging to the medical professional to identify diagnoses accurately most of the time. This is why automation becomes necessary.

2. Artificial Intelligence Artificial intelligence or deep learning has really revolutionized the paradigm of image analysis. It has drawn upon:

Convolutional Neural Networks (CNNs) CNNs are designed to process and analyze image data. These models can automatically learn the features from images, making it highly effective for applying classification and object detection tasks.

Transfer Learning: This enables using models that are pre-trained with huge datasets, like ImageNet. Then the model may be fine-tuned to our particular dataset of X-ray images with good accuracy and relatively fewer data and lesser training

time.

Performance Metrics: The project uses such metrics as accuracy, precision, recall, and F1-score to assess the classification model's performance for so long as the results of the model are reliable and meaningful.

3. Healthcare Informatics Healthcare informatics is the application of information technology and healthcare that aims at improving patient care. In this respect, the project is relevant to the area.

The integration of AI solutions: This project will provide radiologists with faster and more accurate diagnostic analyses while reducing the associated cognitive workload, as an automated X-ray classification system is developed. These can lead to improved workflows in diagnosis.

Eventually, it should eventually enable conditions to be caught early and therefore interventions to be made early on, which may have a better health result. Perhaps more important in emergency practice, decision times are significantly shorter and more vital.

Future Directions to Pursue: Perspectives from this project into AI application in healthcare may motivate further investigation into how deep learning might be applied within clinical practice.

1.4 Scope of the Project

The scope of this project will include all the dimensions of X-ray image classification using deep learning that should alleviate technical as well as practical challenges in medical diagnostics. The major areas of focus are:

1. Data Acquisition and Preprocessing Dataset Selection: use publicly available datasets, for example, NIH Chest X-ray, ChestX-ray14, so that applicability and reproducibility are vast.

Image Preprocessing:Normalization, resizing, and augmentation are used in order to improve image quality and increase the variability in the data set.

2. Model Development Deep Learning Architecture: Design and fine-tune convolutional neural networks (CNNs) to be effective in image-based classification tasks.
Implementation of Transfer Learning: Leverage pre-trained models to speed up training and generally lead to better performance and then fine-tune these pre-trained models specifically for use on the X-ray data.

3. Model Training and Evaluation Training Strategies: Train using robust training methods like validation and cross-validation to be quite sure that the model is generalizable.
Performance Metrics: Training model accuracy, precision, recall, and F1-score are metrics that go towards measuring the effectiveness of the model in the classification task for the images in the X-ray.

4. Application and Integration Clinical Relevance: Inevitably, pneumonia and fractures would have been encountered almost regularly to be eligible for modeling. Ensure that the model addresses relevant clinical needs.
Decision Support System: Develop a prototype of a decision support tool to help radiologists in diagnosis.

5. Future Research Directions Model Enhancement: Understand better the application of ensemble learning, attention mechanisms, or GANs for further improvements in the classification performance.
Clinical Integration: Identify the ways of integrating the model into the existing health-care systems so that its functionality of real-time analysis and feedback may be exercised.

6. Ethical and Regulatory Considerations Some of the key issues that need to be addressed include: - Data Privacy: This involves establishing a level of maintaining some ethical standards for dealing with patient data pertaining to their anonymity and security. B. - Bias and Fairness: Mitigate any form of bias existing in the data and in the model so that there is uniform diagnostic functionality for everyone across every single population.

Chapter 2

LITERATURE REVIEW

2.1 Literature Review

The application of deep learning in X-ray classification has become very popular in the last few years, thanks to the developments in neural network architectures and the deployment of large-scale medical imaging datasets. This literature review combines the main contributions to the field, discussing some of the methods used, tools applied, and concluding with the main findings.

1. Foundational Deep Learning Architectures Convolutional Neural Networks (CNNs) are the main components of image classification technologies. LeNet-5 (Yann LeCun et al., 1998) was among the first to introduce the use of CNNs in medical imaging. More recent models like AlexNet (Krizhevsky et al., 2012), VGGNet (Simonyan & Zisserman, 2014), and ResNet (He et al., 2015) have the better result thanks to the deeper networks and residual connections. They have the ability to detect the intricate features hidden in X-ray images and as a result, they show high classification performance.

2. Transfer Learning and Fine-Tuning Thus, considering the challenges of obtaining labeled datasets in health care, transfer learning has become a serious and effective solution. Model pre-training of public dataset such as Image net is a short-cut to fine-tuning on specific X-ray datasets, besides reducing the time for training and wasting less resources. The experiment in the work of Tajbakhsh et al. (2016) showed that when fine-tuning the model pre-trained on chest X-ray the pneumonia detection classification accuracy was higher compared to from-scratch training. This method eliminates over-fitting and helps generalize.

3. Multi-Class and Multi-Label Classification X-rays are often in a setting where several diseases need to be recognized at the same time, multi-class and multi-label classification approaches are used for this. The Chest X-ray14 data set (Wang et al., 2017) is an essential tool that has been added to the set of benchmarks used to evaluate models for a variety of pathologies, including pneumonia, tuberculosis, cardiomegaly. In the past, various techniques were applied to solve this kind of issue

such as deep learning and ensemble methods. For example, a combination of both CNN and ensemble strategies has been used to tackle this, thus producing some powerful multi-label classifiers.

4. Data Augmentation and Synthetic Data Methods of data augmentation, that is, transformations like rotating, scaling and flipping, have been greatly used to diversify the training datasets and diminish overfitting. Da Vinci dissects medicine and engineers have been able to develop the bike and mobile x-ray machine. A Separate study by Shorten and Khoshgoftaar (2019) stressed how these methods played a vital role in augmenting medical model performance. Moreover, another way to deal with limited datasets has been through use of synthetic data that is generated by Generative Adversarial Networks (GANs), which aims to supplement those datasets, thus making models more robust.

5. Interpretability and Explainability Black-box nature of deep learning models is a concern in medical applications where interpretability is essential. Some works have put emphasis on model explainability improvement methods, for instance, medical image classification, which uses Grad-CAM as the visual analysis tool (Selvaraju et al., 2017). Besides which, the author of the study Chen et al. (2020) observed that interpretable AI in clinical settings was important because the transparency of model decisions could enlighten the healthcare workers and thus develop trust among them.

6. Clinical Integration and Challenges The use of deep learning model Shuman clinical settings has some practical challenges, such as the need to verity during real hospital operating and compliance with the standards of production. A study by Topol (2019) was the very one who stressed the importance of performing comprehensive clinical evaluations leading to the deployment of AI in a given setting. Moreover, patient privacy and the issue of algorithmic bias as well as the question of accountability for AI-powered decisions are other things that should be brought out as healthcare moves towards these technologies.

2.2 Gap Identification

Gap Identification

Until now, although tremendous steps have been taken with deep learning for X-ray classification, there are still lots of gaps to be filled to achieve perfectly efficient and trustworthy diagnostic models. The most important gap, according to the research work done thus far, relates to the thorough datasets on which the models are

trained and tested. There exist many available public databases, but class imbalances and a very limited diversity on demographics and pathologies often characterize them. This may result in biased models that degrade the performance on rarer conditions and, by implication, diagnostic accuracy.

Another critical gap is the interpretability of deep learning models. The ”black box” nature of these algorithms presents challenges for healthcare professionals to explain to patients and caregivers how models determine specific classifications. The need for methods to enhance AI explainability with regard to trust and validation by healthcare practitioners who will rely on AI-assisted diagnoses exists.

These models are usually not directly applicable to real practice. Most models that perform well in a controlled environment do not easily integrate into real clinical workflows because such models lack a mechanism for continuous learning from new data that is encountered during practice. Therefore, this limitation may hinder the uptake of such models in clinical practice settings.

Another significant gap is performance variability across populations. Trainability on specific datasets may not ensure generalizability to different demographics, and thus the risk of disparities in care. Effective globally requires robust AI systems that address the different cultural and contextual variations of disease presentation.

The last but not the least concern pertains to ethical and regulatory issues. Bias within AI models due to nonrepresentative training data would exacerbate health disparities. It is also important to adhere to the regulations of healthcare in terms of patient privacy and rights that are not satisfactorily deployed in model development.

These gaps must be filled to provide effective, trustworthy, and equitable AI-driven diagnostic tools for medical imaging with direct implications on enhancing patient outcomes and healthcare efficiency.

Chapter 3

PROJECT DESCRIPTION

3.1 Existing System

The current state of the art of X-ray classification is more based on the use of classical image analysis techniques and basic machine learning models. For this initial evaluation process, X-ray images are interpreted manually with the assistance of radiologists, a procedure which, by definition, is bound to be time-consuming and inconsistent with variable precision in diagnosis. In addition, some systems depend on the use of conventional algorithms such as SVM or decision trees, which tend to rely on manually extracted features for the classification task. However, these models often fail to capture the complex patterns inherent in the data. Deep learning has yet to make a break-through, and most of the current systems fail to tap all the potential of the CNNs, since these can directly learn from raw images.
Current systems are largely siloed, where modality focus does not seem to easily lend itself to a workflow of health care, leading to delays either in making a diagnosis or treatment. Most of the time, the model is limited to small datasets with a very poor diversity in terms of demographics and pathology. This makes it unsuitable for practical use, particularly in the clinical setting. In addition, most of these models are ”black boxes,” making it problematic for the healthcare practitioners to understand the decision-making process behind them, leading to distrust and hindrance towards the adoption of such. Lastly, the existing systems do not consider the ethical issues of AI in health care regarding bias and data privacy matters critical to both compliance and safety of the patient. Overall, from the information above, there are indeed foundational systems that classify x-rays, but such systems are inefficient, with little accuracy and integration. Advanced data-driven approaches are therefore required to improve the diagnostic process and outcomes of the patients.

3.2 Problem statement

Even though most recent developments in deep learning techniques for X-ray classification can be unveiled, considerable challenges persist, which form the basis for poor implementation of such models in clinical settings. Most current models are seen to suffer from these shortcomings: smaller datasets that lack diversity and have imbalanced classes, hence leading to biased performance and reduced generalization across different populations. Interpretability is another crucial issue with these models. The healthcare professionals find it difficult to understand and to rely on diagnoses from AI systems because of the ”black box” nature of deep learning algorithms.

The integration of these models into clinical practice is often lagging, which means there can be considerable delays in diagnostics and treatment. Many high-performance models fail to include mechanisms for continuous learning from new data, which is crucial for adapting to the evolving needs of the healthcare sector. More over ethical issues, such as possible biases in AI-based algorithms, data privacy regulation, add complexity to deploying these technologies.

Thus, the problem statement is: There is a great need for robust, interpretable, and clinically applicable deep learning models to classify X-rays with minimal constraints of datasets, increased integration into clinical workflows, and ensuring ethics compliance that results in higher diagnostic accuracy and superior patient outcomes.

3.3 System Specification

3.3.1 Hardware Specification

Minimum Hardware Requirements:

- **CPU:** Intel Core i5 / AMD Ryzen 5 or equivalent.
- **Memory:** Minimum 16 GB RAM for smooth operation of analytics and monitoring software.
- **GPU:** Optional (CPU-only training, but very slow for CNN)
- **Storage:** 100 GB HDD/SSD(to accommodate datasets and model checkpoints)
- **Training Time:** Can take hours or days depending on model and dataset size.

Cloud Options:

- **Google Colab Pro/Pro+:** Offers GPUs like NVIDIA Tesla T4/P100.
- **AWS EC2 Instances:** Use p2 or p3 instances with Tesla GPU.
- **Azure / GCP:** Provide access to TPUs and GPUs for large-scale training.

3.3.2 Software Specification

Operating System:

- **Ubuntu:** Preferred for machine learning due to better GPU driver support.
- **Windows 10/11:** If using TensorFlow with CPU or CUDA-enabled GPU.
- **MacOS:** if using TensorFlow-Metal, though limited GPU support.

Cloud Platforms for Training/Hosting :

- **Google Colab:** Free GPU/TPU access.
- **AWS / GCP / Azure:** For scalable compute resources.
- **Streamlit Cloud / Heroku:** To deploy your Streamlit app.

Programming Language:

- **Python 3.x:** Make sure to install the appropriate libraries with compatible versions(recommended version: 3.8 or higher).

Dataset Handling Tools:

- **Kaggle API:** To download datasets directly from Kaggle.

3.3.3 Standards and Policies

Standards and policies for the X-ray classification project are vital to compliance, security, and quality all its lifecycle. The project complies with HIPAA and other industry regulations to protect sensitive medical data by applying strong data protection methods including encryption for both at rest and in transit. Furthermore, strong data access and user authentication policies are put in place to ensure that healthcare staff will only be able to manage patient information with proper authorization. This is to ensure patient privacy as well credibility of the system.

On the technical side, it keeps on following software development best practices (Version Control, Code Review...), and comes with a complete documentation to

help people understand better this project. A combination of functionality and performance tests are used to test the application thoroughly, with key components covered by automated testing. This allows for continuous improvement and adaptation to newer security threats, while being compliant with established standards and policies which are enforced by regular audits and assessments. This sort of rigorous process allows for the whole system to qualify not only against regulatory requirements, but also on accuracy and consistency in X-ray classification. Here are some common limitations of existing systems for X-ray classification using deep learning:

1. Data Limitations

. Class Imbalance: Many datasets may have a disproportionate number of images for different conditions, leading to biased models.

. Quality and Variability: Variations in image quality, equipment, and patient demographics can affect model performance.

. Limited Annotations: High-quality labeled data is often scarce, making it challenging to train robust models.

2. Interpretability

. Black Box Nature: Deep learning models often lack transparency, making it difficult for radiologists to understand decision-making processes.

. Trust Issues: Clinicians may be hesitant to rely on model predictions without clear reasoning.

3. Generalization

. Overfitting: Models trained on specific datasets may not generalize well to new, unseen data, especially from different sources or populations.

. Domain Shift: Changes in imaging technology or protocols can lead to reduced performance.

4. Computational Requirements

. Resource Intensive: Training deep learning models requires significant computational power and memory, which may not be available in all clinical settings.

. Latency in Deployment: Real-time analysis can be challenging due to processing times.

5. Regulatory and Ethical Issues

. Compliance: Meeting healthcare regulations (like HIPAA) can complicate data handling and model deployment.
. Bias and Fairness: Models may inadvertently perpetuate biases present in training data, affecting fairness in diagnostics.

6. Integration Challenges

. Workflow Integration: Difficulty in seamlessly integrating AI systems into existing clinical workflows can hinder adoption. . User Training: Radiologists and technicians may require additional training to effectively use AI tools.

Chapter 4

METHODOLOGY

4.1 Proposed System

The idea is an advanced deep learning model in the area of X-ray classification to improve the accuracy of diagnosis and overcome existing weakness. As it will use diverse and high-quality datasets to minimize biasing and improve generalizability, this system will undergo comprehensive preprocessing techniques, such as normalization and augmentation. The structure for the architecture of the proposed system will include convolutional neural networks (CNNs) and will employ transfer learning from pre-trained models, such as ResNet or DenseNet, for optimization. Besides this, explainable AI techniques like Grad-CAM will be used in the system for making the model's decision-making process understandable to health practitioners, hence enhancing trust in it. It will be designed and developed with user-centered clinical workflows for real-time analysis and active, continuous learning within a robustly evaluated framework that is practically applicable. The effective ethics analysis will include de-biasing and adherence to data privacy regulations as well as deployment in the cloud to achieve scalability and accessibility. The entire scheme ends with the provision of a stable and interpretable solution for X-ray classification, further improving patient outcomes as well as the roles of AI in medical imaging.

4.2 General Architecture

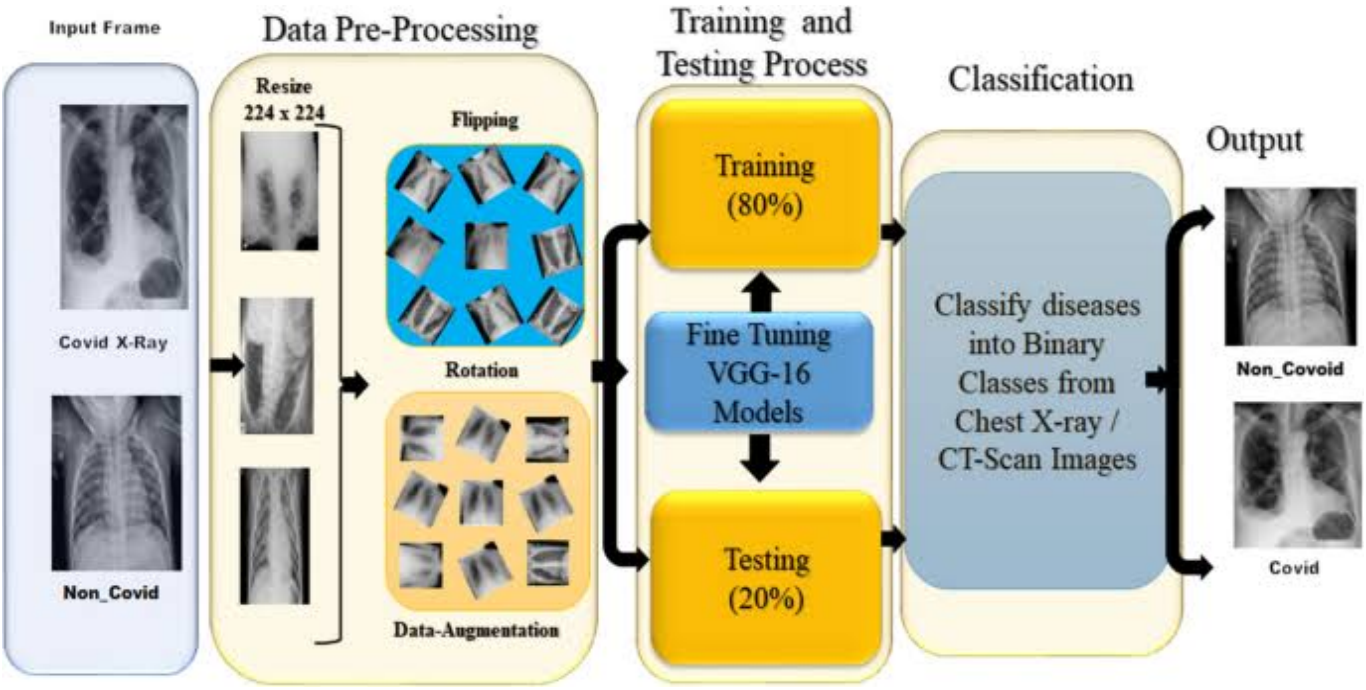


Figure 4.1: **X-ray Architecture**

Figure 4.1 shows the architecture of the proposed transport services system is designed to create an efficient, scalable, and integrated framework that enhances the overall functionality of transportation networks. This architecture consists of multiple layers, each addressing specific aspects of transport management, user interaction, data processing, and integration with existing infrastructure.

4.3 Design Phase

During the design phase of the X-ray classification project, a well-structured framework is established to streamline the overall workflow. The system architecture is built with ease of use in mind, incorporating intuitive data input methods, such as image uploads through a web-based interface. It also features a reliable processing pipeline, which handles essential tasks like resizing, normalization, and data augmentation to prepare the images for analysis.

When it comes to model design, the focus is on selecting the most suitable deep learning architectures, like Convolutional Neural Networks (CNNs) or pre-trained models, while specifying the layer configurations and dropout rates to enhance model performance and mitigate overfitting risks.

User experience is carefully planned out with wireframes that map the process from image upload to result presentation, ensuring a seamless and intuitive interaction. A data flow diagram is also crafted to show how the components interact, ensuring that data transitions smoothly between each phase. Since medical data is sensitive, the system integrates strong security and privacy protocols that adhere to regulations such as HIPAA.

On the technical side, the stack includes frontend technologies like React and backend frameworks like Flask. A comprehensive testing strategy is put in place to ensure that all functions are working correctly and that the system is reliable before going live.

4.3.1 Data Flow Diagram

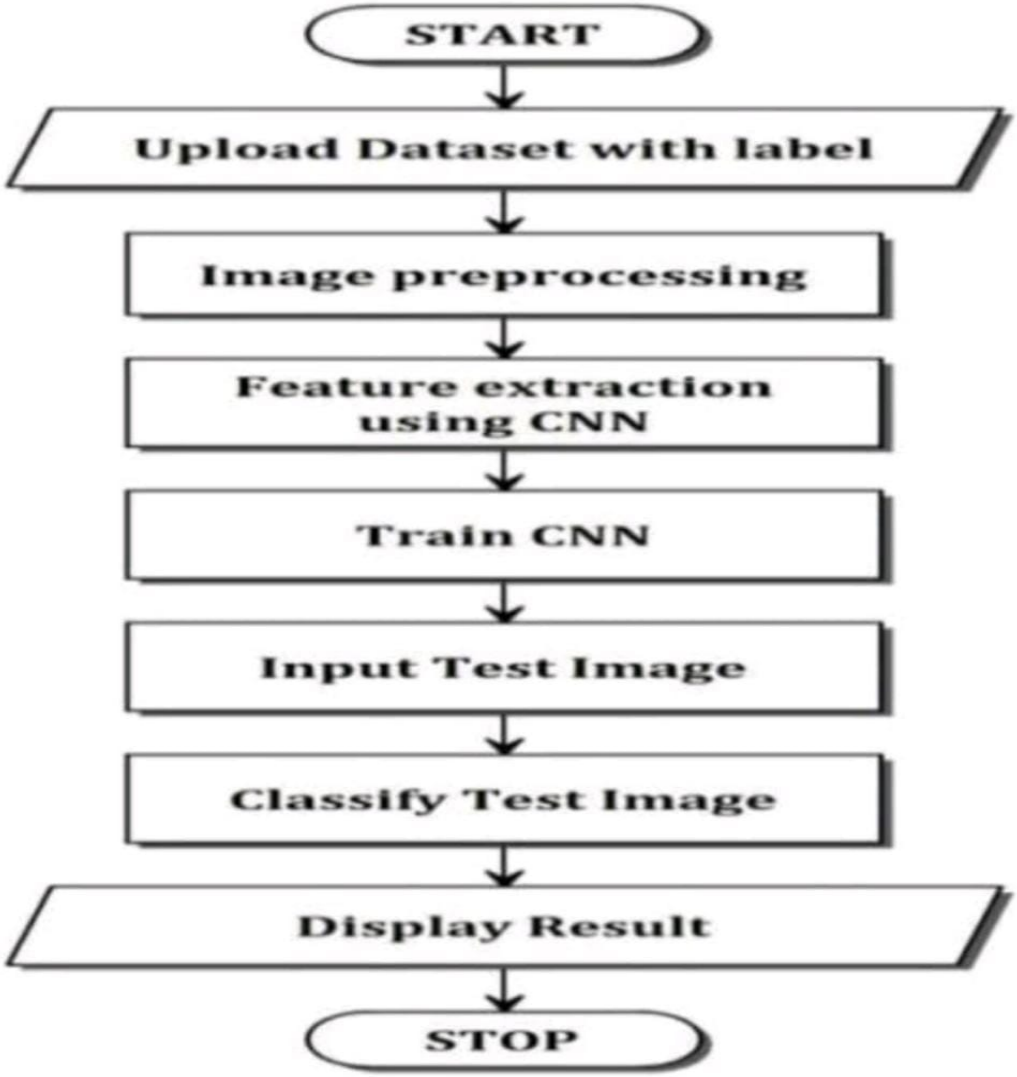


Figure 4.2: **Dataflow Diagram**

Figure 4.2 shows the Data Flow Diagram (DFD) for the X-ray classification project shows how data moves through the system. It begins with the user uploading an X-ray image via the web interface, which is processed by the backend server. The image is then preprocessed through resizing, normalization, and augmentation before being analyzed by the deep learning model. After generating predictions, the results module formats the output, including classification results and confidence scores, and presents them back to the user. This DFD clarifies the data flow from input to output, supporting the system's design.

4.3.2 Use Case Diagram

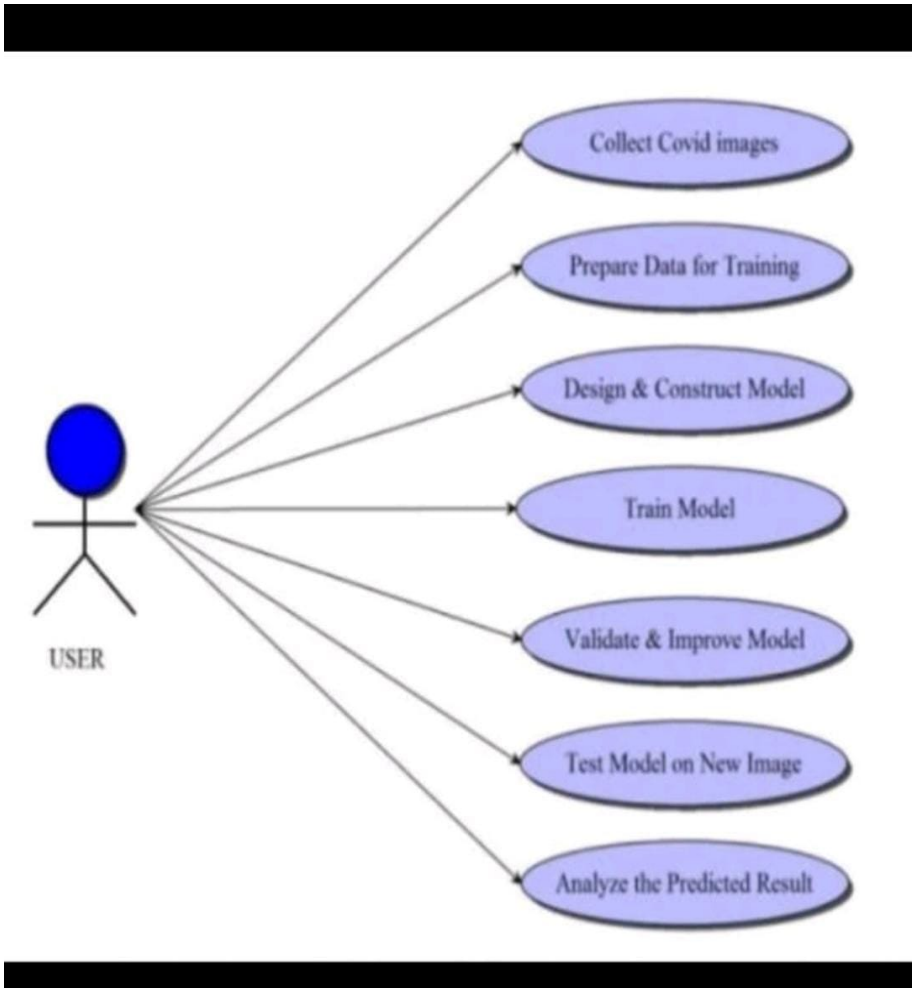


Figure 4.3: **UseCase Diagram**

Figure 4.3 shows the use case diagram for transport services illustrates the interactions between users (actors) and the system, highlighting the various functionalities that the system will offer. The primary actors in this context include passengers, transport operators, fleet managers, and administrators. Each actor interacts with the system to fulfill specific needs and tasks, contributing to the overall functionality of the transport services.

4.3.3 Class Diagram

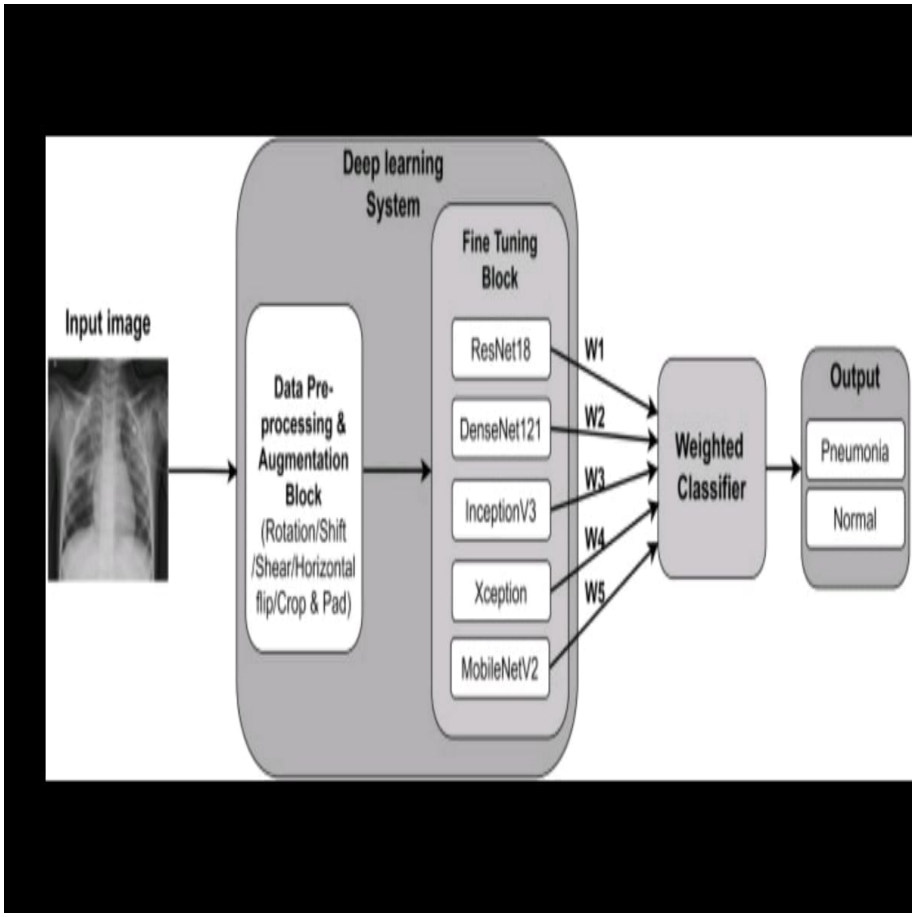


Figure 4.4: **Class Diagram**

Figure 4.4 shows the Class Diagram for the X-ray classification project outlines the system's structure by showcasing key classes, attributes, methods, and their relationships. The User class contains attributes like user ID and user Name, with methods for uploading images and accessing results. The Image class represents X-ray images with attributes such as image ID, file Path, and methods for processing. The Classifier class encapsulates the deep learning model, offering methods for training and predicting. Finally, the Result class holds classification outcomes, including classification label and confidence score. These classes illustrate how components interact to enable the application's functionality.

4.3.4 Sequence Diagram

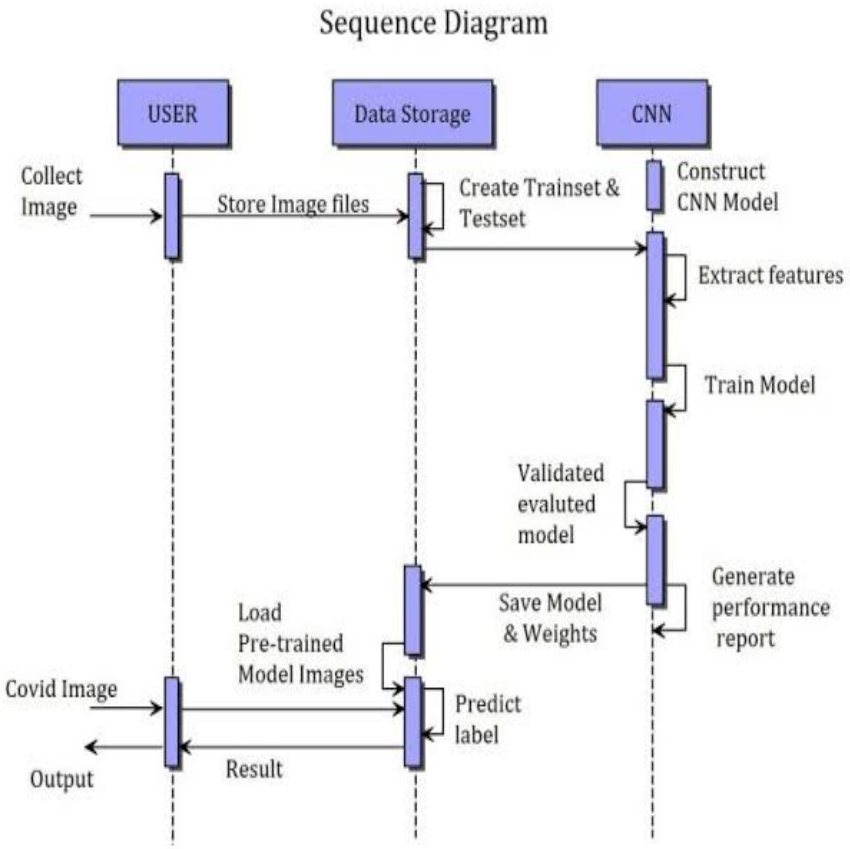


Figure 4.5: **Sequence Diagram**

Figure 4.5 shows the Sequence Diagram for the X-ray classification project shows the interactions during the classification process. It begins with the User uploading an image via the web interface, which sends a request to the Backend Server. The server forwards the image to the Preprocessing Module for resizing and normalization. After processing, the image is sent to the Classifier for analysis, which returns predictions to the server. Finally, the server formats the results and displays the classification outcomes and confidence scores to the User. This diagram effectively captures the flow of interactions throughout the process.

4.3.5 Collaboration diagram:

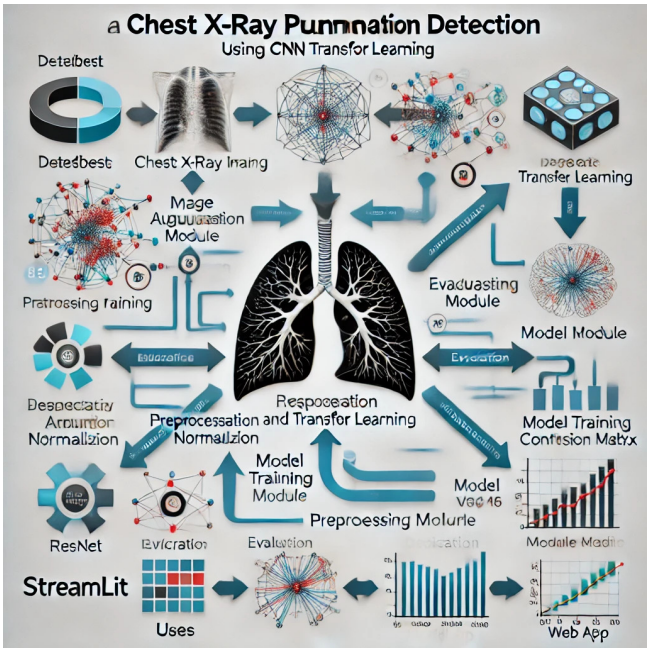


Figure 4.6: **Collaboration Diagram**

Figure 4.6 shows the collaboration diagram illustrates interactions between key components: dataset pre-processing, CNN model training with transfer learning, evaluation metrics, and Streamlit deployment. Clinicians use the web app to input X-rays, with feedback loops enabling model retraining for improved performance.

4.3.6 Activity Diagram

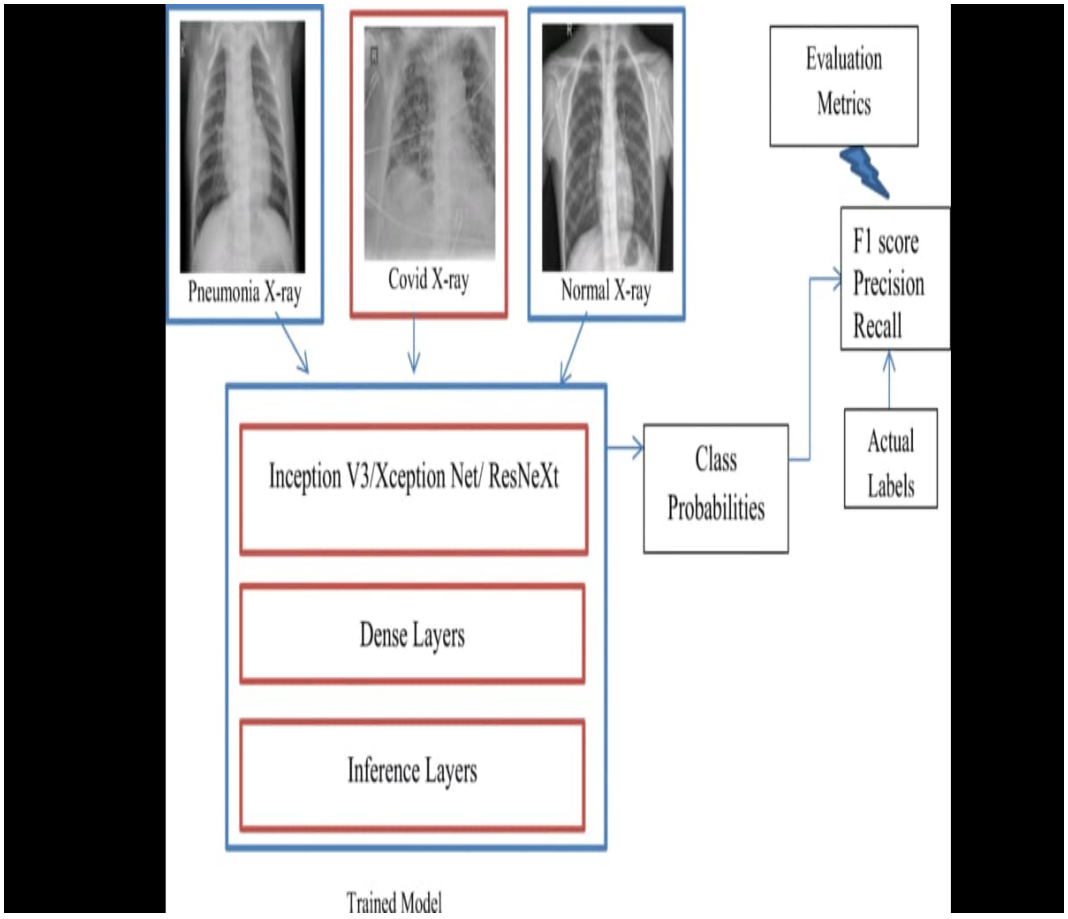


Figure 4.7: **Activity Diagram**

4.4 Algorithm & Pseudo Code

4.4.1 Algorithm

Creating an algorithm for transport services involves defining the steps and processes required to manage tasks like booking, dispatching, and payment processing. Below is a basic outline for an algorithm that covers these key operations in a transport service system, such as a ride-hailing app.

Algorithm for X-ray classfication

1.Data Collection

- **Data Sources:** Collect clinical, radiographic, or laboratory data related to pneumonia cases. In this dataset 5863 images are used for testing and training.

• **Imaging Data:** Chest X-rays or CT scans (especially useful for identifying location-based classification, e.g., lobar or interstitial pneumonia).

2. Data Preprocessing

• Normalize image pixel values to [0, 1].
• Resize images to a fixed dimension (e.g., 150x150 pixels).
• Apply data augmentation (e.g., rotations, flips) to increase dataset variability.

3.Dataset Splitting:

• Split the dataset into training, validation, and test sets.

4. Model Selection:

• Choose a convolutional neural network (CNN) architecture (e.g., VGG16, ResNet).
• Optionally, use transfer learning with a pre-trained model.

5. Model Building:

• Convolutional layers with ReLU activation.
• Max pooling layers.
• Flatten layer.
• Fully connected (Dense) layers with activation functions.
• Output layer (e.g., sigmoid for binary classification).

6. Model Compilation:

• Loss function Binary Crossentropy .
• Optimizer Adam or SGD.
• Metrics Accuracy.

7. Model Training:

• Train the model on the training set.
• Validate using the validation set.
• Monitor performance metrics (accuracy, loss).

8. Model Evaluation:

Evaluate the model on the test set.

Analyze performance using confusion matrix, precision, recall, F1-score, and ROC curve.

4.4.2 Pseudo Code

- **Import necessary libraries:**
  - TensorFlow, Keras, NumPy, Pandas, Matplotlib, Sklearn, Streamlit.
- **Load the dataset:**
  - Download dataset (e.g., Kaggle Chest X-ray dataset).
  - Split into TRAIN, VALIDATION, and TEST sets.
- **Data Preprocessing:**
  - Resize images to input size (e.g., 224x224)
  - Normalize pixel values (scale between 0 and 1)
  - Apply Data Augmentation (rotation, flipping, etc.) to avoid overfitting.
- **Load Pre-trained Model:**
  - Choose model (e.g., VGG16, ResNet50) with pre-trained weights.
  - Remove the top layer (classification head).
  - Add new layers (Dense, Dropout, Output) for pneumonia classification.
- **Compile the Model**
  - Define loss function (e.g., binary crossentropy).
  - Select optimizer (e.g., Adam) and evaluation metrics (e.g., accuracy).
- **Train the Model**
  - Use the TRAIN set for training and VALIDATION set for validation.
  - Save the best model using ModelCheckpoint (based on validation performance).
- **Evaluate the Model**
  - Use the TEST set to calculate accuracy, precision, recall, F1-score.
  - Generate confusion matrix and ROC-AUC curve for performance analysis.
- **Save and Load the Trained Model**
  - Save the trained model for deployment (e.g., model.h5).
- **Deploy Model with Streamlit**
  - Create Streamlit web app interface.

• Load the saved model in Streamlit.
• Take user input (upload X-ray image).
• Preprocess the image and predict if pneumonia is present

– **Display Results**
• Show prediction result and probability.
• Visualize Grad-CAM heatmaps (optional) to explain model prediction.

– **Monitor and Retrain**
• Continuously monitor model performance.
• Retrain with new data if performance drops or bias is detected.

4.4.3 Data Set / Generation of Data (Description only)

The dataset used for this Chest X-Ray Pneumonia Detection project typically consists of labeled chest X-ray images indicating the presence or absence of pneumonia. A commonly used dataset is the Chest X-Ray Images (Pneumonia) dataset available on Kaggle or open-source medical databases like NIH or MIMIC-CXR.

1. Data Composition:

* **categories:** The dataset is divided into two categories Pneumonia and Normal chest X-rays.
* **P**neumonia class may be further divided into bacterial and viral pneumonia, adding granularity to the classification task.

2. Image Properties:

* **I**mages are in grayscale or RGB, typically with varying resolutions (e.g., 1024x1024 pixels).
* **T**hey are resized to a uniform size (e.g., 224x224 pixels) during preprocessing to match the input size of the neural network

3. Data Splitting:

* **T**he dataset is divided into Training (70
* **T**raining Set: Used to fit the CNN model.

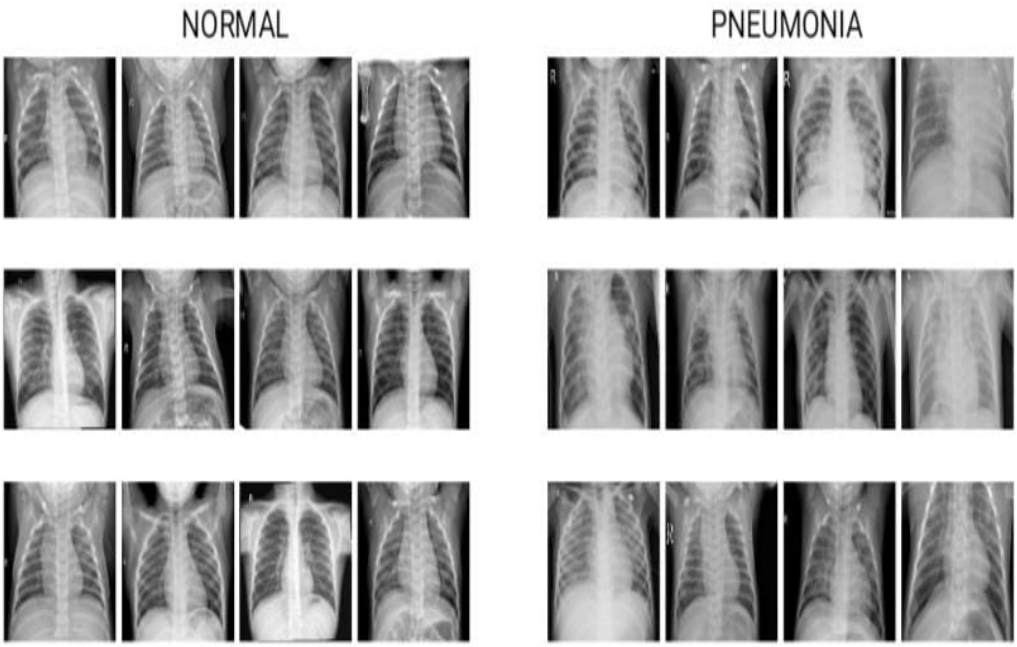


Figure 4.8: **Image Data**

* **V**alidation Set: Monitors model performance and helps tune hyperparameters.
* **T**est Set: Evaluates the final model's generalizability.

4.5 Module Description

4.5.1 MODULE 1: Dataset

Step 1: Data Collection

The data collection process for a chest X-ray pneumonia detection system is a critical step, especially when dealing with medical images. Below are the key considerations and steps involved in collecting relevant data.

Step 2: NIH Chest X-ray Dataset

Information about the population using public transport, including age, gender, and frequency of travel.•

Step 3:Transportation Patterns

Data on common routes, peak travel times, and preferred modes of transportation.

4.5.2 MODULE 2

1.**Predictive Modeling**:

Regression algorithms (e.g., Linear Regression, Random Forest) for predicting bus arrival times and delays.

2.**Natural Language Processing (NLP)**:

NLP algorithms for translating and interpreting multi-language support, such as translation models (e.g., Google Translate API) or language identification tools.

3.**Real-Time Data Processing**:

Algorithms for real-time data handling, such as streaming analytics tools or event-driven architectures for updating information dynamically.

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Input and Output

5.1.1 Input Design

The input consists of chest X-ray images uploaded by users through a web interface created using Streamlit. Users can select and upload images in standard formats like JPEG or PNG, which are then preprocessed to ensure uniformity in size and format. Preprocessing steps include resizing the images to a fixed dimension (e.g., 224x224 pixels), normalizing pixel values to a scale between 0 and 1, and applying any necessary data augmentation techniques to enhance model robustness. Once the images are prepared, they are fed into the trained Convolutional Neural Network (CNN) model for prediction.The output of the system is a clear and concise diagnosis indicating whether pneumonia is present or absent in the uploaded X-ray image. This output is presented as a binary classification result: ”Pneumonia” or ”Normal,” accompanied by a confidence score that reflects the model’s certainty in its prediction. Additionally, for enhanced interpretability, the system can provide visual explanations using Grad-CAM heatmaps, highlighting the regions of the X-ray that contributed most significantly to the model’s decision. This combination of diagnostic results and visual feedback allows clinicians and users to better understand and trust the model’s predictions, facilitating informed decision-making in medical contexts.

5.1.2 Output Design

The output design for the Chest X-Ray Pneumonia Detection System delivers clear insights for clinicians. After uploading a chest X-ray image, users receive a diagnosis—either ”Pneumonia” or ”Normal”—with a confidence score . Grad-CAM heatmaps highlight areas that influenced the

model's decision, aiding in interpretability. The intuitive interface allows for quick understanding, and options to save or export results support effective patient management and documentation.

5.2 Testing

Testing for the Chest X-Ray Pneumonia Detection System includes several phases to ensure accuracy and reliability. It starts with unit testing of components, followed by evaluating the model on a test dataset to compute metrics like accuracy, precision, recall, F1-score, and ROC-AUC, along with a confusion matrix to identify errors. User acceptance testing then assesses the usability of the Streamlit interface and the clarity of outputs. Continuous monitoring post-deployment ensures the system adapts to new data and user feedback, maintaining its effectiveness.

5.3 Types of Testing

The Chest X-Ray Pneumonia Detection System employs various testing types to ensure reliability. Unit testing verifies individual components like data preprocessing and model predictions, while integration testing ensures seamless collaboration between these parts. System testing evaluates the entire application against performance requirements, and regression testing checks that updates do not harm existing functionality. Finally, user acceptance testing (UAT) assesses usability from the end-user's perspective, ensuring an intuitive interface and clear outputs. This comprehensive approach identifies and resolves issues, ensuring the system's effectiveness in clinical use.

5.3.1 Unit testing

Input

```
import pandas as pd
import matplotlib as mat
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
```

```
%matplotlib inline

pd.options.display.max_colwidth = 100

import random
import os

from numpy.random import seed
seed(42)

random.seed(42)
os.environ['PYTHONHASHSEED'] = str(42)
os.environ['TF_DETERMINISTIC_OPS'] = '1'

from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.metrics import accuracy_score

import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras import callbacks
from tensorflow.keras.models import Model
from tensorflow.keras.preprocessing.image import ImageDataGenerator

import glob
import cv2

from tensorflow.random import set_seed
set_seed(42)

import warnings
warnings.filterwarnings('ignore')
```

5.3.2 Integration testing

Input

```
main_path ="/content/drive/MyDrive/MINOR/CHEST/chest_xray"

train_path = os.path.join(main_path,"train")
test_path=os.path.join(main_path,"test")

train_normal = glob.glob(train_path+"/NORMAL/*.jpeg")
train_pneumonia = glob.glob(train_path+"/PNEUMONIA/*.jpeg")

```

```
test_normal = glob.glob(test_path+"/NORMAL/*.jpeg")
test_pneumonia = glob.glob(test_path+"/PNEUMONIA/*.jpeg")
```

5.3.3 System testing

Input

```
print('Train Set - Normal')

plt.figure(figsize=(12,12))

for i in range(0, 12):
    plt.subplot(3,4,i + 1)
    img = cv2.imread(train_normal[i])
    img = cv2.resize(img, (IMG_SIZE,IMG_SIZE))
    plt.imshow(img)
    plt.axis("off")

plt.tight_layout()

plt.show()
```

5.3.4 Test Result

```
df_train
```

	class	image
0	Normal	/content/drive/MyDrive/MINOR/CHEST/chest_xray/train/NORMAL/NORMAL-3360834-0001.jpeg
1	Normal	/content/drive/MyDrive/MINOR/CHEST/chest_xray/train/NORMAL/NORMAL-338872-0001.jpeg
2	Normal	/content/drive/MyDrive/MINOR/CHEST/chest_xray/train/NORMAL/NORMAL-32326-0001.jpeg
3	Normal	/content/drive/MyDrive/MINOR/CHEST/chest_xray/train/NORMAL/NORMAL-3156332-0001.jpeg
4	Normal	/content/drive/MyDrive/MINOR/CHEST/chest_xray/train/NORMAL/NORMAL-3387483-0002.jpeg
...	...	...
5240	Pneumonia	/content/drive/MyDrive/MINOR/CHEST/chest_xray/train/PNEUMONIA/BACTERIA-4213815-0001.jpeg
5241	Pneumonia	/content/drive/MyDrive/MINOR/CHEST/chest_xray/train/PNEUMONIA/BACTERIA-4327797-0001.jpeg
5242	Pneumonia	/content/drive/MyDrive/MINOR/CHEST/chest_xray/train/PNEUMONIA/BACTERIA-4092125-0003.jpeg
5243	Pneumonia	/content/drive/MyDrive/MINOR/CHEST/chest_xray/train/PNEUMONIA/BACTERIA-428676-0001.jpeg
5244	Pneumonia	/content/drive/MyDrive/MINOR/CHEST/chest_xray/train/PNEUMONIA/BACTERIA-4320648-0001.jpeg

5245 rows × 2 columns

```
df_test
```

	class	image
0	Normal	/content/drive/MyDrive/MINOR/CHEST/chest_xray/test/NORMAL/NORMAL-1049278-0001.jpeg
1	Normal	/content/drive/MyDrive/MINOR/CHEST/chest_xray/test/NORMAL/NORMAL-2107985-0001.jpeg
2	Normal	/content/drive/MyDrive/MINOR/CHEST/chest_xray/test/NORMAL/NORMAL-1368583-0001.jpeg
3	Normal	/content/drive/MyDrive/MINOR/CHEST/chest_xray/test/NORMAL/NORMAL-115218-0001.jpeg
4	Normal	/content/drive/MyDrive/MINOR/CHEST/chest_xray/test/NORMAL/NORMAL-11419-0001.jpeg
...	...	...
619	Pneumonia	/content/drive/MyDrive/MINOR/CHEST/chest_xray/test/PNEUMONIA/VIRUS-9890836-0001.jpeg
620	Pneumonia	/content/drive/MyDrive/MINOR/CHEST/chest_xray/test/PNEUMONIA/VIRUS-7014527-0001.jpeg
621	Pneumonia	/content/drive/MyDrive/MINOR/CHEST/chest_xray/test/PNEUMONIA/VIRUS-6207158-0001.jpeg
622	Pneumonia	/content/drive/MyDrive/MINOR/CHEST/chest_xray/test/PNEUMONIA/VIRUS-8427429-0001.jpeg
623	Pneumonia	/content/drive/MyDrive/MINOR/CHEST/chest_xray/test/PNEUMONIA/VIRUS-7788460-0001.jpeg

624 rows × 2 columns

Figure 5.1: **Test Image**

Chapter 6

RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System

The envisioned X-ray classification framework makes it possible to improve diagnostic accuracy by performing X-Ray images analysis using complex deep learning algorithms. Such widespread use of automation helps to cut down the image classification timeline, enabling health care providers to make the required steps basing on available information as quickly as it is required. In contrast, the normal way of doing things is that the radiologists read and interpret the images which may take quite some time. The automated system, on the other hand, takes seconds to run the images and still remains accurate and reliable having been tested across various datasets for long.Apart from helping in quickening the processes, the system incorporates preprocessing approaches such as image resizing, normalization, and augmentation in order to enhance the quality of images and also assist in better classification. This helps in delivering the same output even in cases where the image quality tends to change. By removing manual scoring from the equation, the system allows the clinician to deal with interpreting the results and the patients, which in turn helps in shortening the period for a diagnosis and improving the quality of care. To conclude, there is no doubt that the X-ray classification system for X-ray images is a notable enhancement in medical imaging as it improves processes without compromising the level of accuracy and reliability.

6.2 Comparison of Existing and Proposed System

Existing System: Traditional Image Classification Methods

In the existing system for X-ray classification, traditional image classification methods such as Support Vector Machines (SVM) and basic neural

networks are employed. These approaches rely on handcrafted features extracted from the X-ray images, which require significant domain expertise and are often limited in capturing complex patterns in the data. While these methods can provide reasonable accuracy, they struggle with variations in image quality, noise, and the diversity of conditions presented in the X-rays. Additionally, the reliance on manual feature extraction makes the models less adaptive to new data or changes in diagnostic criteria. As a result, the existing system may demonstrate inconsistent performance and limited scalability in clinical settings, hindering its effectiveness in delivering reliable and timely diagnoses.

Here are some advantages of a proposed deep learning system for X-ray classification:

Proposed System: Deep Learning with Convolutional Neural Networks (CNNs)

To overcome the limitations of the existing system, the proposed system utilizes Convolutional Neural Networks (CNNs) for X-ray classification. CNNs automatically learn hierarchical feature representations from raw pixel data, allowing them to effectively capture intricate patterns in the images without the need for manual feature extraction. This approach enhances accuracy and robustness against variations in image quality and different pathological conditions. The proposed system is designed to leverage a deeper architecture with multiple convolutional and pooling layers, which facilitates better generalization and improves performance on large datasets. By implementing CNNs, the proposed system is expected to significantly enhance diagnostic accuracy, reduce processing time, and ultimately provide more reliable and timely classifications, leading to improved patient outcomes and greater efficiency in clinical workflows. 1. Enhanced Accuracy

Improved Diagnostic Precision: Deep learning models can learn complex patterns, leading to higher accuracy in detecting conditions compared to traditional methods.

2. Scalability

High Throughput: Automated analysis allows for processing large volumes of X-ray images quickly, supporting busy clinical environments.

3. Consistency

Reduced Human Error: AI can provide consistent results, minimizing variability caused by human interpretation.

4. Early Detection

Timely Diagnosis: Automated systems can facilitate early detection of conditions, improving patient outcomes.

5. Continuous Learning

Adaptability: Models can be continuously updated with new data, allowing them to improve over time and adapt to new conditions or imaging techniques.

6. Resource Optimization

Assist Radiologists: AI can act as a second reader, helping radiologists prioritize cases and reducing their workload.

7. Interpretability Improvements

Visualization Tools: Incorporating explainability tools can help clarify model decisions, increasing trust and understanding among clinicians.

8. Integration Potential

Seamless Integration: Proposed systems can be designed to fit into existing workflows, enhancing clinical efficiency.

9. Cost-Effectiveness

Reduced Costs: Automation can lower costs associated with manual analysis and improve resource allocation.

10. Support for Research

Data Insights: Analyzing large datasets can uncover new insights and trends in radiology, supporting ongoing research efforts.

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

In conclusion, the X-ray classification project utilizing deep learning techniques represents a significant advancement over traditional diagnostic methods. By employing Convolutional Neural Networks (CNNs), the proposed system not only automates the classification process but also enhances accuracy and robustness against variations in image quality. This innovation addresses key limitations of existing systems, such as reliance on manual feature extraction and susceptibility to overfitting. The streamlined workflow and rapid processing capabilities enable healthcare professionals to obtain timely and reliable diagnostic insights, ultimately improving patient care. As the field of medical imaging continues to evolve, the integration of deep learning into X-ray classification is poised to set new standards in efficiency, accuracy, and clinical effectiveness, paving the way for more responsive healthcare solutions.

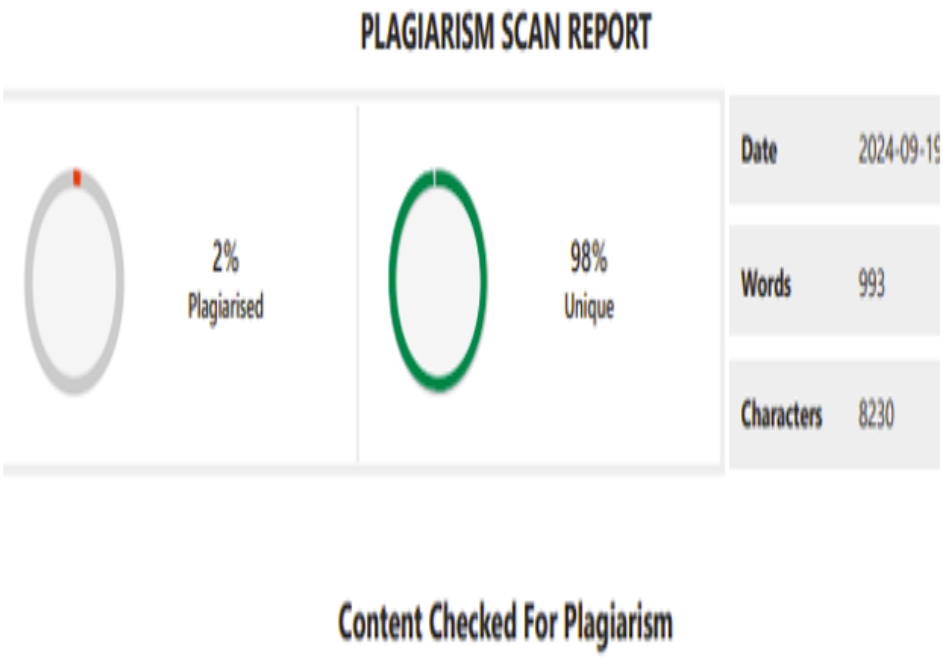
7.2 Future Enhancements

The upgrades titled X-ray classification system is essential to address the numerous areas for clinical effectiveness of the system. One such enhancement could be the ability to apply transfer learning where existing models are updated by training with the available medical data to raise the level of accuracy and reduce the training time. Furthermore, the analysis could be extended to cover the patient's history and laboratory results alongside X-ray imaging.It will be equally important to envisage the implementation of explainable AI (XAI) techniques as they explain the rationale of decisions

made by the model thus enabling clinicians to trust the model without fear of false decision making. The same Brian will be useful in the training process by incorporating updates through new data without major changes in the schedules. Lastly, addressing the real-time processing and mobile application aspects could promote usability by having the system used in different levels of health care delivery even in remote regions. The correction shall serve to entrench the X-ray classification system as one of the components that can help to enhance the patient outcome and diagnostic effectiveness.

Chapter 8

PLAGIARISM REPORT



Chapter 9

Appendices

Appendix A: Data Collection Methodology

1 Survey Instruments: Overview of surveys used, including sample questions on passenger preferences.

2 Data Annotation: Description of labeling methods for medical conditions, including how classes were defined and validated.

3 Sampling Techniques: Explanation of sampling methods, sample sizes, and the demographics of images included in the dataset.

Appendix B: Technical Specifications

1 Model Architectures: Details on the Convolutional Neural Networks (CNNs) used, including layer configurations and hyperparameters.

2 System Architecture: Diagrams illustrating the overall system architecture and the technology stack (e.g., TensorFlow, Keras).

Appendix C: Experimental Results

1 Training and Validation Metrics: Summary of model performance metrics, including accuracy, precision, recall, and F1-score.

2 Comparison of Models: Comparative analysis of different deep learning architectures and their performance metrics.

Appendix D: User Feedback and Testimonials

1 Clinician Surveys: Summary of feedback collected from healthcare professionals regarding the model's usability and accuracy.

2 Stakeholder Testimonials: Quotes from users and stakeholders about their experiences with the classification system.

Appendix E: Financial Analysis

1 Areas for Further Study: Identification of gaps in current research and recommendations for future studies in medical image classification.

2 Emerging Technologies: Discussion of new technologies that could enhance the project, such as advancements in AI and imaging techniques.

Appendix F: Future Research Directions

1 Areas for Further Study: Identification of gaps in current research and recommendations for future studies in medical image classification.

2 Emerging Technologies: Discussion of new technologies that could enhance the project, such as advancements in AI and imaging techniques.

Appendix G: Relevant Legislation and Policies

1 Healthcare Regulations: Summary of regulations affecting the use of AI in healthcare and compliance requirements.

2 Ethical Considerations: Suggested initiatives to ensure ethical use of AI in medical diagnostics, including patient consent and data privacy.

Appendix H: Glossary of Terms

1 Terminology: Definitions of key terms related to X-ray classification, deep learning, and medical imaging for clarity.

Chapter 10

Complete Data / Sample Data / Sample Source Code / etc

```
import pandas as pd
import matplotlib as mat
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
%matplotlib inline

pd.options.display.max_colwidth = 100

import random
import os

from numpy.random import seed
seed(42)

random.seed(42)
os.environ['PYTHONHASHSEED'] = str(42)
os.environ['TF_DETERMINISTIC_OPS'] = '1'

from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.metrics import accuracy_score

import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras import callbacks
from tensorflow.keras.models import Model
from tensorflow.keras.preprocessing.image import ImageDataGenerator

import glob
import cv2

from tensorflow.random import set_seed
set_seed(42)

import warnings
warnings.filterwarnings('ignore')
```

```
IMG_SIZE = 224
BATCH = 32
SEED = 42
main_path = "/content/drive/MyDrive/MINOR/CHEST/chest_xray"


train_path = os.path.join(main_path,"train")
test_path=os.path.join(main_path,"test")

train_normal = glob.glob(train_path+"/NORMAL/*.jpeg")
train_pneumonia = glob.glob(train_path+"/PNEUMONIA/*.jpeg")

test_normal = glob.glob(test_path+"/NORMAL/*.jpeg")
test_pneumonia = glob.glob(test_path+"/PNEUMONIA/*.jpeg")
train_list = [x for x in train_normal]
train_list.extend([x for x in train_pneumonia])

df_train = pd.DataFrame(np.concatenate([['Normal']*len(train_normal) , ['Pneumonia']*len(
    train_pneumonia)]), columns = ['class'])
df_train['image'] = [x for x in train_list]

test_list = [x for x in test_normal]
test_list.extend([x for x in test_pneumonia])

df_test = pd.DataFrame(np.concatenate([['Normal']*len(test_normal) , ['Pneumonia']*len(
    test_pneumonia)]), columns = ['class'])
df_test['image'] = [x for x in test_list]
plt.figure(figsize=(6,4))

ax = sns.countplot(x='class', data=df_train, palette="mako")

plt.xlabel("Class", fontsize= 12)
plt.ylabel("# of Samples", fontsize= 12)
plt.ylim(0,5000)
plt.xticks([0,1], ['Normal', 'Pneumonia'], fontsize = 11)

for p in ax.patches:
    ax.annotate((p.get_height()), (p.get_x()+0.30, p.get_height()+300), fontsize = 13)

plt.show()
plt.figure(figsize=(7,5))

df_train['class'].value_counts().plot(kind='pie',labels = ['',''], autopct='%1.1f%%',
    colors = ['darkcyan','blue'], explode = [0,0.05], textprops = {"fontsize":15})

plt.legend(labels=['Pneumonia', 'Normal'])
plt.show()
plt.figure(figsize=(6,4))

ax = sns.countplot(x='class', data=df_test, palette="mako")
```

```
86
87 plt.xlabel("Class", fontsize= 12)
88 plt.ylabel("# of Samples", fontsize= 12)
89 plt.ylim(0,500)
90 plt.xticks([0,1], ['Normal', 'Pneumonia'], fontsize = 11)
91
92 for p in ax.patches:
93     ax.annotate((p.get_height()), (p.get_x()+0.32, p.get_height()+20), fontsize = 13)
94
95 plt.show()
96 print('Train Set - Normal')
97
98 plt.figure(figsize=(12,12))
99
100 for i in range(0, 12):
101     plt.subplot(3,4,i + 1)
102     img = cv2.imread(train_normal[i])
103     img = cv2.resize(img, (IMG_SIZE,IMG_SIZE))
104     plt.imshow(img)
105     plt.axis("off")
106
107 plt.tight_layout()
108
109 plt.show()
110 print('Train Set - Pneumonia')
111
112 plt.figure(figsize=(12,12))
113
114 for i in range(0, 12):
115     plt.subplot(3,4,i + 1)
116     img = cv2.imread(train_pneumonia[i])
117     img = cv2.resize(img, (IMG_SIZE,IMG_SIZE))
118     plt.imshow(img)
119     plt.axis("off")
120
121 plt.tight_layout()
122
123 plt.show()
124 print('Test Set - Normal')
125
126 plt.figure(figsize=(12,12))
127
128 for i in range(0, 12):
129     plt.subplot(3,4,i + 1)
130     img = cv2.imread(test_normal[i])
131     img = cv2.resize(img, (IMG_SIZE,IMG_SIZE))
132     plt.imshow(img)
133     plt.axis("off")
134
135 plt.tight_layout()
```

```
136
137 plt.show()
138 print('Test Set - Pneumonia')
139
140 plt.figure(figsize=(12,12))
141
142 for i in range(0, 12):
143     plt.subplot(3,4,i + 1)
144     img = cv2.imread(test_pneumonia[i])
145     img = cv2.resize(img, (IMG_SIZE,IMG_SIZE))
146     plt.imshow(img)
147     plt.axis("off")
148
149 plt.tight_layout()
150
151 plt.show()
152 train_datagen = ImageDataGenerator(rescale=1/255.,
153                                    zoom_range = 0.1,
154                                    #rotation_range = 0.1,
155                                    width_shift_range = 0.1,
156                                    height_shift_range = 0.1)
157
158 val_datagen = ImageDataGenerator(rescale=1/255.)
159
160 ds_train = train_datagen.flow_from_dataframe(train_df,
161                                              #directory=train_path, #dataframe contains
                                                     the full paths
162                                              x_col = 'image',
163                                              y_col = 'class',
164                                              target_size = (IMG_SIZE, IMG_SIZE),
165                                              class_mode = 'binary',
166                                              batch_size = BATCH,
167                                              seed = SEED)
168
169 ds_val = val_datagen.flow_from_dataframe(val_df,
170                                          #directory=train_path,
171                                          x_col = 'image',
172                                          y_col = 'class',
173                                          target_size = (IMG_SIZE, IMG_SIZE),
174                                          class_mode = 'binary',
175                                          batch_size = BATCH,
176                                          seed = SEED)
177
178 ds_test = val_datagen.flow_from_dataframe(df_test,
179                                          #directory=test_path,
180                                          x_col = 'image',
181                                          y_col = 'class',
182                                          target_size = (IMG_SIZE, IMG_SIZE),
183                                          class_mode = 'binary',
184                                          batch_size = 1,
```

```
                                                shuffle = False)
 early_stopping = callbacks.EarlyStopping(
    monitor='val_loss',
    patience=5,
    min_delta=1e-7,
    restore_best_weights=True,
)

plateau = callbacks.ReduceLROnPlateau(
    monitor='val_loss',
    factor = 0.2,
    patience = 2,
    min_delt = 1e-7,
    cooldown = 0,
    verbose = 1
    def get_model():

    #Input shape = [width, height, color channels]
    inputs = layers.Input(shape=(IMG_SIZE, IMG_SIZE, 3))

    # Block One
    x = layers.Conv2D(filters=16, kernel_size=3, padding='valid')(inputs)
    x = layers.BatchNormalization()(x)
    x = layers.Activation('relu')(x)
    x = layers.MaxPool2D()(x)
    x = layers.Dropout(0.2)(x)

    # Block Two
    x = layers.Conv2D(filters=32, kernel_size=3, padding='valid')(x)
    x = layers.BatchNormalization()(x)
    x = layers.Activation('relu')(x)
    x = layers.MaxPool2D()(x)
    x = layers.Dropout(0.2)(x)

    # Block Three
    x = layers.Conv2D(filters=64, kernel_size=3, padding='valid')(x)
    x = layers.Conv2D(filters=64, kernel_size=3, padding='valid')(x)
    x = layers.BatchNormalization()(x)
    x = layers.Activation('relu')(x)
    x = layers.MaxPool2D()(x)
    x = layers.Dropout(0.4)(x)

    # Head
    #x = layers.BatchNormalization()(x)
    x = layers.Flatten()(x)
    x = layers.Dense(64, activation='relu')(x)
    x = layers.Dropout(0.5)(x)

    #Final Layer (Output)
    output = layers.Dense(1, activation='sigmoid')(x)
```

```
235
236     model = keras.Model(inputs=[inputs], outputs=output)
237
238     return model
239     keras.backend.clear_session()
240
241 model = get_model()
242 model.compile(loss='binary_crossentropy',
243               optimizer=keras.optimizers.Adam(learning_rate=3e-5),
244               metrics=['binary_accuracy']) # Changed to a list
245 model.summary()
246 history = model.fit(ds_train,
247           batch_size = BATCH, epochs = 50,
248           validation_data=ds_val,
249           callbacks=[early_stopping, plateau],
250           # Use integer division //  to ensure steps_per_epoch and validation_steps are
                integers
251           steps_per_epoch=len(train_df) // BATCH,
252           validation_steps=len(val_df) // BATCH);
253
254 )
255 fig, ax = plt.subplots(figsize=(20,8))
256 sns.lineplot(x = history.epoch, y = history.history['loss'])
257 sns.lineplot(x = history.epoch, y = history.history['val_loss'])
258 ax.set_title('Learning Curve (Loss)')
259 ax.set_ylabel('Loss')
260 ax.set_xlabel('Epoch')
261 ax.set_ylim(0, 0.5)
262 ax.legend(['train', 'val'], loc='best')
263 plt.show()
264 score = model.evaluate(ds_val, steps=len(val_df) // BATCH, verbose=0)  # Use // for
      integer division
265 print('Val loss:', score[0])
266 print('Val accuracy:', score[1])
267 score = model.evaluate(ds_test, steps = len(df_test), verbose = 0)
268
269 print('Test loss:', score[0])
270 print('Test accuracy:', score[1])
271 base_model = tf.keras.applications.ResNet152V2(
272     weights='imagenet',
273     input_shape=(IMG_SIZE, IMG_SIZE, 3),
274     include_top=False)
275
276 base_model.trainable = False
277
278 def get_pretrained():
279
280     #Input shape = [width, height, color channels]
281     inputs = layers.Input(shape=(IMG_SIZE, IMG_SIZE, 3))
282
```

```
    x = base_model(inputs)

    # Head
    x = layers.GlobalAveragePooling2D()(x)
    x = layers.Dense(128, activation='relu')(x)
    x = layers.Dropout(0.1)(x)

    #Final Layer (Output)
    output = layers.Dense(1, activation='sigmoid')(x)

    model = keras.Model(inputs=[inputs], outputs=output)

    return model
    keras.backend.clear_session()

model_pretrained = get_pretrained()
model_pretrained.compile(loss='binary_crossentropy',
              optimizer=keras.optimizers.Adam(learning_rate=5e-5),
              metrics=['binary_accuracy'])

model_pretrained.summary()
IMG_SIZE = 224 # Define IMG_SIZE before using it

import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers

def get_pretrained():
    """
    Returns a pre-trained ResNet152V2 model with a custom head.
    """
    # Input shape = [width, height, color channels]
    inputs = layers.Input(shape=(IMG_SIZE, IMG_SIZE, 3))

    # Include the base model instantiation here
    base_model = tf.keras.applications.ResNet152V2(
        weights='imagenet',
        input_shape=(IMG_SIZE, IMG_SIZE, 3),
        include_top=False)
    base_model.trainable = False

    x = base_model(inputs)

    # Head
    x = layers.GlobalAveragePooling2D()(x)
    x = layers.Dense(128, activation='relu')(x)
    x = layers.Dropout(0.1)(x)

    # Final Layer (Output)
    output = layers.Dense(1, activation='sigmoid')(x)
```

```
333
334     model = keras.Model(inputs=[inputs], outputs=output)
335
336     return model
337
338 keras.backend.clear_session()
339
340 model_pretrained = get_pretrained()
341 model_pretrained.compile(loss='binary_crossentropy',
342                 optimizer=keras.optimizers.Adam(learning_rate=5e-5),
343                 metrics=['binary_accuracy'])
344
345 model_pretrained.summary()
346 def history_plot():
347     # Indent the code within the function
348     fig, ax = plt.subplots(figsize=(20,8))
349     sns.lineplot(x = history.epoch, y = history.history['binary_accuracy'])
350     sns.lineplot(x = history.epoch, y = history.history['val_binary_accuracy'])
351     ax.set_title('Learning Curve (Accuracy)')
352     ax.set_ylabel('Accuracy')
353     ax.set_xlabel('Epoch')
354     ax.set_ylim(0.80, 1.0)
355     ax.legend(['train', 'val'], loc='best')
356     plt.show()
357     def history_plot():
358     # The code within the function should be indented
359     fig, ax = plt.subplots(figsize=(20,8))
360     sns.lineplot(x = history.epoch, y = history.history['loss'])
361     sns.lineplot(x = history.epoch, y = history.history['val_loss'])
362     ax.set_title('Learning Curve (Loss)')
363     ax.set_ylabel('Loss')
364     ax.set_xlabel('Epoch')
365     ax.set_ylim(0, 0.5)
366     ax.legend(['train', 'val'], loc='best')
367     plt.show()
368     def history_plot():
369     # Indent the code within the function by 4 spaces
370     fig, ax = plt.subplots(figsize=(20,8))
371     sns.lineplot(x = history.epoch, y = history.history['binary_accuracy'])
372     sns.lineplot(x = history.epoch, y = history.history['val_binary_accuracy'])
373     ax.set_title('Learning Curve (Accuracy)')
374     ax.set_ylabel('Accuracy')
375     ax.set_xlabel('Epoch')
376     ax.set_ylim(0.80, 1.0)
377     ax.legend(['train', 'val'], loc='best')
378     plt.show()
379     data_dir = '/content/test_images'  # Replace with the actual path to your test images
380 import os
381 if os.path.exists(data_dir):
382     print("Path exists")
```

```
else:
    print("Path does not exist")

train_class1_dir = '/test_images/class1'  # Path to class 1 images
train_class2_dir = '/test_images/class2'  # Path to class 2 images
# ... add paths for other classes if needed
IMG_SIZE = 224  # Define image size

import tensorflow as tf
import numpy as np
import os

# Define the correct path to your test dataset
# Make sure this path is correct! Use an absolute path if unsure.
data_dir = '/content/test_images'

# Print the current working directory for debugging
print(f"Current working directory: {os.getcwd()}")

# Check if the path exists
if not os.path.exists(data_dir):
    # Instead of raising an error, provide instructions to upload the data
    print(f"Directory '{data_dir}' does not exist.")
    print("Please upload your 'test_images' folder to the Colab environment.")
    # You can add instructions on how to upload using the files tab or code
    # For example:
    print("You can upload it by clicking on the 'Files' tab on the left and then clicking
        'Upload'.")
    # Optionally, you can halt execution here to allow the user to upload
    # import sys; sys.exit(1)
else:
    print(f"Directory '{data_dir}' found.")

# ... (rest of your code)
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression

# Load the digits dataset
digits = load_digits()
X, y = digits.data, digits.target

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train a logistic regression model
model = LogisticRegression(max_iter=10000)  # Increase max_iter if needed
model.fit(X_train, y_train)
```

```
432
433 # Make predictions on the test set
434 y_pred = model.predict(X_test)
435
436 # Generate confusion matrix
437 cm = confusion_matrix(y_test, y_pred)
438 disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=digits.target_names)
439 disp.plot(cmap=plt.cm.Blues)
440 plt.show()
```

References

[1] Agrawal T, Choudhary P (2022) Segmentation and classification on chest radiography: a systematic survey. Vis Comput 39(3):875–913. https://doi.org/10.1007/s00371-021-02352-7

[2] Candemir S, Antani S (2019) A review on lung boundary detection in chest X-rays. Int J Comput Assist Radiol Surg 14(4):563–576. https://doi.org/10.1007/s11548-019-01917-1

[3] Sharma H, Jain JS, Bansal P, Gupta S (2020) Feature extraction and classification of chest X-ray images using CNN to detect pneumonia. Proc. Conflu. –10th Int. Conf. Cloud Comput. Data Sci Eng, pp 227–231. https://doi.org/10.1109/Confluence47617.2020.9057809

[4] Litjens G et al (2017) A survey on deep learning in medical image analysis. Med Image Anal 42(1995):60–88. https://doi.org/10.1016/j.media.2017.07.005

[5] Palani S, Kulkarni A, Kochara A (2020) Detection of thoracic diseases using deep learning. ITM Web Conf 32:03024

[6] Ching T et al (2018) Opportunities and obstacles for deep learning in biology and medicine, 15(141). https://doi.org/10.1098/rsif.2017.0387

[7] Meedeniya D, Kumarasinghe H, Kolonne S, Fernando C, Díez ID, Marques G (2022) Chest X-ray analysis empowered with deep learning: A systematic review. Appl Soft Comput 126:109319. https://doi.org/10.1016/j.asoc.2022.109319

[8] Page M. J et al (2021) The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. BMJ. 372. https://doi.org/10.1136/bmj.n71

[9] Hasanah U et al (2023) CheXNet and feature pyramid network: a fusion deep learning architecture for multilabel chest X-Ray clinical diagnoses classification. Int J Cardiovasc Imaging 40(4):709–722. https://doi.org/10.1007/s10554-023-03039-x

[10] Albahli S, Rauf HT, Algosaibi A, Balas VE (2021) AI-driven deep CNN approach for multilabel pathology classification using chest X-Rays. PeerJ Comput Sci 7:1–17. https://doi.org/10.7717/peerj-cs.495