A

Major Project On

**MULTI-DISEASE DETECTION SYSTEM WITH X-RAY IMAGES USING DEEP LEARNING TECHNIQUES**

(Submitted in partial fulfillment of the requirements for the award of Degree)

### BACHELOR OF TECHNOLOGY

In

### COMPUTER SCIENCE AND ENGINEERING

By

1. MEGHANA (217R1A05L2)

N. MANOJ KUMAR (217R1A05P2)

1. AMRUTHA VARSHINI (217R1A05N0)

Under the Guidance of

#### Dr. J. NARASIMHARAO

(Associate Professor)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**CMR TECHNICAL CAMPUS**

**UGC AUTONOMOUS**

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi)

Recognized Under Section 2(f) & 12(B) of the UGCAct.1956, Kandlakoya (V), Medchal Road, Hyderabad-501401.

**April, 2025.**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

****

## CERTIFICATE

This is to certify that the project entitled “**MULTI-DISEASE DETECTION SYSTEM WITH X-RAY IMAGES USING DEEP LEARNING TECHNIQUES**” being submitted by **B. MEGHANA (217R1A05L2), N. MANOJ KUMAR (217R1A05P2) & G. AMRUTHA VARSHINI (217R1A05N0)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, during the year 2024-25.

The results embodied in this project have not been submitted to any other University or Institute for the award of any degree or diploma.

**Dr. J. Narasimharao Dr. Nuthanakanti Bhaskar Associate Professor HoD**

**INTERNAL GUIDE**

**Dr. A. Raji Reddy Signature of External Examiner DIRECTOR**

**Submitted for viva voice Examination held on**

## ACKNOWLEDGEMENT

We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project, we take this opportunity to express our profound gratitude and deep regard to our guide **Dr. J. Narasimharao,** Associate Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We take this opportunity to extend our heartfelt appreciation to the Project Review Committee (PRC) Coordinators—**Dr. K. Maheswari, Dr. J. Narasimharao, Ms. K. Shilpa, and Mr. K. Ranjith Reddy**—for their unwavering support, insightful guidance, and valuable inputs, which played a crucial role in steering this project through its various stages.

Our sincere appreciation also goes to **Dr. Nuthanakanti Bhaskar**, Head, for his encouragement and continuous support in ensuring the successful completion of our project.

We are deeply grateful to **Dr. A. Raji Reddy**, Director, for his cooperation throughout the course of this project. Additionally, we extend our profound gratitude to Sri. **Ch. Gopal Reddy**, Chairman, Smt. **C. Vasantha Latha**, Secretary and Sri. **C. Abhinav Reddy**, Vice-Chairman, for fostering an excellent infrastructure and a conducive learning environment that greatly contributed to our progress.

We also acknowledge and appreciate the guidance and assistance provided by the faculty and staff of **CMR Technical Campus**, whose contributions have been invaluable in bringing this project to fruition.

Lastly, we sincerely thank our families for their unwavering support and encouragement. We also extend our gratitude to the teaching and non-teaching staff of CMR Technical Campus for their guidance and assistance. Their contributions, along with the support of everyone who helped directly or indirectly, have been invaluable in the successful completion of this project.

**B. MEGHANA (217R1A05L2)**

**N. MANOJ KUMAR (217R1A05P2)**

**G. AMRUTHA VARSHINI (217R1A05N0)**

## VISION AND MISSION

### INSTITUTE VISION:

To Impart quality education in serene atmosphere thus strive for excellence in Technology and Research.

### INSTITUTE MISSION:

1. To create state of art facilities for effective Teaching- Learning Process.
2. Pursue and Disseminate Knowledge based research to meet the needs of Industry & Society.
3. Infuse Professional, Ethical and Societal values among Learning Community.

### DEPARTMENT VISION:

To provide quality education and a conducive learning environment in computer engineering that foster critical thinking, creativity, and practical problem-solving skills.

### DEPARTMENT MISSION:

1. To educate the students in fundamental principles of computing and induce the skills needed to solve practical problems.
2. To provide State-of-the-art computing laboratory facilities to promote industry institute interaction to enhance student’s practical knowledge.
3. To inculcate self-learning abilities, team spirit, and professional ethics among the students to serve society.

## ABSTRACT

Multi-disease detection system that utilizes deep learning techniques to analyze X-ray images for the identification of various medical conditions, including pneumonia, tuberculosis, and fractures. By leveraging a comprehensive dataset of labeled X-ray images, we implement convolutional neural networks (CNNs) and explore transfer learning with pre-trained models to enhance classification accuracy. The project undergoes rigorous training, validation, and testing, incorporating image normalization and augmentation techniques to improve robustness. Performance metrics such as accuracy, precision, recall, and F1-score are employed to evaluate the model's effectiveness. The dataset used for training and evaluation consists of a diverse collection of medical images representing different stages and manifestations of the target diseases. The performance of the proposed framework is assessed through comprehensive experimentation, including accuracy, sensitivity, specificity, and area under the receiver operating characteristic curve (AUC-ROC) metrics. It demonstrate promising performance in multi-class classification tasks, indicating the potential of deep learning to assist radiologists and healthcare professionals in making timely and accurate diagnoses. This project underscores the importance of integrating artificial intelligence into clinical workflows to enhance diagnostic efficiency while adhering to healthcare regulations and ethical standards.

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **FIGURE NAME** | **PAGE NO** |
| Figure 3.1 | Project Architecture of Multi-Disease Detection System with X-Ray Images Using Deep Learning Techniques. | 14 |
| Figure 3.2 | Dataflow Diagram of Multi-Disease Detection System with X-Ray Images Using Deep Learning Techniques. | 17 |
| Figure 4.1 | Dataset directory structure with folders "Pneumonia dataset" having all the training examples. | 21 |
| Figure 4.2 | Screenshot of the "Pneumonia" Folder Showing Sample Image Files. | 22 |
| Figure 4.3 | Screenshot of the "Normal" Folder Showing Sample Image Files. | 21 |

|  |  |  |
| --- | --- | --- |
| Figure 5.1 | GUI/Main Interface of f Multi-Disease Detection System With X-Ray Images Using Deep Learning Techniques. | 33 |
| Figure 5.2 | Dataset of A Multi-Disease Detection System With X-Ray Images Using Deep Learning Techniques. | 34 |
| Figure 5.3 | Dataset Preprocessing of A Multi-Disease Detection System With X-Ray Images Using Deep Learning Techniques. | 35 |
| Figure 5.4 | Evaluation Metrics and Confusion Matrix of the VGG16 algorithm of A Multi-Disease Detection System With X-Ray Images Using Deep Learning Techniques. | 36 |
| Figure 5.5 | Evaluation Metrics and Confusion Matrix of the VGG19 algorithm of A Multi-Disease Detection System With X-Ray Images Using Deep Learning Techniques. | 37 |
| Figure 5.6 | Comparison graph between the VGG16 and the VGG19 algorithm of A Multi-Disease Detection System With X-Ray Images Using Deep Learning Techniques. | 38 |

|  |  |  |
| --- | --- | --- |
| Figure 5.7 | Tumor detected in the given image of A Multi-Disease Detection System With X-Ray Images Using Deep Learning Techniques. | 39 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **TABLE NO** | **TABLE NAME** | **PAGE NO** |
| Table 6.2.1 | Uploading Dataset | 41 |
| Table 6.2.2 | Classification | 41 |

**TABLE OF CONTENTS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ABSTRACT** | |  |  | i |
| **LIST OF FIGURES** | | |  | ii |
| **LIST OF TABLES** | | |  | v |
| **1.** | **INTRODUCTION** | |  | 1 |
|  | 1.1 | PROJECT PURPOSE | | 1 |
|  | 1.2 | PROJECT FEATURES | | 2 |
| **2.** | **LITERATURE SURVEY** | | | 3 |
|  | 2.1 | REVIEW OF RELATED WORK | | 7 |
|  | 2.2 | DEFINITION OF PROBLEM STATEMENT | | 9 |
|  | 2.3 | EXISTING SYSTEM | | 10 |
|  | 2.4 | PROPOSED SYSTEM | | 11 |
|  | 2.5 | OBJECTIVES | | 12 |
|  | 2.6 | HARDWARE & SOFTWARE REQUIREMENTS | | 13 |
|  |  | 2.6.1 | HARDWARE REQUIREMENTS | 13 |
|  |  | 2.6.2 | SOFTWARE REQUIREMENTS | 13 |
| **3.** | **SYSTEM ARCHITECTURE & DESIGN** | | | 14 |
|  | 3.1 | PROJECT ARCHITECTURE | | 14 |
|  | 3.2 | DESCRIPTION | | 15 |
|  | 3.3 | DATA FLOW DIAGRAM | | 16 |
| **4.** | **IMPLEMENTATION** | | | 18 |
|  | 4.1 | ALGORITHMS USED | | 18 |
|  | 4.2 | SAMPLE CODE | | 24 |
| **5.** | **RESULTS & DISCUSSION** | | | 33 |
| **6.** | **VALIDATION** | |  | 40 |
|  | 6.1 | INTRODUCTION | | 40 |
|  | 6.2 | TEST CASES | | 41 |
|  |  | 6.2.1 | UPLOADING DATASET | 41 |
|  |  | 6.2.2 | CLASSIFICATION | 41 |
| **7.** | **CONCLUSION & FUTURE ASPECTS** | | | 42 |
|  | 7.1 | PROJECT CONCLUSION | | 42 |
|  | 7.2 | FUTURE ASPECTS | | 43 |
| **8.** | **BIBLIOGRAPHY** | |  | 44 |
|  | 8.1 | REFERENCES | | 44 |
|  | 8.2 | GITHUB LINK | | 46 |

# INTRODUCTION

## INTRODUCTION

A **multi-disease detection system** using **X-ray images** and **deep learning techniques** aims to revolutionize the way diseases are diagnosed by automating the process of identifying multiple conditions from medical images. X-ray imaging is one of the most common diagnostic tools used in healthcare to detect various diseases, including pneumonia, tuberculosis, and COVID-19. However, traditional methods rely heavily on manual analysis by radiologists, which can be slow, error-prone, and challenging, especially when diseases exhibit similar symptoms. Deep learning offers a solution by allowing computers to automatically learn patterns from X-ray images and classify them into different disease categories.

By employing advanced **deep learning models**, such as **Convolutional Neural Networks (CNNs)**, these systems can process and analyze X-ray images more accurately and efficiently. The key advantage is the ability to detect multiple diseases in a single image, improving diagnostic accuracy and reducing the time needed to identify potential health issues. These systems are particularly beneficial in settings where access to trained medical professionals is limited, such as rural or resource-constrained environments. Ultimately, the **multi-disease detection system** seeks to assist healthcare providers in making faster, more reliable diagnoses, leading to better patient care and outcomes.

### PROJECT PURPOSE

The primary purpose of this project is to develop an automated system that can efficiently and accurately detect multiple diseases from a single X-ray image. By leveraging advanced deep learning models, the system aims to assist healthcare professionals in identifying conditions such as lung infections, fractures, tumors, and other abnormalities, improving diagnostic speed and accuracy. This system seeks to reduce the workload of radiologists, minimize the risk of misdiagnosis, and ultimately enhance patient care and outcomes. It also aims to be adaptable, capable to integrate new data and emerging diseases over time.

### PROJECT FEATURES

This project incorporates several key features to improve the accuracy and efficiency of video classification and inappropriate content detection:

**Deep Learning-based Image Classification**: Using advanced deep learning models (like CNNs), the system automatically analyzes and classifies X-ray images, ensuring high accuracy in detecting diseases based on subtle image features.

### **Multi-Class Classification & Multi-Label Detection:**The system supports **multi-class classification**, meaning it can predict multiple possible diseases in a single image (e.g., it can detect both pneumonia and tuberculosis simultaneously if they exist in the same X-ray).The system can handle **multi-label detection**, indicating whether multiple diseases are present in the same X-ray image.

Scalability & Real-Time Detection: The system can predict the likelihood of a disease's presence in a given image, with a confidence score. The system should be able to handle large datasets of X-ray images, supporting healthcare institutions with thousands of patients.Cloud-based systems can scale as required based on the demand.

**Interpretability and Visual Explanations**: Offers tools like heatmaps to explain the model’s decision-making process, ensuring transparency and trust in results.

By integrating these features, this deep learning-based system have a tremendous impact on the healthcare sector by assisting medical professionals in diagnosing diseases quickly, improved accurately and efficiently, thus improving patient outcomes.

# LITERATURESURVEY

## LITERATURE SURVEY

The field of automated disease detection using X-ray images has seen significant advancements with the application of deep learning techniques. Studies such as Wang et al.’s work on the ChestX-ray14 dataset have demonstrated the effectiveness of convolutional neural networks (CNNs) in multi-label classification for detecting chest diseases. The use of transfer learning with pre-trained models like ResNet and VGG16 has enhanced diagnostic performance, especially with limited datasets. Techniques like attention mechanisms and ensemble learning have further improved model sensitivity and specificity. Current research focuses on detecting multiple diseases, addressing issues of imbalanced datasets, and ensuring interpretability of model predictions to aid clinical decision-making.

Deep learning (DL) has emerged as a powerful tool in the field of medical image analysis, particularly for detecting diseases using X-ray images. X-ray images are commonly used in healthcare because they are affordable, quick, and effective for diagnosing many conditions. However, detecting multiple diseases from these images is challenging due to the complexity of medical imaging and the variations in image quality. Deep learning offers a promising solution by automating the detection process and improving diagnostic accuracy. In this literature survey, we review recent advancements in using deep learning for multi-disease detection from X-ray images.

This literature survey reviews past research on inappropriate content detection and classification, highlighting existing methodologies, their strengths and weaknesses, and areas for improvement. The focus is on traditional machine learning models, deep learning architectures, transfer learning techniques, to enhance classification performance.

Deep learning, particularly Convolutional Neural Networks (CNNs), has proven to be highly effective in analyzing medical images, including X-rays. CNNs automatically learn hierarchical features from images, eliminating the need for manual feature extraction. This capability is especially useful for medical image analysis, where the complexity and subtlety of features can be challenging for traditional methods. Early works like those of **LeCun et al. (2015)** demonstrated the success of CNNs in visual tasks, laying the groundwork for their use in medical imaging.

In recent years, the focus has shifted toward building systems that can detect multiple diseases from a single X-ray image, instead of just one. This multi-disease detection approach can enhance the efficiency of diagnostic systems and help clinicians make quicker decisions. For instance, a chest X-ray might show signs of pneumonia, tuberculosis, and other lung diseases simultaneously. **Generative Adversarial Networks (GANs)** to generate synthetic images, helping balance datasets by producing more examples of rare diseases.By training a model to recognize multiple conditions, such systems can save time and reduce the workload on radiologists.

One of the primary challenges in multi-disease detection is **multi-class classification**, where a model is trained to classify X-ray images into multiple distinct categories, such as pneumonia, tuberculosis, or normal. A significant advancement in this area was **Rajpurkar et al. (2017)**, who introduced **CheXNet**, a CNN-based model for detecting pneumonia in chest X-rays. Their work showed that deep learning models can achieve performance comparable to experienced radiologists, pushing the boundaries of AI in healthcare.

In deep learning models for medical image analysis is the **lack of large, labeled datasets**. Annotating medical images requires expert radiologists, which makes it difficult to gather enough data for training deep learning models. **Transfer learning** is a popular solution to this issue. By using pre-trained models (such as those trained on large, non-medical datasets like ImageNet) and fine-tuning them for specific tasks, researchers can leverage existing knowledge without needing vast amounts of medical data. Studies like **Tajbakhsh et al. (2020)** and **Chouhan et al. (2021)** demonstrated that transfer learning could significantly improve the performance of models for multi-disease detection from X-ray images.ome studies have also explored **Generative Adversarial Networks (GANs)** to generate synthetic images, helping balance datasets by producing more examples of rare diseases.

Deep learning models, especially CNNs, are often seen as "black boxes," making it difficult to understand how they reach their conclusions. In healthcare, it's essential for clinicians to trust AI models, so interpretability is crucial. Techniques like **Grad-CAM** (which visualizes the regions of the image that most influence the model’s decision) and **attention mechanisms** (which help the model focus on relevant parts of the image) are being explored to make deep learning models more interpretable and transparent.

Deep learning techniques, particularly **CNNs**, have become the backbone of many state-of-the-art medical imaging systems. **LeCun et al. (2015)** first demonstrated the power of CNNs in visual tasks, setting the foundation for their application in medical image analysis. Later, **Litjens et al. (2017)** conducted a comprehensive review of CNNs, showing that they could significantly outperform traditional methods in tasks like disease detection from radiological images, including X-rays. CNNs automatically extract important features from the images without requiring manual intervention, making them highly effective in analyzing complex patterns in medical images.

Recent research has focused on advancing the performance of multi-disease detection models by exploring new architectures and techniques. One exciting development is the application of **transformer-based models**, which have shown exceptional performance in natural language processing tasks and are now being adapted for medical image analysis. These models can capture long-range dependencies and contextual relationships within images, potentially improving multi-disease detection.

The primary techniques used for multi-disease detection include **multi-class classification** and **multi-label classification**. In **multi-class classification**, the model is trained to assign an X-ray image to one of several distinct categories, such as pneumonia, tuberculosis, or normal lungs. In **multi-label classification**, the model can assign multiple labels to a single image, allowing for the detection of more than one disease at once. Multi-label classification is more complex and requires the model to identify overlapping or co-occurring diseases, making it a more advanced task in medical imaging.

The application of deep learning to multi-disease detection using X-ray images has seen significant progress in recent years. The development of models for **multi-class** and **multi-label** classification has enhanced the diagnostic potential of X-ray imaging. However, challenges such as **data imbalance**, **interpretability**, and **generalization** across datasets remain. Recent advancements in **transfer learning**, **transformer models**, and **federated learning** show great promise in addressing these issues. As the field continues to evolve, the integration of AI with medical imaging will likely lead to faster, more accurate, and more reliable disease detection, benefiting both clinicians and patients.

Recent researchers have been exploring advanced techniques such as **transfer learning**, where models trained on large, general image datasets are fine-tuned for medical tasks. This helps mitigate the problem of small or unbalanced medical datasets. **Multimodal learning**, which combines X-ray images with other patient data (such as medical history or demographic information), is also being used to improve diagnostic accuracy. Furthermore, **federated learning** has emerged as a solution to privacy concerns. It allows models to be trained across multiple institutions without sharing sensitive patient data, enabling collaborative development while maintaining patient privacy.

Traditional techniques relied heavily on **manual feature extraction**, where experts would design specific algorithms to detect features like edges or textures in X-ray images. In contrast, modern **deep learning models**, particularly CNNs, automatically learn to extract features from the raw data, making them more efficient and capable of detecting complex patterns without human intervention.

These methods focused on detecting specific, predefined patterns like edges and textures using algorithms designed by radiologists. They were typically designed for detecting **one disease at a time**, requiring separate processes for each condition, and were limited by the need for expert knowledge. Modern deep learning techniques, such as **Convolutional Neural Networks (CNNs)**, automate the entire process, learning features directly from raw data without manual intervention. These models are capable of detecting **multiple diseases simultaneously** from a single image, significantly improving diagnostic efficiency.

Unlike traditional methods, which required curated datasets and struggled with **generalization**, deep learning models utilize large, diverse datasets and can adapt to various medical environments. While traditional techniques offered **clearer interpretability**, modern deep learning models are more complex and often operate as "black boxes." However, advancements in **model interpretability** are helping to make these systems more transparent. Overall, modern techniques offer higher accuracy, adaptability, and speed, though they face challenges like data requirements and interpretability.

### REVIEW OF RELATED WORK

The application of deep learning in medical image analysis, particularly for **multi-disease detection in X-ray images**, has garnered significant attention in recent years. Previous research has explored a wide range of techniques, from traditional methods to cutting-edge deep learning models, aiming to enhance diagnostic accuracy and automate the detection of multiple diseases. This review discusses the evolution of methodologies used in this domain, emphasizing their strengths and limitations.

* + - 1. **Early Approaches and Traditional Techniques**:

Early methods of disease detection in X-ray images relied on **rule-based systems** and **handcrafted feature extraction**. These systems used **image processing techniques** such as edge detection, thresholding, and texture analysis to identify certain conditions. While these methods were an important step toward automation, they were limited by the need for manual intervention and expert knowledge. The inability to adapt to varying image qualities and patient demographics further constrained the effectiveness of these techniques.

* + - 1. Machine Learning-Based Approaches:

With machine learning algorithms such as **Support Vector Machines (SVM)**, **Random Forests**, or **Gradient Boosting Machines (GBM)** can be used to classify the extracted features into multiple disease categories. These models are especially useful for handling challenges like **data imbalance** and **multi-label classification**, improving the overall accuracy of disease detection. M**achine learning algorithms** can refine the final predictions, enhance multi-disease detection, and help manage challenges like **data imbalance**. Additionally, hybrid models that combine deep learning for feature extraction and machine learning for classification can provide better performance and robustness in diagnosing multiple diseases simultaneously.

* + - 1. Deep Learning-Based Approaches:

Recent advancements in deep learning, particularly **Convolutional Neural Networks (CNNs)**, revolutionized disease detection in X-rays. A CNN model trained to detect pneumonia from chest X-rays, which outperformed radiologists in some cases. Since then, deep learning models have been extended to detect a wide range of diseases, including **tuberculosis**, **lung cancer**, **pleural effusion**, and more. These models automatically extract relevant features from images and can learn to recognize complex patterns, offering a significant improvement over traditional methods.

One of the most effective architectures for image processing is CNN-LSTM hybrids, where CNNs extract spatial features, and LSTMs analyze temporal relationships. Studies have demonstrated the effectiveness of pre-trained CNN models like VGG-16, ResNet, and EfficientNet for feature extraction, followed by LSTM or GRU layers for sequence modeling. Despite their success, these models often require large-scale annotated datasets and suffer from computational complexity, limiting their real-time application.

* + - 1. Recent Advances: Attention Mechanisms & Transformer-Based Models:

Transformer-based models, such as the **Vision Transformer (ViT)**, employ a mechanism that processes the entire image as a sequence, capturing long-range dependencies and global context. Additionally, **hybrid models** that combine **CNNs with transformers** and **Attention U-Net** architectures are being used to enhance feature extraction and disease classification accuracy. **Vision Transformer (ViT)** and **Swin Transformer** are examples of transformer-based models that have demonstrated strong performance in medical image tasks, including **disease segmentation** and classification.

However, these models require massive computational resources and pre-training on large datasets.

* + - 1. Comparison with the Proposed Approach:

While existing methods have made significant progress in multi-disease detection, challenges remain in terms of accuracy, scalability, and real-time detection. The proposed approach in this project can identify multiple diseases simultaneously. Moreover, current systems often require extensive manual feature extraction, whereas the deep learning model reduces the need for this, offering faster and more accurate results. Real-time detection is another key advantage, as the deep learning system can provide immediate results, making it suitable for clinical environments where time is crucial. The proposed approach also adapts better to various datasets, ensuring better generalization across diverse patient populations.

This review highlights the evolution of content moderation techniques, emphasizing the machine learning approaches to deep learning-based models. The proposed methodology aims to address the limitations of previous research by offering a highly accurate, scalable. The system’s ability to use deep learning, particularly CNNs, to automatically analyze and classify diseases is a significant advancement over traditional methods. The review focuses on the accuracy, efficiency, and scalability of the approach, highlighting its potential to improve diagnostic processes in healthcare.

### DEFINITION OF PROBLEM STATEMENT

The primary goal of this project is to create an **automated system** that can detect multiple diseases like pneumonia, tuberculosis, and Brain Tumor from **X-ray images** using deep learning. This system will help doctors by providing faster and more accurate diagnoses. The key objectives include achieving **high accuracy** and **sensitivity** while identifying various diseases from X-ray scans. The system must also **handle diverse and imbalanced datasets** effectively to ensure reliable results. It will use **advanced deep learning models** to process images and make predictions. The model should be able to **differentiate between diseases** that may have similar symptoms. Additionally, the system aims to be **interpretable**, so doctors can trust its results. By automating the detection process, it will help save time and support medical professionals, especially in areas with limited access to experts.

### EXISTING SYSTEM

Existing systems for disease detection using X-ray images mainly focus on identifying specific conditions like pneumonia, tuberculosis, or fractures. These systems often rely on deep learning models, particularly Convolutional Neural Networks (CNNs), which can analyze images and detect abnormalities. Some systems, like CheXNet, are trained to identify multiple diseases from chest X-rays, but they are limited to a few conditions. Many models still struggle with variations in image quality, differences in equipment, and the need for large, labeled datasets. Although progress has been made, most systems are not fully capable of detecting a wide range of diseases from a single X-ray image. Multi-disease detection using deep learning algorithms represents a significant advancement in medical diagnostics, offering enhanced accuracy, efficiency, and scalability compared to traditional methods. By harnessing the power of artificial intelligence and medical imaging data, these innovative approaches have the potential to revolutionize disease detection and management, ultimately improving patient outcomes and healthcare delivery worldwide.The challenge remains to create an efficient, multi-disease detection system that can handle diverse image data and provide accurate diagnoses across different medical conditions.

#### Limitations of Existing System

Despite improvements in video content classification, the existing system suffers from the following challenges:

* + - * Data Limitations: Many systems need a lot of labeled data to train, but for rare diseases, this data may not be available. If the data is limited, the model may become overfitted, meaning it works well on the training data but struggles with new, unseen data.
      * Computational Resource Requirements: Training deep learning models requires a lot of computational power and memory, which may not be available in all healthcare settings. This limits the ability to use these systems, especially in areas with fewer resources.
      * Limited Multi-Disease Detection Capability: While some systems focus on single disease detection, fewer models effectively address multi-disease scenarios. This limitation may necessitate the development of separate models for each condition, complicating clinical decision-making.
      * Need for Continuous Learning:As medical knowledge and disease patterns change, models need regular updates and retraining to stay accurate, which can be costly and time- consuming.

### PROPOSED SYSTEM

The proposed system advocates for the development of transfer learning techniques and data augmentation strategies to mitigate the need for large annotated datasets. By leveraging pre-trained deep learning models on related tasks or domains and fine-tuning them with limited labeled data, the proposed system can overcome data scarcity issues and improve model generalization across diverse patient populations and disease manifestations. Additionally, data augmentation methods such as geometric transformations, random noise injection, and generative adversarial networks can artificially increase the diversity and size of training datasets, enhancing model robustness and reducing the risk of overfitting. By incorporating attention mechanisms, explainable AI techniques, and uncertainty quantification methods into deep learning architectures, the system aims to provide clinicians with insights into the decision-making process of the model and identify salient features contributing to disease predictions.

A multi-label classification framework will be implemented to allow simultaneous detection of multiple diseases, providing healthcare professionals with comprehensive diagnostic insights. Adherence to privacy regulations, transparent data sharing practices, and bias mitigation strategies are integral components of the proposed system's design. To enhance interpretability, it include visualization tools, such as Grad-CAM, to illustrate model decision-making processes, thereby fostering trust among clinicians. By addressing key challenges while prioritizing patient safety, data privacy, and clinical utility, the proposed system aims to accelerate the translation of AI-driven innovations into routine clinical practice, improving healthcare outcomes and enhancing the quality of patient care.

#### Advantages of the Proposed System:

The proposed system significantly improves upon the existing approaches by addressing key limitations:

* Multi-Disease Detection Capability: This system can detect multiple diseases from one X-ray image at the same time, unlike other systems that can only identify one disease, making it a more complete diagnostic tool.
* Adaptability to Emerging Diseases: The system can be quickly updated with new data and retrained to detect emerging diseases or changes in how diseases appear, keeping it effective in a constantly changing healthcare environment.
* Data-Driven Insights for Clinical Decisions: The system can provide valuable data-driven insights to assist healthcare professionals in making informed clinical decisions, ultimately enhancing the quality of patient care.
* Transfer Learning Efficiency: Utilizing pre-trained models through transfer learning allows the system to achieve robust performance even with limited labeled data.

### OBJECTIVES

* Develop Disease Detection System – Develop a deep learning model to accurately classify multiple diseases, enabling automated diagnosis from a single scan.
* Enhance System Efficiency – Reduce reliance on manual review by implementing DL techniques.
* Improve Detection Accuracy – Utilize ResNet-152V2, VGG-16 and VGG-19 with attention to achieve higher accuracy than traditional models.
* Ensure Scalability – Design the system to handle handle increasing data volumes, users, and computational demands without compromising performance.

### HARDWARE & SOFTWARE REQUIREMENTS

* + 1. **HARDWARE REQUIREMENTS:**

Hardware interfaces specifies the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements,

|  |  |  |
| --- | --- | --- |
| * Processor | : | i3 processor or above. |
| * Hard disk | : | 40GB. |
| * RAM | : | 512GB. |

### SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements,

* + - * Operating system : Windows 10 or above.
      * Coding Language : Python(3.7.0)
      * Frame Work : Tkinter.

# 3. SYSTEM ARCHITECTURE &

**DESIGN**

## SYSTEM ARCHITECTURE & DESIGN

Project architecture refers to the structural framework and design of a project, encompassing its components, interactions, and overall organization. It provides a clear blueprint for development, ensuring efficiency, scalability, and alignment with project goals. Effective architecture guides the project's lifecycle, from planning to execution, enhancing collaboration and reducing complexity.

### PROJECT ARCHITECTURE

This project architecture shows the procedure followed multi-disease detection system using X-ray images and deep learning techniques, starting from input to final detection.

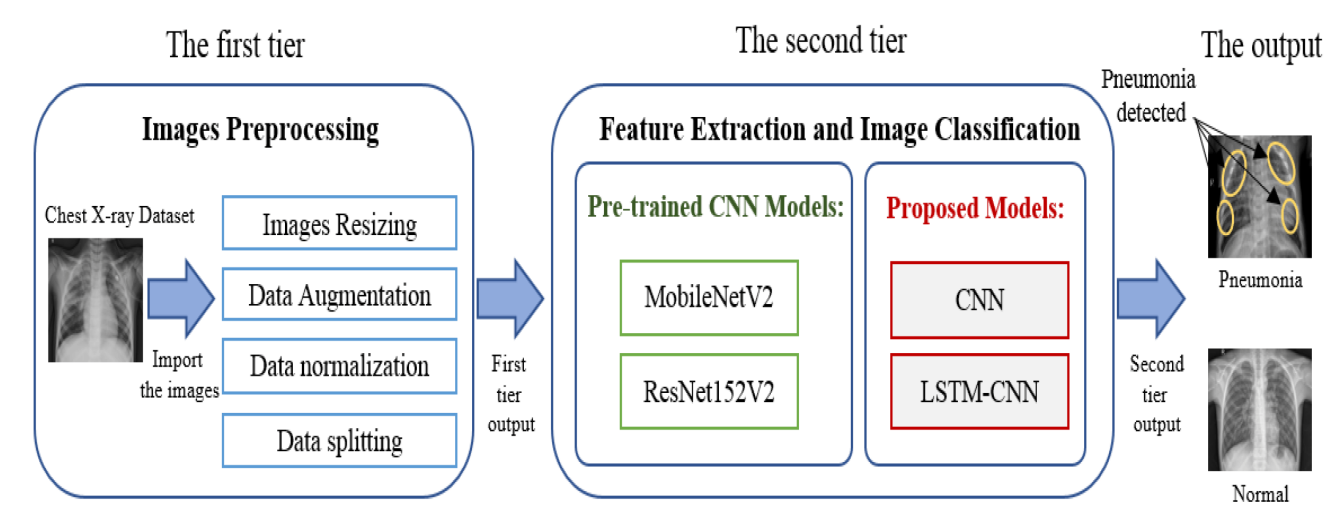


Figure 3.1: Project Architecture of Multi-disease detection system using X-ray images and deep learning techniques.

### DESCRIPTION

This image presents a two-tiered deep learning-based approach for pneumonia detection from chest X-ray images. Here’s a breakdown of the process:

The First Tier: Images Preprocessing

* The chest X-ray dataset is first imported.
* Several preprocessing techniques are applied:
  + Images Resizing: Standardizing image sizes for consistent input.
  + Data Augmentation: Enhancing the dataset using transformations like rotation, flipping, and contrast adjustments.
  + Data Normalization: Scaling pixel values to a standard range to improve model performance.
  + Data Splitting: Dividing the dataset into training, validation, and testing sets.

The Second Tier: Feature Extraction and Image Classification

* The processed images are passed through deep learning models for feature extraction and classification.
* Two types of models are used:
  + Pre-trained CNN Models: MobileNetV2 and ResNet152V2, which are well-established architectures known for efficient image classification.
  + Proposed Models: A basic CNN and an LSTM-CNN hybrid model, where LSTM (Long Short-Term Memory) is integrated with CNN for sequential image processing.

The Output:

* The system classifies the X-ray image as either Pneumonia (highlighting affected regions) or Normal (indicating a healthy lung).
* If pneumonia is detected, the affected areas in the lung are marked.

### DATA FLOW DIAGRAM

A Data Flow Diagram (DFD) is a graphical representation that illustrates how data flows within a system, showcasing its processes, data stores, and external entities. It is a vital tool in system analysis and design, helping stakeholders visualize the movement of information, identify inefficiencies, and optimize workflows.

A Data Flow Diagram comprises Four primary elements:

* + - External Entities: Represent sources or destinations of data outside the system.
    - Processes: Indicate transformations or operations performed on data.
    - Data Flows: Depict the movement of data between components.
    - Data Stores: Represent where data is stored within the system.

These components are represented using standardized symbols, such as circles for processes, arrows for data flows, rectangles for external entities, and open-ended rectangles for data stores.

**Benefits:**

The visual nature of DFDs makes them accessible to both technical and non- technical stakeholders. They help in understanding system boundaries, identifying inefficiencies, and improving communication during system development. Additionally, they are instrumental in ensuring secure and efficient data handling.

**Applications:**

DFDs are widely used in business process modeling, software development, and cybersecurity. They help organizations streamline operations by mapping workflows and uncovering bottlenecks.

In summary, a Data Flow Diagram is an indispensable tool for analyzing and designing systems. Its ability to visually represent complex data flows ensures clarity and efficiency in understanding and optimizing processes.

**Levels of DFD:**

DFDs are structured hierarchically:

* Level 0 (Context Diagram): These images are collected from a medical dataset, ensuring they contain both normal and pneumonia-affected samples.
* Level 1: Data augmentation such as rotation and contrast adjustments.
* Level 2: The system outputs either "Pneumonia Detected" or "Normal".

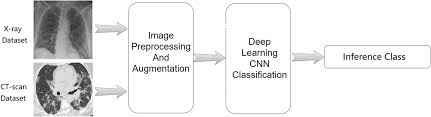


Figure 3.2: Dataflow Diagram of Multi-disease detection system using X-ray images and

deep learning techniques.

# 4. IMPLEMENTATION

## IMPLEMENTATION

The implementation phase of a project involves executing the planned strategies and tasks. It requires meticulous coordination, resource allocation, and monitoring to ensure that objectives are met efficiently. Effective implementation is crucial for achieving project goals and delivering expected outcomes within the set timeline and budget constraints.

### ALGORITHMS USED

#### CNN-Based Models for Feature Extraction

Convolutional Neural Networks (CNNs) are widely used for video frame classification as they are highly effective at capturing spatial features. Models such as ResNet-50, VGG-16, and EfficientNet have been employed to analyze individual frames to determine whether the content is safe or unsafe. CNNs, particularly EfficientNet-B7, extract spatial features from video frames to detect inappropriate content. By identifying patterns in images, CNNs help recognize explicit or violent elements. These captures temporal relationships between frames, improving detection accuracy.

Advantages of CNN-Based Models:

* Pre-trained models like MobileNetV2 and ResNet152V2 have been trained on large datasets, improving accuracy in pneumonia detection.
* Pooling layers help reduce computational complexity while retaining essential features.
* They can generalize well across different X-ray datasets and real-world scenarios

Disadvantages of CNN-Based Models:

* CNNs, especially deep architectures like ResNet152V2, require significant computational power for training and inference.
* CNN models need a vast amount of labeled chest X-ray images to generalize well.

**4.1.1. MobileNetV2**

* A lightweight Convolutional Neural Network (CNN) designed for mobile and embedded devices.
* Uses depthwise separable convolutions to reduce computational cost while maintaining accuracy.
* Implements inverted residual blocks with linear bottlenecks for efficient feature extraction.
* Works well for image classification with fewer parameters and lower memory requirements.
* Used in this project as a pre-trained feature extractor for pneumonia detection.

**4.1.2. ResNet152V2**

* A deep residual network with 152 layers, designed to tackle the vanishing gradient problem.
* Uses skip connections (residual learning) to allow gradients to flow easily during backpropagation.
* Improves training stability and accuracy, even in deep networks.
* Suitable for high-performance image classification tasks, including medical imaging.
* Utilized as a pre-trained CNN model for extracting pneumonia-related features.

#### ****4.1.3. CNN (Convolutional Neural Network)****

* A neural network designed for **image processing and feature extraction** using convolutional layers.
* Uses **convolution, pooling, and activation functions** to detect spatial patterns in chest X-rays.
* Helps in identifying key pneumonia-related features like lung abnormalities.
* Requires a large amount of labeled data to perform well.
* Used as a **custom model** in this project for pneumonia classification.

**4.1.4. LSTM-CNN (Hybrid Model)**

* A combination of CNN for spatial feature extraction and LSTM (Long Short-Term Memory) for sequential learning.
* CNN processes chest X-ray images to extract important visual features.
* LSTM captures temporal dependencies, which helps in cases where X-ray images show progressive pneumonia stages.
* Improves accuracy by retaining long-term feature dependencies compared to standard CNNs.
* Proposed as an advanced model for more precise pneumonia classification.

The identification and treatment of lung disease infection have gotten great interest around the world. Most scientific studies revealed that different lung disease symptoms can be seen in X-Ray images of the lungs. The wide availability of X-Ray images made imaging modalities a suitable way for early detection of lung disease infection. The major achievements of deep learning (DL) approaches in detecting certain irregularities in medical images, motivated researchers to learn more about deep CNN architectures for lung disease classification in both X-Ray images.

The pre-trained CNN models, VGG16Net and VGG19Net, are developed for discriminating the lung diseases like COVID-19, pneumonia, and normal (healthy) lung both X-Ray and CT scans.

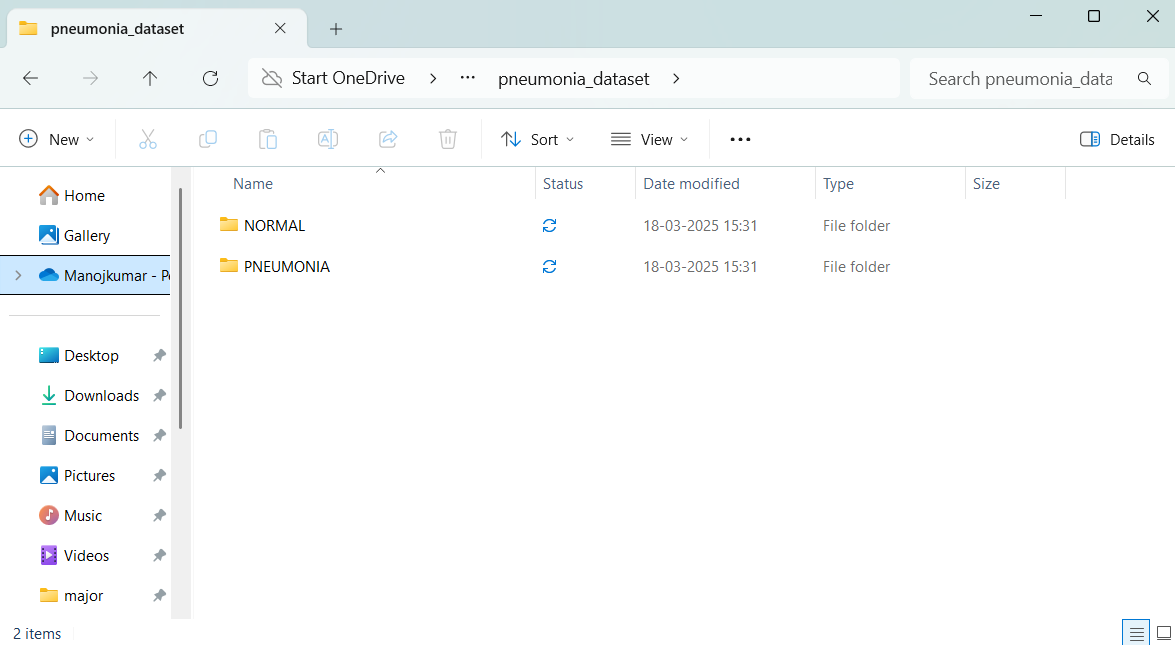
Chest X-ray images are collected from medical datasets and undergo preprocessing to enhance model performance. The images are first resized to a fixed dimension to ensure uniformity. Normalization is applied to scale pixel values between 0 and 1, reducing computation complexity.

Preprocessed images are passed through pre-trained CNN models like MobileNetV2 and ResNet152V2, which extract meaningful visual patterns from X-ray scans. These models use convolutional layers to detect lung abnormalities while leveraging transfer learning to enhance accuracy. Additionally, a custom CNN model is trained to capture specific disease-related features. In some cases, an LSTM-CNN hybrid model is used, combining CNN’s spatial analysis with LSTM’s ability to retain sequential dependencies for better classification.

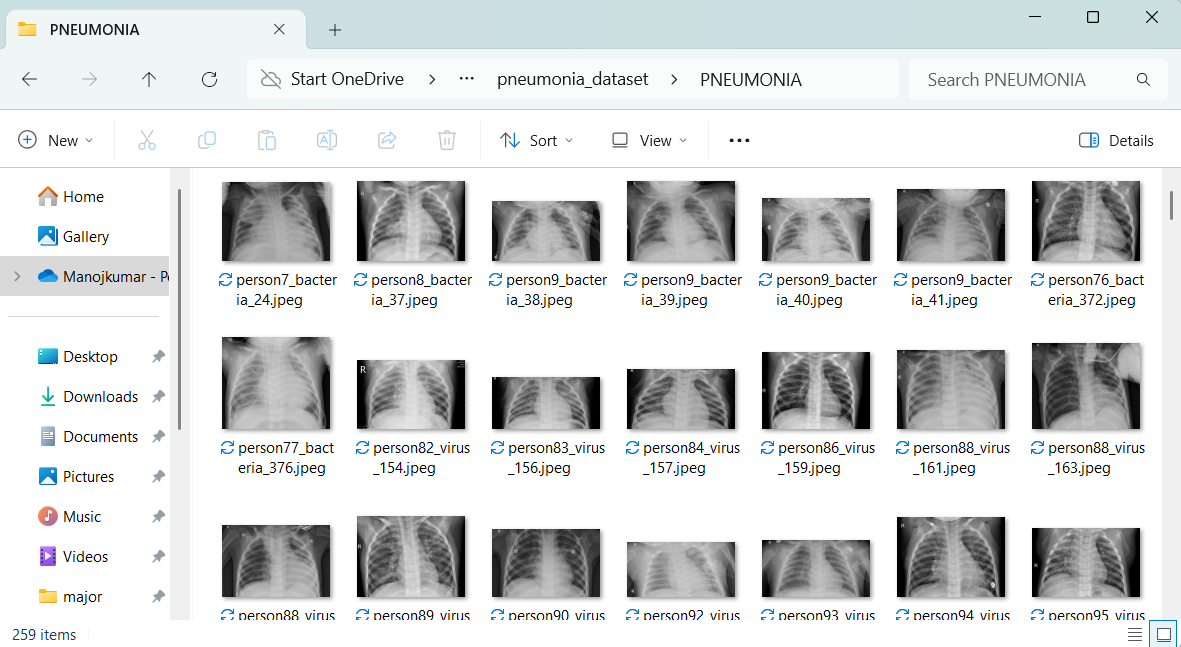
The extracted features are fed into a fully connected layer, where a classification algorithm predicts the disease type. The system outputs whether the X-ray is normal or affected by diseases such as pneumonia, tuberculosis, lung cancer, or COVID-19. In cases of disease detection, visualization techniques are used to highlight infected regions for better interpretability. The final predictions help radiologists and healthcare professionals make informed decisions regarding patient diagnosis and treatment.

We have used ‘X-ray image’ type of dataset to train above algorithms. For the problematic human x-ray images the dataset which is used as Pneumonia (Figure 4.2)and for the normal human x-ray images the dataset which is used as Normal images (Figure 4.3).

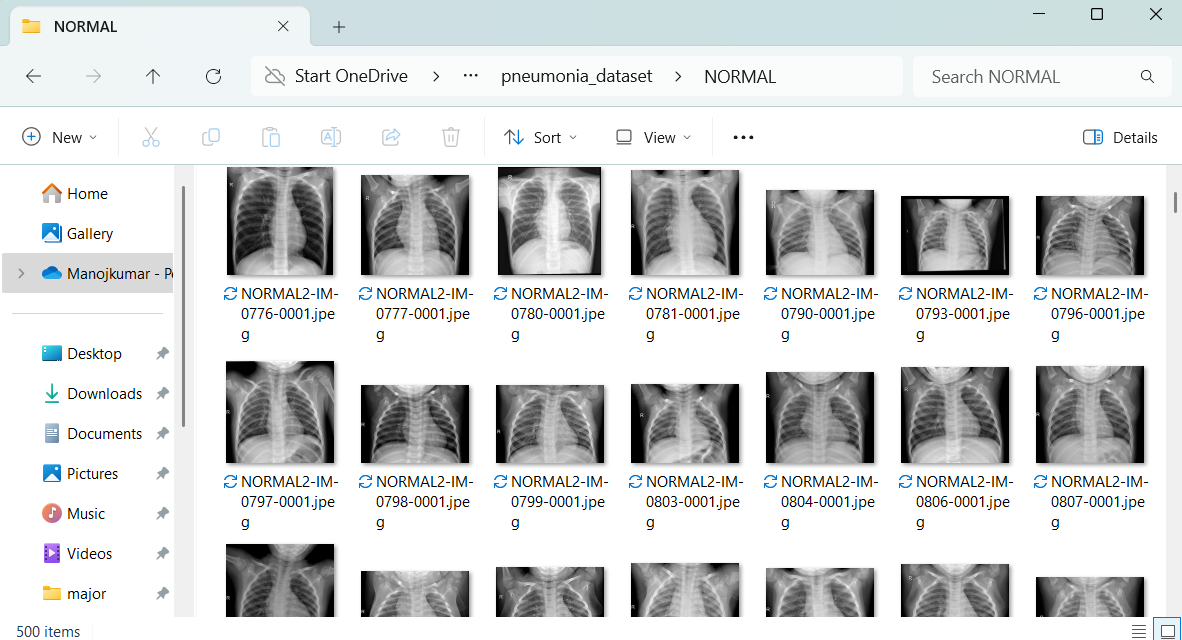
To train algorithm we have used below pneumonia dataset images consist of two folders (Figure 4.1) and below screen showing dataset details



**Figure 4.1**: Dataset directory structure with folders “Pneumonia dataset” having all the training examples



**Figure 4.2**: Screenshot of the "Pneumonia" Folder Showing Sample Image Files.



**Figure 4.3**: Screenshot of the "Normal image" Folder Showing Sample Image Files.

To implement this project, we have designed following modules:

1. **Upload X-Ray Images Dataset**: Upload x-ray images dataset to the application.

(Figure 4.1).

1. **Dataset Preprocessing**: Preprocess and extract x-ray images and then shuffle and split dataset where application using 80% dataset for training and 20% for testing
2. **Run VGG16 Algorithms:** Input 80% training data to VGG16 to train a model and this model will be applied on test data to calculate prediction accuracy.
3. **Run VGG19 Algorithms:** Input 80% training data to VGG19 to train a model and this model will be applied on test data to calculate prediction accuracy.
4. **Comparison graph:**Plot accuracy graph between VGG16 and VGG19 algorithms.
5. **Disease detection:** Upload test image dataset to application to extract x-ray images and then predict weather image is normal or diseases are detected.

### SAMPLE CODE

from tkinter import messagebox

from tkinter import \*

from tkinter import simpledialog

import tkinter

from tkinter import filedialog

from tkinter.filedialog import askopenfilename

import cv2

import random

import numpy as np

from keras.utils.np\_utils import to\_categorical

from keras.layers import  MaxPooling2D

from keras.layers import Dense, Dropout, Activation, Flatten

from keras.layers import Convolution2D

from keras.models import Sequential

from sklearn.model\_selection import train\_test\_split

from keras.applications import VGG16

from keras.applications import VGG19

from sklearn.metrics import accuracy\_score

from keras.callbacks import ModelCheckpoint

import pickle

import os

from keras.models import load\_model

from sklearn.metrics import precision\_score

from sklearn.metrics import recall\_score

from sklearn.metrics import f1\_score

from sklearn.metrics import accuracy\_score

import matplotlib.pyplot as plt

from sklearn.metrics import confusion\_matrix

import seaborn as sns

import pydicom as dicom

main = tkinter.Tk()

main.title("Multi disease detection using x-ray images")

main.geometry("1300x1200")

global filename

global classifier

global labels, X, Y, X\_train, y\_train, X\_test, y\_test, vgg16\_model

def uploadDataset():

    global filename

    global labels

    labels = []

    filename = filedialog.askdirectory(initialdir=".")

    pathlabel.config(text=filename)

    text.delete('1.0', END)

    text.insert(END,filename+" loaded\n\n");

    labels = ['Normal', 'Alzheimer Brain Tumor']

    text.insert(END,"Tumor found in dataset are\n\n")

    for i in range(len(labels)):

        text.insert(END,labels[i]+"\n")

def processDataset():

    text.delete('1.0', END)

    global filename, X, Y, X\_train, y\_train, X\_test, y\_test

    if os.path.exists("model/X.txt.npy"):

        X = np.load('model/X.txt.npy')

        Y = np.load('model/Y.txt.npy')

    else:

        for root, dirs, directory in os.walk(filename):

            for j in range(len(directory)):

                name = os.path.basename(root)

                if 'Thumbs.db' not in directory[j]:

                    ds = dicom.dcmread(root+'/'+directory[j])

                    img = ds.pixel\_array

                    cv2.imwrite("test.png", img\*255)

                    img = cv2.imread("test.png")

                    img = cv2.resize(img, (32,32))

                    im2arr = np.array(img)

                    im2arr = im2arr.reshape(32,32,3)

                    X.append(im2arr)

                    if name == 'Normal':

                        Y.append(0)

                    else:

                        Y.append(1)

                    print(name+" "+str(label))

        X = np.asarray(X)

        Y = np.asarray(Y)

        np.save('model/X.txt',X)

        np.save('model/Y.txt',Y)

    X = X.astype('float32')

    X = X/255

    text.insert(END,"Dataset Preprocessing Completed\n")

    text.insert(END,"Total images found in dataset : "+str(X.shape[0])+"\n\n")

    indices = np.arange(X.shape[0])

    np.random.shuffle(indices)

    X = X[indices]

    Y = Y[indices]

    Y = to\_categorical(Y)

    X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size=0.2)

    text.insert(END,"80% images are used to train VGG16 & 19 :"+str(X\_train.shape[0])+"\n")

    text.insert(END,"20% images are used to test  : "+str(X\_test.shape[0])+"\n")

    text.update\_idletasks()

    img = X[0]

    img = cv2.resize(img, (200, 200))

    cv2.imshow("Brain Image", img)

    cv2.waitKey(0)

def trainVGG16():

    text.delete('1.0', END)

    global filename, X, Y, X\_train, y\_train, X\_test, y\_test, labels, vgg16\_model

    vgg16 = VGG16(include\_top=False, weights='imagenet', input\_shape=(X\_train.shape[1], X\_train.shape[2], X\_train.shape[3]))

    for layer in vgg16.layers:

        layer.trainable = False

    vgg16\_model = Sequential()

    vgg16\_model.add(vgg16)

    vgg16\_model.add (Convolution2D(32, (1, 1), input\_shape = (X\_train.shape[1], X\_train.shape[2], X\_train.shape[3]), activation = 'relu'))

    vgg16\_model.add(MaxPooling2D(pool\_size = (1, 1)))

    vgg16\_model.add(Convolution2D(32, (1, 1), activation = 'relu'))

    vgg16\_model.add(MaxPooling2D(pool\_size = (1, 1)))

    vgg16\_model.add(Flatten())

    vgg16\_model.add(Dense(units = 256, activation = 'relu'))

    vgg16\_model.add(Dense(units = y\_train.shape[1], activation = 'softmax'))

    vgg16\_model.compile(optimizer = 'adam', loss = 'categorical\_crossentropy', metrics = ['accuracy'])

    if os.path.exists("model/vgg16\_weights.hdf5") == False:

        model\_check\_point = ModelCheckpoint(filepath='model/vgg16\_weights.hdf5', verbose = 1, save\_best\_only = True)

        hist = vgg16\_model.fit(X\_train, y\_train, batch\_size = 32, epochs = 20, validation\_data=(X\_test, y\_test), callbacks=[model\_check\_point], verbose=1)

        f = open('model/vgg16\_history.pckl', 'wb')

        pickle.dump(hist.history, f)

        f.close()

    else:

        vgg16\_model = load\_model("model/vgg16\_weights.hdf5")

    predict = vgg16\_model.predict(X\_test)

    predict = np.argmax(predict, axis=1)

    testY = np.argmax(y\_test, axis=1)

p = precision\_score(testY, predict,average='macro') \* 100

    r = recall\_score(testY, predict,average='macro') \* 100

    f = f1\_score(testY, predict,average='macro') \* 100

    a = accuracy\_score(testY,predict)\*100

    text.insert(END,"VGG16 Accuracy  : "+str(a)+"\n")

    text.insert(END,"VGG16 Precision : "+str(p)+"\n")

    text.insert(END,"VGG16 Recall    : "+str(r)+"\n")

    text.insert(END,"VGG16 FSCORE    : "+str(f)+"\n\n")

    conf\_matrix = confusion\_matrix(testY, predict)

    plt.figure(figsize =(6, 6))

    ax = sns.heatmap(conf\_matrix, xticklabels = labels, yticklabels = labels, annot = True, cmap="viridis" ,fmt ="g");

    ax.set\_ylim([0,len(labels)])

    plt.title("VGG16 Confusion matrix")

    plt.ylabel('True class')

    plt.xlabel('Predicted class')

    plt.show()

def trainVGG19():

    global filename, X, Y, X\_train, y\_train, X\_test, y\_test, labels, vgg19\_model

    vgg19 = VGG19(include\_top=False, weights='imagenet', input\_shape=(X\_train.shape[1], X\_train.shape[2], X\_train.shape[3]))

    for layer in vgg19.layers:

        layer.trainable = False

    vgg19\_model = Sequential()

    vgg19\_model.add(vgg19)

    vgg19\_model.add(Convolution2D(32, (1, 1), input\_shape = (X\_train.shape[1], X\_train.shape[2], X\_train.shape[3]), activation = 'relu'))

    vgg19\_model.add(MaxPooling2D(pool\_size = (1, 1)))

    vgg19\_model.add(Convolution2D(32, (1, 1), activation = 'relu'))

    vgg19\_model.add(MaxPooling2D(pool\_size = (1, 1)))

    vgg19\_model.add(Flatten())

    vgg19\_model.add(Dense(units = 256, activation = 'relu'))

vgg19\_model.add(Dense(units = y\_train.shape[1], activation = 'softmax'))

    vgg19\_model.compile(optimizer = 'adam', loss = 'categorical\_crossentropy', metrics = ['accuracy'])

    if os.path.exists("model/vgg19\_weights.hdf5") == False:

        model\_check\_point = ModelCheckpoint(filepath='model/vgg19\_weights.hdf5', verbose = 1, save\_best\_only = True)

        hist = vgg19\_model.fit(X\_train, y\_train, batch\_size = 32, epochs = 30, validation\_data=(X\_test, y\_test), callbacks=[model\_check\_point], verbose=1)

        f = open('model/vgg19\_history.pckl', 'wb')

        pickle.dump(hist.history, f)

        f.close()

    else:

        vgg19\_model = load\_model("model/vgg19\_weights.hdf5")

    predict = vgg19\_model.predict(X\_test)

    predict = np.argmax(predict, axis=1)

    testY = np.argmax(y\_test, axis=1)

    p = precision\_score(testY, predict,average='macro') \* 100

    r = recall\_score(testY, predict,average='macro') \* 100

    f = f1\_score(testY, predict,average='macro') \* 100

    a = accuracy\_score(testY,predict)\*100

    text.insert(END,"VGG19 Accuracy  : "+str(a)+"\n")

    text.insert(END,"VGG19 Precision : "+str(p)+"\n")

    text.insert(END,"VGG19 Recall    : "+str(r)+"\n")

    text.insert(END,"VGG19 FSCORE    : "+str(f)+"\n\n")

    conf\_matrix = confusion\_matrix(testY, predict)

    plt.figure(figsize =(6, 6))

    ax = sns.heatmap(conf\_matrix, xticklabels = labels, yticklabels = labels, annot = True, cmap="viridis" ,fmt ="g");

    ax.set\_ylim([0,len(labels)])

    plt.title("VGG19 Confusion matrix")

    plt.ylabel('True class')

    plt.xlabel('Predicted class')

plt.show()

def graph():

    f = open('model/vgg16\_history.pckl', 'rb')

    graph = pickle.load(f)

    f.close()

    vgg16\_accuracy = graph['val\_accuracy']

    f = open('model/vgg19\_history.pckl', 'rb')

    graph = pickle.load(f)

    f.close()

    vgg19\_accuracy = graph['val\_accuracy']

    vgg19\_accuracy = vgg19\_accuracy[10:30]

    plt.figure(figsize=(10,6))

    plt.grid(True)

    plt.xlabel('EPOCH')

    plt.ylabel('Accuracy')

    plt.plot(vgg16\_accuracy, 'ro-', color = 'green')

    plt.plot(vgg19\_accuracy, 'ro-', color = 'blue')

    plt.legend(['VGG16 Accuracy', 'VGG19 Accuracy'], loc='upper left')

    plt.title('VGG16 & 19 Training Accuracy Graph')

    plt.show()

def predictTumor():

    global vgg19\_model, labels

    filename = filedialog.askopenfilename(initialdir="testData")

    ds = dicom.dcmread(filename)

    img = ds.pixel\_array

    cv2.imwrite("test.png", img\*255)

    img = cv2.imread('test.png')

    img = cv2.resize(img, (32, 32))

    im2arr = np.array(img)

    im2arr = im2arr.reshape(1,32,32,3)

    img = np.asarray(im2arr)

img = img.astype('float32')

    img = img/255

    preds = vgg19\_model.predict(img)

    predict = np.argmax(preds)

    img = cv2.imread("test.png")

    img = cv2.resize(img, (700,400))

    cv2.putText(img, 'x-ray of Brain  : '+labels[predict], (10,25),  cv2.FONT\_HERSHEY\_SIMPLEX,0.7, (0, 0, 255), 2)

    cv2.imshow('Alzhimer Brain Tumor : '+labels[predict], img)

    cv2.waitKey(0)

def close():

main.destroy()

font = ('times', 16, 'bold')

title = Label(main, text='Multi disease detection using x-ray images',anchor=W, justify=CENTER)

title.config(bg='yellow4', fg='white')

title.config(font=font)

title.config(height=3, width=120)

title.place(x=0,y=5)

font1 = ('times', 13, 'bold')

upload = Button(main, text="Upload x-ray images Dataset", command=uploadDataset)

upload.place(x=50,y=100)

upload.config(font=font1)

pathlabel = Label(main)

pathlabel.config(bg='yellow4', fg='white')

pathlabel.config(font=font1)

pathlabel.place(x=50,y=150)

processButton = Button(main, text="Preprocess Dataset", command=processDataset)

processButton.place(x=50,y=200)

processButton.config(font=font1)

trainvggButton = Button(main, text="Train VGG16 Algorithm", command=trainVGG16)

trainvggButton.place(x=50,y=250)

trainvggButton.config(font=font1)

vggButton = Button(main, text="Train VGG19 Algorithm", command=trainVGG19)

vggButton.place(x=50,y=300)

vggButton.config(font=font1)

graphButton = Button(main, text="Comparison Graph", command=graph)

graphButton.place(x=50,y=350)

graphButton.config(font=font1)

predictButton = Button(main, text="Tumor Detection", command=predictTumor)

predictButton.place(x=50,y=400)

predictButton.config(font=font1)

font1 = ('times', 12, 'bold')

text=Text(main,height=25,width=78)

scroll=Scrollbar(text)

text.configure(yscrollcommand=scroll.set)

text.place(x=370,y=100)

text.config(font=font1)

main.config(bg='magenta3')

main.mainloop()

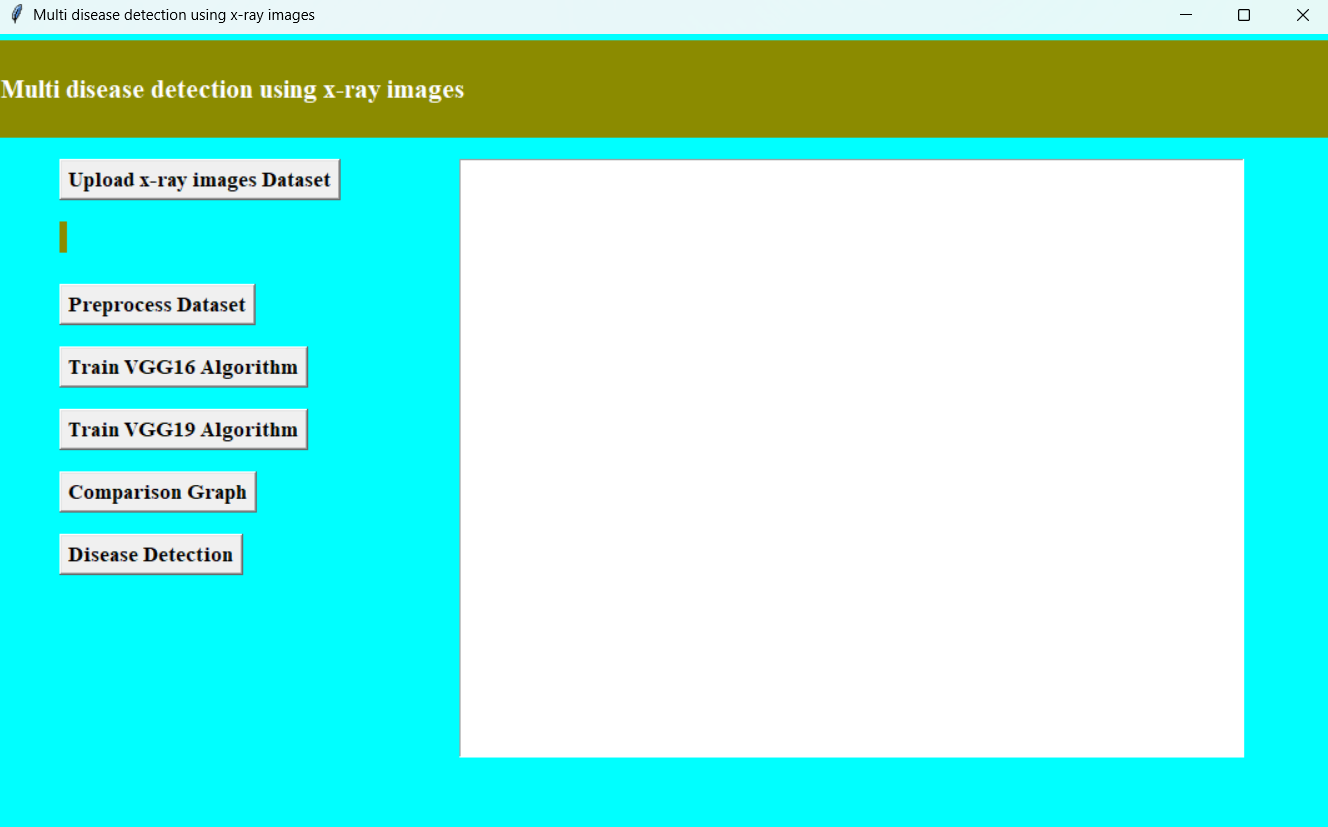
# 5. RESULTS & DISCUSSION

## RESULTS & DISCUSSION

The following screenshots showcase the results of our project, highlighting key features and functionalities. These visual representations provide a clear overview of how the system performs under various conditions, demonstrating its effectiveness and user interface. The screenshots serve as a visual aid to support the project's technical and operational achievements.

#### GUI/Main Interface :

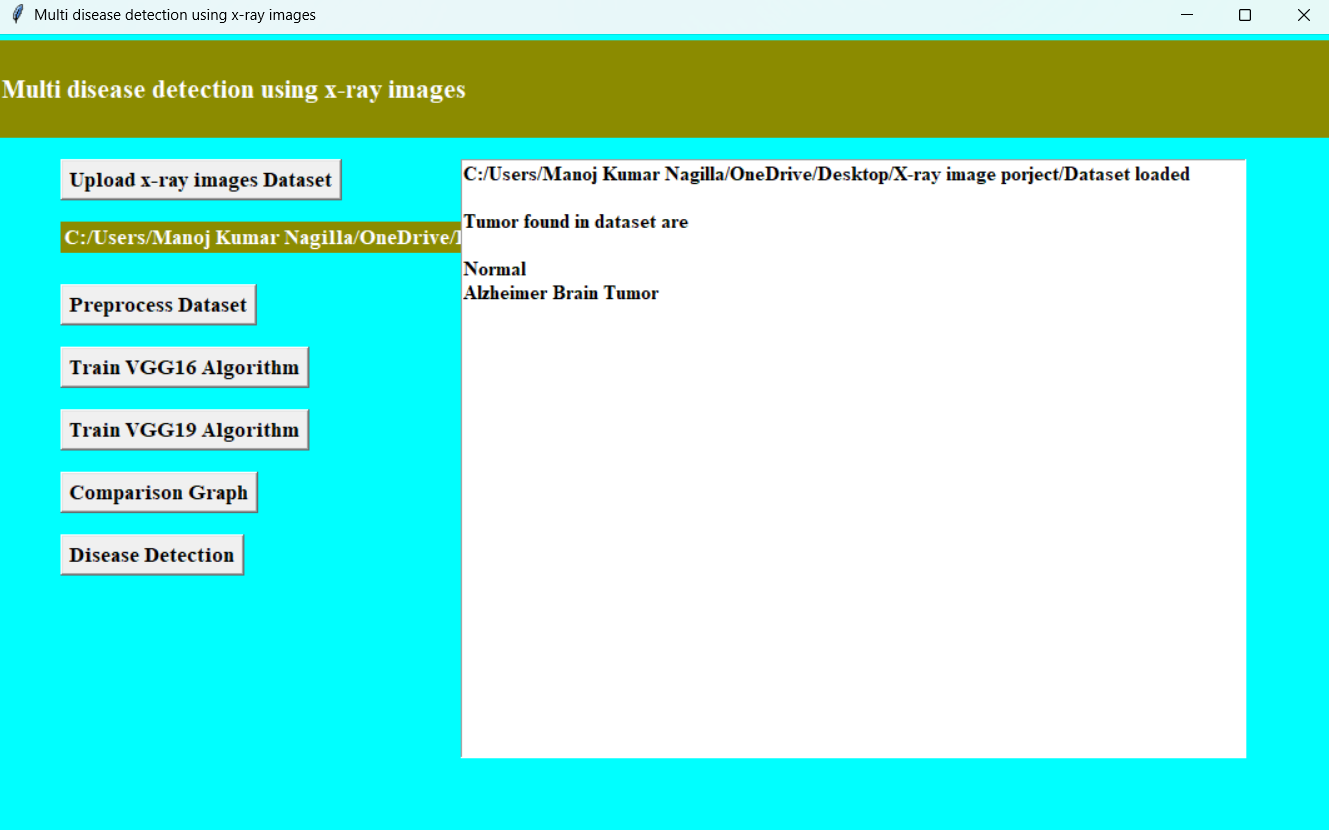
In below screen, click on ‘Upload x-ray images Dataset’ button to upload dataset.



**Figure 5.1 :** GUI/Main Interface of Multi-Disease Detection System With X-Ray Images Using Deep Learning Techniques.

#### Dataset :

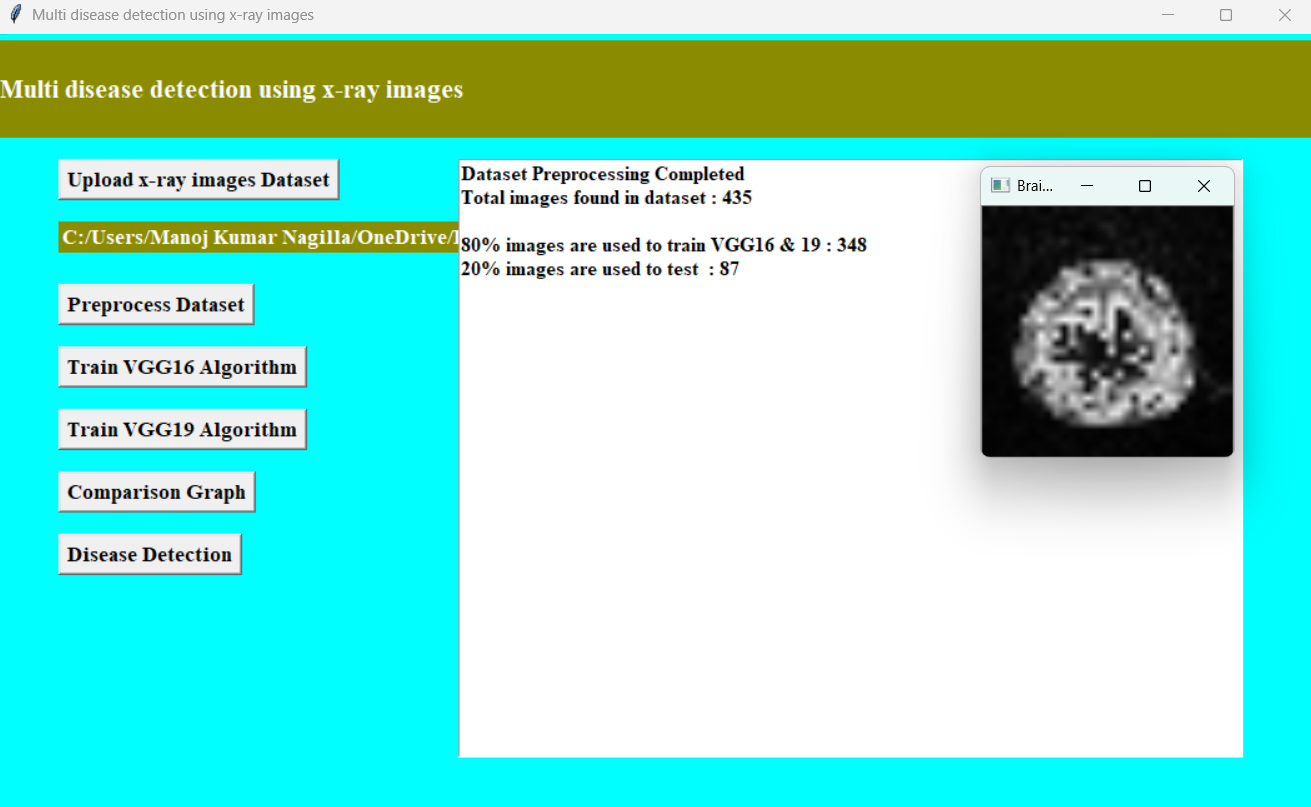
In below screen, click dataset upload button to upload all images in the dataset.



**Figure 5.2 :** Dataset of A Multi-Disease Detection System With X-Ray Images Using Deep Learning Techniques.

#### Dataset Preprocessing:

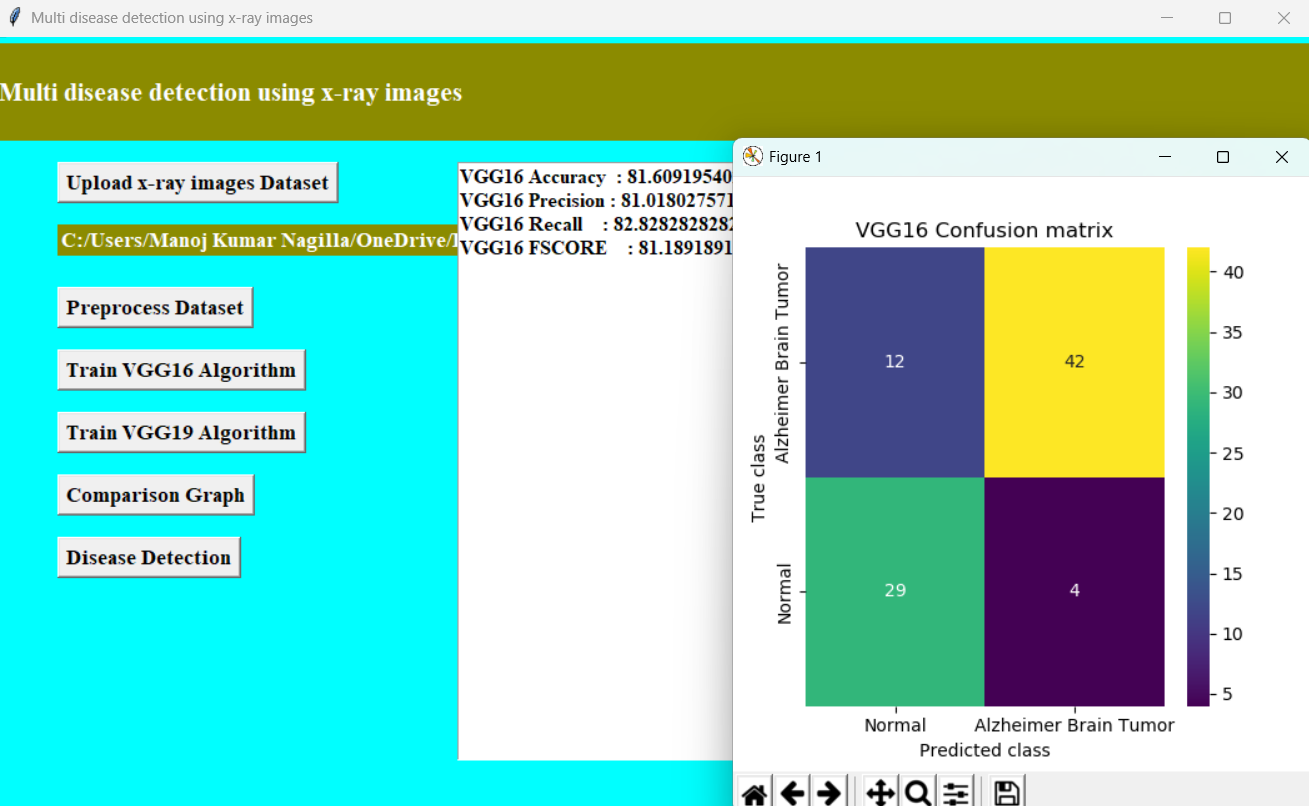
In below screen, dataset loaded and now click on Dataset Preprocessing button to read all images and then processes those images for training.



**Figure 5.3 :** Dataset Preprocessing of A Multi-Disease Detection System With X-Ray Images Using Deep Learning Techniques.

#### Evaluation Metrics and Confusion Matrix of the VGG16 algorithm

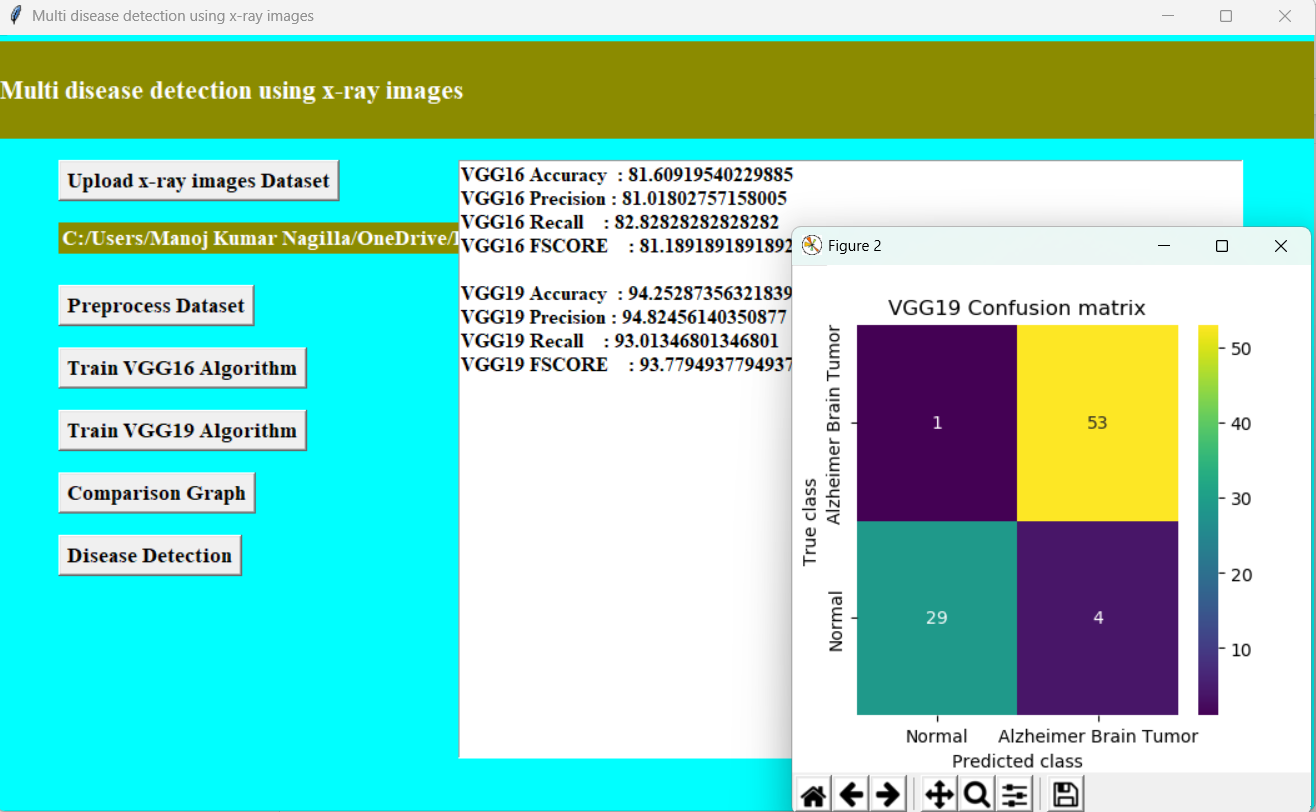
In below screen, with propose EfficientNetB7-BI-LSTM we got 99.04% accuracy and in confusion matrix graph x-axis represents Predicted Labels and y-axis represents True Labels and green and yellow boxes contains correct prediction count and blue boxes contains incorrect prediction count which is 2 only. Now close above graph and then click on ‘Run EfficientNet-SVM Algorithm’ button.



**Figure 5.4 :** Evaluation Metrics and Confusion Matrix of the VGG16 algorithm of A Multi-Disease Detection System With X-Ray Images Using Deep Learning Techniques.

#### Evaluation Metrics and Confusion Matrix of the VGG19 algorithm:

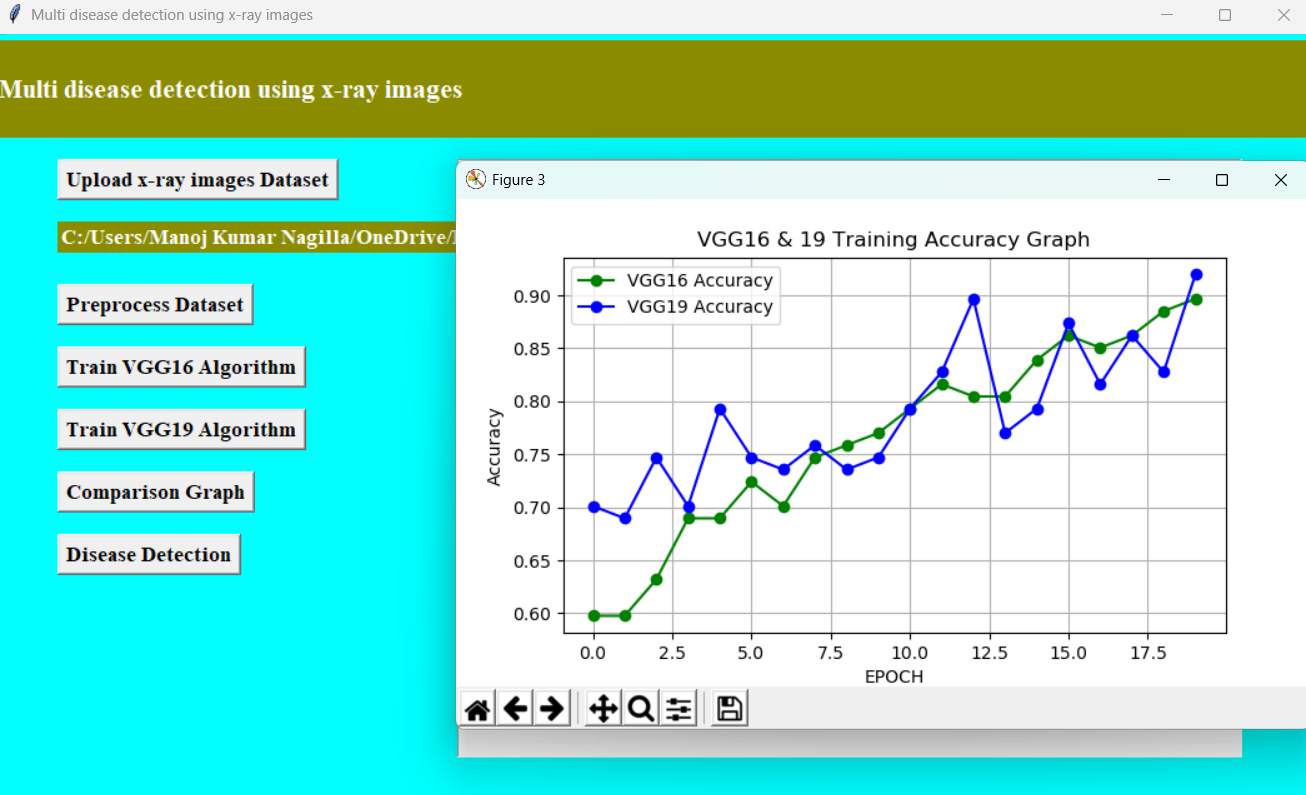
In below screen, with EfficientNetB7-SVM we got 88% accuracy and in confusion matrix graph we can see in blue boxes that SVM predicted total 24 incorrect prediction so its accuracy is less. Now close above graph and then click on ‘Comparison Graph’ button.



**Figure 5.5:** Evaluation Metrics and Confusion Matrix of the VGG19 algorithm of A Multi-Disease Detection System With X-Ray Images Using Deep Learning Techniques.

#### Comparison graph between the VGG16 and the VGG19 algorithm:

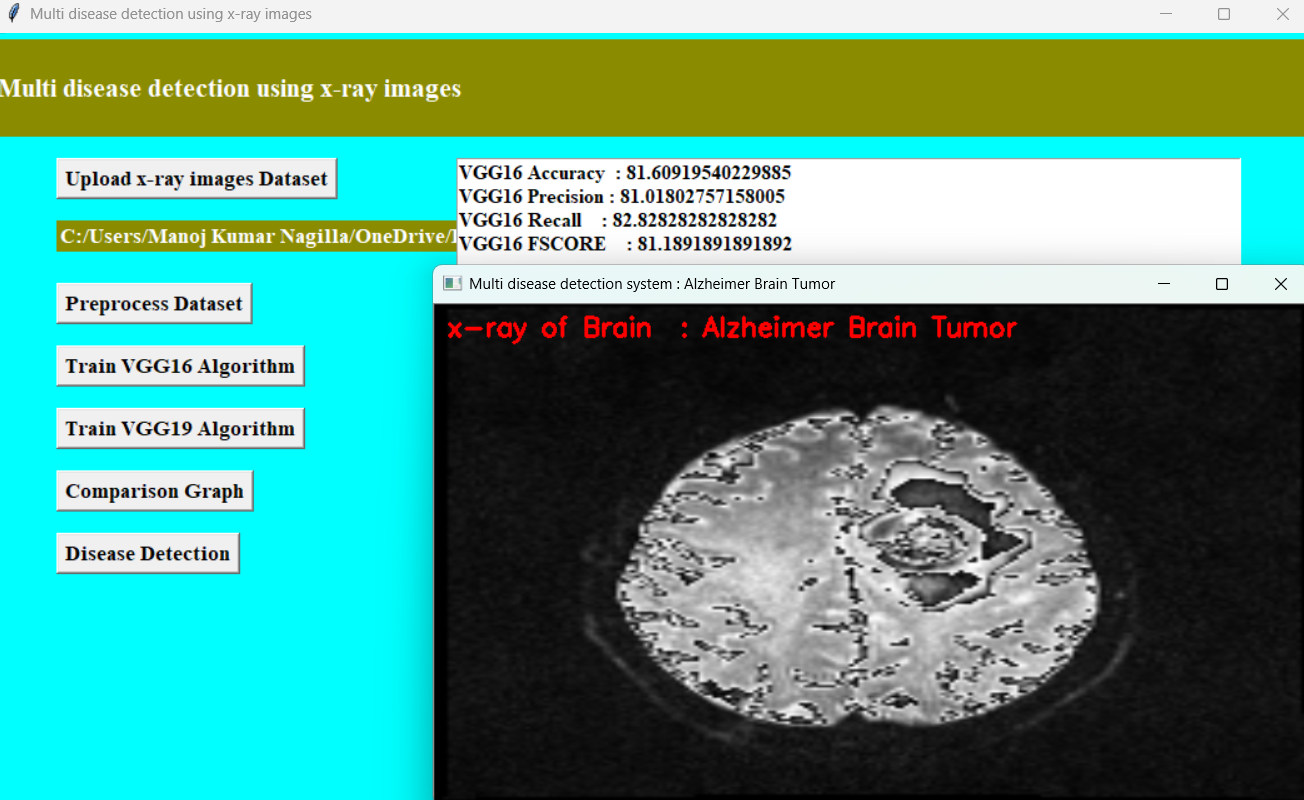
In below graph, x-axis represents algorithm names and y-axis represents accuracy and other metrics in different colour bars. In both algorithms propose EfficientNetB7-BI- LSTM got high accuracy. Now close above graph and then click on ‘Inappropriate Content Prediction from Test Video’ button to upload test video and classify it as Safe or inappropriate.



**Figure 5.6 :** Comparison graph between the VGG16 and the VGG19 algorithm of A Multi-Disease Detection System With X-Ray Images Using Deep Learning Techniques

#### Disease detection :

In below screen, selecting and uploading video and then click on “Open’ button to play video and perform classification.



**Figure 5.7 :** Tumor detected in the given image of A Multi-Disease Detection System With X-Ray Images Using Deep Learning Techniques

**6. VALIDATION**

## VALIDATION

The validation of this project is a robust strategy which is essential to ensure the model’s accuracy and generalization capabilities . It is a comprehensive and well-structured approach which is critical to ensure .The model not only performs well on the training data but also generalizes effectively to unseen X-ray images. Stratified splitting should be used to maintain the balance across various disease classes, especially in cases where the data is imbalanced.

### INTRODUCTION

The validation process begins with an appropriate dataset split. Typically, the dataset is divided into three subsets: 70% for training, 15% for validation, and 15% for testing. This split helps in optimizing the model during training while reserving a portion of the data to assess the model's final performance. Since multi-disease detection often deals with imbalanced datasets—where some diseases are significantly more frequent than others—stratified splitting is recommended to ensure that all disease classes are well-represented across the training, validation, and test sets.

Another essential step in validation is generating multi-label confusion matrices, which provide a class-wise breakdown of true positives, false positives, true negatives, and false negatives. This helps to identify which diseases are more prone to misclassification and to detect any systemic biases in the model's predictions. Beyond numerical validation, visual interpretability is crucial, especially in medical imaging. Techniques like Grad-CAM or Grad-CAM++ can be used to produce visual explanations in the form of heatmaps, highlighting the specific areas in the X-ray images that influenced the model’s predictions. This not only improves model transparency but also allows radiologists or healthcare practitioners to verify if the model is focusing on clinically relevant regions.

Finally, it is vital to document training vs. validation loss and accuracy curves to detect signs of overfitting or underfitting. A holistic validation process such as this not only improves model trustworthiness but also ensures readiness for deployment in clinical settings where decision-making accuracy is paramount.

**6.2 TEST CASES**

**TABLE 6.2.1 UPLOADING DATASET**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case ID | Test case name | Purpose | Test Case | Output |
| 1 | User uploads Dataset. | To test dataset upload and labeling . | The user uploads the Dataset, on which the content is detected. | Dataset successfully loaded. |

**TABLE 6.3.2 CLASSIFICATION**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case ID | Test case name | Purpose | Input | Output |
| 1 | Classification test 1 | To check if the classifier performs its task | X-Ray image is selected | Normal (No disease detected). |
| 2 | Classification test 2 | To check if the classifier performs its task | Inappropriate X-Ray image is selected | Disease Detected (Alzheimer Brain Tumor). |

# 7. CONCLUSION & FUTURE ASPECTS

## CONCLUSION & FUTURE ASPECTS

In conclusion, the project has successfully achieved its objectives, showcasing significant progress and outcomes. Multi-disease detection using deep learning algorithms represents a transformative paradigm shift in medical diagnostics, offering unparalleled capabilities for early disease detection, personalized medicine, and improved patient outcomes. Moreover, the interpretability of deep learning models enables clinicians to understand the rationale behind diagnostic decisions, fostering trust and collaboration between AI systems and healthcare practitioners. By harnessing the power of AI and big data analytics, these innovative approaches have the potential to revolutionize healthcare delivery, accelerate medical discoveries, and ultimately improve the quality of life for patients worldwide.

### PROJECT CONCLUSION

This research presents the integration of deep learning algorithms into multi-disease detection frameworks holds tremendous promise for revolutionizing medical diagnostics across a spectrum of conditions including Alzheimer's disease, brain tumors, COVID-19, and pneumonia. The advancements in deep learning techniques have enabled the development of highly accurate, efficient, and scalable diagnostic tools capable of analyzing complex medical imaging data with unprecedented precision. These innovative approaches facilitate early disease detection, personalized treatment planning, and improved patient outcomes, thereby enhancing the quality of healthcare delivery and reducing healthcare burdens.

Despite the remarkable progress achieved in multi-disease detection using deep learning algorithms, several challenges and opportunities for future research remain. Addressing issues such as data scarcity, model interpretability, computational resource constraints, and ethical dilemmas requires interdisciplinary collaboration and ongoing innovation in AI, healthcare, and regulatory domains. Additionally, efforts to enhance model robustness, generalizability, and real-world deployment are essential to ensure the widespread adoption and clinical impact of AI-driven diagnostic tools in diverse healthcare settings.

### FUTURE ASPECTS

One promising direction for future research involves the development of multimodal fusion techniques to leverage complementary information from diverse data sources. Integrating data from multiple imaging modalities, such as MRI, PET, CT, and molecular imaging, can provide a more comprehensive characterization of disease states and improve diagnostic accuracy.

The future of multi-disease detection using deep learning algorithms entails advancing multimodal fusion techniques, enhancing interpretability and transparency, improving model robustness and generalizability, and upholding ethical and regulatory standards. By addressing these challenges and opportunities, future research endeavors have the potential to accelerate the translation of AI-driven innovations into routine clinical practice, ultimately improving healthcare outcomes and enhancing the quality of life for patients worldwide.

# BIBLIOGRAPHY

## BIBLIOGRAPHY

### REFERENCES

1. Alzheimer's Association. (2020). 2020 Alzheimer's disease facts and figures.
2. Suk, H. I., Shen, D., & Alzheimer's Disease Neuroimaging Initiative. (2017). Deep learning-based feature representation for AD/MCI classification.
3. Liu, M., Cheng, D., & Yan, W. (2018). Alzheimer's Disease Diagnosis Based on Multiple Cluster Dense Deep Learning Neural Networks.
4. Litjens, G., Kooi, T., & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis.
5. Bakas, S., Reyes, M., & Jakab, A. (2018). Identifying the Best Machine Learning Algorithms for Brain Tumor Segmentation, Progression Assessment, and Overall Survival Prediction in the BRATS Challenge.
6. Prasoon, A., Petersen, K., & Igel, C. (2013). Deep feature learning for knee cartilage segmentation using a triplanar convolutional neural network.
7. Wynants, L., Van Calster, B., Collins, G. S., & Riley, R. D. (2020). Prediction models for diagnosis and prognosis of covid-19: systematic review and critical appraisal.
8. Apostolopoulos, I. D., & Mpesiana, T. A. (2020). Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural network
9. Li, L., Qin, L., & Xu, Z. (2020). Using artificial intelligence to detect COVID-19 and community-acquired pneumonia based on pulmonary CT: evaluation of the diagnostic accuracy.
10. Rajpurkar, P., Irvin, J., & Zhu, K. (2017). CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning.
11. Islam, M. M., Karray, F., & Alhajj, R. (2020). Zareapoor, M., & Shahbahrami, A. (2020). Detection of Pneumonia in Chest X-Ray Images Using a Novel CNN-Based Approach.
12. Li, X., Chen, H., & Liu, M. (2020). Retrospective study on the value of chest CT in diagnosis of COVID-19.

### GITHUB LINK

<https://github.com/amrutha1405/Multi-disease-detection-system-with-X-Ray-Images-using-deep-learning-techniques>