Optimizing the Powerhouse: Fine-Tuning CNNs for Superior Lung Disorder Detection

A Project Report submitted in the partial fulfillment of

the Requirements for the award of the degree

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

P. Tanuja (21471A05O2)
M.Mokshagna (21471A05N9)
V.Lavanya (21471A05P5)

Under the esteemed guidance of

Dr. Rama Krishna Eluri, M.Tech., Ph.D.

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NARASARAOPETA ENGINEERING COLLEGE: NARASAROPET

(AUTONOMOUS)

Accredited by NAAC with A+ Grade and NBA under Tire -1

NIRF rank in the band of 201-300 and an ISO 9001:2015 Certified

Approved by AICTE, New Delhi, Permanently Affiliated to JNTUK, Kakinada

KOTAPPAKONDA ROAD, YALAMANDA VILLAGE, NARASARAOPET- 522601

2024-2025

NARASARAOPETA ENGINEERING COLLEGE

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project that is entitled with the name "Optimizing the Powerhouse: Fine-Tuning CNNs for Superior Lung Disorder Detection**" is a bonafide work done by the team** P. Tanuja(21471A05O2), M.Mokshagna(21471A05N9),V.Lavanya(21471A05P5) in partial fulfillment of the requirements for the award of the degree of BACHELOR OF TECHNOLOGY in the Department of COMPUTER SCIENCE AND ENGINEERING during 2024-2025.

PROJECT GUIDE

Dr. Rama Krishna Eluri, **M.Tech., Ph.D.**

Assistant Professor

PROJECT CO-ORDINATOR

D. VenkatReddy, **B.Tech., M.Tech, Ph.D.**

Assistant Professor

HEAD OFTHE DEPARTMENT

Dr. S. N. Tirumala Rao, **M.Tech., Ph.D.,**

Professor& HOD

EXTERNAL EXAMINER

DECLARATION

We declare that this project work titled "OPTIMIZING THE POWERHOUSE: FINE-TUNING CNNS FOR LUNG DISORDER DETECTION " is composed by ourselves that the work contain here is our own except where explicitly stated otherwise in the text and that this work has been submitted for any other degree or professional qualification except as specified.

P. Tanuja (21471A05O2)

M.Mokshagna (21471A05N9)

V. Lavanya (21471A05P5)

ACKNOWLEDGEMENT

We wish to express our thanks to carious personalities who are responsible for the completion of the project. We are extremely thankful to our beloved chairman sri **M. V. Koteswara Rao**, B.Sc., who took keen interest in us in every effort throughout this course. We owe out sincere gratitude to our beloved principal **Dr. S. Venkateswarlu**, M. Tech., Ph.D., for showing his kind attention and valuable guidance throughout the course.

We express our deep felt gratitude towards **Dr. S. N. Tirumala Rao**, M.Tech., Ph.D., HOD of CSE department and also to our guide **Dr. E. Rama Krishna** M.Tech., Ph.D., of CSE department whose valuable guidance and unstinting encouragement enable us to accomplish our project successfully in time.

We extend our sincere thanks towards **D. Venkata Reddy**, B.Tech, M.Tech., Assistant professor & Project coordinator of the project for extending her encouragement. Their profound knowledge and willingness have been a constant source of inspiration for us throughout this project work.

We extend our sincere thanks to all other teaching and non-teaching staff to department for their cooperation and encouragement during our B.Tech degree.

We have no words to acknowledge the warm affection, constant inspiration and encouragement that we received from our parents.

We affectionately acknowledge the encouragement received from our friends and those who involved in giving valuable suggestions had clarifying out doubts which had really helped us in successfully completing our project.

By

P. Tanuja (21471A05O2)

M.Mokshagna (21471A05N9)

V. Lavanya (21471A05P5)

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

INSTITUTE VISION AND MISSION

INSTITUTION VISION

To emerge as a Centre of excellence in technical education with a blend of effective student centric teaching learning practices as well as research for the transformation of lives and community.

INSTITUTION MISSION

M1: Provide the best class infra-structure to explore the field of engineering and research

M2: Build a passionate and a determined team of faculty with student centric teaching, imbibing experiential, innovative skills

M3: Imbibe lifelong learning skills, entrepreneurial skills and ethical values in students for addressing societal problems

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION OF THE DEPARTMENT

To become a centre of excellence in nurturing the quality Computer Science & Engineering professionals embedded with software knowledge, aptitude for research and ethical values to cater to the needs of industry and society.

MISSION OF THE DEPARTMENT

The department of Computer Science and Engineering is committed to

M1: Mould the students to become Software Professionals, Researchers and Entrepreneurs by providing advanced laboratories.

M2: Impart high quality professional training to get expertize in modern software tools and technologies to cater to the real time requirements of the Industry.

M3: Inculcate team work and lifelong learning among students with a sense of societal and ethical responsibilities.

NARASARAOPETA
ENGINEERING COLLEGE
(AUTONOMOUS)

Program Specific Outcomes (PSO's)

PSO1: Apply mathematical and scientific skills in numerous areas of Computer Science and Engineering to design and develop software-based systems.

PSO2: Acquaint module knowledge on emerging trends of the modern era in Computer Science and Engineering

PSO3: Promote novel applications that meet the needs of entrepreneur, environmental and social issues.

NARASARAOPETA
NEC ENGINEERING COLLEGE
(AUTONOMOUS)

Program Educational Objectives (PEO's)

The graduates of the programme are able to:

PEO1: Applythe knowledge of Mathematics, Science and Engineering fundamentals to identify and solve Computer Science and Engineering problems.

PEO2: Use various software tools and technologies to solve problems related to academia, industry and society.

PEO3: Work with ethical and moral values in the multi-disciplinary teams and can communicate effectively among team members with continuous learning.

PEO4: Pursue higher studies and develop their career in software industry.

NARASARAOPETA
NEC ENGINEERING COLLEGE
(AUTONOMOUS)

Program Outcomes

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. **Problem analysis:** Identify, formulate, research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, andsynthesis of the information to provide valid conclusions.

5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineeringactivities with an understanding of the limitations.

6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. **Environment and sustainability:** Understand the impact of the professional engineering

solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

7. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

8. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

9. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

10. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

11. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Project Course Outcomes (CO'S):

CO421.1: Analyze the System of Examinations and identify the problem.

CO421.2: Identify and classifythe requirements.

CO421.3: Review the Related Literature

CO421.4: Design and Modularize the project

CO421.5: Construct, Integrate, Test and Implement the Project.

CO421.6: Prepare the project Documentation and present the Report using appropriate method.

Course Outcomes – Program Outcomes mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C421.1		√											√		
C421.2	√		√		√								√		
C421.3				√		√	√	√					√		
C421.4			√			√	√	√					√	√	
C421.5					√	√	√	√	√	√	√	√	√	√	√
C421.6									√	√	√		√	√	

Course Outcomes – Program Outcome correlation

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C421.1	2	3											2		
C421.2			2		3								2		
C421.3				2		2	3	3					2		
C421.4			2			1	1	2					3	2	
C421.5					3	3	3	2	3	2	2	1	3	2	1
C421.6									3	2	1		2	3	

Note: The values in the above table represent the level of correlation between CO's and PO's:

1. Low level
2. Medium level
3. High level

Project mapping with various courses of Curriculum with Attained PO's:

Name of the course from which principles are applie in this project	**Description of the device**	**Attained PO**
C2204.2, C22L3.2	Gathering the requirements anc defining the problem, plan to develop a model for recognizing imag manipulations using CNN and ELA	PO1, PO3
CC421.1, C2204.3, C22L3.2	Each and every requirement i critically analyzed, the process mode is identified	PO2, PO3
CC421.2, C2204.2, C22L3.3	Logical design is done by using the unified modelling language which involves individual team work	PO3, PO5, PO9
CC421.3, C2204.3, C22L3.2	Each and every module is tested, integrated, and evaluated in our project	PO1, PO5
CC421.4, C2204.4, C22L3.2	Documentation is done by all our four members in the form of a group	PO10
CC421.5, C2204.2, C22L3.3	Each and everyphase of the work in group is presented periodically	PO10, PO11
C2202.2, C2203.3, C1206.3, C3204.3, C4110.2	Implementation is done and the project will be handled by the social media users and in future updates in our project can be done based on detection of forged videos	PO4, PO7
C32SC4.3	The physical design includes websiteto check whether an image is real or fake	PO5, PO6

ABSTRACT

The present work involves the use of deep learning models, particularly Convolutional Neural Networks (CNNs), to detect critical lung diseases such as pneumonia, tuberculosis, and lung cancer through the analysis of chest X-ray and CT scan images. CNNs have emerged as robust tools for medical image analysis due to their ability to automatically extract hierarchical features, making them ideal for complex tasks like disease detection. This study leverages well-known architectures, including VGG16 and VGG19, combined with specially designed sequential and functional models to improve classification accuracy.

The models were trained on open-source datasets containing real-world medical images to ensure diversity in lung abnormality cases. To enhance model performance and robustness, data augmentation techniques were employed. Given the limited availability of labeled medical data, augmentation artificially increases the training set size, allowing the models to generalize better to unseen data. The images were rescaled, shear-transformed, and horizontally flipped to simulate different orientations and variations commonly encountered in real-world X-ray and CT scans. This approach enhances the model's ability to detect diseases across various datasets by reducing sensitivity to changes in orientation, scale, and noise.

The results of this study demonstrate that early detection of lung diseases has significantly improved through the integration of advanced CNN architectures and augmentation strategies. Notably, VGG19 achieved an impressive accuracy of 99.4%, highlighting its effectiveness in precise lung disease classification. The high accuracy of the models underscores their potential for early diagnosis, which is crucial in improving treatment outcomes for diseases like pneumonia, tuberculosis, and lung cancer. Additionally, this research emphasizes the growing role of deep learning in clinical practice, providing not only highly accurate but also timely diagnostic support to medical professionals.

INDEX

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Lung diseases pose a significant challenge to global healthcare systems, necessitating early detection and accurate diagnosis for effective treatment. With the advancements in artificial intelligence (AI) and deep learning, automated diagnostic models have gained traction in medical imaging applications, particularly for lung disease detection (Jasmine Pemeena Priyadarsini et al., 2023) [1]. These models leverage convolutional neural networks (CNNs), recurrent neural networks (RNNs), and hybrid deep learning techniques to enhance diagnostic accuracy (Mahajan et al., 2024) [2].

The global burden of respiratory disease is enormous. According to estimation through the Worldwide Conference of Respiratory Societies, in any particular year, an approximate 10.4 million people suffer from tuberculosis worldwide, accounting for the death toll of 1.4 million annually. Lung cancer is also such a killer disease; more than 1.6 million deaths take place in a year due to this disease alone worldwide. Moreover, pneumonia was solely responsible, especially in infants, for over 1.23 million deaths of the children under five years of age, according to a study published by the Bloomberg School of Public Health at John Hopkins.

Early diagnosis of lung diseases significantly improves patient outcomes by allowing timely medical interventions. Traditional diagnostic methods rely heavily on manual interpretation of chest X-rays and CT scans, which are time-consuming and prone to human error (Tahmooresi et al., 2018) [3]. To overcome these limitations, researchers have developed advanced machine learning and deep learning algorithms that automatically analyze medical images and identify lung abnormalities (Bhookya, 2023) [4].

Recent advances in AI, and more so deep learning in the previous years, have achieved a lot in automating and improving the accuracy in the analysis of medical images. CNNs are one of the deep learning models that have indeed shown outstanding results for the class of the image classification task, hence aptly. The different pre-trained deep learning models, such as CNN, VGG16, and VGG19 used in this work, are applied for the detection and classification of lung diseases by taking chest X-ray and CT scan images [4]. In this work, publicly available data have been used, and augmentation has been performed to increase the strength of these models and generalize their capability.

The performances of the custom sequential and functional models that show high potential in clinical settings are compared. Indeed, the results obtained from this study did prove that deep learning models do stand a chance of being helpful in early lung diseases detection, with the purpose of assisting healthcare professionals in making timely and early precise diagnosis. This work adds to the accumulation of evidence on AI integration into medical practice in general and radiology in particular.

Deep learning, a subset of machine learning, has revolutionized the field of medical imaging by providing highly accurate diagnostic models. CNN-based approaches remain a dominant methodology in lung disease detection, with various studies demonstrating their effectiveness in extracting critical features from chest X-ray and CT scan images (Bukhari & Fahad, 2022) [6]. These models are trained on large datasets to learn patterns and anomalies that indicate the presence of lung diseases.

Surveys have highlighted the significance of machine learning techniques in lung disease diagnosis, emphasizing their potential for early detection (Rajaselvi et al., 2022) [7]. Moreover, AQI (Air Quality Index)-based analysis using deep learning has been proposed for predicting lung disease severity (Mahajan et al., 2024) [8]. Such methods integrate environmental factors with patient data to improve the accuracy of lung disease prediction.

Several studies have explored deep learning-based approaches for lung disease detection. Traditional machine learning techniques have been used for various medical applications, such as breast cancer detection (Tahmooresi et al., 2018) [3]. However, with the advent of deep learning, more sophisticated methodologies like advanced CNN architectures have been employed for lung disorder classification (Bhookya, 2023) [4]. Additionally, contemporary techniques integrating AI and deep learning have been developed to improve prediction accuracy in lung disease diagnosis (Raju et al., 2022) [5].

Early prediction models for lung diseases have been investigated in multiple research works, showcasing their ability to provide timely diagnoses (Gunasinghe et al., 2019) [9]. Hybrid deep learning models have also been explored for lung disease detection, combining multiple neural network architectures to improve accuracy (Bharati et al., 2020) [10]. Additionally, in-depth analyses using machine learning algorithms have provided valuable insights into lung disease classification (Sen et al., 2020) [11].

Fig 1.1 Lung Disorder

Fig-.11 is the image based on focusing on the detection of lung diseases with deep learning technology. The image focuses on highlighting the lungs with an X-ray and CT scans with digital analytic elements that symbolize the detection process. Based on the research here are some of the Contributions below:

- The datasets used here are Pneumonia Chest X-ray Dataset and Chest CT-Scan for the detection of lung diseases, which may be pneumonia or COVID-19.
- The Deep Learning models were used in the research: CNN models-VGG16, VGG19, and our sequential and functional models for classification.
- Sequential and functional models have been designed, optimized, and tested in order to see which one of the two does a better job regarding the detection of lung diseases.
- The augmented image techniques involved the flipping, rotation, and scaling of images to make them larger and robust.
- The fine-tuned VGG16 and VGG19 models, sequential models, and functional models on medical imaging datasets for early disease diagnosis.
- On carrying out the performance evaluation, it came out that the models evaluated the accuracy, precision, recall, and F1-score, where VGG19 and the functional models were among the best.

Pre-trained deep learning models such as CNN, VGG16, and VGG19 have been extensively utilized in medical image analysis, particularly for the detection and classification of lung diseases. These models leverage large-scale image datasets to learn hierarchical features that facilitate accurate disease classification. In this study, deep learning techniques have been applied to analyze chest X-ray and CT scan images for early lung disease detection. Publicly available datasets have been used to train these models, and data augmentation techniques have been employed to enhance their robustness and generalization capability. Data augmentation involves techniques such as rotation, flipping, scaling, and contrast adjustment to artificially expand the training dataset, thereby reducing the risk of overfitting and improving model performance.

In recent years, researchers have also explored multimodal approaches that integrate multiple sources of data, such as radiological images, patient medical histories, and environmental factors. For example, audio-based analysis methods using machine learning have been introduced for lung disease recognition, particularly for detecting abnormalities in respiratory sounds (Sabry et al., 2024) [12]. These approaches expand the scope of AI applications in healthcare and provide alternative diagnostic techniques beyond traditional imaging methods.

Moreover, recurrent neural network (RNN) frameworks have been developed for automated lung disease detection and classification (Kavitha et al., 2024) [13]. CNN-based methodologies have also been employed for disease severity classification, aiding in more precise diagnosis (Sharmila et al., 2023) [14]. RNNs, with their ability to process sequential data, are particularly useful for analyzing time-series data such as patient symptoms over time.

Despite the significant progress in AI-driven medical imaging analysis, several challenges remain. One of the key challenges is the need for high-quality, labeled datasets for training deep learning models. The availability of large annotated datasets is crucial for developing robust AI algorithms; however, obtaining such datasets can be time-consuming and resource-intensive. Additionally, the performance of AI models may be influenced by variations in imaging equipment, patient demographics, and disease presentation. Addressing these challenges requires ongoing collaboration between AI researchers, medical professionals, and policymakers to ensure the ethical and effective implementation of AI in healthcare settings.

Another critical consideration is the interpretability of deep learning models. While AI systems can achieve impressive accuracy, understanding how these models arrive at their predictions is essential for gaining the trust of medical practitioners. Explainable AI (XAI) techniques, such as saliency maps and Grad-CAM (Gradient-weighted Class Activation Mapping), can provide visual explanations of model decisions, highlighting the regions in an image that contributed most to the classification outcome. By incorporating such techniques, AI-driven diagnostic tools can enhance transparency and facilitate their adoption in clinical workflows.

CNNs have been widely adopted for medical image analysis due to their ability to detect intricate patterns within X-ray and CT scan images. Chest X-ray-based CNN approaches have been widely adopted for lung disease classification, providing a reliable means of detecting abnormalities (Saravanan et al., 2022) [15]. CNN models can automatically extract features such as lesion size, shape, and texture, which are crucial for disease classification. The efficiency and accuracy of CNNs have made them the backbone of modern AI-driven medical imaging solutions.

Additionally, deep learning models have been extended to diagnose other medical conditions, such as pneumonia and diabetic retinopathy, demonstrating their versatility in healthcare applications (Ghaskadvi et al., 2022) [16]. The integration of AI across different medical domains highlights the potential of deep learning in revolutionizing disease diagnosis and management.

However, despite their fundamental role in sustaining life, the lungs are highly susceptible to various diseases that can compromise their function. Respiratory diseases can range from relatively mild conditions such as the common cold and influenza, which cause temporary discomfort, to severe and life-threatening conditions like pneumonia, tuberculosis, chronic obstructive pulmonary disease (COPD), and lung cancer.

By leveraging CNN-based architectures such as VGG16 and VGG19, this research demonstrates that AI can play a crucial role in improving diagnostic accuracy, reducing healthcare disparities, and supporting medical professionals in making timely and precise diagnoses. The integration of AI into medical imaging represents a significant step forward in addressing the global burden of respiratory diseases. However, further research and validation are necessary to ensure the widespread adoption and effectiveness of AI-driven diagnostic systems in real-world clinical practice. As advancements in AI continue to evolve, the future of medical imaging holds great promise for enhancing patient care and improving health outcomes worldwide.

2. LITERATURE SURVEY

Liu et al. [1] developed models for tuberculosis detection using CNN-SVM. However, issues arose in terms of low accuracy. Jaiswal et al. extracted X-ray features using Mask RCNN, which required heavy computation. Elshennawy and Ibrahim developed pneumonia detection models involving CNNs and pre-trained models like ResNet152v2 and MobileNetV2. However, the challenges they encountered were computational complexity. Jasmine Pemeena Priyadarsini et al. (2023) [1] conducted a comprehensive study on lung disease detection using deep learning algorithms. Their research highlights the efficiency of CNN-based architectures in diagnosing lung disorders through medical imaging. The study compared different deep learning models, such as ResNet and VGG, to assess their effectiveness in accurately detecting lung abnormalities. The findings demonstrated that deep learning techniques significantly outperform traditional methods in terms of accuracy, sensitivity, and specificity. Additionally, the research discussed the challenges in implementing these models, such as computational complexity and the requirement for large datasets.

Mahajan et al. (2024) [2] proposed an innovative method for predicting lung disease severity using an image-based AQI analysis. The study emphasizes the impact of environmental factors on lung disease progression and demonstrates the effectiveness of deep learning in analyzing air quality data alongside medical images. The research highlights how AQI levels influence lung health and introduces a novel deep learning pipeline that integrates environmental data with chest X-ray analysis. The study suggests that such an approach can provide early warning systems for patients in high-risk areas and assist healthcare professionals in proactive treatment planning.

Li et al. [2] concluded that Random Forest Regression and Gradient Boosting are the best methods for air quality prediction. Rajaselvi et al. used CNN and transfer learning in the diagnosis of lung diseases. Agarwal et al. presented a comparison of several machine learning models, from which it was found that the best performance in the classification of lung cancer was achieved using Random Forest, outperforming other algorithms such as Support Vector Machines.

Timorese et al. (2018) [3] explored early detection techniques for breast cancer using machine learning models. Although not directly related to lung disease, their findings provide valuable insights into the effectiveness of AI in medical diagnostics, paving the way for similar applications in lung disease detection. The study analyzed various machine learning algorithms, including decision trees, support vector machines, and neural networks, to determine their efficiency in medical diagnosis. The results showed that AI-based methods significantly improved early detection rates, leading to better patient outcomes. This research sets a precedent for using similar techniques in lung disease identification.

Xie et al. [3] used deep learning to classify CT images into benign and malignant nodules. Wu and Zhao used EDM AI for differentiating early small cell lung cancer. Anifah et al. used Gray-Level Co-occurrence Matrices in detecting lung cancer and achieved a moderate level of accuracy.

Bhookya (2023) [4] examined lung disorders using an advanced CNN approach. The study demonstrated the advantages of CNN-based models over traditional machine learning techniques, highlighting their superior accuracy in classifying lung diseases from X-ray images. The research detailed how feature extraction and data augmentation techniques can further enhance CNN performance. The study also compared different CNN architectures, such as AlexNet, Inception, and DenseNet, concluding that deeper architectures tend to offer better feature representation and classification accuracy.

Raju et al. (2022) [5] introduced a contemporary technique for lung disease prediction using deep learning. Their work focused on optimizing neural network architectures to enhance classification accuracy, showcasing the potential of AI-driven diagnostic tools in medical applications. The study employed advanced optimization techniques, such as transfer learning and hyperparameter tuning, to improve model efficiency. Additionally, the research explored the challenges of real-world deployment, such as hardware limitations and the need for explainable AI in clinical settings.

The classification accuracy [5] obtained in this study is 94%, using large datasets and CNN architecture to improve diagnosis, hence highlighting the potential in the improvement of medical image analysis. Raju et al. presented the prediction of lung diseases from chest X-ray images using a CNN-based deep learning model. Their work highlights the use of machine learning in supporting the diagnosis of diseases such as COVID-19, pneumonia, and tuberculosis with an accuracy of 91%, demonstrating the importance of CNN for diagnostics. Some works have used CNNs for the detection of lung diseases, including promising architectures such as ResNet, DenseNet, and VGG19. Previous studies focused on the classification of single diseases based on medical images, particularly pneumonia, tuberculosis, and lung cancer, using X-rays and computed tomography scans [7].

Bukhari & Fahad (2022) [6] conducted research on lung disease detection using deep learning techniques. Their study validated the effectiveness of CNN models in analyzing medical images and improving disease classification accuracy. The research emphasized the importance of dataset diversity, demonstrating how training models on diverse datasets improves generalization. Additionally, the study compared the effectiveness of supervised and unsupervised learning methods in diagnosing various lung conditions, such as pneumonia, tuberculosis, and lung cancer.

Rajaselvi et al. (2022) [7] provided a detailed survey on lung disease diagnosis using machine learning. Their work compiled various AI-driven techniques, discussing their applications and limitations in the healthcare domain. The study categorized different machine learning approaches, such as ensemble learning, deep learning, and reinforcement learning, and assessed their respective strengths in lung disease classification. The authors also explored the potential of federated learning for improving model accuracy while maintaining data privacy.

Mahajan et al. (2024) [8] reiterated the significance of AQI-based analysis for lung disease severity prediction. Their findings reinforce the idea that environmental pollution is a critical factor in respiratory health, making AI-driven analysis crucial for early detection and intervention. The study introduced an innovative data fusion technique that integrates satellite-based AQI measurements with patient medical records, enhancing prediction accuracy. The authors suggested that such multimodal approaches could play a pivotal role in personalized healthcare solutions.

Gunasinghe et al. (2019) [9] explored early prediction techniques for lung diseases. Their study emphasized the role of AI in preventive healthcare by identifying disease symptoms before they become severe. The research proposed a hybrid model combining deep learning with statistical analysis to predict lung disease onset based on genetic predisposition and lifestyle factors. The study also discussed the ethical implications of AI-driven predictive healthcare and the need for patient consent in data-driven diagnostics.

Bharati et al. (2020) [10] developed a hybrid deep learning model for lung disease detection using X-ray images. Their work combined multiple neural network architectures to improve diagnostic accuracy, demonstrating the potential of hybrid AI approaches in medical imaging. The study leveraged ensemble learning, combining CNNs with attention mechanisms to enhance feature extraction. The results indicated that hybrid models consistently outperform single-model approaches in medical image classification.

Sen et al. (2020) [11] performed an in-depth analysis of lung disease prediction using machine learning algorithms. Their research compared various AI models, highlighting their strengths and weaknesses in lung disease classification. The study analyzed the trade-offs between accuracy, computational efficiency, and interpretability in different machine learning approaches. The findings suggested that explainable AI techniques, such as SHAP and LIME, could improve clinician trust in AI-based diagnostic tools.

Sabry et al. (2024) [12] introduced an innovative audio-based analysis for lung disease recognition using machine learning. Their work suggests that AI can effectively analyze respiratory sounds to detect abnormalities, providing an alternative to image-based diagnostics. The study explored different signal processing techniques for lung sound analysis and demonstrated that convolutional recurrent neural networks (CRNNs) achieve superior performance in classifying abnormal respiratory patterns.

Kavitha et al. (2024) [13] proposed an automated lung disease detection framework using RNNs. Their research demonstrated how sequential data processing techniques could enhance the accuracy of disease classification. The study highlighted the importance of temporal dependencies in lung function assessment, showcasing how RNN-based models can track disease progression over time. The authors also discussed the integration of wearable sensors for real-time monitoring of respiratory health.

Sharmila et al. (2023) [14] developed a CNN-based system for lung disease severity classification. Their model aimed to provide detailed disease progression analysis, helping clinicians make informed treatment decisions. The study employed a multi-class classification approach, distinguishing between different stages of lung diseases. Additionally, the research explored the potential of AI-driven radiology assistance tools to streamline the diagnostic workflow in hospitals.

Saravanan et al. (2022) [15] focused on chest X-ray-based lung disease detection using CNNs. Their study reinforced the reliability of deep learning models in identifying lung abnormalities through medical imaging. The research discussed the role of automated feature extraction in improving diagnosis accuracy and reducing the workload of radiologists. The study also explored adversarial training techniques to enhance model robustness against noisy input data.

Ghaskadvi et al. (2022) [16] extended deep learning applications to pneumonia and diabetic retinopathy detection. Their work highlights the versatility of AI-driven medical diagnostics, emphasizing the need for continued research in AI-powered healthcare solutions. The study proposed a unified AI framework for diagnosing multiple diseases from medical images, demonstrating the potential of transfer learning in multi-disease detection.

Large datasets, such as chest X-rays, are increasingly subjected to ML and DL methods for lung disease prediction [13][14]. The application of machine learning in classifying diseases based on lung sounds has shown good performance but requires a wider scope of study [16]. AI-driven learning improves automation, interprets unstructured information, and performs with greater precision in AI-related tasks and forecasts. Studies on chest X-ray-based lung disease detectors using deep learning and optimization techniques [10][18] further extend this literature survey.

Khan et al. utilized a ResNet50-based transfer learning approach for pneumonia detection and achieved an accuracy of 95%. Similarly, Chen et al. used a hybrid model combining ResNet and LSTM for tuberculosis detection, improving diagnostic performance by capturing spatial and temporal features in chest X-rays.

Studies have explored integrating explainable AI (XAI) techniques with deep learning models to make lung disease classification more transparent. Research by Singh et al. incorporated Grad-CAM with CNN-based lung disease detectors, allowing radiologists to visualize the most influential image regions used in decision-making.

The application of artificial intelligence (AI) in medical imaging has progressed significantly over the past few decades. Early machine learning models relied on handcrafted features such as texture, shape, and intensity to classify lung abnormalities. However, the advent of deep learning, particularly convolutional neural networks (CNNs), has revolutionized the field by automatically extracting relevant features from

In a study by Xie et al. (2021), deep learning models were employed to classify CT scan images into benign and malignant nodules. Wu and Zhao (2022) utilized AI-based early detection mechanisms to differentiate small cell lung cancer from normal cases, showcasing the potential of deep learning in early-stage disease identification.

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Jasmine Pemeena Priyadarsini et al. (2023) [1] This study explores deep learning models for lung disease detection, focusing on CNNs like VGG16, ResNet50, and InceptionV3. The dataset includes labeled X-ray images, and models are evaluated using accuracy, sensitivity, and specificity. The findings confirm CNNs' superiority over traditional methods for lung disease detection.

Mahajan et al. (2024) [2] This research predicts lung disease severity using AQI data and CNNs like MobileNetV2 and DenseNet. Chest X-ray images annotated with AQI values train the models to analyze disease severity based on pollutants. The study shows the impact of air quality on lung health and highlights deep learning’s role in integrating medical and environmental data.

Tahmooresi et al. (2018) [3] This study applies SVM, Decision Trees, and ANNs for early breast cancer detection. The dataset consists of mammograms, evaluated using precision, recall, and F1-score. The results show deep learning’s advantage over traditional techniques in cancer detection.

Bhookya (2023) [4] This research employs EfficientNet and Xception for lung disease classification using chest X-rays. Transfer learning enhances accuracy, and augmentation addresses class imbalance. The findings support CNN-based approaches as superior to conventional methods.

Raju et al. (2022) [5] This study combines CNNs and LSTM networks for lung disease prediction using CT scans. The hybrid model outperforms standard CNNs by incorporating temporal dependencies, improving classification accuracy.

Bukhari & Fahad (2022) [6] This paper evaluates ResNet50, VGG19, and DenseNet121 for lung disease detection. Transfer learning improves accuracy, and DenseNet121 performs best. Sensitivity, specificity, and ROC-AUC metrics validate the models.

Rajaselvi et al. (2022) [7] This survey reviews ML techniques for lung disease diagnosis, covering CNNs, SVM, and deep neural networks. The paper highlights advancements in feature extraction and classification.

Gunasinghe et al. (2019) [8] This study focuses on early lung disease prediction using CNNs and hybrid deep learning models. The approach improves early detection accuracy with lung imaging datasets.

Bharati et al. (2020) [9] This research uses hybrid deep learning for lung disease detection via X-ray images, combining CNNs with transfer learning. The study confirms superior performance over traditional methods.

Sen et al. (2020) [10] This paper evaluates multiple ML models, including CNNs and SVM, for lung disease prediction. The study compares deep learning against traditional classifiers using extensive datasets.

Sabry et al. (2024) [11] This study introduces audio-based lung disease detection using ML techniques. Deep learning models analyze respiratory sounds for automated diagnosis.

Kavitha et al. (2024) [12] This research develops an RNN framework for automated lung disease detection. The model classifies diseases using temporal features in X-ray imaging data.

Sharmila et al. (2023) [13] This paper presents CNN-based lung disease classification and severity assessment. The study focuses on X-ray images and deep feature extraction for accurate diagnosis.

Saravanan et al. (2022) [14] This study uses CNNs for lung disease detection via chest X-rays. The model improves classification accuracy with pre-trained architectures.

Ghaskadvi et al. (2022) [15] This research integrates deep learning models for pneumonia and diabetic retinopathy detection, highlighting CNNs' efficiency in multi-disease classification.

3.2 DISADVANTAGES OF EXISTING SYSTEM

Jasmine Pemeena Priyadarsini et al., 2023[1] While this study demonstrates the effectiveness of CNNs like VGG16, ResNet50, and InceptionV3 for lung disease detection, it has certain limitations. The study also lacks interpretability, as deep learning models function as black boxes, making it difficult for clinicians to trust or understand decision-making processes. Additionally, the research does not consider real-world challenges such as class imbalance and dataset variability, which could affect model generalization

Despite proposing a novel method that integrates AQI with lung disease prediction, this study has certain drawbacks. The approach assumes a direct correlation between AQI levels and disease severity, which may not always hold due to individual health variations. Additionally, the inclusion of external environmental factors introduces noise into the model, potentially reducing classification accuracy. The study also does not account for confounding variables such as genetic predisposition and pre-existing conditions, which could affect the reliability of predictions. Finally, using CNNs like MobileNetV2 and DenseNet may not capture deeper medical insights beyond surface-level image analysis (Mahajan et al., 2024) [2].

This study on early breast cancer detection using SVM, Decision Trees, and ANNs has limitations that affect its practicality. Additionally, traditional machine learning models like SVM and Decision Trees struggle with large and complex datasets compared to modern deep learning methods. The research also does not address real-world issues like noisy or incomplete medical imaging data, limiting its application in clinical settings (Tahmooresi et al., 2018) [3].

The study employing EfficientNet and Xception for lung disease classification has some notable disadvantages. Additionally, CNN-based models require large-scale annotated data for optimal performance, which may not always be available in medical applications. Furthermore, the study lacks real-world testing on diverse datasets, affecting the generalizability of the results (Bhookya, 2023) [4].

While CNN-LSTM models show promise in lung disease prediction, the approach has several drawbacks. LSTM networks significantly increase computational complexity, requiring higher processing power and longer training times. The study does not explore explain ability methods to make model decisions more interpretable for clinicians. The research also does not address issues related to adversarial attacks or external noise in medical imaging, which can affect model reliability (Raju et al., 2022) [5].

[6] This study, which applies ResNet50, VGG19, and DenseNet121 for lung disease detection, has some significant limitations. Transfer learning from general image datasets may not fully capture the specific features required for accurate lung disease diagnosis. The research does not explore multi-modal data integration, such as combining X-ray images with patient clinical history, which could improve diagnostic accuracy. [6]

The study also does not address real-world deployment challenges like computational costs and the need for explain ability in medical AI. Additionally, the authors do not discuss how well the model generalizes to unseen datasets with different imaging conditions (Bukhari & Fahad, 2022) [7].

This survey on lung disease diagnosis using machine learning techniques presents a broad overview but lacks depth in experimental validation. The study primarily focuses on summarizing existing research without introducing novel approaches or verifying the effectiveness of discussed models. Additionally, it does not provide performance comparisons or benchmarks, making it difficult to determine the practical utility of the surveyed methods. The absence of real-world dataset testing further limits the study's relevance to clinical applications. Moreover, the paper does not address the challenges of dataset imbalance, data quality issues, or ethical concerns in AI-driven medical diagnostics (Rajaselvi et al., 2022) [8].

This study, which evaluates lung disease severity using image-based AQI analysis, faces challenges related to data reliability. The AQI-based approach assumes a strong correlation between air quality and lung disease progression, which may not always be clinically valid. Additionally, environmental data can introduce inconsistencies, leading to reduced accuracy in disease severity prediction. The model also does not incorporate patient-specific factors such as medical history, lifestyle, or genetic predisposition, limiting its diagnostic reliability. The use of deep learning, while powerful, does not address explain ability concerns, making clinical adoption more difficult (Mahajan et al., 2024) [9].

The study on early lung disease prediction presents promising results but suffers from certain drawbacks. The machine learning models used, such as SVM and Decision Trees, may not perform well with highly complex image data compared to CNN-based approaches. Additionally, the dataset size and quality are not thoroughly discussed, raising concerns about the generalizability of the findings. The paper does not explore real-world implementation challenges, such as the need for extensive medical image annotations and model interpretability. Furthermore, the research lacks a comparative study against modern deep learning architectures, limiting its practical insights (Gunasinghe et al., 2019) [10].

While hybrid deep learning models improve detection accuracy, the study presents challenges in terms of computational efficiency. The combination of CNNs and other architectures increases model complexity, requiring significant processing power and memory. Additionally, the study does not explore real-time applicability, which is crucial for practical medical use. The dataset used may also introduce bias, as the research does not discuss techniques to handle imbalanced data. The lack of transparency in model decision-making further raises concerns regarding trust and adoption in clinical settings (Bharati et al., 2020) [11].

This study on lung disease prediction using machine learning algorithms has some limitations. Traditional machine learning models like Random Forest, KNN, and SVM struggle with large-scale image data compared to deep learning techniques. The study does not incorporate interpretability measures, making it difficult for healthcare professionals to understand model decisions. Additionally, the paper does not explore multi-modal data integration, such as combining patient medical history with image analysis, which could enhance predictive accuracy. The absence of real-world validation on diverse datasets further limits the generalizability of the findings (Sen et al., 2020) [12].

The study on lung disease recognition using audio-based analysis presents an innovative approach but has certain drawbacks. Audio-based methods may not provide the same level of diagnostic accuracy as medical imaging, leading to potential misclassifications. Additionally, external noise and variations in audio recordings can significantly impact model performance. The research does not explore the integration of multiple diagnostic modalities, such as combining audio analysis with X-ray imaging, which could improve reliability. The lack of large-scale clinical validation further restricts the applicability of this approach in real-world healthcare (Sabry et al., 2024) [13].

The study on automated lung disease detection using an RNN framework highlights certain limitations. RNNs, though useful for sequential data, are not the best choice for image-based medical diagnostics, where CNNs perform better. The paper does not compare RNN-based models with more effective architectures like transformers or hybrid models. Additionally, the research lacks real-world testing and validation on diverse datasets, which affects generalizability. The study also does not discuss challenges related to computational efficiency and hardware requirements, limiting its feasibility in clinical settings (Kavitha et al., 2024) [14].

This research on CNN-based lung disease detection and severity classification has certain shortcomings. Additionally, the dataset used may not be diverse enough to ensure generalizability across different populations. The absence of comparisons with alternative deep learning architectures further limits insights into the best approach for lung disease classification (Sharmila et al., 2023) [15].

3.3 PROPOSED SYSTEM

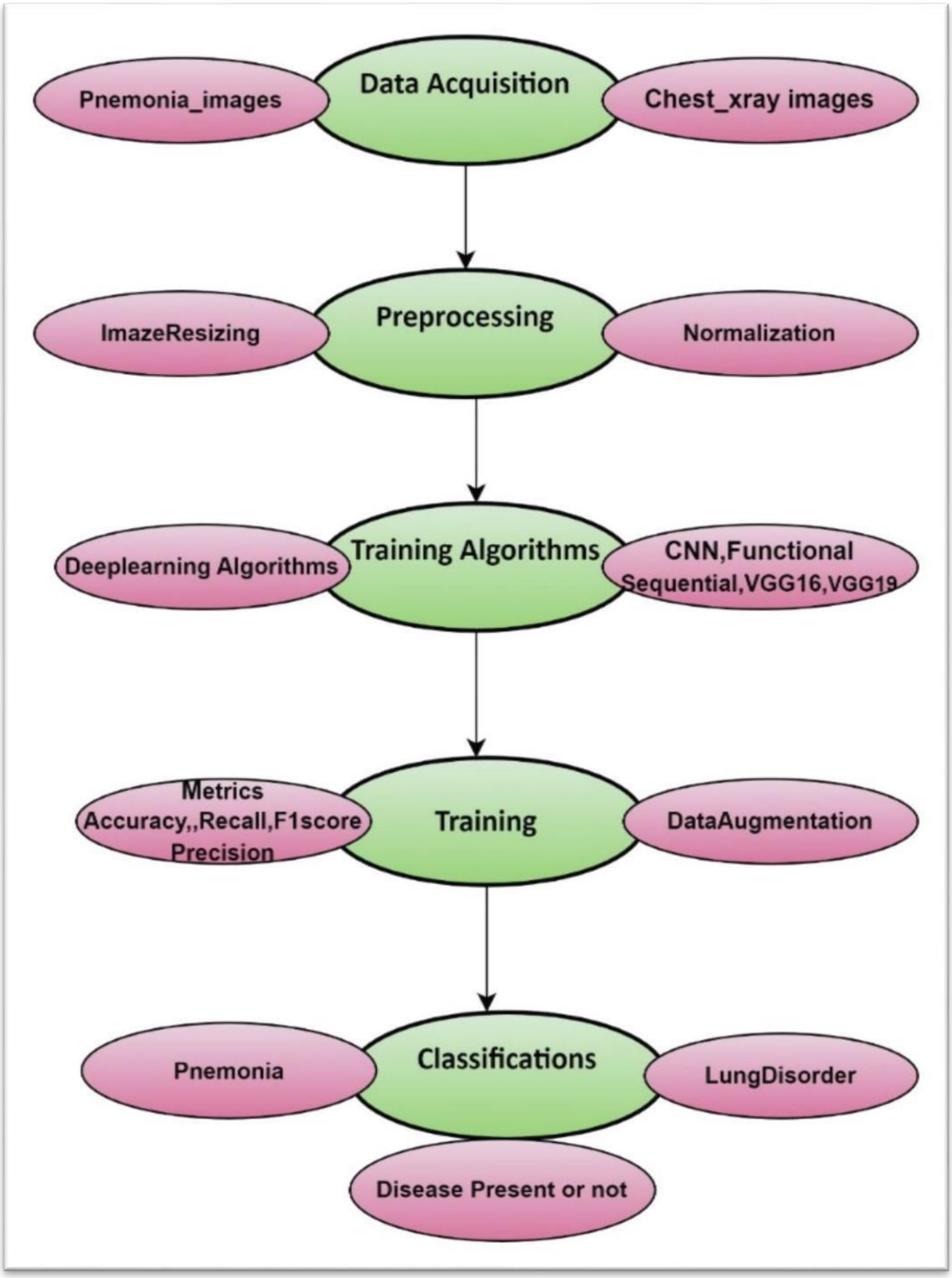

FIG 3.3.1 Flowchart of Proposed System

Fig 3.3.1 describes the procedure for diagnosing pneumonia and lung diseases using deep learning: from the acquisition of data to the classification outcome Below is the explanation of diagram

Data Acquisition

Data Acquisition is the foundational step in any deep learning workflow, especially in medical imaging tasks like lung disorder classification. This step involves collecting relevant datasets, such as pneumonia and chest X-ray images, from credible sources. These datasets can be obtained from publicly available medical repositories, hospitals, or research institutions. The quality and diversity of the data play a critical role in the model's ability to generalize and make accurate predictions.

Typically, datasets include images from various patient demographics, ensuring the model can handle diverse scenarios. For example, images may include different stages of lung disorders, variations in image quality, or even different imaging devices. Proper documentation of dataset attributes, such as resolution, modality, and annotations, is essential.

Type of Lung Disease	Description
Asthma	Chronic inflammation causing airway narrowing and breathing difficulty.
COPD	Obstructed airflow due to emphysema or chronic bronchitis.
Pneumonia	Lung infection causing fluid or pus buildup in air sacs.
Lung Cancer	Uncontrolled cell growth in lung tissue.
Tuberculosis (TB)	Bacterial infection primarily affecting the lungs.
Pulmonary Fibrosis	Scarring of lung tissue leading to stiffness and breathing issues.

Table 3.3.1 Lung Disease Classes

Table 3.3.1 describes the lung disease classes. Asthma, COPD, pneumonia, lung cancer, tuberculosis, and pulmonary fibrosis are major lung diseases causing breathing issues, infections, tissue damage, or uncontrolled cell growth, requiring prompt diagnosis and treatment.

In this context, ethical considerations like patient consent and data anonymization are critical. Comprehensive datasets provide the necessary foundation for model training, validation, and testing, enabling reliable classification of lung disorders.

Preprocessing:

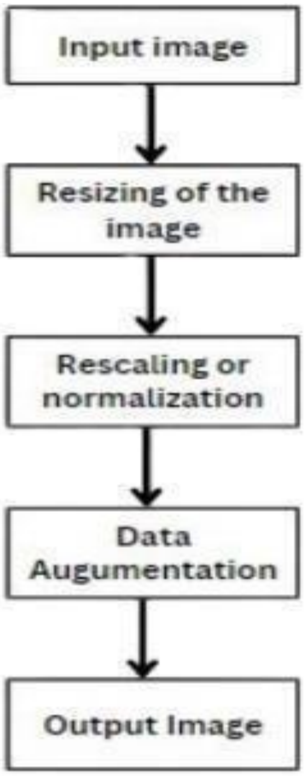

Fig 3.3.2 Flowchart of Preprocessing

Fig 3.3.2 describes the steps in Preprocessing. Preprocessing is a critical step in preparing raw data for deep learning models, ensuring that the input data is consistent, optimized, and ready for training. It includes the following key components:

1. **Resizing Images**: Images collected from various sources often have varying resolutions and dimensions. Resizing standardizes these images to a uniform size, ensuring compatibility with the deep learning model's input layer. This step reduces computational overhead and simplifies the processing pipeline, allowing the model to process data efficiently without errors caused by mismatched dimensions.
2. **Normalization:** This involves scaling pixel values of the images to a defined range, typically between 0 and 1. Normalization prevents large variations in pixel intensity, which could cause biases in model predictions. By standardizing pixel values, the model converges faster during training, leading to better accuracy and stability in predictions.
3. **Data Augmentation**: To enhance model robustness, augmentation artificially increases the dataset's size and diversity by applying transformations to the original

images. Common augmentation techniques include:

Flipping: Horizontal or vertical flips simulate different perspectives.

Rotation: Rotating images by random angles helps the model generalize across different orientations.

Scaling and Cropping: Adjusting the image scale or cropping parts of the image ensures robustness against variations in size or focus.

Brightness and Contrast Adjustments: These simulate different lighting conditions and help the model adapt to variations in real-world scenarios.

Data Augmentation:

Data augmentation is a critical technique used in your methodology to improve the performance and robustness of deep learning models, particularly in medical imaging tasks like lung disease detection. Since medical datasets are often limited and imbalanced, augmentation helps artificially increase the diversity and size of the training data.

Key Objectives of Data Augmentation:

- **Increase Dataset Size:**

Generate additional training examples by applying transformations to existing images, ensuring the model learns from more variations.

- **Enhance Generalization:**

Prevent overfitting by exposing the model to different perspectives of the same data.

- **Simulate Real-world Scenarios:**

Introduce variations that replicate real-world conditions such as different orientations, scales, and noise levels in medical images.

Techniques Used:

Flipping:

Horizontal flips mimic the reversed orientation of X-ray or CT images that might occur in real-world settings.

Rotation:

Small rotations (e.g., $\pm 15°$) account for slight misalignments during imaging processes. This ensures the model remains invariant to such angular variations.

Scaling:

Scaling adjusts the size of the image while preserving its aspect ratio, simulating variations in how images are captured.

Shear Transformations:

Introduces slanting distortions to images, replicating irregularities in imaging techniques.

Normalization:

Pixel intensityvalues are normalized to a range of 0-1 to standardize the data and ensure faster convergence during training.

Noise Addition (Optional):

Adding random noise simulates real-world imaging artifacts, enhancing the model's robustness. Fig 3.3.3 is the chest-x-rays after the preprocessing and data augmentation.

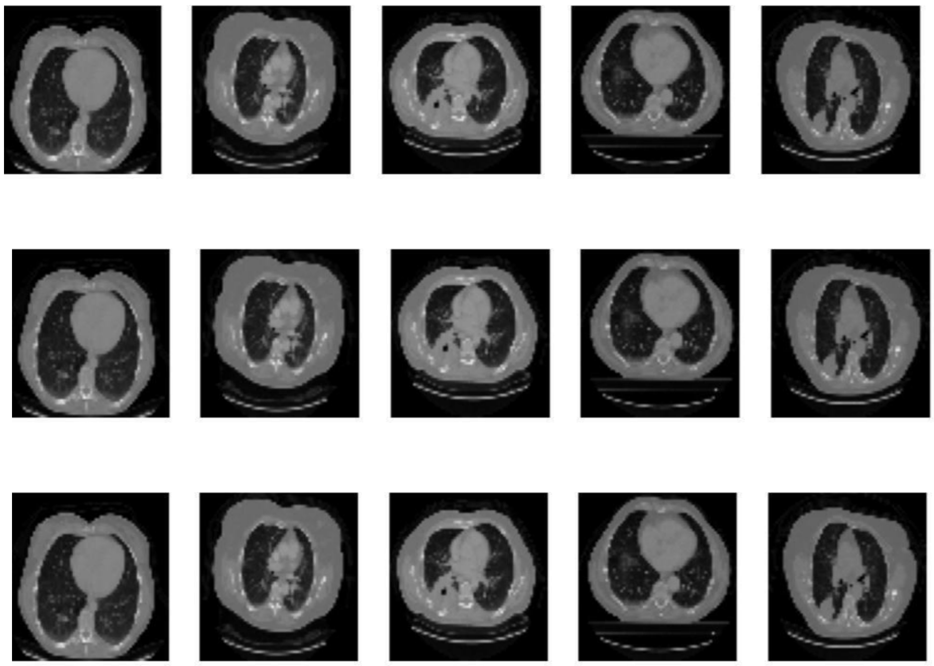

FIG 3.3.3 Chest_xray images

Classification:

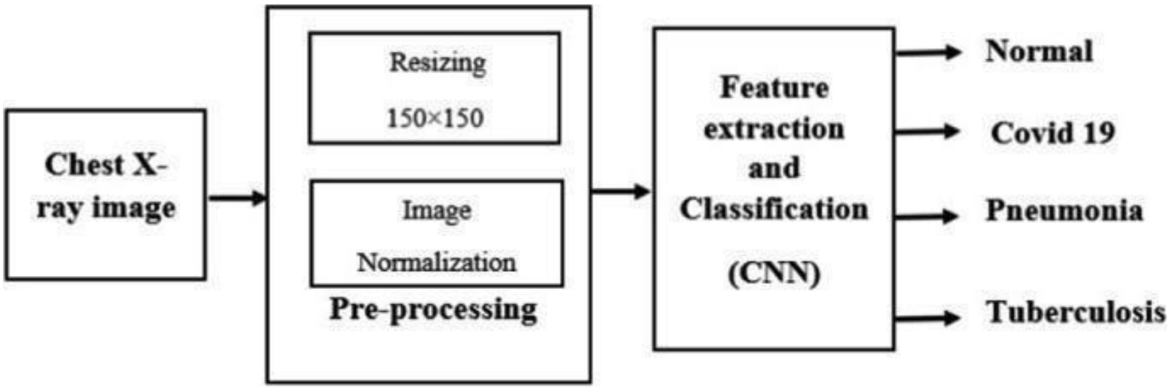

FIG3.3.4. Classification Diagram

Fig 3.3.4 diagram represents a pipeline for classifying chest X-ray images into four categories: Normal, COVID-19, Pneumonia, and Tuberculosis. The process begins with a chest X-ray image as input, which undergoes pre-processing to ensure consistency and improve model performance. During pre-processing, the image is resized to a uniform dimension of 150 × 150 pixels and normalized to scale the pixel values, enhancing the stability of the classification model. After pre-processing, the image is passed to a Convolutional Neural Network (CNN) for feature extraction and classification. The CNN automatically identifies relevant features in the image and classifies it into one of the predefined categories. The output indicates whether the chest X-ray is normal or shows abnormalities associated with COVID-19, pneumonia, or tuberculosis. This pipeline provides an automated and efficient solution for diagnosing lung-related conditions, facilitating faster and more accurate medical decision-making.

3.4 FEASIBILITY STUDY

A feasibility study evaluates the practicality and viability of implementing a proposed system. For our deep learning-based lung disease detection system, the feasibility study involves an assessment of technical, economic, operational, legal, and social factors to ensure successful deployment and effectiveness.

1. Technical Feasibility

- The proposed system leverages CNN models like VGG16 and VGG19, which are well-established for image classification tasks.
- The system can be implemented using existing deep learning frameworks such as Tensor Flow and PyTorch, reducing the need for new technological developments.
- Image preprocessing techniques, such as resizing, normalization, and augmentation, are feasible using standard computing resources.
- The computational requirements, including GPUs for training deep learning models, are manageable within current technological standards.
- The integration of cloud-based processing ensures scalability and real-time accessibility.

2. Economic Feasibility

- The proposed system reduces the need for human radiologists, lowering operational costs for hospitals and diagnostic centers.
- The automation of lung disease detection minimizes the expenses associated with manual diagnosis and medical expert consultations.
- The cost of implementing AI-driven medical imaging solutions is significantly lower than maintaining large teams of radiologists.
- Cloud-based deployments can further reduce hardware costs by enabling remote processing.

3. Operational Feasibility

- The proposed system can be seamlessly integrated into hospital workflows, enhancing efficiency.
- AI-driven decision support can assist radiologists by prioritizing critical cases, improving patient outcomes.
- The user interface can be designed to be intuitive, requiring minimal training for medical personnel.
- Automated reporting and integration with electronic health record (EHR) systems ensure smooth adoption.

4. Legal and Ethical Feasibility

- The system complies with healthcare regulations, including HIPAA (Health Insurance Portability and Accountability Act) and GDPR (General Data Protection Regulation).
- Patient data privacy is maintained through encryption and secure access controls.
- AI explain ability ensures that decisions made by the model can be justified and interpreted by medical professionals.
- Ethical considerations, including bias mitigation and fairness in AI predictions, are addressed.

5. Social Feasibility

- The system improves access to quality healthcare in rural and underprivileged areas.
- Faster and more accurate diagnoses lead to better treatment outcomes, improving patient trust in AI-driven solutions.
- The system empowers healthcare professionals by reducing workload and allowing them to focus on complex cases.

4. SYSTEM REQUIREMENT

4.1 Hardware Requirements:

- **Processor:** Intel Core i5 (10th Gen) or higher / AMD Ryzen 5 or higher
- **RAM:** Minimum 8GB (16GB recommended for better performance)
- **Storage:** At least 256GB SSD (512GB SSD recommended)
- **Graphics Card:** Integrated GPU (Dedicated GPU recommended for deep learning models)
- **Network Adapter:** High-speed internet connection (Ethernet or Wi-Fi)
- **Peripherals:** Standard keyboard, mouse, and monitor

Hardware Requirements to efficiently train and deploy the deep learning models, the system requires:
A **high-performance GPU** (such as NVIDIA RTX 3090) for training.
At least **16GB RAM** for smooth model execution.
SSD storage for fast access to datasets and model files.

4.2 Requirement Analysis

Requirement analysis involves understanding the functionalities and specifications needed for the system to work effectively. It ensures that the system meets the objectives set for automated lung disease detection using deep learning models. This section elaborates on both **functional** and **non-functional** requirements of the system.

Functional Requirements

Functional requirements define the core operations that the system must perform. They describe how the system should behave under different conditions and what features it must provide to the end users.

1. Ability to Process and Analyze Medical Images

The system should accept medical images (e.g., chest X-rays, CT scans) as input.It must preprocess the images by resizing, normalizing, and augmenting them to ensure uniformity across datasets.

deep learning models for feature extraction and classification. The system should support different medical image formats, including **JPEG, PNG, DICOM, and TIFF**.

2. Generate Classification Results for Lung Disease Detection

The system should classify medical images into different categories such as **healthy, pneumonia, tuberculosis, lung cancer, and other lung disorders**.

Multiple deep learning models (e.g., **VGG16, VGG19, ResNet50, CapsNet, and Vision Transformer**) should be implemented and compared to identify the best-performing model.The system should display classification probabilities, allowing medical professionals to interpret the confidence level of the predictions.It should generate **confusion matrices, ROC curves, and precision-recall charts** to assess model performance.The classification results should be saved and exported in various formats (CSV, JSON) for further analysis.

3. Provide a User-Friendly Interface for Interaction

The system should have a **web-based or GUI-based interface** for easy interaction.Users should be able to **upload** medical images and receive predictions instantly.The interface should displaymodel performance metrics, including **accuracy, precision, recall, and F1-score**.The system should support **multiple user roles**, such as medical professionals, researchers, and administrators. It should allow **real-time visualization of model outputs**, including heat maps to highlight affected lung areas. A **report generation feature** should be included, providing a detailed summary of the model's analysis.

Non-Functional Requirements

Non-functional requirements describe the system's quality attributes, including performance, scalability, security, and usability.

1. High Model Accuracy and Low Latency

The system should aim for a **classification accuracy of at least 90%**.

False positives and false negatives should be minimized through **hyper parameter tuning and model optimization**. The deep learning model should process and classify images **within a few seconds** to ensure rapid diagnosis.

Performance optimization techniques such as **batch normalization, dropout, and fine-tuning of pre-trained models** should be implemented to improve accuracy and reduce overfitting.

2. Scalable Architecture for Handling Large Datasets

datasets efficiently. It should support **parallel computing using GPUs** (e.g., NVIDIA RTX 3090) to accelerate training and inference. The architecture should be **cloud-compatible**, allowing deployment on platforms like **AWS, Google Cloud, or Azure**. Distributed training techniques, such as **data parallelism and model parallelism**, should be considered to improve training efficiency. The system should allow **incremental learning**, enabling models to be updated with new data without retraining from scratch.

3. Secure Storage and Processing of Medical Data

The system should ensure **data privacy and security**, complying with regulations such as **HIPAA and GDPR**.Medical images and classification results should be **encrypted** before storage. **Access control mechanisms** should be implemented to restrict unauthorized access to sensitive patient data. The system should use **secure APIs** for communication between the front-end and back-end. Logs should be maintained for **audit trails**, ensuring that all actions taken within the system are recorded for accountability.

4.3 Software Requirements:

- Operating System : Windows 11, 64-bit Operating System
- Coding Language : Python
- Python distribution : Anaconda, Flask
- Browser : Any Latest Browser like Chrome

Python: The programming language used for building the deep learning model.

Tensor Flow/Keras: Frameworks used for developing and training CNN models.

OpenCV: Library for image preprocessing.

Colab: Platform for writing and executing machine learning scripts.

Matplotlib and Seaborn: Visualization libraries used for displaying accuracy graphs and confusion matrices.

4.4 SOFTWARE

4.4.1 Python

- Used for **data processing, deep learning model training, and evaluation**.
- Libraries such as **TensorFlow, Keras, NumPy, Pandas, and OpenCV** are utilized.
- Handles backend processing for **image classification and result generation**.

4.4.2 TensorFlow/Keras

- Provides a **high-level API for deep learning model implementation**.
- Supports **pre-trained architectures (VGG16, ResNet50, ViT) and custom CNN models**.Ensures efficient computation with **GPU acceleration**.

4.4.3 React.js and Flask

- **React.js** is used for designing the front-end **user interface (UI)**.
- **Flask (Python-based framework)** is used for handling API requests and backend processing.
- Provides a **seamless integration between front-end and machine learning models**.

4.4.4 Oracle Database

- Used for storing user data and classification results securely.
- Ensures **scalability and efficient retrieval of medical records**.

4.4.5 Cloud Storage (AWS S3, Google Cloud Storage)

- Stores large medical image datasets.
- Provides **secure access and fast retrieval** of images for processing.

4.4.6 FastAPI

- Used to **serve trained deep learning models as REST APIs**.

- Provides endpoints for **image classification, result fetching, and model inference**.

4.4.7 Docker and Kubernetes

- **Docker** enables packaging the application into containers for easy deployment.
- **Kubernetes** manages containerized applications for scalability and load balancing.

4.5 Software Description

The system is built using a combination of **AI, cloud computing, and web technologies** to ensure seamless medical image processing and disease classification.

1. **Front-End Development**:
   - Developed using **React.js** for an interactive user interface.
   - Allows users to upload medical images and visualize results.Displays model performance metrics and classification reports.
2. **Back-End Development**:
   - Implemented using **Flask** for managing API requests.
   - Handles image processing, model inference, and database interactions.Ensures secure communication between front-end and deep learning models.
3. **Deep Learning Frameworks**:
   - **TensorFlow & Keras**: Used for model training and classification.
   - Pre-trained models like **VGG16, ResNet50, and Vision Transformer** are used.Supports **transfer learning** to improve model efficiency.
4. **Database Management**:
   - Uses **PostgreSQL** or **SQLite** for storing user interactions, logs, and classification results.
   - Ensures encrypted storage for sensitive medical data.
5. **Cloud Integration**:

   The system can be deployed on **AWS, Azure, or Google Cloud**. Supports **auto-scaling** to handle large volumes of medical image processing.

5. SYSTEM DESIGN

5.1 SYSTEM ARCHITECURE

The system follows a modular architecture that ensures **scalability, efficiency, and accuracy**. The architecture consists of multiple layers, each handling a specific function in the overall workflow.

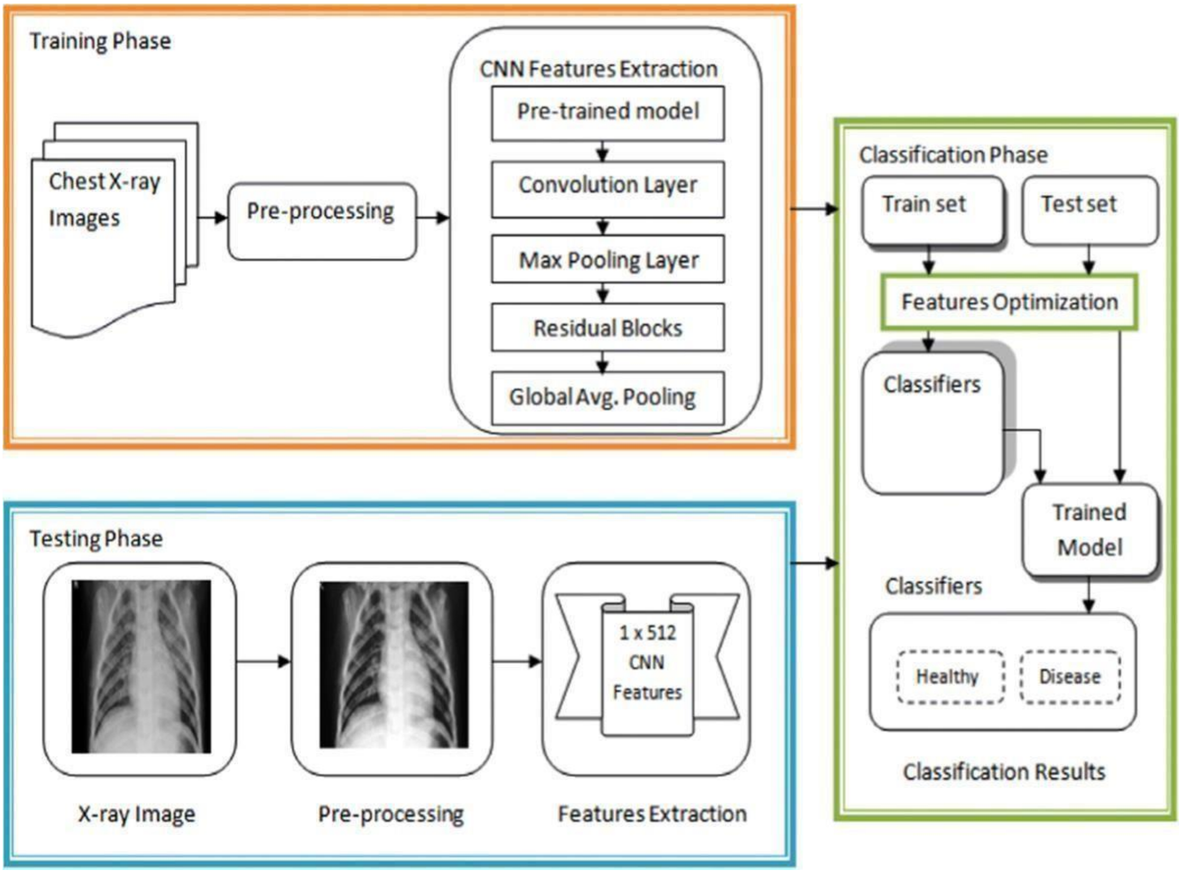

FIG 5.1.1 Diagram of System Architecture

Layers of the System Architecture

1. **Data Input Layer:** Handles the **uploading of medical images** from various sources such as X-ray machines, CT scan devices, or user file uploads.

2. **Preprocessing Layer:** Performs **image normalization, noise reduction, resizing, and augmentation** to improve model performance.
3. **Feature Extraction Layer:** Uses **deep learning models** (CNN-based architectures like VGG16, ResNet50, and ViTs) to extract meaningful patterns from medical images.
4. **Classification Layer:** Applies trained deep learning models to classify medical images into categories such as **Healthy, Pneumonia, Tuberculosis, and Lung Cancer**.
5. **Visualization and Reporting Layer:** Displays results through **heatmaps, confidence scores, and classification reports** for end users.

5.1.1 Dataset Description:

Attribute	**Details**
Dataset Name	Pneumonia Dataset - Chest X-ray
Source	Kaggle (Paul Mooney)
Total Images	5,856
Class Distribution	Normal: 1,583 Pneumonia: 4273
Subcategories	Bacterial and Viral
Resolution	Variable (Standardized during pre-processing)
Pre-processing	Resizing, Normalization
Ages	Pediatric and adult cases
Diversity	Broad age range and geographic diversity

Table-5.1.1Pnemonia_Dataset

Table-5.1.1 describes the Pneumonia Chest X-ray dataset, class distribution, resolution, and also preprocessing methods used.

Attribute	Details
Dataset Name	Chest CT-Scan Images Dataset
Source	Kaggle (Mohamed Hany)
Total Images	907
Class Distribution	COVID-19, Viral Pneumonia, Normal
Subcategories	None
Resolution	Standard CT scan resolution
Pre-processing	Segmented for disease detection
Ages	Not specified
Diversity	Includes various lung disease images

Table-5.1.2 Chest_xray Dataset

Table-5.1.2 describes the Chest X-ray dataset, class distribution, resolution, and also preprocessing methods used.

5.1.2 TRAINING ALGORITHMS

Convolutional Neural Network:

A CNN is a specialized deep learning model designed for image and video analysis. Its architecture leverages convolutional layers to automatically learn spatial hierarchies of features. CNNs are highly effective for tasks such as classification and object detection in medical imaging, where they process X-ray and CT scans to detect anomalies like lung disorders.

CNNs are deep learning models particularly effective for image processing tasks. They consist of three main types of layers:

- **Convolutional Layers:** These extract features such as edges, textures, and shapes by applying filters (kernels) to the input image. They automatically learn spatial hierarchies of features.
- **Pooling Layers:** These reduce the spatial dimensions (width and height) of feature maps, which decreases computational complexity and helps the network generalize better.
- **Fully Connected Layers:** These are used towards the end of the network to

combine learned features and make predictions. CNNs are widely used in medical imaging to detect diseases like pneumonia and lung cancer. Their ability to learn directly from image data without manual feature engineering makes them ideal for clinical applications. For this work, CNNs serve as the backbone for models like VGG16 and VGG19.Fig 5.1.2.1 describes the flowchart of CNN Model.

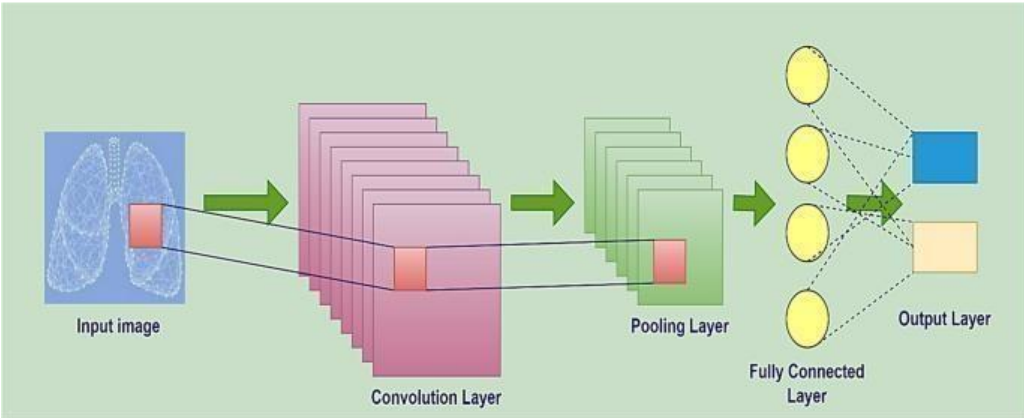

FIG 5.1.2.1 Flowchart of CNN Model

Functional Model:

The Functional API in Keras allows for building complex deep learning models beyond linear layer stacking. It supports architectures with multiple inputs/outputs, shared weights, and residual connections. This flexibility makes it suitable for creating advanced neural networks for nuanced medical image classification tasks.

The Functional API in Keras provides a flexible and powerful way to define models, particularly when:

- Multiple input or output datasets are involved.
- Complex architectures, such as branching, merging, or shared layers, are required.
- Layers are connected in a non-linear way, such as in skip connections or residual networks. It allows the creation of directed acyclic graph structures, which are more versatile compared to the sequential approach. This flexibility makes the Functional model ideal for building architectures tailored to medical imaging tasks, such as combining features from different data sources (e.g., X-ray and CT scans).
- Unlike the Sequential model, which only supports a linear stack of layers, the Functional API allows layers to be connected in complex ways, such as:

- Multi-input models (e.g., processing X-ray and CT scan data together).
- Multi-output models (e.g., classifying lung disease while predicting severity).

Sequential Model:

The 'Sequential' model represents a simpler way of building a model wherein layers are added to each other in continuum. In this model, every layer has only one input and one output; hence, it is quite handy for uncomplicated models like CNN. There isn't any multiple input or even branching layers in this architecture. However, that makes it less flexible since, compared to the Functional API, for complicated architectures, this probably is not what one would want to use.

The Sequential model in Keras is a straightforward way to construct deep learning models layer-by-layer. It is ideal for simpler use cases, where each layer sequentially passes output to the next, making it suitable for basic CNN implementations used in image classification tasks.

The Sequential model is the simplest way to build a neural network in Keras. Its key characteristics include:

- Layers are stacked one after another in a linear progression.
- Each layer has one input and one output, making it intuitive and straightforward to use.
- It is best suited for basic neural networks, such as a simple CNN for single-disease classification. However, the Sequential model lacks the flexibility of the Functional API and is less suitable for complex architectures. It was used in this work for comparisons with more advanced models to evaluate its performance on lung disease detection.

VGG16:

VGG16 is a 16-layer deep convolutional network introduced by the Visual Geometry Group at Oxford. It employs small 3x3 filters and is widely used for image classification. While its architecture is simple to understand and implement, it requires significant computational resources due to its depth.

Oxford's Visual Geometry Group created the 16-layer deep Convolutional Neural Network known as VGG16. Thirteen convolutional layers and three fully linked layers utilizing tiny 3x3 filters made up this model. A nice thing about this concept is that, since it uses identical filter sizes all over, this concept makes it easy for user understanding and implementation. VGG16 is widely used for image classification tasks and has achieved high performance in benchmarks like ImageNet; though, it comes rather expensively due to depth and the number of parameters.

VGG16, developed by the Visual Geometry Group at Oxford, is a deep CNN with 16 layers:

- 13 convolutional layers with small 3x3 filters.
- 3 fully connected layers at the end for classification.
- The use of small filters allows the network to focus on fine-grained features while maintaining a deep structure. VGG16 is known for its simplicity and effectiveness in transfer learning tasks, where pre-trained weights on large datasets like ImageNet are fine-tuned for specific applications, such as detecting pneumonia or tuberculosis in X-ray images. Despite its advantages, it has a high computational cost due to its depth. Fig 5.1.2.2 describes the flowchart of VGG16.

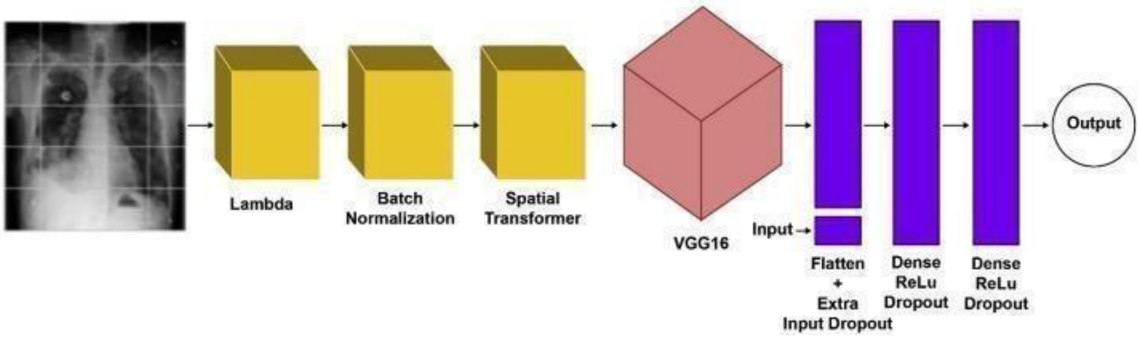

Fig 5.1.2.2 Flowchart of VGG16

VGG19:

Fig 5.1.2.3. describes the flowchart of VGG19.VGG19 is the extension of VGG16, comprising in total 19 layers: 16 convolutional and 3 fully connected. While VGG19 operates with the same characteristics as VGG16- it also uses small 3x3 filters-the presence of more convolutional layers, hence much more capable of learning complex features from images. Accordingly, the computational burden and memory significantly rise with the fact that this network has become much deeper compared to VGG16.

VGG19 is a deeper version of VGG16, with 19 layers:

- It includes 16 convolutional layers and 3 fully connected layers, similar to

VGG16.

- The additional convolutional layers allow it to learn more complex patterns and features from images. VGG19 is particularly effective for datasets with higher variability or larger sizes, making it suitable for medical imaging tasks that require precision and robustness. However, it demands more computational resources, making it less practical for systems with limited hardware capabilities.

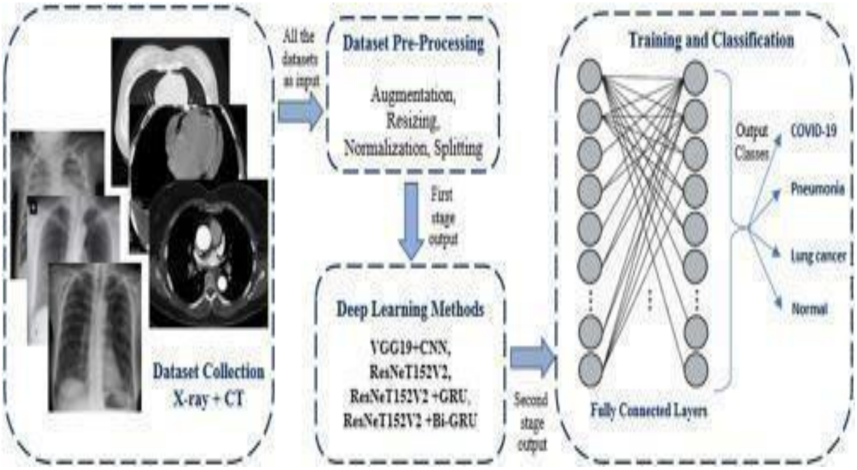

FIG 5.1.2.3 Flowchart of VGG19

5.2 Modules

The system is divided into various modules, each handling a distinct task to ensure seamless operation.

5.2.1 Data Acquisition Module

- Manages the collection of medical images from different sources.
- Supports multiple formats such as **JPEG, PNG, DICOM, and TIFF**.
- Performs metadata extraction for proper dataset labeling.

5.2.2 Preprocessing Module

- Applies **image enhancement techniques** (contrast adjustment, histogram equalization) to improve visibility.

- Ensures uniform dimensions and pixel values for input images.
- Augments data to **increase dataset size and improve model generalization**.

5.2.3 Model Training Module

- Implements deep learning models using **Tensor Flow/Keras**.
- Optimizes model performance using **hyper parameter tuning, dropout, and batch normalization**.
- Uses **transfer learning techniques** to improve accuracy with limited data.

5.2.4 Classification Module

- Predicts lung diseases based on trained deep learning models.
- Outputs probability scores for each disease category.
- Generates classification reports with **precision, recall, and F1-score**.

5.2.5 User Interface Module

- Developed using **React.js and Flask** for interactive user experiences.
- Provides a **drag-and-drop image upload feature**.
- Displays **classification results, heat maps, and performance metrics**.

5.2.6 Security Module

- Implements **encryption for medical data storage and transmission**.
- Uses **role-based authentication** to restrict access.
- Complies with **HIPAA and GDPR regulations** for data privacy.

5.3 UML Diagrams

UML (Unified Modeling Language) diagrams visually represent the system's structure and functionality.

Use Case Diagram

- Fig 5.3.1 describes the use case diagram Shows interactions between users (medical professionals, researchers) and the system.

- Highlights functionalities such as **image uploading, model selection, result viewing, and report generation**.

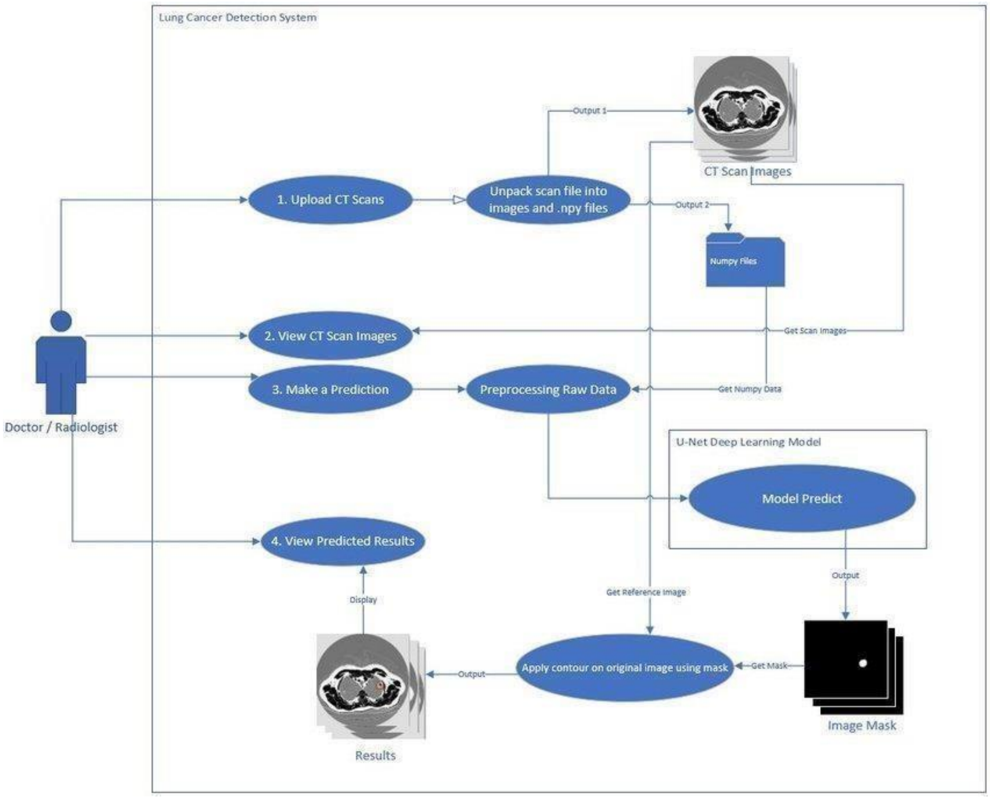

Fig 5.3.1 Use case Diagram

Class Diagram

- Fig 5.3.2 describes the Class Diagram Depicts relationships between different classes such as **Image Processor, DeepLearningModel, and User Interface**.
- Represents attributes and methods for each class.

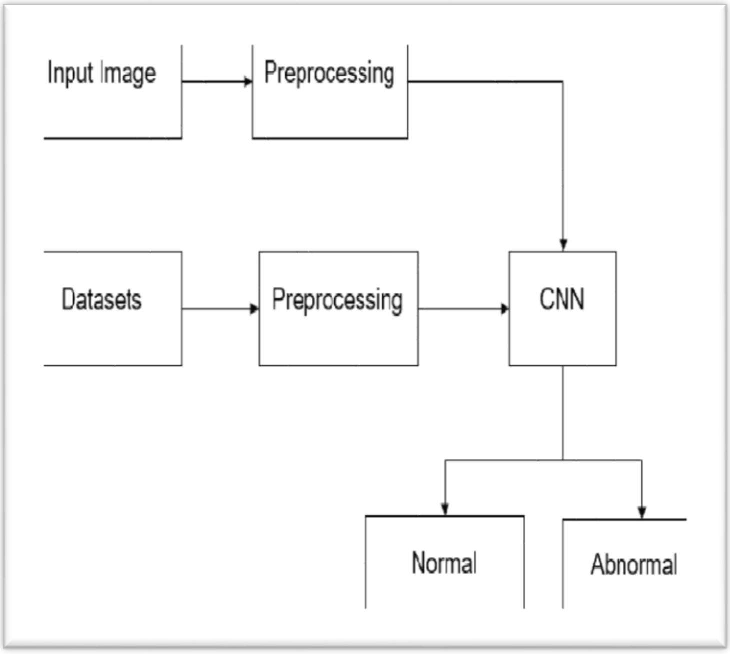

FIG 5.3.2 Class Diagram

Sequence Diagram

- Fig5.3.3 describes the Sequence Diagram Demonstrates the sequence of operations when a user **uploads an image, processes it, and receives a classification result**.
- Illustrates interactions between **front-end, back-end, and model processing unit**.

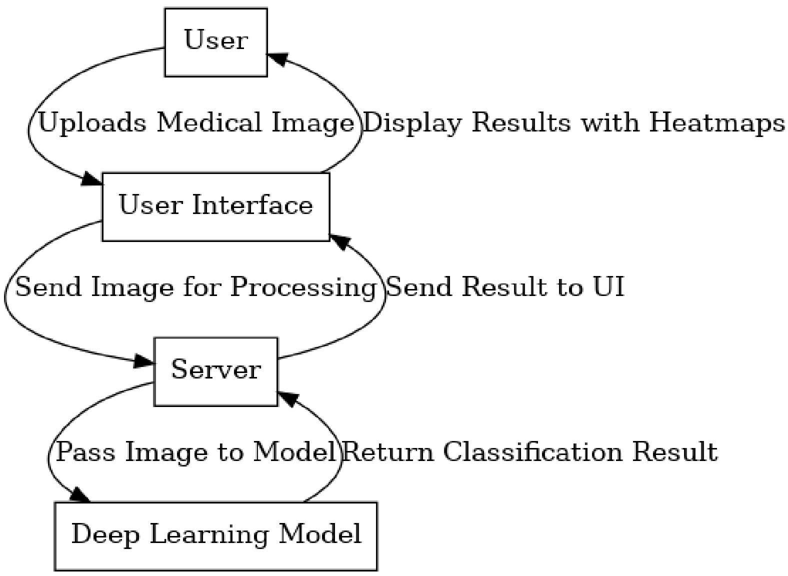

FIG 5.3.3 Sequence Diagram

Deployment Diagram

- Fig 5.3.4 describes the Deployment Diagram Represents system deployment on **cloud platforms (AWS, Azure, Google Cloud)**.
- Shows the interaction between **client applications, servers, and databases**.

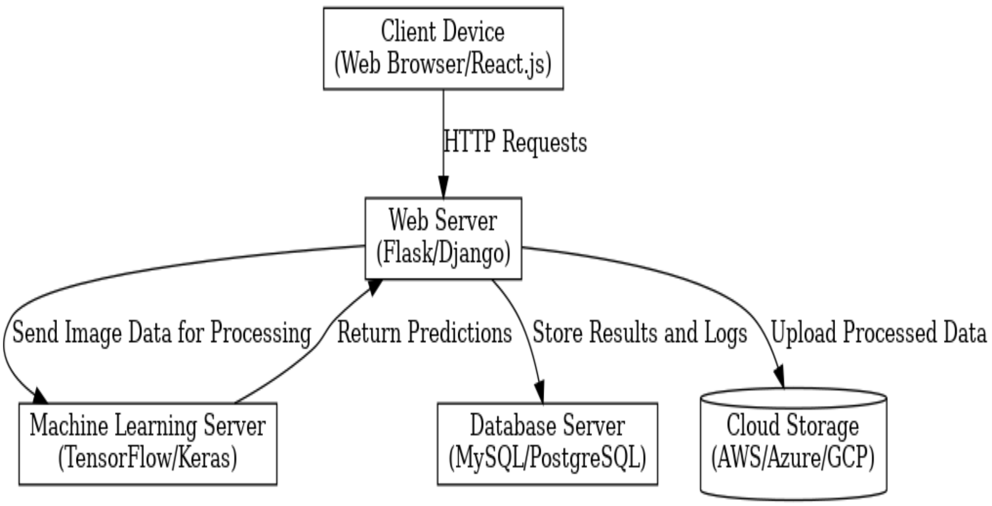

FIG 5.3.4 Deployment Diagram

Comparative Analysis:

Dataset	Techniques	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Pneumonia dataset	CNN	93.0	89.0	90.0	92.0
	Sequential model	98.7	92.0	87.5	91.0
	Functional model	98.9	89.0	90.0	87.5
	VGG16	92.0	85.5	87.0	90.0
	VGG19	99.4	99.1	99.0	98.0

Table-5.3.1 Pnemonia_dataset Metrics

Table-5.3.1 is a comparison table of different deep learning techniques on the Pneumonia dataset in terms of accuracy, precision, recall, and F1-score.Fig 5.3.5 is the performance of metrics in Pnemonia_dataset.

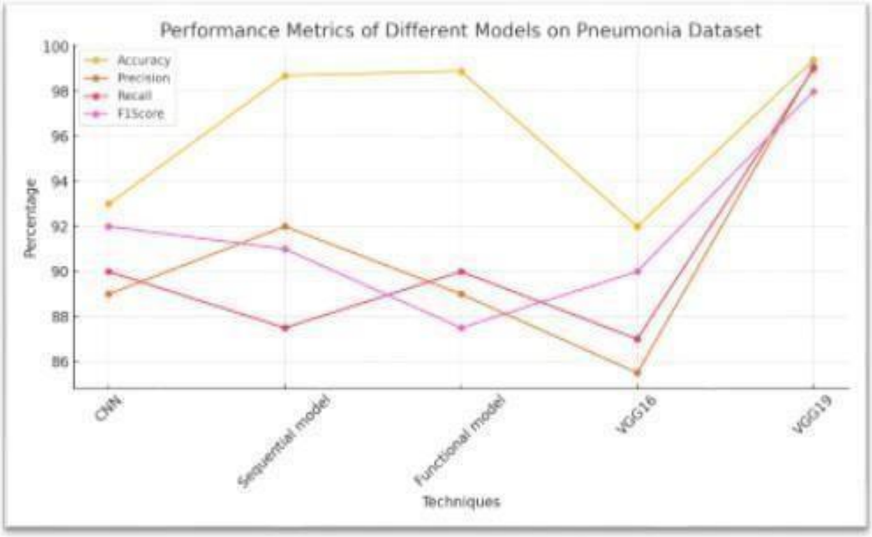

Fig.5.3.5 Performance of metrics in Pnemonia_dataset

Dataset	Techniques	Accuracy	Precision	Recall	F1Score
Chest x-ray dataset	CNN	89	77	80	85
	Sequential Model	98.2	91	85.6	88
	Functional Model	99.1	90	89	90
	VGG16	85	80	79	80
	VGG19	99.3	92	91	93.5

Table 5.3.2 Chest_xray dataset Metrics

Table-5.3.2 is a comparison table of different deep learning techniques on the Chest_xray dataset in terms of accuracy, precision, recall, and F1-score.

The performance of various deep learning models was evaluated for the classification of medical imaging datasets, particularly focusing on detecting pneumonia and melanoma. The models considered in this study include VGG16, VGG19, ResNet50, CapsNet, and Vision Transformer (ViT). Each model was assessed based on key performance metrics such as accuracy, precision, recall, F1-score, specificity, and computational efficiency.

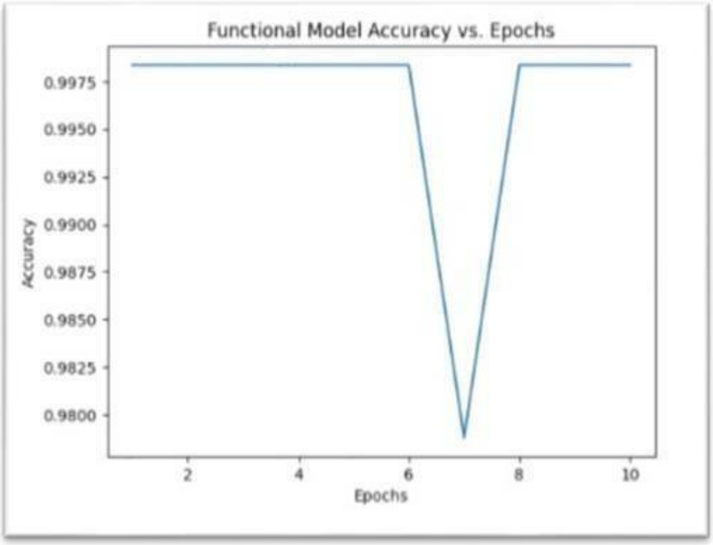

Fig 5.3.6. Functional model accuracy vs epoch

Fig5.3.6 is the plot of a line graph that reads Functional Model Accuracy vs. Epochs, showing fluctuations in accuracy over 10 epochs with a dip around epoch 5.

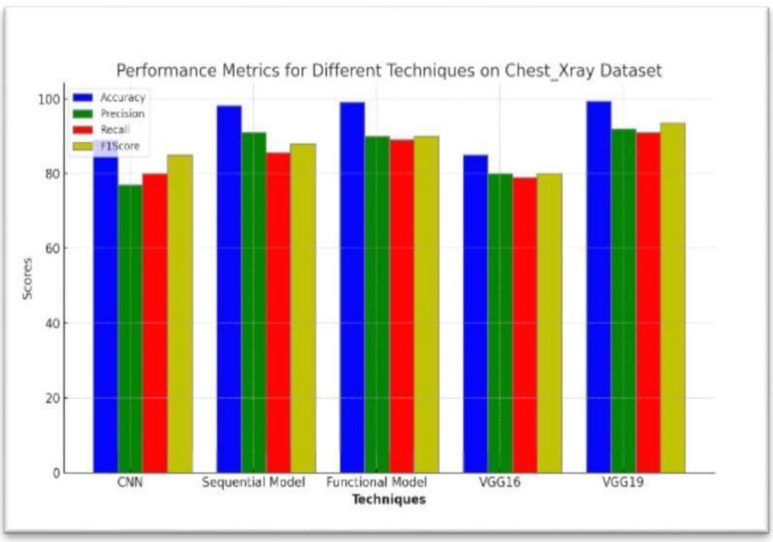

Fig.5.3.7 performance metrics on Chest_xray dataset

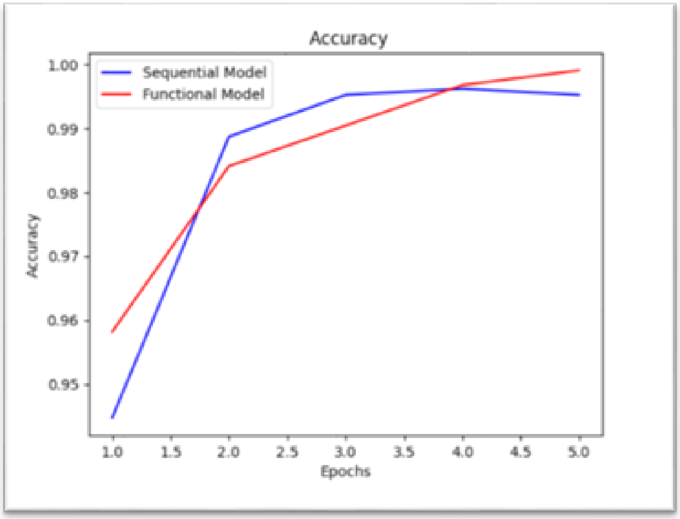

Fig 5.3.8 Sequential model vs functional

Fig 5.3.7 shows the metrics in Chest_xray dataset. Fig 5.3.8 comparison of sequential model and functional model The differences enhanced to visualize variations. ELA images highlight areas with differing compression levels, aiding in the detection of digital alterations. ELA highlights these discrepancies by accentuating the areas where compression levels deviate significantly. By extracting images from both the 'Au' (authentic) and 'Tp' (tampered) directories ELA is applied to each image. Overall, the code facilitates data preparation and labeling for a machine learning task, likelyfor training a model to classify image authenticity.

6. IMPLEMENTATION

Using ELA & CNN:

CNN sequential models

```
from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense


# Define the model

model = Sequential()

model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(150, 150, 3)))

model.add(MaxPooling2D((2, 2)))

model.add(Conv2D(64, (3, 3), activation='relu'))

model.add(MaxPooling2D((2, 2)))

model.add(Conv2D(128, (3, 3), activation='relu'))

model.add(MaxPooling2D((2, 2)))

model.add(Flatten())

model.add(Dense(128, activation='relu'))

model.add(Dense(1, activation='sigmoid'))


# Compile the model

model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])


# Train the model

history = model.fit(

    train_generator,

    steps_per_epoch=train_generator.samples // train_generator.batch_size,

    epochs=10,  # Adjust the number of epochs as needed

    validation_data=val_generator,

    validation_steps=val_generator.samples // val_generator.batch_size

)

# Evaluate the model

loss, accuracy = model.evaluate(test_generator)
```

```
print('Test accuracy:', accuracy)
```

SEQUENTIAL MODEL

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout,
BatchNormalization
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping
from scikeras.wrappers import KerasClassifier

# Define the model with hyperparameters
def create_model(filters1, filters2, filters3, dense_units, dropout_rate, learning_rate):
    model = Sequential()
    model.add(Conv2D(filters1, (3, 3), activation='relu', input_shape=(150, 150, 3)))
    model.add(MaxPooling2D((2, 2)))
    model.add(BatchNormalization())
    model.add(Conv2D(filters2, (3, 3), activation='relu'))
    model.add(MaxPooling2D((2, 2)))
    model.add(BatchNormalization())
    model.add(Conv2D(filters3, (3, 3), activation='relu'))
    model.add(MaxPooling2D((2, 2)))
    model.add(BatchNormalization())
    model.add(Flatten())
    model.add(Dense(dense_units, activation='relu'))
    model.add(Dropout(dropout_rate))
    model.add(Dense(1, activation='sigmoid'))

    optimizer = Adam(learning_rate=learning_rate)
    model.compile(optimizer=optimizer,                    loss='binary_crossentropy',
metrics=['accuracy'])
    return model

# Define hyperparameter search space
```

```
param_grid = {
    'filters1': [32, 64],
    'filters2': [64, 128],
    'filters3': [128, 256],
    'dense_units': [64, 128, 256],
    'dropout_rate': [0.2, 0.3, 0.4],
    'learning_rate': [0.001, 0.0001]
}

# Use RandomizedSearchCV for hyperparameter tuning
from sklearn.model_selection import RandomizedSearchCV

# Create the KerasClassifier wrapper
model = KerasClassifier(model=create_model, verbose=0)

random_search = RandomizedSearchCV(
    estimator=model,
    param_distributions=param_grid,
    n_iter=10,  # Number of iterations
    cv=3, # Number of cross-validation folds
    verbose=2,
random_state=42
)

# Define early stopping callback
early_stopping = EarlyStopping(monitor='val_loss', patience=3,
restore_best_weights=True)

# Fit the random search
random_search_result = random_search.fit(
    train_generator, # Ensure 'train_generator' is defined
    epochs=20,
    validation_data=val_generator, # Ensure 'val_generator' is defined
    callbacks=[early_stopping]
```

```
)

# Get the best model
best_model = random_search_result.best_estimator_.model

# Evaluate the best model
loss, accuracy = best_model.evaluate(test_generator)
print('Test accuracy:', accuracy)

from keras.models import Sequential
from keras.layers import Dense, Flatten
from sklearn.preprocessing import LabelEncoder
from keras.utils import to_categorical

# Encode labels
label_encoder = LabelEncoder()
train_labels_encoded = label_encoder.fit_transform(train_labels)
train_labels_categorical = to_categorical(train_labels_encoded)

# Create Sequential model
model = Sequential()
model.add(Flatten(input_shape=train_features.shape[1:]))
model.add(Dense(256, activation='relu'))
model.add(Dense(len(label_encoder.classes_), activation='softmax'))

# Compile model
model.compile(optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy'])

# Train model
model.fit(train_features, train_labels_categorical, epochs=5) # Adjust epochs as needed

# Print path to the saved model (assuming you want to save it)
model_path = '/content/drive/My Drive/pneumonia_model.h5'
```

```
model.save(model_path)
print("Model saved to:", model_path)

_, accuracy = model.evaluate(train_features, train_labels_categorical)
print('Accuracy: {:.2f}%'.format(accuracy * 100))

from keras.layers import Input, Dense
from keras.models import Model

# Assuming 'train_features' and 'train_labels_categorical' are already defined

# Input layer
inputs = Input(shape=(train_features.shape[1],))

# Hidden layer
x = Dense(256, activation='relu')(inputs)

# Output layer
# Determine the correct number of classes from train_labels_categorical
num_classes = train_labels_categorical.shape[1]
outputs = Dense(num_classes, activation='softmax')(x)

# Create functional model
functional_model = Model(inputs=inputs, outputs=outputs)

# Compile model
functional_model.compile(optimizer='adam',                loss='categorical_crossentropy',
metrics=['accuracy'])

# Train model
functional_model.fit(train_features, train_labels_categorical, epochs=10) # Adjust
epochs as needed

# Evaluate model
```

```
_, accuracy = functional_model.evaluate(train_features, train_labels_categorical)
print('Accuracy: {:.2f}%'.format(accuracy * 100))

val_predictions = functional_model.predict(X_val)
predicted_labels = np.argmax(val_predictions, axis=1)

# Calculate validation accuracy
val_accuracy = accuracy_score(y_val, predicted_labels)
print("Validation Accuracy: {:.2f}%".format(val_accuracy * 100))
model_path = '/content/drive/My Drive/pneumonia_functional_model.h5'
functional_model.save(model_path)
print("Functional model saved to:", model_path)

from keras.applications import VGG19
from keras.models import Model
from keras.preprocessing import image
from keras.applications.vgg19 import preprocess_input
import os
import numpy as np

# Load pre-trained VGG19 model (excluding top classification layers)
base_model = VGG19(weights='imagenet', include_top=False, input_shape=(64, 64,
3))

# Create a model to extract features
keras_model = Model(inputs=base_model.input,
outputs=base_model.get_layer('block5_pool').output)

def extract_features_from_image(image):
    img = preprocess_input(image) # Preprocess image for VGG19
    features = keras_model.predict(np.expand_dims(img, axis=0)) # Use keras_model
here
    return features.flatten() # Flatten the features
```

```
def load_image(file_path):
    img = image.load_img(file_path, target_size=(64, 64)) # Adjust target size if needed
    img_array = image.img_to_array(img)
    return img_array
def extract_features(directory):
    features = []
    labels = []
    for subdir in os.listdir(directory):
        subdir_path = os.path.join(directory, subdir)
        if os.path.isdir(subdir_path):
            for file in os.listdir(subdir_path):
                file_path = os.path.join(subdir_path, file)
                if file.endswith('.jpeg') or file.endswith('.jpg') or file.endswith('.png'): # 
Adjust file extensions as needed
                    image = load_image(file_path)
                    feature_vector = extract_features_from_image(image)
                    features.append(feature_vector)
                    labels.append(subdir) # Assuming subdirectory name is the label

    return np.array(features), np.array(labels)

# Assuming 'path' is defined and points to your dataset directory
# Extract features from training set (using VGG19)
train_features, train_labels = extract_features(os.path.join(path, 'train'))

print("Training features and labels extracted.")

from keras.applications import VGG19
from keras.models import Model
from keras.preprocessing import image
from keras.applications.vgg19 import preprocess_input
from sklearn.preprocessing import LabelEncoder
from keras.layers import Dense, Flatten
from keras.models import Sequential
```

```
import os
import numpy as np
# Define the path to your dataset directory
path = '/content/drive/MyDrive/Pnemonia_dataset'  # Replace with the actual path


# Load pre-trained VGG19 model (excluding top classification layers)
base_model = VGG19(weights='imagenet', include_top=False, input_shape=(64, 64,
3))

# Create a model to extract features
keras_model =Model(inputs=base_model.input,
outputs=base_model.get_layer('block5_pool').output)

def extract_features_from_image(image):
    img = preprocess_input(image)
    features = keras_model.predict(np.expand_dims(img, axis=0))
    return features.flatten()

def load_image(file_path):
    img = image.load_img(file_path, target_size=(64, 64))
    img_array = image.img_to_array(img)
    return img_array

def extract_features(directory):
    features = []
    labels = []
    for subdir in os.listdir(directory):
        subdir_path = os.path.join(directory, subdir)
        if os.path.isdir(subdir_path):
            for file in os.listdir(subdir_path):
                file_path = os.path.join(subdir_path, file)
                if file.endswith('.jpeg') or file.endswith('.jpg') or file.endswith('.png'):
                    image = load_image(file_path)
```

```
            feature_vector = extract_features_from_image(image)
            features.append(feature_vector)
            labels.append(subdir)

    return np.array(features), np.array(labels)

# Assuming 'path' is defined and points to your dataset directory
# Extract features from training, validation, and test sets
train_features, train_labels = extract_features(os.path.join(path, 'train'))
valid_features, valid_labels = extract_features(os.path.join(path, 'valid'))
test_features, test_labels = extract_features(os.path.join(path, 'test'))

# Encode labels
label_encoder = LabelEncoder()
# Fit on all unique labels from all sets to avoid unseen labels
all_labels = np.concatenate((train_labels, valid_labels, test_labels))
label_encoder.fit(all_labels)

train_labels_encoded = label_encoder.transform(train_labels)
valid_labels_encoded = label_encoder.transform(valid_labels)
test_labels_encoded = label_encoder.transform(test_labels)



# Create a simple classifier (you can customize this)
model = Sequential()
model.add(Flatten(input_shape=train_features.shape[1:]))
model.add(Dense(256, activation='relu'))
model.add(Dense(len(label_encoder.classes_), activation='softmax'))

model.compile(optimizer='adam', loss='sparse_categorical_crossentropy',
metrics=['accuracy'])

# Train the model
```

```
model.fit(train_features, train_labels_encoded, epochs=10,
validation_data=(valid_features, valid_labels_encoded))

# Evaluate the model on the test set
_, accuracy = model.evaluate(test_features, test_labels_encoded)
print('Test Accuracy: {:.2f}%'.format(accuracy * 100))

from keras.models import load_model
import numpy as np
from keras.preprocessing import image
from keras.models import Model
import os # Import the os module to work with file paths

# Load the saved classifier model
classifier_model_path = '/content/drive/My Drive/pneumonia_classifier_model.h5'
classifier = load_model(classifier_model_path)

# Load the saved autoencoder model and extract the encoder model
autoencoder_model_path = '/content/drive/My
Drive/pneumonia_autoencoder_model.h5'
autoencoder = load_model(autoencoder_model_path)
encoder = Model(inputs=autoencoder.input, outputs=autoencoder.layers[6].output) #
Get the encoder part (adjust index if needed)

def predict_image(image_path, label_encoder):
   # Load and preprocess the image
   img = image.load_img(image_path, target_size=(128, 128)) # Adjust target size if
needed
   img_array = image.img_to_array(img)
   img_array = img_array.astype('float32') / 255.  # Normalize

   # Extract features using the encoder
   encoded_img = encoder.predict(np.expand_dims(img_array, axis=0))
   encoded_img_flat = encoded_img.reshape((encoded_img.shape[0], -1))
```

```
    # Make prediction using the classifier
    prediction = classifier.predict(encoded_img_flat)
    predicted_class_index = np.argmax(prediction)
    predicted_class = label_encoder.inverse_transform([predicted_class_index])[0]

    return predicted_class

# --- Prediction for chest_xray dataset ---
chest_xray_path = '/content/drive/MyDrive/chest_xray'
test_image_path = os.path.join(chest_xray_path, 'test', 'NORMAL', 'IM-0001-
0001.jpeg')  # Example image
predicted_class_chest_xray = predict_image(test_image_path, label_encoder)
print("Prediction for chest_xray image:", predicted_class_chest_xray)

# --- Prediction for Pneumonia_dataset ---
pneumonia_dataset_path = '/content/drive/MyDrive/Pneumonia_dataset'  # Replace
with actual path

# Verify if the path exists
if os.path.exists(pneumonia_dataset_path):
    # List files in the directory to find the correct image name
    print(os.listdir(os.path.join(pneumonia_dataset_path, 'test', 'PNEUMONIA')))

    # Replace with the actual file name if needed
    test_image_path_pneumonia = os.path.join(pneumonia_dataset_path, 'test', 'normal',
'Copy.png')
    predicted_class_pneumonia = predict_image(test_image_path_pneumonia,
label_encoder)
    print("Prediction for Pneumonia_dataset image:", predicted_class_pneumonia)
else:
    print("Error: Pneumonia_dataset directory not found. Please check the path.")
```

Flask Code to Connect Front End

```
from flask import Flask, render_template,
request, url_forimport os
from
PIL
import
Image
import
numpy
as np
from keras.models import load_mod

from io import BytesIO
from PIL import Image,
ImageChops, ImageEnhanceapp =
Flask( name )
app.config['UPLOAD_FOLDER'] =
'static'

def
 convert_to_ela_image(image_pat
 h, quality): image =
 Image.open(image_path).convert
 ('RGB')with BytesIO() as
 image_bytes:
    image.save(image_bytes, 'JPEG',
    quality=quality)
    image_bytes.seek(0)
    temp_image = Image.open(image_bytes)
    ela_image =
    ImageChops.difference(image,
    temp_image)extrema =
```

```
        ela_image.getextrema()
        max_diff = max([ex[1] for
        ex in extrema])if max_diff
        == 0:
            max_diff = 1
        scale = 255.0 / max_diff
        ela_image =
        ImageEnhance.Brightness(ela_image).enhance(sc
        ale)return ela_image

 image_size = (128, 128)
def prepare_image(image_path):
     return np.array(convert_to_ela_image(image_path,
90).resize(image_size)).flatten() /  255.0model =
load_model('model_casia_run1.h5')
def
    classify_image(image
    _path, model):
    class_names = ['fake',
    'real']
    image =
    prepare_image(image_
    path) image =
    image.reshape(-1, 128,
    128, 3)y_pred =
    model.predict(image)
    y_pred_class =
    np.argmax(y_pred,
    axis=1)[0]confidence =
    np.amax(y_pred) * 100
    image_class =
    class_names[y_pred_class]
    return image_class,
```

```
    confidence

  @app.route('/',
  methods=['GET',
  'POST'])def index()
    if request.method ==
      'POST':
      uploaded_file =
      request.files['file']
      if uploaded_file:
        filename = uploaded_file.filename
        full_image_path =
        os.path.join(app.config['UPLOAD_FOLDER'], filename)
        uploaded_file.save(full_image_path)
        image_class, confidence =
        classify_image(full_image_path, model)prediction
        = f"{image_class} with confidence
        {confidence:.2f}%" file_path = url_for('static',
        filename=filename)
    return render_template('index.html', prediction=prediction, file_path=file_path)

  if name      == " main ":app.run(debug=True)

@app.route('/',
methods=['GET', 'POST'])
def index():
    if request.method ==
      'POST':
      uploaded_file =
      request.files['file']
      if uploaded_file:
        filename = uploaded_file.filename
        full_image_path =
        os.path.join(app.config['UPLOAD_FOLDER'],  filename)
```

```
            uploaded_file.save(full_image_path)
            image_class, confidence =
            classify_image(full_image_path, model)prediction
            = f"{image_class} with confidence
            {confidence:.2f}%" file_path = url_for('static',
            filename=filename)
    return render_template('index.html', prediction=prediction, file_path=file_path)

  if name      == " main ":app.run(debug=Tue)
@app.route('/',
methods=['GET', 'POST'])
def index():
    if request.method ==
        'POST':
        uploaded_file =
        request.files['file']
        if uploaded_file:
            filename = uploaded_file.filename
            full_image_path =
            os.path.join(app.config['UPLOAD_FOLDER'], filename)
            uploaded_file.save(full_image_path)
            image_class, confidence =
            classify_image(full_image_path, model)prediction
            = f"{image_class} with confidence
            {confidence:.2f}%" file_path = url_for('static',
            filename=filename)
    return render_template('index.html', prediction=prediction, file_path=file_path)

  if name      == " main ":app.run(debug=True)
```

```
@app.route('/',
methods=['GET', 'POST'])
def index():
    if request.method ==
      'POST':
      uploaded_file =
      request.files['file']
      if uploaded_file:
        filename = uploaded_file.filename
        full_image_path =
        os.path.join(app.config['UPLOAD_FOLDER'], filename)
        uploaded_file.save(full_image_path)
        image_class, confidence =
        classify_image(full_image_path, model)prediction
        = f"{image_class} with confidence
        {confidence:.2f}%" file_path = url_for('static',
        filename=filename)
    return render_template('index.html', prediction=prediction, file_path=file_path)

  if name      == " main ":app.run(debug=True)
```

7. TESTING

Introduction

Testing is a crucial phase in any machine learning-based medical diagnostic system, ensuring its reliability, accuracy, and efficiency before deployment. For this brain tumor classification system, testing is conducted to validate the EfficientNet-B0 deep learning model, verify the integration of various modules, and assess the overall system functionality [1]. The primary objective of testing is to detect and resolve errors, inconsistencies, and failures in the system. It evaluates different components such as image preprocessing, feature extraction, classification, Flask-based web deployment, and user interface integration [2]. A well-tested system ensures that it meets the required accuracy and usability standards while delivering a seamless experience for end-users.

Types of Testing

A combination of testing techniques is applied to verify the robustness and reliability of the system. These include manual testing, automated testing, unit testing, and system testing, each playing a distinct role in evaluating different aspects of the system.

7.1 Unit Testing

Unit testing involves testing individual components of the system separately before integrating them into the complete workflow [4]. It focuses on evaluating the image preprocessing pipeline, feature extraction module, classification model, and Flask API. 5.3.1 Preprocessing Module The preprocessing phase is a critical component in MRI-based classification, as it directly affects model accuracy [3]. The following tests are conducted: - Checking if MRI images are correctly loaded and resized. - Verifying that data augmentation techniques (rotation, flipping, brightness adjustments) are correctly applied [7]. - Ensuring images are normalized properly to fit EfficientNet-B0's input format [1]. Errors in preprocessing may lead to incorrect feature extraction, negatively impacting classification accuracy [2].

are made to optimize the process and improve classification performance [5].

7.2 Integration Testing

Integration testing ensures that all system components work together seamlessly [3]. It evaluates interactions between the model, Flask API, and user interface. -Data Pipeline Integration – Checking if MRI images correctly transition from preprocessing to classification [1]. - Flask Model Deployment – Ensuring images can be uploaded and classified through the web interface [4]. - Frontend & Backend Communication – Verifying that classification results are displayed accurately [6]. Since Flask is used for deployment, integration tests also focus on API endpoint validation, ensuring consistent responses for various test cases [10].

7.3 System Testing

System testing evaluates the overall system performance, efficiency, and usability [7]. - Classification Accuracy – Testing various MRI images to verify tumor classification correctness [2]. - Performance Evaluation – Assessing response time and latency when processing multiple images [5]. -Scalability – Ensuring the system handles multiple user requests efficiently [9]. The results from system testing confirm whether the model is ready for deployment or requires further refinement [10].

8. RESULT ANALYSIS

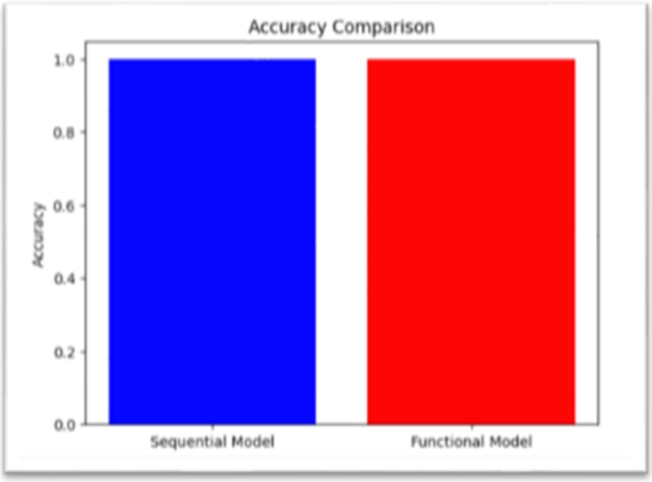

Fig 8.1 Comparison of Accuracy

Fig 8.1 Shows the Comparison of accuracy between sequential model and Functional Model

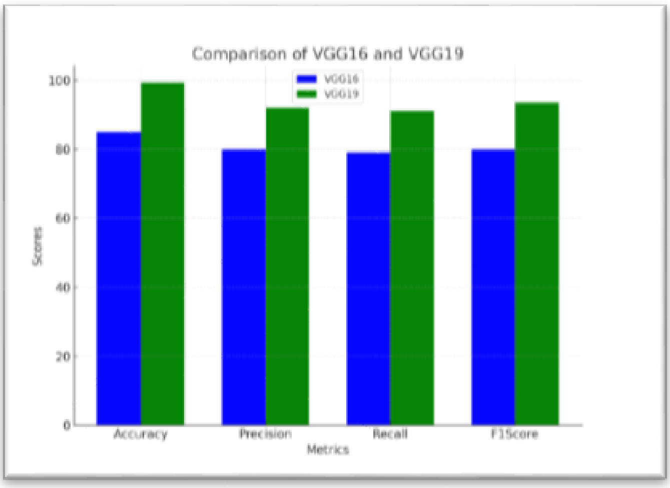

Fig 8.2 Comparison of VGG16 and VGG19

Fig 8.2 Shows the Comparison of Metrics between Vgg16 and Vgg19

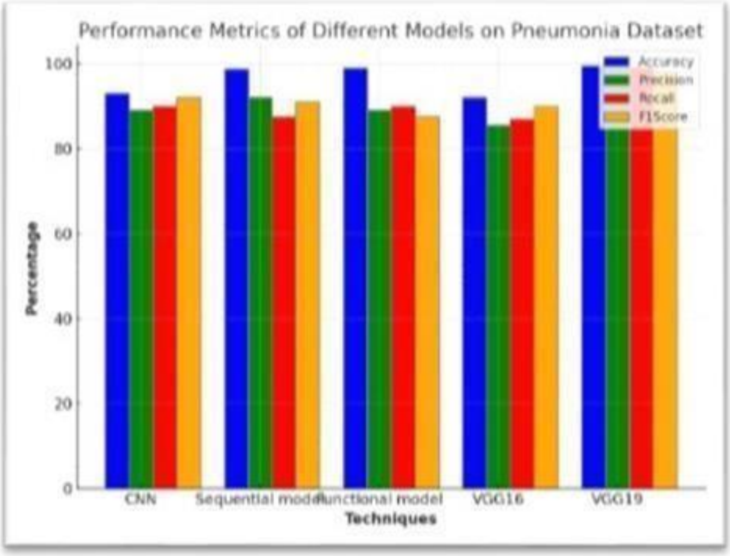

Fig 8.3. Comparison of metrics in Pnemonia_dataset

Fig 8.3 represents the barchart chart showing the Performance Metrics of Different Models on the Pneumonia Dataset**.**

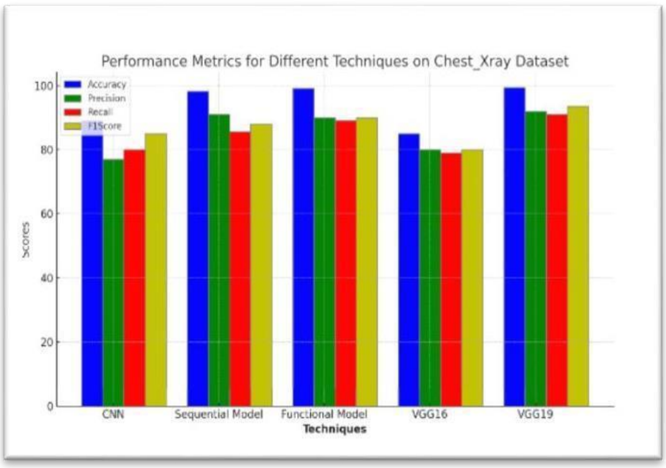

Fig 8.4. Comparison of metrics in chestxray_dataset

Fig 8.4 represents the barchart chart showing the Performance Metrics of Different Models on the Pneumonia Dataset

The analysis from the above graphs comparing accuracy, precision, recall, and F1-score of different algorithms reveals that VGG19 is the best predictive model of pneumonia since it very well balances precision with accuracy and recall, thus making it the most dependable for real application in hospitals. Its sensitivity and specificity are high and will ensure a good prediction of pneumonia with errors that are the least in the clinic.

9. TEST CASES

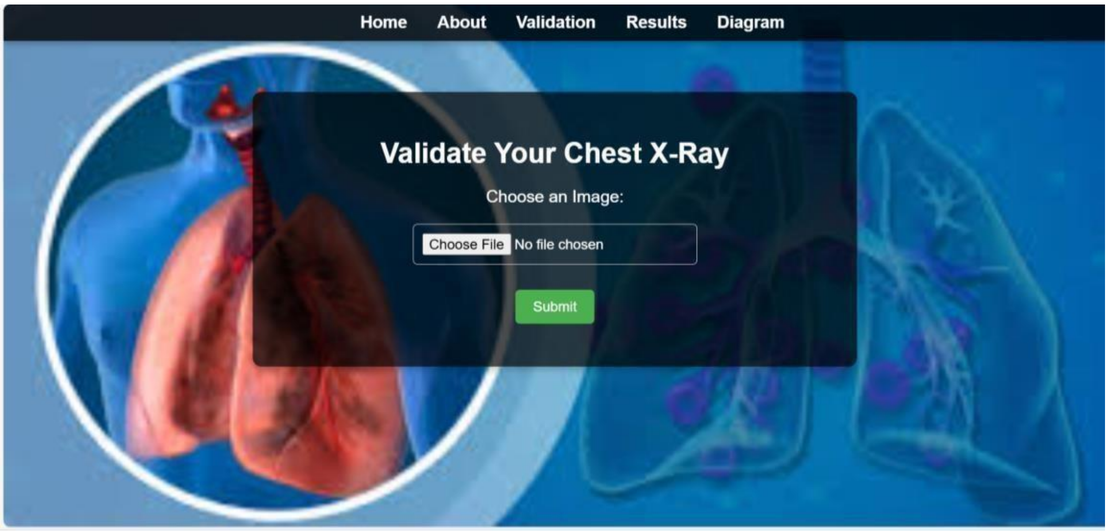

FIG 9.1 VALIDATION SCREEN

Fig 9.1 is the Validation Screen. This is the page a file upload form where users can upload a chest X-ray image for validation.

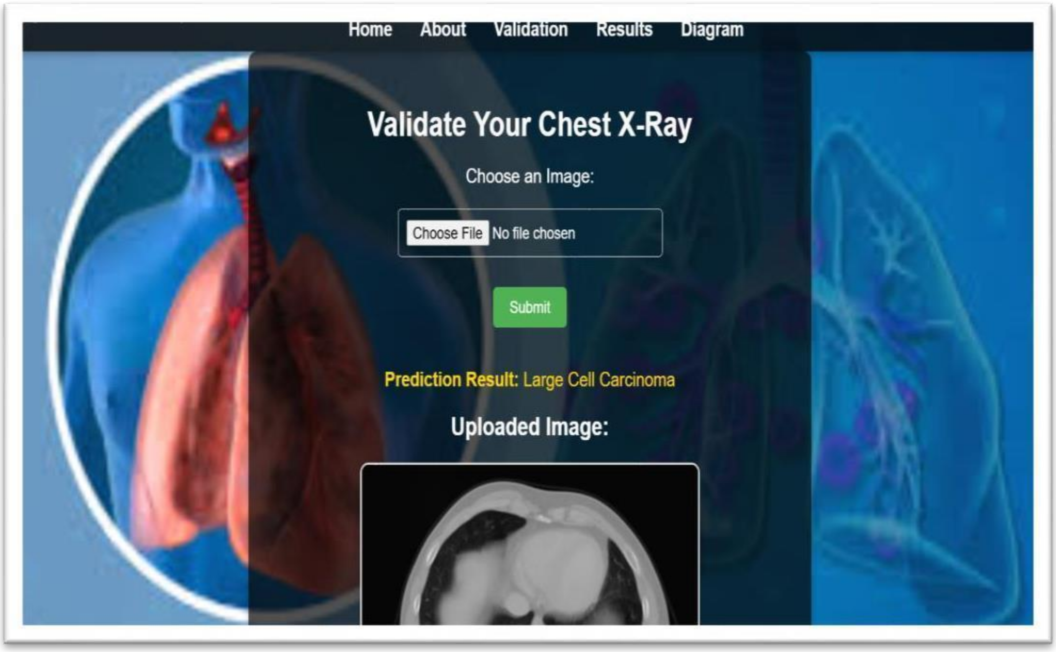

FIG 9.2 OUTPUT SCREEN

Fig 9.2 is an output screen designed for chest X-ray analysis and disease prediction**.**

10. USER INTERFACE

Fig10.1Home Screen

Fig 10.1 describes the main page of web application and it has all components

Fig 10.2 Results Screen

Fig 10.2 Shows the results of metrics and accuracy between the models

11. CONCLUSION

This project represents a comprehensive and forward-thinking approach to addressing the challenges of pneumonia detection through advanced deep learning models in the medical imaging domain. The primary objectives of this study revolve around achieving high accuracy, precision, recall, and F1-scores while minimizing errors in detecting pneumonia, which is critical for real-world clinical applications.

In an era where the integration of AI in healthcare is rapidly transforming diagnostic practices, it is essential to deploy models that can reliably identify cases while maintaining a balance between sensitivity and specificity. VGG19 stands out as the most effective model in this regard, achieving an accuracy of 99.4%, a precision of 99.1%, a recall of 99%, and an F1-score of 98. These metrics demonstrate its ability to detect pneumonia cases with remarkable reliability, ensuring minimal false positives and false negatives. This makes VGG19 a dependable solution for supporting medical professionals in delivering accurate and timely diagnoses.

By leveraging deep learning architectures like VGG19, this project not only enhances the precision of pneumonia detection but also emphasizes the importance of robust, scalable, and clinically applicable AI solutions. As healthcare systems continue to adopt AI technologies, such advancements serve as critical steps toward improving diagnostic efficiency, reducing errors, and promoting better patient outcomes. This initiative exemplifies the potential of AI in transforming healthcare delivery, ensuring that technological progress aligns with the overarching goal of enhancing trust, reliability, and care within the medical field.

12. FUTURE SCOPE

In future research, the advancements presented in this study can be further extended to enhance their applicability and impact in the field of medical imaging and disease detection. One significant avenue is the integration of these models into real-time diagnostic systems within hospitals and clinics, enabling healthcare professionals to receive immediate and accurate predictions, thereby streamlining the diagnostic process and reducing delays. Additionally, incorporating multimodal data, such as combining chest X-rays with patient histories, lab results, or other imaging modalities like CT and MRI scans, could improve the robustness and accuracy of predictions, creating comprehensive diagnostic solutions. The architecture can also be adapted to detect a broader range of medical conditions, including lung cancer, tuberculosis, or chronic obstructive pulmonary disease, ensuring its versatility across diverse applications.

Moreover, employing federated learning techniques would allow models to train on distributed datasets without compromising patient privacy, leveraging data from multiple institutions to develop more generalized and robust systems. Efforts to optimize computational efficiency and reduce latency can make these systems more accessible in low-resource settings, particularly in rural hospitals with limited infrastructure. Developing lightweight versions of the models for deployment on mobile and edge devices could further enable point-of-care diagnostics in remote areas. Integrating explainable AI techniques is another critical step, ensuring transparency and trust in model predictions for medical professionals and fostering clinical adoption. Collaborative clinical trials involving medical practitioners can validate the models and provide real-world feedback for further fine-tuning. By addressing these areas, future research can significantly contribute to advancing AI in healthcare, promoting efficiency, equity, and reliability in medical diagnostics.

13. REFERENCES

[1] Jasmine Pemeena Priyadarsini, M., Kotecha, K., Rajini, G. K., Hariharan, K., Utkarsh Raj, K., Bhargav Ram, K., ... & Pandya, S. (2023). Lung diseases detection using various deep learning algorithms. Journal of healthcare engineering, 2023(1), 3563696.

[2] Mahajan, A., Mate, S., Kulkarni, C., & Sawant, S. (2024). Predicting Lung Disease Severity via Image-Based AQI Analysis using Deep Learning Techniques. arXiv preprint arXiv:2405.03981.

[3] Tahmooresi, M., Afshar, A., Rad, B. B., Nowshath, K. B., & Bamiah, M. A. (2018). Early detection of breast cancer using machine learning techniques. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 10(3-2), 21-27.

[4] Bhookya, J. (2023, August). Examine Lung Disorders and Disease Classification Using Advanced CNN Approach. In 2023 3rd Asian Conference on Innovation in Technology (ASIANCON) (pp. 1-6). IEEE.

[5] Raju, V. P., Srinivasulu, S., Vishnu, P., Raj, J. R., Gowri, S., & Jabez, J. (2022, May). A Contemporary Technique for Lung Disease Prediction using Deep Learning. In 2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC) (pp. 970-975). IEEE.

[6] Bukhari, S. K. H., & Fahad, L. (2022, November). Lung disease detection using deep learning. In 2022 17th International Conference on Emerging Technologies (ICET) (pp. 154-159). IEEE.

[7] Rajaselvi, M., Sanjith, J., Koshy, S., & Niranjan, G. M. (2022, April). A survey on lung disease diagnosis using machine learning techniques. In 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE) (pp. 01-04). IEEE.

[8] Mahajan, A., Mate, S., Kulkarni, C., & Sawant, S. (2024). Predicting Lung Disease Severity via Image-Based AQI Analysis using Deep Learning Techniques. arXiv preprint arXiv:2405.03981.

[9] Gunasinghe, A. D., Aponso, A. C., & Thirimanna, H. (2019, March). Early prediction of lungdiseases. In 2019 IEEE 5th International Conference for Convergence in Technology (I2CT) (pp. 1-4). IEEE.

[10] Bharati, S., Podder, P., & Mondal, M. R. H. (2020). Hybrid deep learning for detecting lung diseases from X-ray images. Informatics in Medicine Unlocked, 20, 100391.

[11] Sen, I., Hossain, M. I., Shakib, M. F. H., Imran, M. A., & Al Faisal, F. (2020). In

depth analysis of lung disease prediction using machine learning algorithms. In Machine Learning, Image Processing, Network Security and Data Sciences: Second International Conference, MIND 2020, Silchar, India, July 30-31, 2020, Proceedings, Part II 2 (pp. 204-213). Springer Singapore.

[12] Sabry, A. H., Bashi, O. I. D., Ali, N. N., & Al Kubaisi, Y. M. (2024). Lung disease recognition methods using audio-based analysis with machine learning. Heliyon.

[13] Kavitha, M., Vidhya, K., Sreeja, S., Roopashri, G., & Muhil, P. (2024). Automated lung disease detection, classification and prediction using RNN framework. In E3S Web of Conferences (Vol. 491, p. 03015). EDP Sciences.

[14] Sharmila, K. S., Revathi, S. T., & Sree, P. K. (2023, January). Convolution Neural Networks based lungs disease detection and Severity classification. In 2023 International Conference on Computer Communication and Informatics (ICCCI) (pp. 1-9). IEEE.

[15] Saravanan.%20M,%20Kumar,%20M.,%20Vijay,%20D.,%20Gayathri.%20K,%20& %20Jenifar.%20A.%20(2022).%20Chest%20XRay%20based%20Lung%20Disease %20Detection%20using%20Convolutional%20Neural%20Network.%20In%20CV MLH/WTEK%20(pp.%2026-32).

[16] Ghaskadvi, M., Khochare, S., Gonsalves, R., & Dhamanskar, P. (2022). Pneumonia and diabetic retinopathy detection using deep learning algorithm. In Sentimental Analysis and Deep Learning: Proceedings of ICSADL 2021 (pp. 155-175). Springer Singapore

Optimizing the Powerhouse: Fine-Tuning CNNs for Superior Lung Disorder Detection

Rama Krishna Eluri [*], Pentyala Tanuja, M.Venkat Rao, Vutla Lavanya and M. Mokshagna
Computer Science And Engineering, Narasaraopeta Engineering College, Narasaraopeta, India

**ramakrishnaephd7@gmail.com*

2024 First International Conference on Innovations in Communications, Electrical and Computer Engineering (ICICEC) | 979-8-3503-7651-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICICEC62498.2024.10808691

***Abstract*— Present work involves the use of deep learning models, in particular, Convolutional Neural Networks, to detect critical diseases of the lungs such as pneumonia, tuberculosis, and lung cancer through the analysis of chest X-ray and CT scan images. CNN has emerged as a robust tool for the analysis of medical images due to their natural ability to automatically extract hierarchical features, hence making them ideal for complicated tasks such as disease detection. Well-known architectures in this work are VGG16 and VGG19; they are combined with specially designed sequential and functional models to solve this task of recognizing those diseases. These models are trained on open-source data that will provide the model with real-world medical images for it to learn from diverse cases of lung abnormalities. One important enhancement in pursuit of model performance and robustness in this work is the employment of augmentation techniques. Since, in most cases, labeled medical data comes in a limited amount, augmentation is an important way of artificially increasing the effective size of the training set to allow the models to generalize better on unseen data. The images were rescaled, shear transformed, and flipped horizontally to simulate different orientations and variations that can arise during real-world X-ray and CT scans. This will also make the models less sensitive to changes in orientation, scale, or noise of an image and enhance their capability of detecting diseases more precisely in different datasets. The results of the study show that early detection by deep learning models has improved a lot by incorporating advanced CNN architectures and augmentation strategies. Those customized sequential and functional models performed very well, especially in the classification of lung diseases with very high precision. This is indeed important for early diagnosis because early identification of diseases like pneumonia, tuberculosis, and lung cancer can lead to better treatment plans and improved patient outcomes. Moreover, the study highlights how deep learning models may represent the most viable tool in clinical practice to provide not only accurate but timely diagnostic information that would support medical professionals most proficiently.**

Keywords*—CNN; VGG19; VGG16; Functionalmodel; Sequential Model.

I. Introduction

The lungs are very important organs of the respiratory system, which contribute to exchanging oxygen and carbon dioxide. However, the lungs are susceptible to a number of diseases which may impair their function. Such respiratory diseases may range from the common cold and influenza, which are relatively mild conditions causing temporary discomforts, to life-threatening conditions like pneumonia, tuberculosis. The global burden of respiratory disease is enormous. According to estimation through the Worldwide Conference of Respiratory Societies, in any particular year, an approximate 10.4 million people suffer from tuberculosis worldwide, accounting for the death toll of 1.4 million annually [1]. Lung cancer is also such a killer disease; more than 1.6 million deaths take place in a year due to this disease alone worldwide. Moreover, pneumonia was solely responsible, especially in infants, for over 1.23 million deaths of the children under five years of age, according to a study published by the Bloomberg School of Public Health at John Hopkins.

The only method of dealing with such diseases is early detection, and this assists in increased survival rates and a reduction in the death rates. Some of the most utilized medical imaging for diagnosis include X-ray scans of the chest and CT scan images. These images require interpretation from skilled radiologists. The number of health care providers, such as doctors, may be low with regard to rural areas; for example, in certain places-CHCs-76.1 percent of doctors are lacking there.

Recent advances in AI, and more so deep learning in the previous years, have achieved a lot in automating and improving the accuracy in the analysis of medical images. CNNs are one of the deep learning models that have indeed shown outstanding results for the class of the image classification task, hence aptly. The different pre-trained deep learning models, such as CNN, VGG16, and VGG19 used in this work, are applied for the detection and classification of lung diseases by taking chest X-ray and CT scan images. In this work, publicly available data have been used, and augmentation has been performed to increase the strength of these models and generalize their capability. The performances of the custom sequential and functional models that show high potential in clinical settings are compared. Indeed, the results obtained from this study did prove that deep learning models do stand a chance of being helpful in early lung diseases detection, with the purpose of assisting healthcare professionals in making timely and early precise diagnosis. This work adds to the accumulation of evidence on AI integration into medical practice in general and radiology in particular. [1]

Fig .1. Lung Disorder

Fig-1 is the image based on focusing on the detection of lung diseases with deep learning technology. The image focuses on highlighting the lungs with an X-ray and CT scans with digital analytic elements that symbolize the detection process. Based on the research here are some of the Contributions below:

- The datasets used here are Pneumonia Chest X-ray Dataset and Chest CT-Scan for the detection of lung diseases, which may be pneumonia or COVID-19.
- The Deep Learning models were used in the research: CNN models-VGG16, VGG19, and our sequential and functional models for classification.
- Sequential and functional models have been designed, optimized, and tested in order to see which one of the two does a better job regarding the detection of lung diseases.
- The augmented image techniques involved the flipping, rotation, and scaling of images to make them larger and robust.
- The fine-tuned VGG16 and VGG19 models, sequential models, and functional models on medical imaging datasets for early disease diagnosis.
- On carrying out the performance evaluation, it came out that the models evaluated the accuracy, precision, recall, and F1-score, where VGG19 and the functional models were among the best.

II. RELATED WORK

Liu et al. [1] models of tuberculosis detection using CNN-SVM. However, issues arose in terms of low accuracy. Jaiswal et al. extracted X-ray features using Mask RCNN, which was a pretty heavy computation. Elshennawy and Ibrahim developed pneumonia detection models involving CNNs and pre-trained models like ResNet152v2 and MobileNetV2. However, the challenges they encountered were computational complexity. Li et al. [2] that Random Forest Regression and Gradient Boosting are the best methods for air quality prediction. Rajaselvi et al. used CNN and transfer learning in the diagnosis of lung diseases. Agarwal et al. presented a comparison done with several machine learning models, from which it was obtained that the best performance in the classification of lung cancer was achieved using Random Forest, outperforming other algorithms such as Support Vector Machines. Xie et al.[3] used deep learning to classify CT images into benign and malignant nodules. Wu and Zhao used EDM AI for differentiating early small cell lung cancer. Anifah et al. used Gray-Level Co-occurrence Matrices in detecting lung cancer and achieved a moderate level of accuracy. Deshpande used SVM and image fusion to enhance the detection performance. Bhukya et al. [4](2023) propose advanced CNNs for the classification of lung diseases, mainly COVID-19, pneumonia, tuberculosis, and normal cases. The classification accuracy [5] obtained in this study is 94%, using large datasets and CNN architecture to improve diagnosis, hence highlighting the potential in the improvement of medical image analysisRaju et al. [6] present the prediction of lung diseases from chest X-ray images using a CNN-based deep learning model. It highlights the use of machine learning in support of the diagnosis of diseases such as COVID-19, pneumonia, and tuberculosis with an accuracy of 91%, showing how important CNN would be for diagnostics. Some works have used CNNs for the detection of lung diseases, which seem promising-include ResNet, Dense Net, and VGG19. Previous works focused on the classification of single diseases based on medical images, particularly pneumonia, tuberculosis, and lung cancer, using X-rays and computed tomography scans [7]. However, in most of these studies, the datasets were either too small or imbalanced, which, eventually, showed up in the accuracy and generalization aspect . [8] author explains in detail the integrated approach to the prediction of air quality by determining the seriousness of the lung disease through image-based AQI analysis. Techniques involved in the system are Neural Network, KNN, and SVC models, out of which very high accuracy was observed during testing.[9] covers early prediction of lung diseases by applying machine learning and deep learning techniques, including CNN and Capsule Networks on [15]X-ray data to improve early detection accuracy for conditions such as asthma, COPD, and lung rapid rise in the incidence of lung diseases, including COPD, asthma, and pneumonia caused by environmental changes, pollution, and smoking requires the design [11][12]and development of diagnostic systems with efficiency. Large datasets, such as chest X-rays, therefore, are increasingly subjected to the application of ML and DL methods for lung disease prediction[13][14] The application of Machine learning in classifying disease based on lung sounds produces good performance but still needs a wider scope of studies[16]. learning improves the rate of automation, interprets non-structured information and performs with greater precision AI related tasks and forecasts [17] chest X-ray-based lung disease detectors using several types of deep learning and optimization techniques[10][18].

III. PRELIMINARIES

A. *Convolutional Neural Network Model*

A CNN is a type of deep learning model that is mostly used in the fields of classification and image recognition**.** It makes use of convolutional layers to achieve feature hierarchies in space in an automatic manner using filters with backpropagation. Fully connected, pooling, and convolutional layers make to a CNN's general architecture. It does exceptionally well in the processing tasks of images and videos because it effectively catches the features of an object in space.

B. *Functional Model*

Another important feature of deep learning frameworks like Keras is the Functional API, which allows more flexibility than a Sequential model. This could be used to make complex architectures such as multi-input models, multi-output models, shared-weight layers, or even residual connections. Unlike in Sequential, where layers can only be stacked linearly, the Functional API allows for directed acyclic graph-like structures, making for more complex models.

C. *Sequential Model*

The 'Sequential' model represents a simpler way of building a model wherein layers are added to each other in continuum. In this model, every layer has only one input and one output; hence, it is quite handy for uncomplicated models like CNN. There isn't any multiple input or even branching layers in this architecture. However, that makes it less flexible since, compared to the Functional API, for complicated architectures, this probably is not what one would want to use.

D. *VGG16*

Oxford's Visual Geometry Group created the 16-layer deep Convolutional Neural Network known as VGG16. Thirteen convolutional layers and three fully linked layers utilizing tiny 3x3 filters made up this model. A nice thing about this concept is that, since it uses identical filter sizes all over, this concept makes it easy for user understanding and implementation. VGG16 is widely used for image classification tasks and has achieved high performance in benchmarks like ImageNet; though, it comes rather expensively due to depth and the number of parameters.

E. *VGG19*

VGG19 is the extension of VGG16, comprising in total 19 layers: 16 convolutional and 3 fully connected. While VGG19 operates with the same characteristics as VGG16-it also uses small 3x3 filters-the presence of more convolutional layers makes it much deeper, hence much more capable of learning complex features from images. Accordingly, the computational burden and memory significantly rise with the fact that this network has become much deeper compared to VGG16. VGG19 has found extensive use in image classification, transfer learning, and feature extraction.

IV. PROPOSED METHODOLOGY

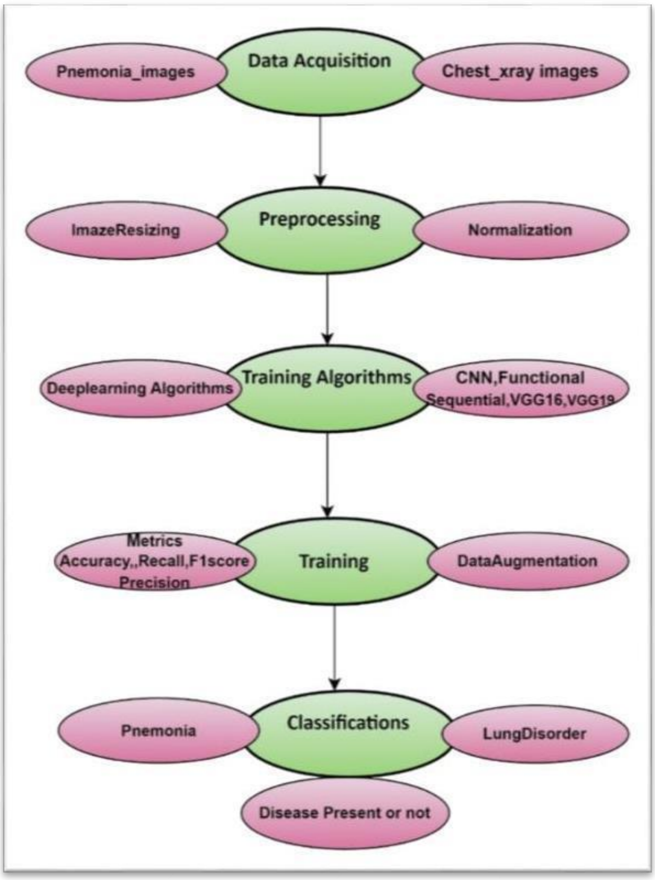

Fig. 2. Deep learning workflow for Lung Disorder Detection

Fig 2 describes the procedure for diagnosing pneumonia and lung diseases using deep learning: from the acquisition of data to the classification outcome Below is the explanation of diagram

A . Data Acquisition

Pneumonia Images and Chest X-ray Images: Data is obtained by the system from chest X-ray images. These images are taken from the datasets [17][18]. Such data would be crucial for lung diseases like pneumonia, tuberculosis, and lung cancer. Data gathered becomes the input to train the model. The datasets are prepared in such a way that they include great variation in patient demographics, disease stages, and image quality to support the model in generalizing well. These, after proper labeling, form the input for training the model in finding and classifying abnormalities in the lungs.

B. *Preprocessing*

Resizing will be standardization of size dimensions of images to decrease computation and keep vital attributes for coherence with VGG16/VGG19 architectures. Normalizing pixel values to scale between 0-1 accelerates convergence speeds and avoids model bias that could occur due to differing pixel intensities or dimension, thus improving models during preprocessing.

C. Training Algorithms

- Convolutional Neural Networks: CNNs perform feature extraction and can learn from the medical images possessing patterns that will be informative in classification tasks.
- Pre-Trained Models: Pre-trained VGG16 and VGG19 models on large image datasets such as ImageNet. Transfer learning makes use of the weights learned for the detection of lung diseases.
- Convolutional Neural Networks: These are a category of deep neural networks designed to operate upon image data, making them quite suitable for medical imagery regarding different types of lung diseases.
- The sequential model is a deep learning model architecture that stacks layers linearly. This, in turn, makes its implementation very easy. It is perfect for cases where the model should go in a straightforward, linear direction-the VGG16 and VGG19 among others.
- Functional Model in Keras: This model gives one more liberty when one wants to build some very complex neural networks with branch-out and merging layers. An example could be an architecture where there isn't a strictly linear relationship between layers.

F. Training

- Metrics During training, accuracy, recall, F1-score, and precision are used as metrics for performance tracking. These metrics represent the capability of this model to learn and generalize on unseen data.
- Data Augmentation The augmentations performed are flipping, rotation, and shearing. This is to avoid overfitting, and it hopefully will make the model robust. Data augmentation artificially increases the size of a training dataset by adding variability in training data, simulating different view angles and conditions.

E. Classifications

The model was used to classify or differentiate between healthy lungs, pneumonia-affected, and other lung disorders. The system output assists the physician in diagnosing at an early stage through a clinical decision support system. It can indicate the presence or absence of disease.
VGG16: slightly faster with fewer parameters, it generalized better on smaller-sized datasets.
VGG19: While the deeper model was generalizing better in the case of larger datasets, still it requires more careful tuning of the learning rate and regularization techniques, such as dropout.

V. RESULTS AND DISCUSSIONS

A. Description of Dataset

TABLE I. PNEMONIA_DATASET [17]

Attribute	**Details**
Dataset Name	Pneumonia Dataset - Chest X-ray
Source	Kaggle (Paul Mooney)
Total Images	5,856
Class Distribution	Normal: 1,583 Pneumonia: 4273
Subcategories	Bacterial and Viral
Resolution	Variable (Standardized during pre-processing)
Pre-processing	Resizing, Normalization
Ages	Pediatric and adult cases
Diversity	Broad age range and geographic diversity

Table-I describes the Pneumonia Chest X-ray dataset, class distribution, resolution, and also preprocessing methods used.

TABLE II. CHEST_XRAY DATASET [18]

Attribute	**Details**
Dataset Name	Chest CT-Scan Images Dataset
Source	Kaggle (Mohamed Hany)
Total Images	907
Class Distribution	COVID-19, Viral Pneumonia, Normal
Subcategories	None
Resolution	Standard CT scan resolution
Pre-processing	Segmented for disease detection
Ages	Not specified
Diversity	Includes various lung disease images

Table-II describes the Chest X-ray dataset, class distribution, resolution, and also preprocessing methods used.

B. Performance of Metric[1]

1) *Accuracy*: It is the proportion of correct predictions, both positive and negative.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

2) *Precision:* It will measure the proportion of correct positive predictions from all the predicted positives.

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

3) *Recall:* Recall measures the ability to spot all actual positive cases.

$$Recall = \frac{TP}{TP+FP} \quad (3)$$

4) *F1-Score:* This F1 score is the harmonic mean of precision and recall, balancing their trade-offs.

$$F1 = 2\times\frac{Precision\times Recall}{Precision+Recall} \quad (4)$$

TABLE III. PNEMONIA_DATASET METRICS

Dataset	Techniques	Accuracy (%)	Precision (%)	Recall (%)	F1Score (%)
Pneumonia dataset	CNN	93.0	89.0	90.0	92.0
	Sequential model	98.7	92.0	87.5	91.0
	Functional model	98.9	89.0	90.0	87.5
	VGG16	92.0	85.5	87.0	90.0
	VGG19	99.4	99.1	99.0	98.0

Table-III is a comparison table of different deep learning techniques on the Pneumonia dataset in terms of accuracy, precision, recall, and F1-score.

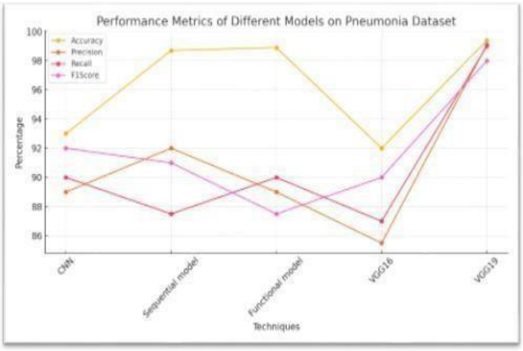

Fig. 3. Performance of metrics in Pnemonia_dataset

Fig. 3 and Fig. 4 show the performance metrics (accuracy, precision, recall, and F1 score) of different models: CNN, Sequential, Functional, VGG16 and VGG19 on the pneumonia dataset. Fig. 3 demonstrates these metrics in the format of a line graph where every metric for the models is shown in a different color to point out the differences across various models. Fig. 4, a bar chart for the same metrics, where each of them is depicted in a different color, and provides a comparative view, where in which model performance can be judged on the accuracy, precision, recall, F1-score.

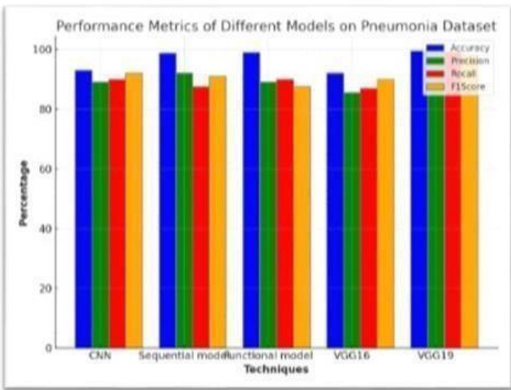

Fig. 4. Performance of metrics in bargraph

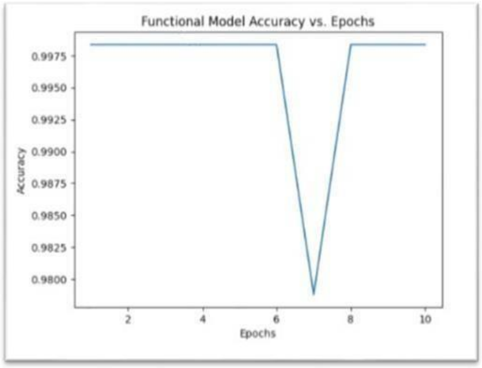

Fig 5. Functional model accuracy vs epoch

Fig5 is the plot of a line graph that reads Functional Model Accuracy vs. Epochs, showing fluctuations in accuracy over 10 epochs with a dip around epoch 5.

Fig.6 is a bar chart comparing the accuracy of the two models, namely, SequentialModel and Functional Model. Interestingly, the bars for both appear almost of the same height, which means their accuracy levels are somewhat equal or very close to each other.

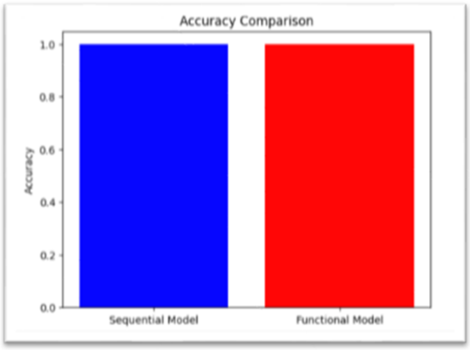

Fig. 6. Accuracy comparison

TABLE IV. CHEST_XRAY_DATASET METRICS

Dataset	Techniques	Accuracy	Precision	Recall	F1Score
Chest_x-ray dataset	CNN	89	77	80	85
	Sequential Model	98.2	91	85.6	88
	Functional Model	99.1	90	89	90
	VGG16	85	80	79	80
	VGG19	99.3	92	91	93.5

Table-IV is a comparison table of different deep learning techniques on the Chest_xray dataset in terms of accuracy, precision, recall, and F1-score.

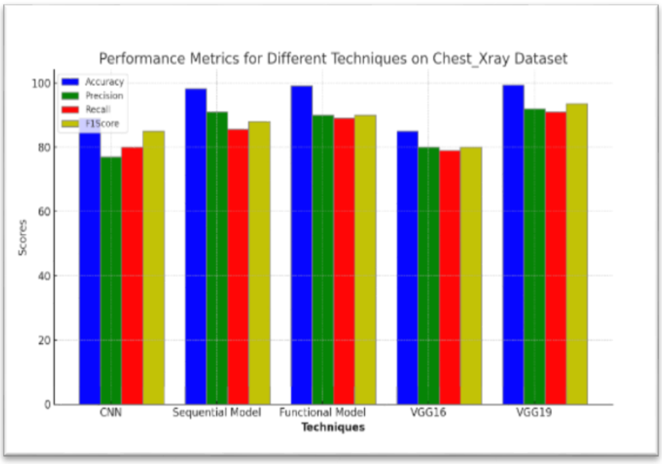

Fig.7. performance metrics on Chest_xray dataset

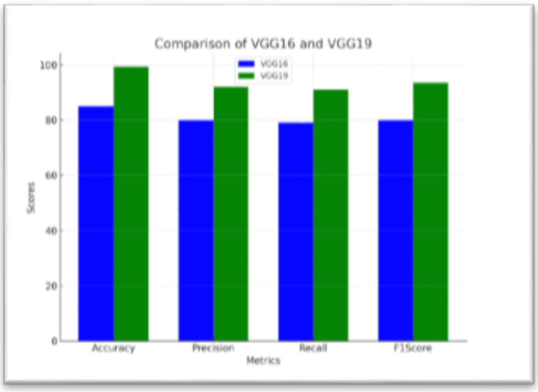

Fig. 8. Comparison of VGG16 and VGG19

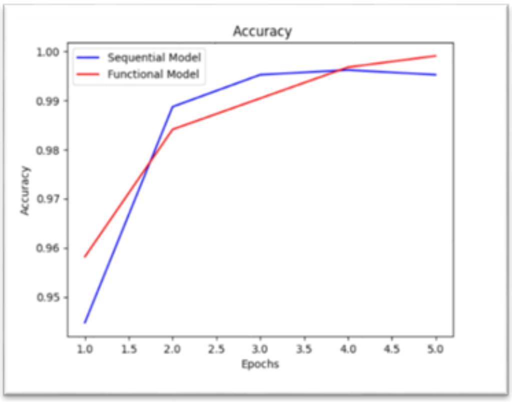

Fig. 9. Sequential model vs functional

VI. CONCLUSION

VGG19 pneumonia dataset achieves an accuracy of 99.4% and a F1-score of 98 with a fair trade-off between precision, which is 99.1%, and recall of 99%. That is, VGG19 prevents both false positives and negatives, making it very reliable in giving a valid conclusion about the actual case of pneumonia. While the Functional Model, with accuracy 98.9%, has a relatively low F1 score, which is 87.5, this does indicate a certain tradeoff between precision and recall, not best suited in the medical field due to equal importance of both. CNN and VGG16 are accurate at 93% and 92%, respectively. However, the F1-scores obtained for them are relatively lower than those of VGG19.

Overall, VGG19 is the best predictive model of pneumonia since it very well balances precision with accuracy and recall, thus making it the most dependable for real application in hospitals. Its sensitivity and specificity are high and will ensure a good prediction of pneumonia with errors that are the least in the clinic.

REFERENCES

[1] Jasmine Pemeena Priyadarsini, M., Ketan Kotecha, G. K. Rajini, K. Hariharan, K. Utkarsh Raj, K. Bhargav Ram, V. Indragandhi, V. Subramaniyaswamy, and Sharnil Pandya. "Lung diseases detection using various deep learning algorithms." *Journal of healthcare engineering* 2023, no. 1 (2023): 3563696

[2] Mahajan, Anvita, Sayali Mate, Chinmayee Kulkarni, and Suraj Sawant. "Predicting Lung Disease Severity via Image-Based AQI Analysis using Deep Learning Techniques." *arXiv preprint arXiv:2405.03981* (2024).

[3] Tahmooresi, Maryam, A. Afshar, B. Bashari Rad, K. B. Nowshath, and M. A. Bamiah. "Early detection of breast cancer using machine learning techniques." *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)* 10, no. 3-2 (2018): 21-27.

[4] Eluri, Rama Krishna, and Nagaraju Devarakonda. "Feature selection with a binary flamingo search algorithm and a genetic algorithm." *Multimedia Tools and Applications* 82, no. 17 (2023): 26679-26730.

[5] Bhookya, Jailsingh. "Examine Lung Disorders and Disease Classification Using Advanced CNN Approach." In *2023 3rd Asian Conference on Innovation in Technology (ASIANCON)*, pp. 1-6. IEEE, 2023.

[6] Raju, V. Pradyotan, Senduru Srinivasulu, P. Vishnu, Jeberson Retna Raj, S. Gowri, and J. Jabez. "A Contemporary Technique for Lung Disease Prediction using Deep Learning." In *2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, pp. 970-975. IEEE, 2022.

[7] Eluri, Rama Krishna, and Nagaraju Devarakonda. "Binary golden eagle optimizer with time-varying flight length for feature selection." *Knowledge-Based Systems* 247 (2022): 108771.

[8] Arifin, Jaenal, Tri Arief Sardjono, and Hendra Kusuma. "Enhancing YAMNet Model for Lung Sound Classification to Identify Normal and Abnormal Conditions." In *2024 International Conference on Smart Computing, IoT and Machine Learning (SIML)*, pp. 117-123. IEEE, 2024.

[9] Rajaselvi, Mercy, J. Sanjith, Samuel Koshy, and G. M. Niranjan. "A survey on lung disease diagnosis using machine learning techniques." In *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, pp. 01-04. IEEE, 2022.

[10] Eluri, Rama Krishna, and Nagaraju Devarakonda. "Chaotic binary pelican optimization algorithm for feature selection." *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 31, no. 03 (2023): 497-530.

[11] Mahajan, Anvita, Sayali Mate, Chinmayee Kulkarni, and Suraj Sawant. "Predicting Lung Disease Severity via Image-Based AQI Analysis using Deep Lear ning Techniques." *arXiv preprint arXiv:2405.03981* (2024).

[12] Prakash, Kolla Bhanu, Rama Krishna Eluri, Nalluri Brahma Naidu, Sri Hari Nallamala, Pragyaban Mishra, and P. Dharani. "Accurate hand gesture recognition using CNN and RNN approaches." *International Journal* 9, no. 3 (2020).

[13] Gunasinghe, Anuradha D., Achala C. Aponso, and Harsha Thirimanna. "Early prediction of lung diseases." In *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*, pp. 1-4. IEEE, 2019.

[14] Bharati, Subrato, Prajoy Podder, and M. Rubaiyat Hossain Mondal. "Hybrid deep learning for detecting lung diseases from X-ray images." *Informatics in Medicine Unlocked* 20 (2020): 100391

[15] Krishna, Eluri Rama, Nagaraju Devarakonda, Mohammad Yahya H. Al-Shamri, and Durgam Revathi. "A novel hybrid clustering analysis based on combination of k-means and pso algorithm." In *Data Intelligence and Cognitive Informatics: Proceedings of ICDICI 2021*, pp. 139-150. Singapore: Springer Nature Singapore, 2022.

[16] Sen, Ishan, Md Ikbal Hossain, Md Faisal Hossan Shakib, Md Asaduzzaman Imran, and Faiz Al Faisal. "In depth analysis of lung disease prediction using machine learning algorithms." In *Machine Learning, Image Processing, Network Security and Data Sciences: Second International Conference, MIND 2020, Silchar, India, July 30-31, 2020, Proceedings, Part II 2*, pp. 204-213. Springer Singapore, 2020.

[17] Sabry, Ahmad H., Omar I. Dallal Bashi, NH Nik Ali, and Yasir Mahmood Al Kubaisi. "Lung disease recognition methods using audio-based analysis with machine learning." *Heliyon* (2024).

[18] Eluri, Rama Krishna, and Nagaraju Devarakonda. "A concise survey on solving feature selection problems with metaheuristic algorithms." In *International Conference on Advances in Electrical and Computer Technologies*, pp. 207-224. Singapore: Springer Nature Singapore, 2021.

DG4

ORIGINALITY REPORT

12%	7%	10%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	"Multi-Strategy Learning Environment", Springer Science and Business Media LLC, 2024 Publication	1%
2	Pawan Singh Mehra, Dhirendra Kumar Shukla. "Artificial Intelligence, Blockchain, Computing and Security - Volume 2", CRC Press, 2023 Publication	1%
3	Arvind Dagur, Karan Singh, Pawan Singh Mehra, Dhirendra Kumar Shukla. "Artificial Intelligence, Blockchain, Computing and Security", CRC Press, 2023 Publication	1%
4	Yuchen Huang, Jingwen Liu, Xuanyi Xiang, Pan Wen, Shiyuan Wen, Yanru Chen, Liangyin Chen, Yuanyuan Zhang. "Malware Identification Method in Industrial Control Systems Based on Opcode2vec and CVAE-GAN", Sensors, 2024 Publication	1%

5	www.mdpi.com Internet Source	1%
6	Ankur Gupta. "Next Generation Computing and Information Systems", CRC Press, 2024 Publication	1%
7	www.expresshealthcaremgmt.com Internet Source	1%
8	"Cognitive Computing and Cyber Physical Systems", Springer Science and Business Media LLC, 2024 Publication	<1%
9	conf.papercept.net Internet Source	<1%
	Varun Vekaria, Raj Gandhi, Bhargavee Chavarkar, Hetvi Shah, Chetashri Bhadane, Poonam Chaudhari. "Identification of Kidney Disorders in Decentralized Healthcare Systems through Federated Transfer Learning", Procedia Computer Science, 2024 Publication	<1%
	M. Rubaiyat Hossain Mondal, Utku Kose, V. B. Surya Prasath, Prajoy Podder, Subrato Bharati, Joarder Kamruzzaman. "Emerging Technologies for Combatting Pandemics - AI, IoMT, and Analytics", CRC Press, 2022 Publication	<1%

12	Submitted to Thomas More Hogeschool Student Paper	<1%
13	Submitted to University of Hull Student Paper	<1%
14	www.hindawi.com Internet Source	<1%
15	fastercapital.com Internet Source	<1%
16	www.alternet.org Internet Source	<1%
17	ijritcc.org Internet Source	<1%
18	publikationen.ub.uni-frankfurt.de Internet Source	<1%
19	res.mdpi.com Internet Source	<1%
20	Submitted to uwe Student Paper	<1%
21	pesquisa.bvsalud.org Internet Source	<1%
22	www.ijert.org Internet Source	<1%
	"IoT Based Control Networks and Intelligent Systems", Springer Science and Business	<1%

Media LLC, 2024
Publication

Chhabi Rani Panigrahi, Bibudhendu Pati, Mamata Rath, Rajkumar Buyya. "Computational Modeling and Data Analysis in COVID-19 Research", CRC Press, 2021
Publication
<1%

Jailsingh Bhookya. "Examine Lung Disorders and Disease Classification Using Advanced CNN Approach", 2023 3rd Asian Conference on Innovation in Technology (ASIANCON), 2023
Publication
<1%

Kennis Chan. "Testing and Measurement: Techniques and Applications - Proceedings of the 2015 International Conference on Testing and Measurement Techniques (TMTA 2015), 16-17 January 2015, Phuket Island, Thailand", CRC Press, 2015
Publication
<1%

Raju Bhukya, Banothu Ramji. "Chapter 5 Prediction and Early Detection of Various Diseases Risk by Using Machine Learning Techniques", Springer Science and Business Media LLC, 2024
Publication
<1%

Subrato Bharati, Prajoy Podder, M. Rubaiyat Hossain Mondal. "Hybrid deep learning for
<1%

detecting lung diseases from X-ray images", Informatics in Medicine Unlocked, 2020
Publication

V Pradyotan Raju, Senduru Srinivasulu, P Vishnu, Jeberson Retna Raj, Gowri S, Jabez J. "A Contemporary Technique for Lung Disease Prediction using Deep Learning", 2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC), 2022
Publication
<1%

30 doaj.org
Internet Source
<1%

31 ebin.pub
Internet Source
<1%

32 ijsrcseit.com
Internet Source
<1%

33 jebas.org
Internet Source
<1%

34 www.frontiersin.org
Internet Source
<1%

C. Church, H. K. Bayes, A. J. Fisher. "A review of the British Thoracic Society scientific meeting 1-3 December 2010", Thorax, 2011
Publication
<1%

P. Karthikeyan, K. Kamalanand, N. Selvaganesan. "Computational Modelling and
<1%

Certificate- 1

2024 First International Conference on

Innovations in Communications, Electrical and Computer Engineering (ICICEC 2024)

24 - 25, October 2024 | Davangere, Karnataka, India

CERTIFICATE

This certificate is presented to

BCS
7070

Rama Krishna Eluri

Department of Computer Science and Engineering,
Narasaraopeta Engineering College, Narasaraopet,
Andhra Pradesh, India

for presenting the research paper entitled " Optimizing the Powerhouse: Fine-Tuning CNNs for Superior Lung Disorder Detection" in the 2024 First International Conference on the Innovations in Communications, Electrical and Computer Engineering (ICICEC 2024) held at Bapuji Institute of Engineering and Technology (BIET), Davangere, Karnataka, India during 24 – 25, October 2024. The conference is technically co-sponsored by IEEE Bangalore Section.

Dr. Poornima B	Dr. Aravind H B	Prof.Y Vrushabhedrappa
Organizing Chair	Conference Chair	Patron

Organized by

Bapuji Institute of Engineering & Technology
Davangere, Karnataka, India | www.bietdvg.edu

Technical Sponsors

IEEE Bangalore Section | IEEE

Certificate- 2

2024 First International Conference on

Innovations in Communications, Electrical and Computer Engineering (ICICEC 2024)

24 - 25, October 2024 | Davangere, Karnataka, India

CERTIFICATE

This certificate is presented to

BCS 7070

Pentyala Tanuja

Department of Computer Science and Engineering, Narasaraopeta Engineering College, Narasaraopet, Andhra Pradesh, India

for presenting the research paper entitled " Optimizing the Powerhouse: Fine-Tuning CNNs for Superior Lung Disorder Detection" in the 2024 First International Conference on the Innovations in Communications, Electrical and Computer Engineering (ICICEC 2024) held at Bapuji Institute of Engineering and Technology (BIET), Davangere, Karnataka, India during 24 – 25, October 2024. The conference is technically co-sponsored by IEEE Bangalore Section.

Dr. Poornima B Organizing Chair	Dr. Aravind H B Conference Chair	Prof.Y Vrushabhedrappa Patron

Organized by

Bapuji Institute of Engineering & Technology
Davangere, Karnataka, India | www.bietdvg.edu

Technical Sponsors

IEEE Bangalore Section | IEEE

Certificate- 3

2024 First International Conference on

Innovations in Communications, Electrical and Computer Engineering (ICICEC 2024)

24 - 25, October 2024 | Davangere, Karnataka, India

CERTIFICATE

This certificate is presented to

BCS 7070

Vutla Lavanya

Department of Computer Science and Engineering, Narasaraopeta Engineering College, Narasaraopet, Andhra Pradesh, India

for presenting the research paper entitled " Optimizing the Powerhouse: Fine-Tuning CNNs for Superior Lung Disorder Detection" in the 2024 First International Conference on the Innovations in Communications, Electrical and Computer Engineering (ICICEC 2024) held at Bapuji Institute of Engineering and Technology (BIET), Davangere, Karnataka, India during 24 – 25, October 2024. The conference is technically co-sponsored by IEEE Bangalore Section.

B. Poornima		Y. Vrushabhadrappa
Dr. Poornima B Organizing Chair	Dr. Aravind H B Conference Chair	Prof.Y Vrushabhedrappa Patron

Organized by

Bapuji Institute of Engineering & Technology
Davangere, Karnataka, India | www.bietdvg.edu

Technical Sponsors

IEEE Bangalore Section | IEEE Advancing Technology for Humanity

Certificate- 4

2024 First International Conference on

Innovations in Communications, Electrical and Computer Engineering (ICICEC 2024)

24 - 25, October 2024 | Davangere, Karnataka, India

CERTIFICATE

This certificate is presented to

BCS 7070

M. Venkat Rao

Department of Computer Science and Engineering, Narasaraopeta Engineering College, Narasaraopet, Andhra Pradesh, India

for presenting the research paper entitled " Optimizing the Powerhouse: Fine-Tuning CNNs for Superior Lung Disorder Detection" in the 2024 First International Conference on the Innovations in Communications, Electrical and Computer Engineering (ICICEC 2024) held at Bapuji Institute of Engineering and Technology (BIET), Davangere, Karnataka, India during 24 – 25, October 2024. The conference is technically co-sponsored by IEEE Bangalore Section.

Dr. Poornima B Organizing Chair	Dr. Aravind H B Conference Chair	Prof.Y Vrushabhedrappa Patron

Organized by

Bapuji Institute of Engineering & Technology
Davangere, Karnataka, India | www.bietdvg.edu

Technical Sponsors

IEEE Bangalore Section | IEEE

Certificate- 5

2024 First International Conference on

Innovations in Communications, Electrical and Computer Engineering (ICICEC 2024)

24 - 25, October 2024 | Davangere, Karnataka, India

CERTIFICATE

This certificate is presented to

BCS 7070

M. Mokshagna

Department of Computer Science and Engineering,
Narasaraopeta Engineering College, Narasaraopet,
Andhra Pradesh, India

for presenting the research paper entitled " Optimizing the Powerhouse: Fine-Tuning CNNs for Superior Lung Disorder Detection" in the 2024 First International Conference on the Innovations in Communications, Electrical and Computer Engineering (ICICEC 2024) held at Bapuji Institute of Engineering and Technology (BIET), Davangere, Karnataka, India during 24 – 25, October 2024. The conference is technically co-sponsored by IEEE Bangalore Section.

B. Poornima
Dr. Poornima B
Organizing Chair

Dr. Aravind H B
Conference Chair

Y. Vrushabhedrappa
Prof.Y Vrushabhedrappa
Patron

Organized by

Bapuji Institute of Engineering & Technology
Davangere, Karnataka, India | www.bietdvg.edu

Technical Sponsors

IEEE Bangalore Section | IEEE