P&H(PNEUMONIA AND HEART) ANALYSIS THROUGH DEEP LERNING

A Capstone Project report submitted
in partial fulfillment of requirement for the award of degree

BACHELOR OF TECHNOLOGY
In
COMPUTER SCIENCE & ENGINEERING (CSE)
By

Vontela Dhanush	**19K41A0506**
Amogh Varsh Raju Ambati	**19K41A0590**
Sardar Kamaljith Singh	**19K41A05E3**
Mohd Amaan	**19K41A05G3**
Thallapaly Vinay Prakash	**19K41A05H2**

Under the guidance of
Mr.D.RAMESH
Assistant Professor, Department of CSE.

SR Engineering College
Innovation . Creativity . Entrepreneurship

SR ENGINEERING COLLEGE

Ananthasagar, Warangal.

CERTIFICATE

This is to certify that this report entitled **"P&H(Pneumonia and Heart) Analysis Through Deep Learning"** is the bonafide work carried out by **Vontela Dhanush, Ambati Amogh Varsh Raju, Sardar Kamaljeeth Singh, Mohd Amaan,Thallapaly Vinay prakash** as a Capstone Project for the partial fulfillment to award the degree **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE & ENGINEERING (CSE)** during academic year 2022-2023 under our guidance and supervision.

D.RAMESH
Ass.Prof.,Department of CSE
S R Engneering College
Ananthasagar, Warangal

Dr.M.Sheshikala
Assoc.Prof.&Head(CSE)
S R Engineering College
Ananthasagar,Warangal

External Examiner

ACKNOWLEDGMENT

We owe an enormous debt of gratitude to our project guide **D. RAMESH, Assistant Professor,** as well as Head of the CSE Department **Dr. Sheshikala, Associate Professor** for guiding us from the beginning through the end of the major project with their intellectual advice and insightful suggestions. We truly value their consistent feedback on our progress, which was always constructive and encouraging and ultimately drove us to the right direction.

We express our thanks to project coordinators **Mr. Sallauddin Md, Asst. Prof.,and Y.Chanti , Asst. Prof.** for their encouragement and support.

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved principal, **Dr. V. Mahesh,** for his continuous support and guidance to complete this project in the institute.

Finally, we express our thanks to all the teaching and non-teaching staff of the department for their suggestions and timely support.

ABSTRACT

Pneumonia is a common respiratory infection that affects the lungs and can be caused by various infectious agents such as bacteria, viruses, and fungi. It is characterized by inflammation and infection of the air sacs in the lungs, leading to symptoms such as fever, cough, chest pain, and difficulty breathing. Pneumonia can range from mild to severe, and it can be a life-threatening condition, especially in vulnerable populations such as the elderly, infants, and individuals with weakened immune systems. The lungs and heart play a critical role in the respiratory system, as they work together to ensure the exchange of oxygen and carbon dioxide, which is essential for the body's overall functioning and well-being.

Custom CNN, VGG16, Resnet-152 are deep learning models that have been widely used in the field of medical image analysis, including the analysis of chest X-ray images for pneumonia diagnosis. These models have been utilized to acquire different perspectives in analyzing chest X-ray images and aid in the accurate detection of pneumonia. Custom CNN allows for the customization of the Convolutional neural network architecture to suit specific requirements, while VGG16 and Resnet-152 are pre-trained models that have demonstrated high accuracy in image recognition tasks. By leveraging these models, researchers and clinicians can benefit from their ability to learn complex patterns from large datasets, leading to improved accuracy and efficiency in pneumonia diagnosis. These models have the potential to enhance the early detection and management of pneumonia, leading to better patient outcomes and reduced healthcare costs. Further research and development in the field of deep learning and medical image analysis are expected to continue to advance our understanding and management of pneumonia, ultimately benefiting patients and healthcare systems worldwide.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

ACRONYM	ABREVIATION
AI	Artificial Intelligence
DL	Deep Learning
CNN	Convolutional Neural Network
VGG-16	Visual Geometry Group
CT	Computer Tomography
CXR’s	Chest X Ray Images
CAD4TB	Computer-Aided Detection for Tuberculosis
XGB	Xtreme Gradient Boosting
ANN	Artificial Neural Network

1. INTRODUCTION

1.1 OVERVIEW:

Pneumonia is acute pneumonia caused by bacteria, viruses, or fungi that infect the lungs, causing inflammation of the air sacs and fluid filling of the lungs, pleural effusion. More than 15% of his deaths in children under the age of 5 are due to this disease. Pneumonia is most common in underdeveloped and developing countries where overpopulation, pollution, and unsanitary environmental conditions exacerbate the situation and lack medical resources. Therefore, early diagnosis and treatment can play an important role in preventing the disease from becoming fatal.Computer tomography (CT), magnetic resonance imaging (MRI), or radiography. Radiological examination of the lungs using (x-rays) is often used for diagnosis. X-ray imaging is a non-invasive and relatively inexpensive examination of the lungs. Figure 1 shows examples of X-rays of a lung and a healthy lung. A white spot (marked by a red arrow) on a chest x-ray called an infiltrate distinguishes pneumonia from a healthy condition. However, chest radio-graphs for detecting pneumonia are prone to subjective variability. Therefore, we need an automated system to detect pneumonia. In this study, we developed a computer-aided diagnosis (CAD) system that uses an ensemble of deep transfer learning models for accurate classification of chest radio-graphs.

Deep learning is an important artificial intelligence tool that plays a key role in solving many complex computer vision problems. Deep learning models, especially Convolutional neural networks (CNN's), are widely used for various image classification problems. However, such models work best only when presented with large amounts of data. The X-ray image analysis problem requires an experienced physician to classify each image, which is costly and time consuming, making it difficult to collect such large amounts of labeled data. Transfer learning is a workaround to overcome this obstacle. In this technique, a model trained on a large data set is reused and the network weights determined by that model are applied to solve problems associated with small data sets. CNN models trained on large datasets like Image Net consisting of over 14 million images are widely used for Chest X-Ray image analysis tasks.

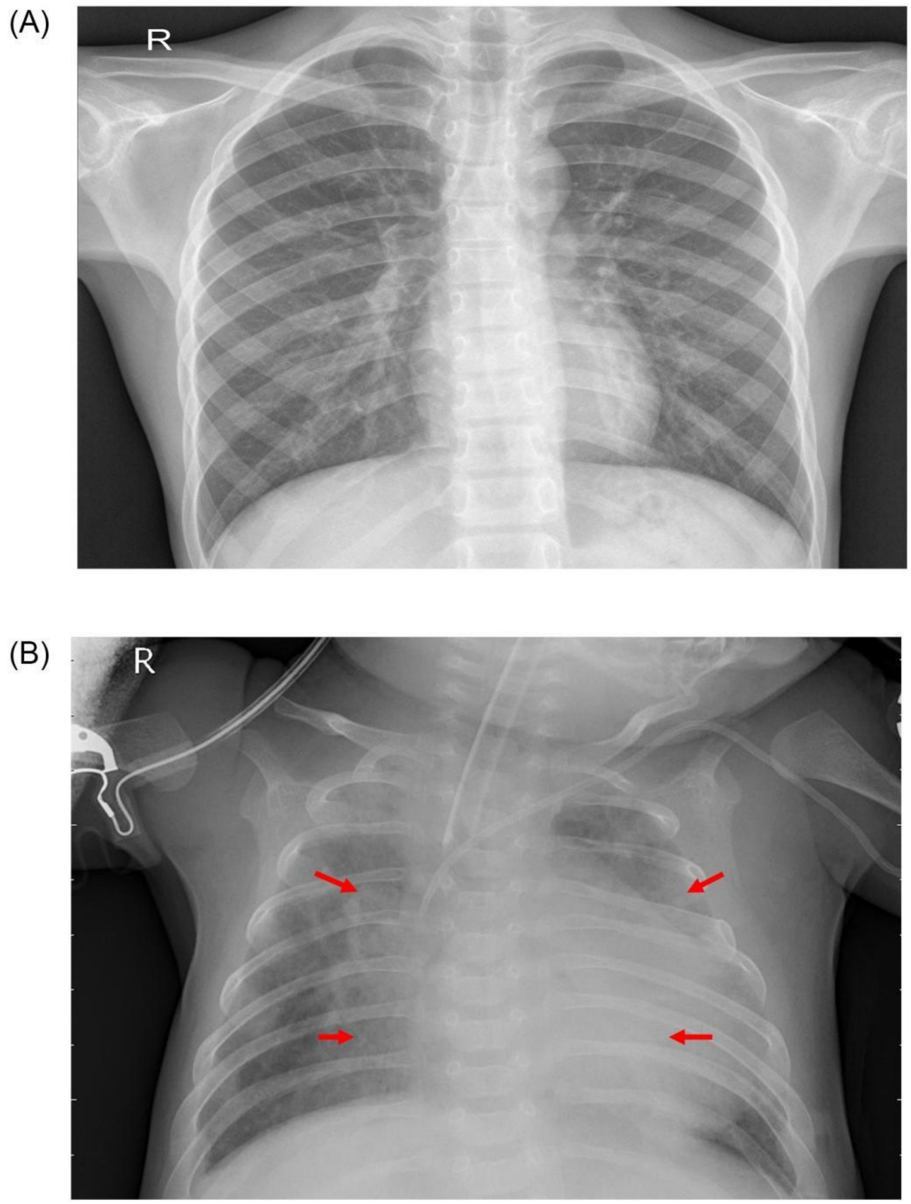


Figure 1.1: Examples of two X-ray plates that display (a) a healthy lung and (b) a pneumonia lung.

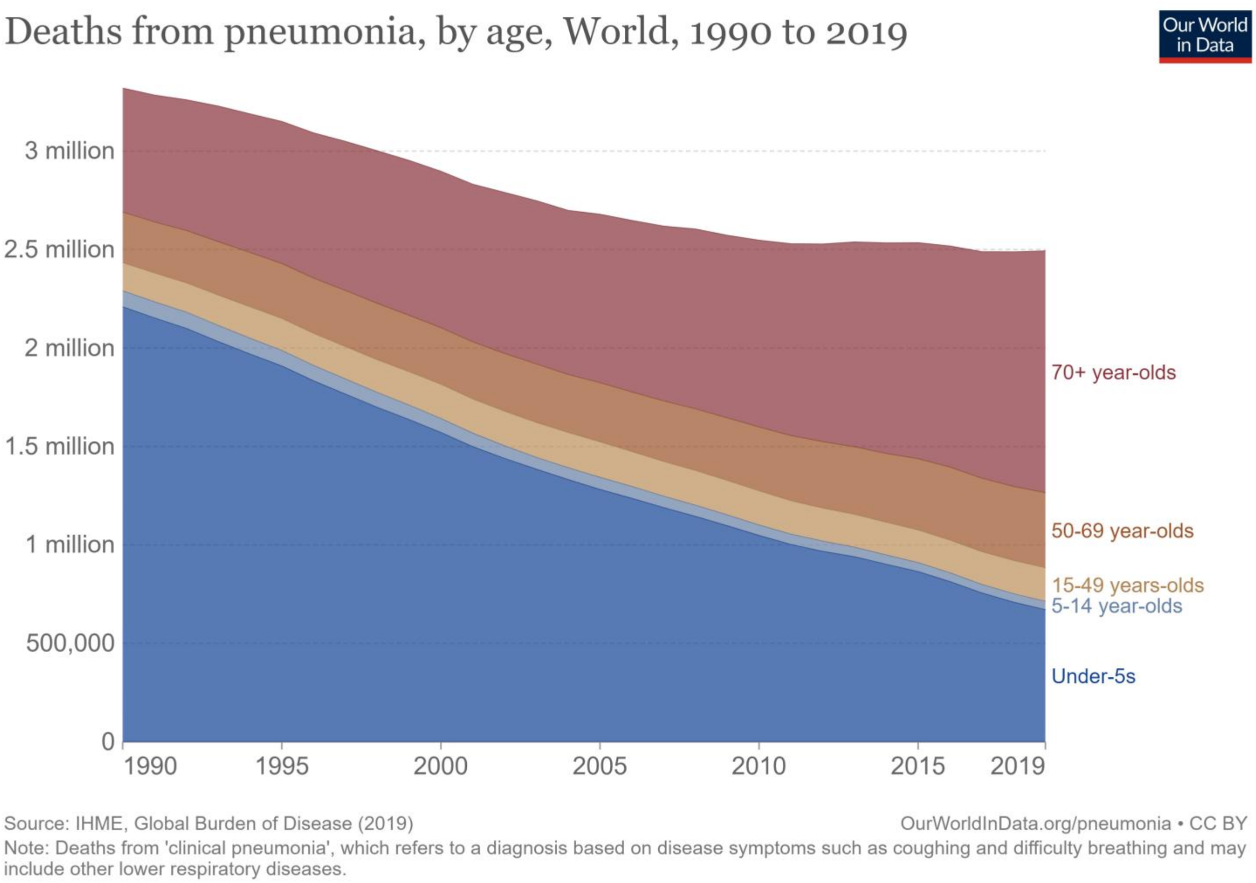


Figure 1.2:Deaths from pneumonia, by age

1.2 MOTIVATION AND SCOPE OF THE PROJECT

The Main Reason to uphold and propose this particular project segment is the opportunity of taking a leap of technique in creating an image analysis and new innovation in health care segment, not only that but also the fact that people who face problems in analyzing the disease as soon as possible. Building a chest X-ray analysis system using deep learning offers several benefits. Firstly, it allows for faster and easier analysis of chest X-ray images, helping clinicians make more efficient and accurate diagnoses. Deep learning models, such as CNN's, are capable of learning complex patterns and features from large datasets, enabling them to detect abnormalities in chest X-ray images with high accuracy. Moreover, pre-trained models like VGG16, ResNet152, and others, provide a head start by leveraging their learned features from vast datasets, making it easier to fine-tune them for the specific task of chest X-ray analysis. Additionally, deep learning models can be easily accessed and implemented using popular deep learning libraries like Keras or TensorFlow, making it accessible to a wide range of researchers and practitioners. The interpretability and explainability of deep learning models can also aid clinicians in understanding and trusting the model's predictions. Overall, building a chest X-ray analysis system through deep learning offers advantages such as easy analysis, faster results, and accessibility, making it a promising approach for improving diagnostic accuracy in chest X-ray analysis. In addition, the accessibility of deep learning libraries like Keras and TensorFlow makes it easier for researchers and practitioners to implement and experiment with chest X-ray analysis systems. These libraries provide a wide range of pre-processing functions, optimization algorithms, and visualization tools that facilitate the development, training, and evaluation of deep learning models for chest X-ray analysis.

Moreover, the interpretability and explainability of deep learning models can enhance trust and acceptance among healthcare professionals. Understanding how the model arrives at its predictions can provide valuable insights and aid in the decision-making process. This can help clinicians better understand the model's strengths, limitations, and potential biases, leading to more informed and confident diagnoses.

Overall, the adoption of deep learning for chest X-ray analysis represents a significant technological leap in the health sector, offering advanced capabilities, accessibility, and interpretability that can potentially revolutionize the field of medical imaging and improve patient care.

1.3: PROBLEM STATEMENT

The accurate and timely diagnosis of chest X-ray images plays a crucial role in the detection and management of various respiratory and cardiovascular diseases. However, manual interpretation of chest X-rays can be subjective, time-consuming, and prone to human error. Therefore, there is a need for an automated and efficient system that can analyze chest X-ray images with high accuracy and provide rapid results to aid in timely patient care.

1.4: Existing methods

There are several existing methods used in various fields in biomedical field, surgical field, Research sector and much more , the main existing methods involved training through mainly

ResNet-34: First, there is a convolution layer containing 64 filters with kernel size 7×7, this is the first convolution, followed by a max pooling layer. I specified a stride length of 2 in both cases. The following conv2_x has a pooling layer and the following Convolutional layers.

ANN,CNN: Various types of neural networks in deep learning, such as Convolutional neural networks (CNN's), recurrent neural networks (RNNs), and artificial neural networks (ANNs), are changing the way we interact with the world.

VGG19+CNN: VGG-19 is a 19-layer deep Convolutional neural network. You can load a pre-trained version of a network trained with over 1 million images from the ImageNet database. A pre-trained network can classify images into 1000 object categories, such as keyboards, mice, pens, and many animals.

GoogLeNet: Google Net (or Inception V1) was proposed in 2014 at Google (in collaboration with various universities) in a research paper "Going Deeper with Convolutions". This architecture won the ILSVRC 2014 Image Classification Challenge. Significantly lower error rate compared to previous winners AlexNet (2012 ILSVRC winner) and his ZF-Net (2013 ILSVRC winner) and significantly lower than VGG (2014 runner-up) was the error rate. This architecture uses techniques such as 1 × 1 convolution and global average pooling in the middle of the architecture.

ResNet-18: ResNet-18 is a Convolutional neural network with a depth of 18 layers. You can load a pre-trained version of the network trained with over 1 million images from the ImageNet database . A pre-trained network can classify images into 1000 object categories, such as keyboards, mice, pens, and many animals.

CAD4TBv6:CAD4TB is a CE-certified AI software that supports cost-effective, fast, easy, and accurate automated tuberculosis detection

DenseNet:A DenseNet is a type of Convolutional neural network that uses dense connections between layers through dense blocks. Here, we directly connect all layers (with matching feature map sizes).

ResNet152: ResNet is one of the most powerful deep neural networks which has achieved fantabulous performance results in the ILSVRC 2015 classification challenge.

EfcientNet-B7: It is a Convolutional neural network architecture and scaling method that uniformly scales all dimensions of depth/width/resolution using a compound coefficient

DeepLabv3: DeepLabv3 is a semantic segmentation architecture that improves upon DeepLabv2 with several modifications. To handle the problem of segmenting objects at multiple scales, modules are designed which employ atrous convolution in cascade or in parallel to capture multi-scale context by adopting multiple atrous rates.

XGB-linear: XGBoost Linear is an advanced implementation of a gradient boosting algorithm with a linear model as the base model. Boosting algorithms iteratively learn weak classifiers and then add them to a final strong classifier.

1.5 LITERATURE SURVEY:

There are several solutions and approaches designed and are achieved but most of the projects, proposals have bee proven to be either ineffective, only classification based outputs with best score, better analysis but with low score etc... , the main draw back of the following projects was the challenges faced for managing the given data and it is difficult for the image to be completely balanced and is farther more complicated than imagined.

The number of collected papers are 40 and are quality papers which clearly mention whole process of how did they approach the solution and get the best results and challenges faced. The papers also portray different kinds of data-sets being used , the results perspective shows a lot of difference from one to one .

There are certain papers which show only whether the given input x-ray image consists of a disease or not , some other projects give x-ray image as output but with minimal information for example: Heat map output highlighting region of disease affected, marked locations of tumors etc.

The detection of pneumonia using chest radiographs has been an open problem for many years, with the main limitation being the paucity of published data. Traditional machine learning methods have been extensively studied. Chandra et al. Segmenting lung regions from chest X-rays, from these regions he extracted eight statistical features and used them to classify them. They implemented his five conventional classifiers:

Multilayer Perceptron (MLP), Random Forest, Sequential Minimal Optimization (SMO), Classification by Regression, and Logistic Regression. They evaluated the method on 412 images and achieved a curacy rate of 95.39 using the MLP classifier. Kuo et al. used 11 features to detect pneumonia in 185 schizophrenic patients. We applied these features to various regression and classification models, such as decision trees, support vector machines, and logistic regression, and compared the results of the models. They achieved the highest accuracy of 94.5% using a decision tree classifier. Other models lagged far behind. Similarly, Yue et al. Six features were used to detect pneumonia on chest CT scan images of 52 patients. The highest achieved her AUC value was 97%. However, these methods cannot be generalized and have been evaluated on small data sets.

Citation	Title	Model used	Feature extraction	Data set	Output metrics
[1]	Deep learning for chest X-ray analysis: A survey	CNN	image-level prediction (classification and regression), segmentation, localization, image generation and domain adaptation	Kaggle challenge	Finding the high accuracy using image-level prediction (classification and regression), segmentation, localization, image generation and domain adaptation
[2]	Application of deep learning techniques for detection of COVID-19 cases using chest X-ray images: A comprehensive study	ResNet-34	CNN	Kaggle challenge	98.33%
[3]	PNEUMONIA DETECTION USING CNN THROUGH CHEST X-RAY	ANN, CNN	CNN	Kaggle challenge	Architecture 5 works better
[4]	CHEST X-RAYS IMAGE CLASSIFICATION IN MEDICAL IMAGE ANALYSIS	ResNet-50	CNN	Chest X-Ray14	ResNet-50 achieved state-of-the-art results in four out of fourteen classes.
[5]	Deep-Pneumonia Framework Using Deep Learning Models Based on Chest X-Ray Images	ResNet152V2	CNN	Kaggle challenge	99.22%
[6]	Deep-chest: Multi-classification deep learning model for diagnosing COVID-19, pneumonia, and lung cancer chest diseases	VGG19+CNN	CNN	Kaggle challenge	98.05%
[7]	Pneumonia detection in chest X-ray images	GoogLeNet,	convolutional neural	Kaggle	98.81%

	using an ensemble of deep learning models	ResNet-18, and DenseNet-121	network	challenge	
[8]	Chest x-ray analysis with deep learning-based software as a triage test for pulmonary tuberculosis: a prospective study of diagnostic accuracy for culture-confirmed disease	CAD4TBv6	Deep learning based software , regression model	Kaggle challenge	High accuracy than qXRv2
[9]	Identifying Pneumonia in Chest XRays: A Deep Learning Approach	(ResNet50 + ResNet101)	Mask-RCNN	Kaggle challenge	0.218051
[10]	Chest Radiograph Interpretation with Deep Learning Models: Assessment with Radiologist-adjudicated Reference Standards and Population-adjusted Evaluation	CNN	convolutional neural networks	Google cloud, kaggle	The model demonstrated population-adjusted areas under the receiver operating characteristic curve of 0.95 (pneumothorax), 0.72 (nodule or mass), 0.91 (opacity), and 0.86 (fracture)
[11]	Detection of tuberculosis patterns in digital photographs of chest X-ray images using Deep Learning: feasibility study	ANN, ViDi classification tool, ViDi detection tool	Deep Learning image analysis software (Suite v2.0; ViDi Systems, Villaz-Saint-Pierre, Switzerland	Kaggle challenge	area under the ROC curve 0.82 & AUC 0.98
[12]	Deep learning based detection and analysis of COVID-19 on chest X-ray images	Xception	CNN	Kaggle challenge	97.97%
[13]	A machine	CNN	CNN	Kaggl	The proposed method

	learningbased framework for diagnosis of COVID19 from chest Xray images	+PCA		e challe nge	achieved high accuracy of 100% using CNN +PCA when variance of 0.99 was used
[14]	Machinelearning classifcation of texture features of portable chest Xray accurately classifes COVID19 lung infection	XGB-linear	DCNN	Kaggl e challe nge	100%
[15]	A deep-learning pipeline for the diagnosis and discrimination of viral, non-viral and COVID-19 pneumonia from chest X-ray images	DeepL abv3	Semantic Segmentation	Kaggl e challe nge	DeepLabv3 outperformed both FCN and U-Net
[16]	Deep learning for distinguishing normal versus abnormal chest radiographs and generalization to two unseen diseases tuberculosis and COVID19	Efcient Net-B7	CNN	NH	EfcientNet-B7 performs better than other advanced networks
[17]	Machine learning applied on chest x-ray can aid in the diagnosis of COVID-19: a first experience from Lombardy, Italy	Image analysi s for deep learnin g classifi er	CNN	aggle challe nge	99%
[18]	Pneumonia Detection and Classification Using Deep Learning on Chest X-Ray Images	ResNe t152	CNN	Kaggl e challe nge	97%
[19]	Predicting COVID-19 Pneumonia Severity on Chest X-ray With Deep Learning	Dense Net	CNN	Chest X-ray8 from NIH	The use of a score combining geographical extent and degree of opacity allows clinicians to compare CXR images

[20]	CoroNet: A deep neural network for detection and diagnosis of COVID-19 from chest x-ray images	Xception Classification model	normalization	Image Net data set	Overall accuracy of 89.6% precision and recall rate is 93% and 98.2% for 4 classes and classification accuracy is 95%
[21]	AI for radiographic COVID-19 detection selects shortcuts over signal	CNN Classification model	normalization	Github Covid repository Chest x-ray14 repository	Outlined area of disease on the x-ray through generated images from the original x-reays
[22]	Public Covid-19 X-ray datasets and their impact on model bias - a systematic review of a signifificant problem		PROBAST , TRIPOD and TREE.		
[23]	Predicting COVID-19 Pneumonia Severity on Chest X-ray With Deep Learning	Dense net Model from torchx-ray vision library	Normalization Image resized to 224*224 and scaled to -1024 1024	RSNA , cheXpert, chestxray8	Quantitative performance metrics- correlation is 0.8 with output of 4 labels (lung opacity, pneumonia, infiltration and consolidation)
[24]	Determination of disease severity in COVID-19 patients using deep learning in chest X-ray images	EfficentNet @ modified U-net	Manual classification of types of x-rays	X-rays of 48 patients between the age of	Classification and output of percentage of RT-PCR results
[25]	Deep transfer learning artificial intelligence accurately stages COVID-19 lung disease severity on portable chest radiographs	CNN, VGG16.	Five-fold cross validation	CXR images from open download	Predicted vs radiology scores,correlation analysis,mean absolute error analysis are optimal to the research previous papers

[26]	A promising approach for screening pulmonary hypertension based on frontal chest radiographs using deep learning: A retrospective study	Resnet 50, xception, inception v3	tune learning rate to 0.0008 according to the best score on validation data.	PASP measured Doppler transthoracic echocardiography from 762 patients (357 healthy controls and 405 with PH) from three institutes in China from January 2013 to May 2019	The AUC performed by the best model (Inception V3) achieved 0.970 in the internal test, and slightly declined in the external test (0.967) when using deep learning algorithms to clas sify PH from normal based on chest X-rays. The mean absolute error (MAE) of the best model for prediction of PASP value was smaller in the internal test (7.45) compared to 9.95 in the external test
[27]	Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study	CNN	A cross-sectional design with multiple model training cohorts was used to evaluate model generalizabili	chest X-rays from the National Institutes of Health Clinical	Pneumonia-screening CNNs achieved better internal than external performance in 3 out of 5 natural comparisons. When models were trained on pooled data from sites with different pneumonia prevalence

			ty to external sites using split-sample validation. A total of 158,323 chest radio graphs were drawn from three institutions	Center	
[28]	Artificial Intelligence-Based Diagnosis of Cardiac and Related Diseases	Mask-Rcnn Resnet	Feature pyramid network decoder	CXR image s	Semantic segmentation and output mask of input CXR images
[29]	Deep learning to automate Brasfifield chest radiographic scoring for cystic fifibrosis	DCNN model Resnet -18	XR650 digital radiography system	Bras-field datase ts	Accuracy of 90.3% Correlation of 0.79 - 0.83
[30]	Diagnosis of Corona-virus Disease 2019 Pneumonia by Using Chest Radiography: Value of Artificial Intelligence	CV19-net	(ACU) Area under characteristic curve used to differentiate based on disease and date taken from certasin period of time to fixed period of time.	Covid -19 chest x-ray	Accuracy of 94% and was able to classify the items from 2019 pneumonia to other related pneumonia
[31]	Ensemble learning based automatic detection of tuberculosis in chest X-ray images using hybrid feature descriptors	CNN with enseamble learnin g	K-fold cross validation scheme and rezied images to 300 * 300	Shenz hen and Montg omery datase ts	Achieved 80% accuracy through the model valuation at model evaluation
[32]	Synthesis of	VGG-	Normalizatio	RSNA	Model evaluation and

	COVID-19 chest X-rays using unpaired image-to-image translation	16 and ResNet-50	n and resize of images to 256 * 256 and scaled pixel value to [0,1]	Pneumonia detection challenge datasets with CXR'S	comparison shown through using the two models no accuracy achieved as for the work nature
[33]	Reproducibility of abnormality detection on chest radiographs using convolutional neural network in paired radiographs obtained within a short-term interval	CNN	normalization	PACS at ASAN medical center all CXR's With MRI scans	Image output with coloured outlined box around the affected area but with minimal exposure and enhancement
[34]	Deep learning-based model for screening and staging pneumoconiosis	U-Net semantic segmentation with Res-net as backbone	2 stages of image reading with final dropped images with	Manual collection of data-set through their domain websites through user registrations.	Achieved accuracy of 92% with
[35]	Identifying Cardiomegaly in ChestX-ray8 Using Transfer Learning	Transfer learning	Images re-sized to (224,224) with dimension 3	Chest X-ray8 data-set issued by NIH	Histogram equalization and cropped images of gray scale images

[36]	Pneumonia detection in chest X-ray images using an ensemble of deep learning models	GoogLeNet, ResNet-18, and DenseNet-121	Deep learning model	Kermany et al. and the Radiological Society of North America (RSNA)	Obtained an accuracy rate of 98.81%, a sensitivity rate of 98.80%, a precision rate of 98.82%, and an f1-score of 98.79% on the Kermany dataset and an accuracy rate of 86.86%, a sensitivity rate of 87.02%, a precision rate of 86.89%,
[37]	Covid-19 Classification Using Deep Learning in Chest X-Ray Images	ResNet - 50	Deep learning model		Accuracy rate of 99.5%
[38]	Deep Learning Approach for Analyzing the COVID-19 Chest X-Rays	VGG 16	Deep learning model	Kaggle	Accuracy 94.96%
[39]	Pneumothorax detection in chest radiographs using convolutional neural networks	CNN	Deep learning model	Google Cloud	Acquired AUC with values between 0.92 and 0.96
[40]	A deep learning approach for classification of COVID and pneumonia using DenseNet-201	DenseNet - 121	Deep learning model	Kaggle	accuracy of 99.1%, sensitivity of 98.5%, and specificity of 98.95%.

Table 1.1

1.1 Comparison Table

2. HARDWARE / SOFTWARE TOOLS

2.1 PYTHON:

Python: An Indispensable Tool for Modern Computing

Python is a high-level, general-purpose, interpreted programming language that has gained immense popularity in recent years due to its versatility, ease of use, and extensive libraries. Originally created by Guido van Rossum in the late 1980s, Python has become one of the most widely used programming languages, offering a wide range of applications in various domains, including data science, machine learning, web development, scientific computing, and more.

Key Features of Python:

1. Easy to Read and Write: Python's clean and readable syntax, using indentation to define blocks of code, makes it easy to understand and write, even for beginners. This reduces the likelihood of errors and enhances code readability, leading to more maintainable and robust code.

2. Large Standard Library: Python comes with a vast standard library that includes modules for file I/O, regular expressions, data manipulation, networking, and more. These modules provide a rich set of tools and functionalities, making Python a powerful language for a wide range of tasks without the need for additional external libraries.

3. Dynamically Typed and Interpreted: Python is a dynamically typed language, allowing for flexibility and ease of use as variable types can be changed on the fly. Additionally, Python is interpreted, allowing for quick and efficient development cycles with no need for compiling, making it ideal for rapid prototyping and development.

4. Extensive Third-Party Libraries: Python has a vibrant ecosystem of third-party libraries, such as NumPy, Pandas, Matplotlib, TensorFlow, and Scikit-learn, which provide additional functionalities for

data analysis, machine learning, visualization, and more. These libraries make Python a popular choice for data science and machine learning tasks, offering a rich toolkit for developers.

Advantages of Python:

1. Productivity and Rapid Development: Python's simplicity and readability enable developers to write code quickly, reducing development time and increasing productivity. Its large standard library and extensive third-party libraries also provide pre-built functionalities, further speeding up development cycles.

2. Versatility and Scalability: Python's versatility allows it to be used for a wide range of applications, from small scripts to complex applications. It can be used for web development, scientific computing, machine learning, data analysis, network programming, and more. Python's scalability makes it suitable for both small-scale projects and large-scale enterprise applications.

3. Strong Community Support: Python has a strong and active community of developers, users, and contributors who regularly contribute to the language's development and maintenance. This results in a vast pool of resources, documentation, forums, and communities that provide support, making Python an easy language to learn and use.

4. Cross-Platform Compatibility: Python is a cross-platform language, meaning that Python code can run on different operating systems, including Windows, macOS, and Linux, without any changes. This makes it highly portable and suitable for multi-platform development.

Use Cases of Python:

1. Data Science and Machine Learning: Python is widely used in data science and machine learning tasks due to its extensive libraries such as NumPy, Pandas, and Scikit-learn, which provide tools for data manipulation, analysis, visualization, and machine learning algorithms. Python's ease of use and rich ecosystem make it a preferred choice for building and deploying machine learning models.

2. Web Development: Python is used for web development, utilizing frameworks such as Flask, Django, and Pyramid, which provide tools for building web applications, API, and web services. Python's simplicity and versatility make it a popular choice for developing scalable and efficient web applications.

2.1.1 LIBRARIES:

2.1.1.1: PANDAS: Pandas is a powerful and widely used open-source data manipulation library for Python. It provides data structures such as Data-frame and Series, which are designed for efficient data analysis and manipulation. Pandas offers a wide range of functionalities for data cleaning, data transformation, data visualization, and data analysis. With Pandas, developers can easily load, manipulate, and analyze data in various formats, including CSV, Excel, SQL, and more. It offers intuitive and flexible data indexing, slicing, and filtering options, making it easy to perform complex data operations. Pandas also integrates well with other popular libraries such as NumPy and Matplotlib, enabling developers to create end-to-end data analysis workflows in Python. Its comprehensive documentation and strong community support make Pandas a top choice for data analysis tasks in fields such as data science, machine learning, finance, and business analytics.

2.1.1.2: NUMPY: NumPy is a widely used open-source numerical computing library for Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to perform operations on these arrays efficiently. NumPy is the foundation for many other popular data science and machine learning libraries in Python, as it provides fast, memory-efficient operations for numerical computations. NumPy's key data structure is the NumPy array, which is a homogeneous, fixed-size array that allows for vectorized operations, making it highly efficient for numerical computations. NumPy also provides a rich set of functions for mathematical operations such as linear algebra, Fourier analysis, statistical analysis, and more. NumPy's array manipulation capabilities, such as reshaping, indexing, and broadcasting, make it convenient for working with complex data sets and performing advanced computations. Its performance and ease of use make it a preferred choice for tasks that require fast and efficient numerical computations in fields such as scientific computing, data analysis, machine learning, and image processing. Additionally, NumPy has a large community of users and developers, providing extensive documentation and support, making it a popular tool in the data science and numerical computing communities.

2.1.1.3: cv2(OpenCV): OpenCV, also known as cv2, is a powerful open-source computer vision library for Python. It provides a wide range of tools and functions for image and video processing, making it a popular choice for tasks such as image manipulation, object detection, face recognition, and more. OpenCV is designed to be highly optimized and efficient, allowing for real-time processing of images and videos. It provides support for a wide variety of image formats and supports common image processing operations such as filtering, transformation, edge detection, and color space conversions. OpenCV also includes advanced features such as feature detection, feature matching, and camera calibration, making it suitable for complex computer vision tasks. The library is extensively used in fields such as computer vision, robotics, augmented reality, and machine learning. OpenCV has a large community of developers and users, which provides continuous support, documentation, and a wide range of resources for learning and development. Its ease of use, performance, and versatility make OpenCV a popular choice for a wide range of computer vision applications in both academia and industry.

2.1.1.4: KERAS(TENSORFLOW BACKEND): Keras is a high-level neural networks API written in Python that provides a user-friendly interface for building and training deep learning models. With TensorFlow backend, Keras offers a powerful and flexible deep learning framework for developing a wide variety of applications, ranging from image and speech recognition to natural language processing and more. TensorFlow, a popular open-source machine learning framework, serves as the computational backend for Keras, providing efficient implementation of neural network operations and optimizations for training large-scale deep learning models. Keras with TensorFlow backend allows for seamless integration with other popular Python libraries such as NumPy, Pandas, and Matplotlib, making it easy to preprocess data, visualize results, and perform various model evaluations. Keras provides a wide range of pre-processing functions, activation functions, loss functions, and optimization algorithms, making it highly customizable for building different types of neural network architectures. Keras also supports both sequential and functional model building approaches, providing flexibility in designing complex neural network architectures. Additionally, Keras with TensorFlow backend offers extensive documentation, tutorials, and a supportive community, making it a popular choice for both beginners and experienced deep learning practitioners. Its user-friendly interface, flexibility, and integration with TensorFlow make Keras a powerful tool for developing deep learning models with ease and efficiency.

3. PROJECT IMPLEMENTATION

3.1 DESCRIPTION:

We have used 3 models and 2 per-processing techniques to perform the operation the dateset which we gathered are complete CXR's which plays the important role in training the models and divided into sets.

Chest X-ray (CXR) analysis has been transformed by deep learning techniques and libraries such as Pandas, NumPy, OpenCV (cv2), and Keras with TensorFlow backend. These tools enable efficient handling, preprocessing, and analysis of CXR images. Deep learning models, including custom CNN's and pre-trained models like VGG16 and ResNet-152, offer improved accuracy and efficiency in diagnosing respiratory conditions. CXR images are crucial for health assessment, and deep learning-based analysis has the potential to automate and improve the accuracy of diagnosis. This can revolutionize radiology, leading to enhanced healthcare delivery and improved patient outcomes.

3.2 DESIGN

Our project design processes and architecture involves 3 segments:

1. **Data Analysis**
2. **Multilevel classification**
3. **Model saving**

In this particular design segments we had to resort to both conventional and modern solutions to overcome the complexity challenges put in-front by the project. So, we had to go with this particular process design where in the first phase we have done

1. DATA ANALYSIS: In here we have first tried to read the data into the kernel to start analyzing stuff it didn't work out as we planned . As the data is high resolution images it was hard to read and initialize at once so we first tried to compress them and loosen as much as necessary and UN-necessary pixels as discarded.

Then we moved on to second stage and we decided to use Multilevel classification architecture
But yet the problem preceded of kernel crashes and other errors so we had to train the model by parts after training and saving the models by parts. Below flow chart shows you the flow of process precisely.

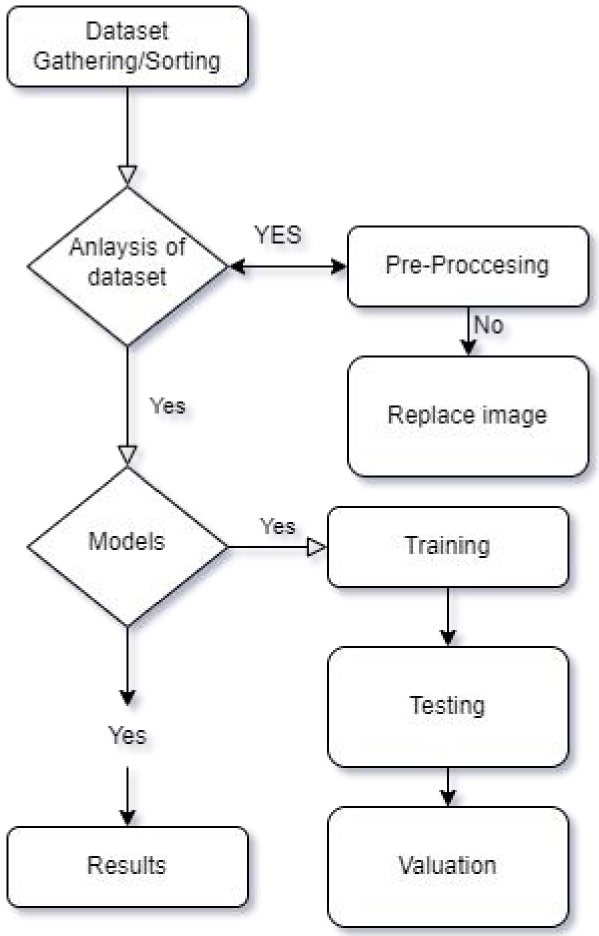


FIG:3.1 PROCESS FLOW

The project likely involves analyzing CXR images for the detection and classification of pneumonia, an important respiratory disease that affects the lungs. The use of deep learning techniques, such as Convolutional neural networks (CNN's) and pre-trained models like VGG16 and ResNet152, suggests a sophisticated approach to image analysis.

The project likely begins with data preparation using Pandas and NumPy libraries for data manipulation and processing. The CXR images may be loaded and pre-processed using OpenCV (cv2) library, which provides tools for image manipulation, such as re-sizing, normalization, and augmentation. Image normalization may involve converting the pixel values to a standardized range, while augmentation may involve applying random or systematic transformations to create new variations of the original images, thereby augmenting the training data.

Next, the project likely involves building a deep learning model using Keras with a TensorFlow backend. The model architecture may include Conv2D, MaxPooling2D, Batch Normalization, Flatten, and Dense layers, among others, to define the CNN architecture. The project may also use pre-trained models like VGG16 or ResNet152, which have been trained on large datasets and are known for their effectiveness in image classification tasks.

During the model training phase, the project may utilize techniques such as cross-validation, model check-pointing, and evaluation using metrics like accuracy, confusion matrix, and classification report to assess the model's performance. The project may also involve fine-tuning the model by adjusting hyper-parameters, tuning the model architecture, or using transfer learning techniques to improve the model's accuracy and generalization.

Finally, the project may involve interpreting the results and drawing insights from the trained model's predictions. The importance of CXR images in diagnosing respiratory diseases like pneumonia may be highlighted, and the potential of deep learning techniques in revolutionizing healthcare by enabling automated and accurate analysis of medical images may be discussed.

In summary, the project likely focuses on building a deep learning-based CXR analysis system using Python libraries like Pandas, NumPy, OpenCV (cv2), and Keras with a TensorFlow backend. The project involves data preparation, model building, model training, and result interpretation, with an emphasis on the importance of CXR images and the potential of deep learning in revolutionizing healthcare.

3.2.1 DATA-SET ACQUIZITION:

The data set we have gathered manually is taken many sources and is arranged into 3 files with each specific to its feature perspective.The data set consists of chest-x-ray images.

The data is the combination of various sources Firstly the data-sets chosen were from kaggle but the first dateset "pneumonia challenge" was of 2 GB and has pneumonia data images but the data-set was imbalanced to be worked with so we moved onto another data-set namely Chest X-ray 14 which was of 42 GB which was optimized to 3.6 GB but was completely imbalanced so we checked other data-sets and resources and the results were are for off the expectations. So, to overcome this manual and software based image filtration has been done and the first data-set has been altered to cover the imbalance it had and finally we have achieved the supposed and projected results

The data set consists of 3 folders namely:

1. **Train**
2. **Test**
3. **Val**

1. **Train** : this folder further consists of two sub folders which are named as abnormal and normal which consists of separate x-ray images of diseased and non-diseased respectively. The total number of images together are 5,218 images respectively.

2. **Test:** this folder also consists of two sub-folders with normal and ab-normal file names and also the number of x-rays in these folders together are 624 images

3. **Val:** This also consists of same sub folders but only 18 images in them we call them augmented data which are completed filtered data for stronger training of data

Note: we have made some subtle changes to the data-set i.e we have have taken images from the other data-set mainly named as Chest X-ray 14 datasets created by NIH. This datasets consist of images worth of around 42GB which was too heavy to be even initiated to be read and operated so we had to resort to this data set and edit it according to the balance of the data-set.

The main challenges were to analyze the images and filter the best one so we had to use image data-generator to analyze each picture individual and extract the important pixels which are more intensive and elaborated.

Fig: 3.2 Main folders

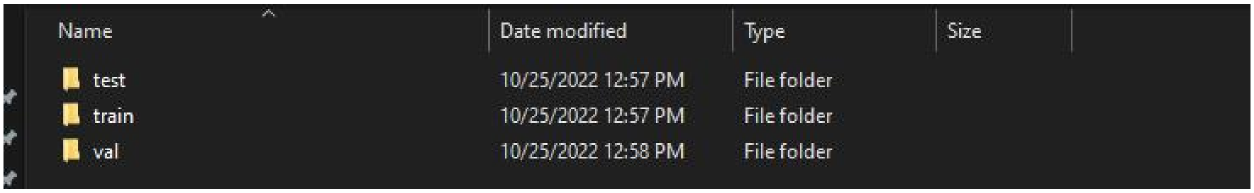


Fig 3.3 Sub folders

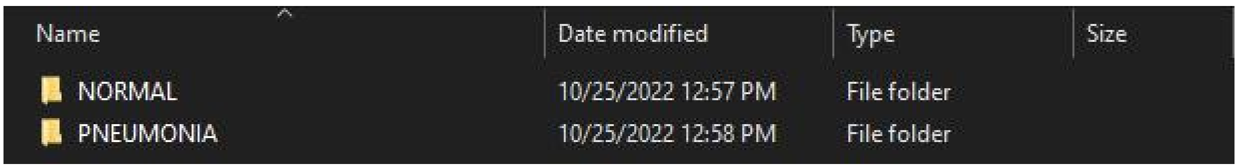
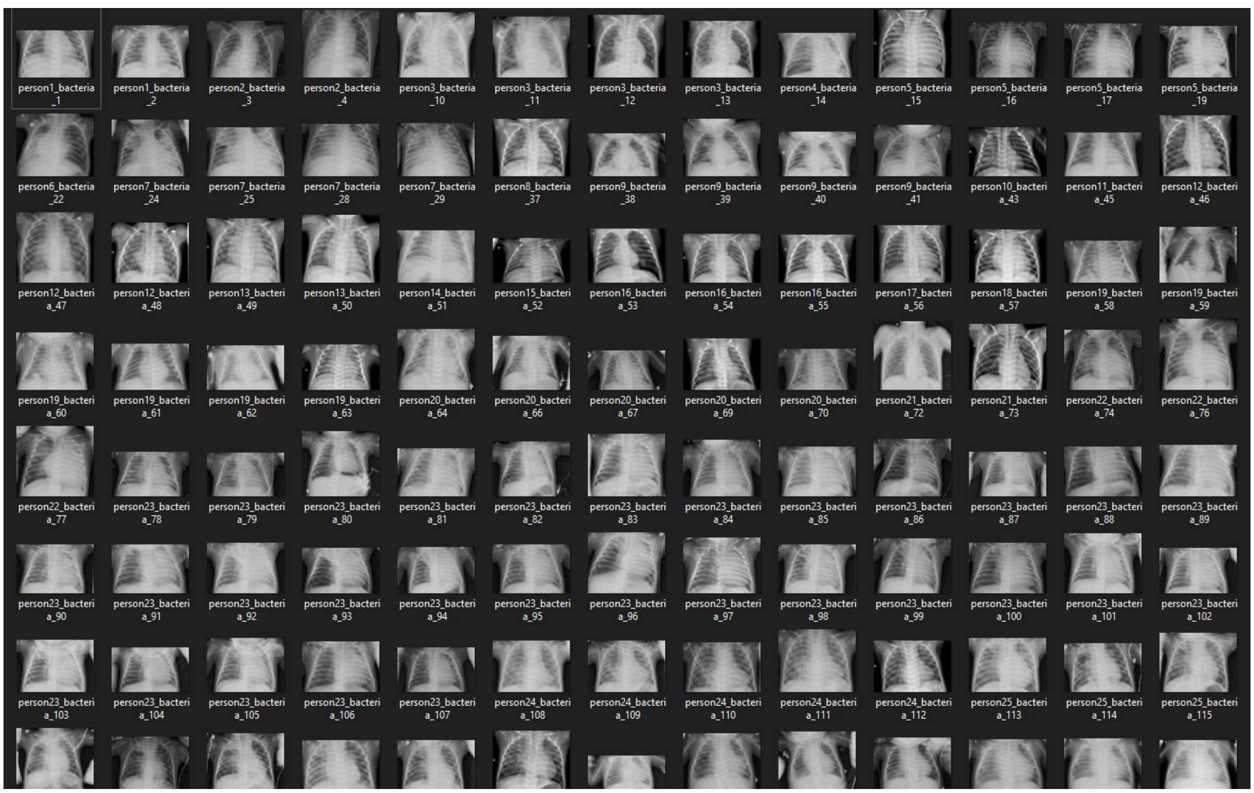


Fig 3.4: X-ray images from data set



S.No	Data-Set Used	Folders Count	Size
1.	ChestX-ray 14	14	45 GB
2.	Pneumonia challeneg	3	2.1 GB
3.	Custom Collection	1	500+ MB
4.	Final Dataset size	3	1.75 GB

Table 3.1 Data-set Comparison

3.2.2 DATASET - PROCCESING

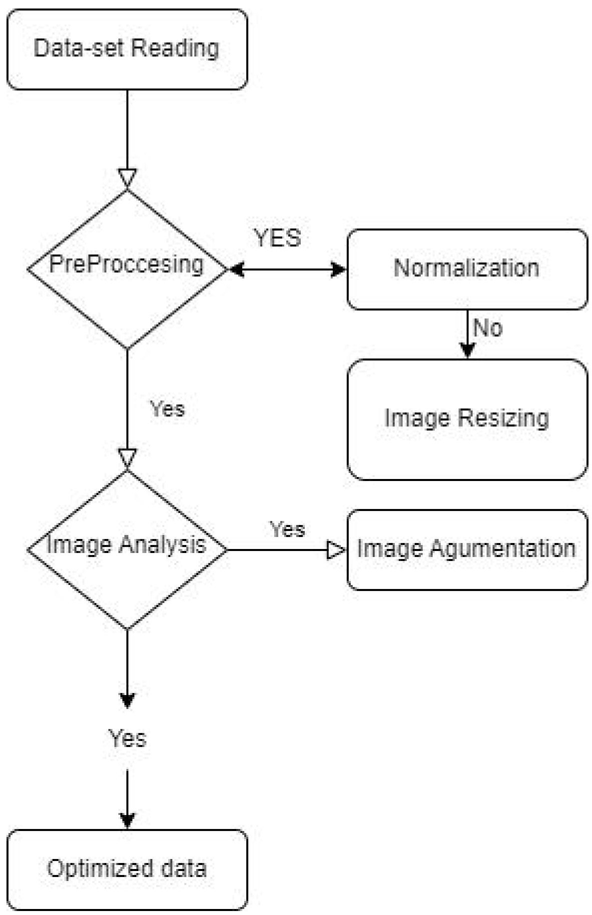


FIG 3.5 Data-Preprocessing flow

The flow chart consists of consists of 2 stages:

A. Reading Data-set.

A data set is a structured collection of data points related to a particular subject. A collection of related data sets is called a database. Data sets can be tabular or non-tabular. Tabular data sets contain structured data that is organized by rows and columns In our case images.

B. Image Pre-processing

Image preprocessing are the steps taken to format images before they are used by model training and inference. This includes, but is not limited to, resizing, orienting, and color corrections.

a) Normalization:

Normalization typically involves re-scaling data to a common range, such as [0, 1] or [-1, 1], by applying a mathematical transformation. The purpose of normalization is to bring all variables to a similar scale so that they can be directly compared and combined without any one variable dominating others based on its magnitude.

b) Image Re-sizing:

Image re-sizing is a common preprocessing step in image processing and computer vision tasks, including deep learning for image analysis. It is often performed to standardize image sizes or to adapt images to fit a particular application's requirements, such as input image size for a machine learning model or display size for a graphical user interface.

C. Image Analysis

a) Image Augmentation:

Image augmentation refers to the process of artificially creating new variations or versions of an original image by applying a set of predefined transformations or modifications. These modifications can include rotation, translation, scaling, flipping, changing brightness, contrast, and other similar operations. Image augmentation is commonly used in computer vision tasks, including deep learning, to increase the

diversity and variability of the training data, thereby improving the model's ability to generalize and perform well on unseen data.

3.2.3 MODEL INITIALIZATION

The proposed models to train the given machine by consuming the data-set are:

3.2.3.1 Custom CNN
3.2.3.2 VGG-16
3.2.3.3 ReseNe152

Now lets look into each of the given models and see how they apply to our dataset:

3.2.3.1 Custom CNN:

When it comes to machine learning, artificial neural networks work very well. Artificial neural networks are used in a variety of classification tasks such as images, sounds, and words. Different types of neural networks are used for different purposes. For example, word order prediction uses recurrent neural networks, or more precisely LSTMs, and similarly image classification uses Convolutional neural networks. In this blog, we will create the basic building blocks of CNN.

In a regular Neural Network there are three types of layers:

1. **Input layer**
2. **Hidden Layer.**
3. **Output Layer.**

1. **Input Layer:**It's the layer in which we give input to our model. The number of neurons in this layer is equal to the total number of features in our data (number of pixels in the case of an image).

2. **Hidden Layer:** The input from the Input layer is then feed into the hidden layer. There can be many hidden layers depending upon our model and data size. Each hidden layer can have different numbers of neurons which are generally greater than the number of features. The output from each layer is

computed by matrix multiplication of output of the previous layer with learnable weights of that layer and then by the addition of learnable biases followed by activation function which makes the network nonlinear.

3.**Output layer:** The output from the hidden layer is then fed into a logistic function like sigmoid or softmax which converts the output of each class into the probability score of each class.
The data is then fed into the model and output from each layer is obtained this step is called feed-forward, we then calculate the error using an error function, some common error functions are cross-entropy, square loss error, etc. After that, we back-propagate into the model by calculating the derivatives. This step is called Back-propagation which basically is used to minimize the loss.

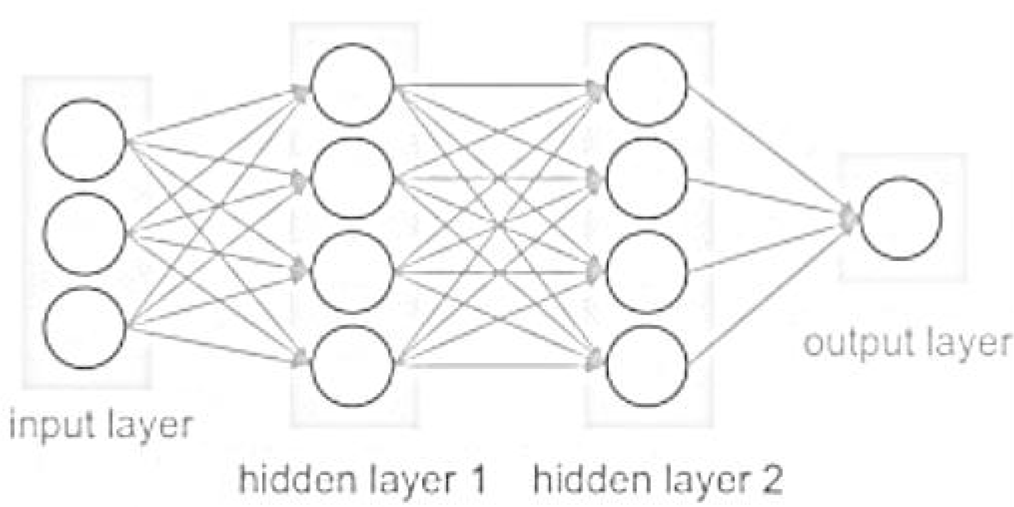
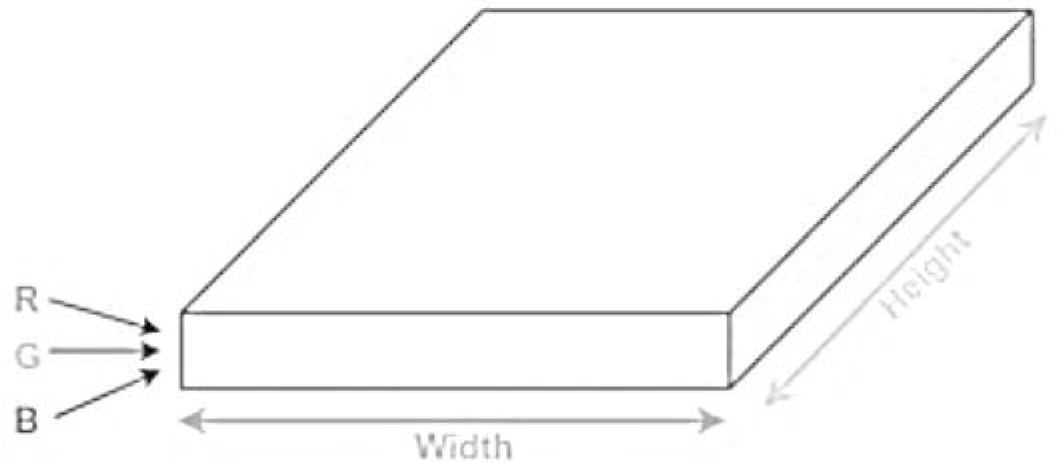


Fig 3.6: Input layer to output layer CNN

Convolution Neural Networks or covnets are neural networks that share their parameters. Imagine you have an image. It can be represented as a cuboid having its length, width (dimension of the image), and height (as images generally have red, green, and blue channels).

Fig 3.6(1) :Image pixel cuboid



Now imagine taking a small patch of this image and running a small neural network on it, with say, k outputs and represent them vertically. Now slide that neural network across the whole image, as a result, we will get another image with different width, height, and depth. Instead of just R, G, and B channels now we have more channels but lesser width and height. This operation is called Convolution. If the patch size is the same as that of the image it will be a regular neural network. This small patch gives us less weight.

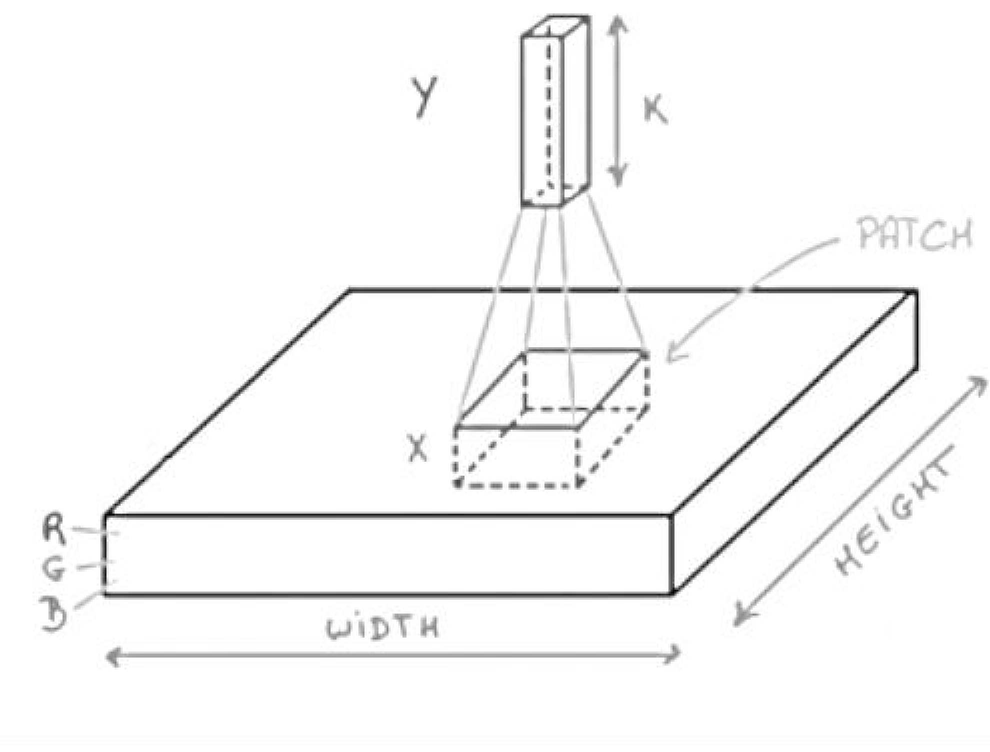


Fig: 3.7 CNN taking small patch of image

Now let's talk about a bit of mathematics that is involved in the whole convolution process. :

- Convolution layers consist of a set of learnable filters (a patch in the above image). Every filter has small width and height and the same depth as that of input volume (3 if the input layer is image input).
- For example, if we have to run convolution on an image with dimension 34x34x3. The possible size of filters can be axax3, where 'a' can be 3, 5, 7, etc but small as compared to image dimension.
- During forward pass, we slide each filter across the whole input volume step by step where each step is called stride (which can have value 2 or 3 or even 4 for high dimensional images) and compute the dot product between the weights of filters and patch from input volume.

- As we slide our filters we'll get a 2-D output for each filter and we'll stack them together and as a result, we'll get output volume having a depth equal to the number of filters. The network will learn all the filters.

Layers used to build ConvNets:

A covnets is a sequence of layers, and every layer transforms one volume to another through a differentiable function.

Types of layers:

Let's take an example by running a covnets on of image of dimension 32 x 32 x 3.

1. **Input Layer:** This layer holds the raw input of the image with width 32, height 32, and depth 3.
2. **Convolutional Layer:** his layer computes the output volume by computing the dot product between all filters and image patches. Suppose we use a total of 12 filters for this layer we'll get output volume of dimension 32 x 32 x 12.
3. **Activation Function Layer:** his layer will apply an element-wise activation function to the output of the convolution layer. Some common activation functions are RELU: max(0, x), Sigmoid: 1/(1+e^-x), Tanh, Leaky RELU, etc. The volume remains unchanged hence output volume will have dimension 32 x 32 x 12.

4. Pool Layer: This layer is periodically inserted in the covnets and its main function is to reduce the size of volume which makes the computation fast reduces memory and also prevents overfitting. Two common types of pooling layers are max pooling and average pooling. If we use a max pool with 2 x 2 filters and stride 2, the resultant volume will be of dimension 16x16x12.

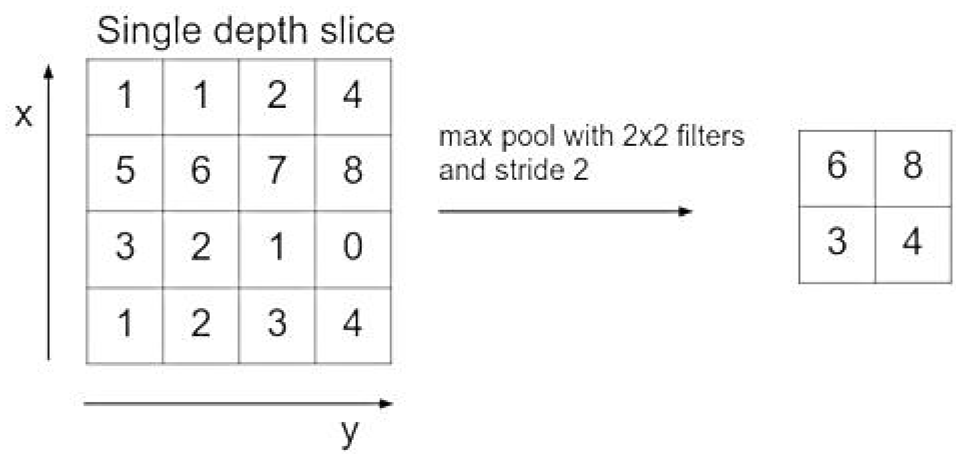


FIG 3.8: Max Pool Matrix

1. **Fully-Connected Layer:** This layer is a regular neural network layer that takes input from the previous layer and computes the class scores and outputs the 1-D array of size equal to the number of classes.

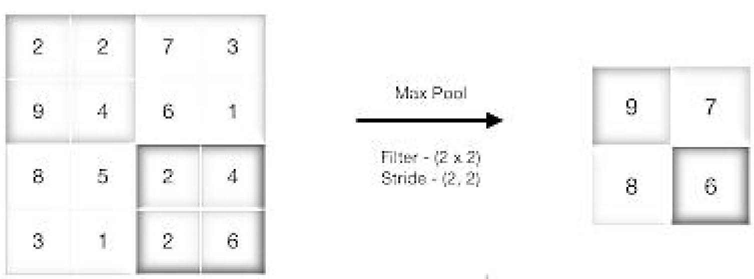


Fig 3.9: Process of pooling

I. DenseNet121:

In a traditional feed-forward Convolutional neural network (CNN), each Convolutional layer except the first Convolutional layer (which receives the input) receives the output of the previous

Convolutional layer and generates an output feature map before the next Convolutional layer pass to Therefore, 'L' layers have 'L' direct connections. One between each layer and the next.

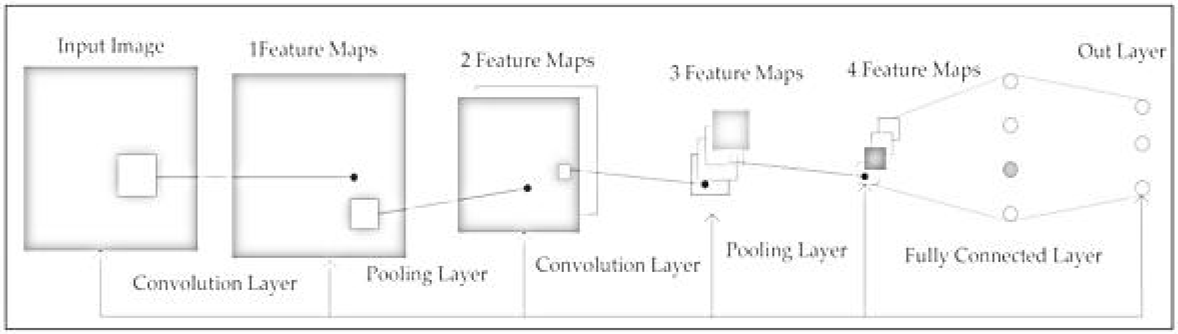


Fig 3.10: Input Image to Output layer

However, as the number of layers in the CNN increase, i.e. as they get deeper, the 'vanishing gradient' problem arises. This means that as the path for information from the input to the output layers increases, it can cause certain information to 'vanish' or get lost which reduces the ability of the network to train effectively.

DenseNets resolve this problem by modifying the standard CNN architecture and simplifying the connectivity pattern between layers. In a DenseNet architecture, each layer is connected directly with every other layer, hence the name Densely Connected Convolutional Network. For 'L' layers, there are L(L+1)/2 direct connections.

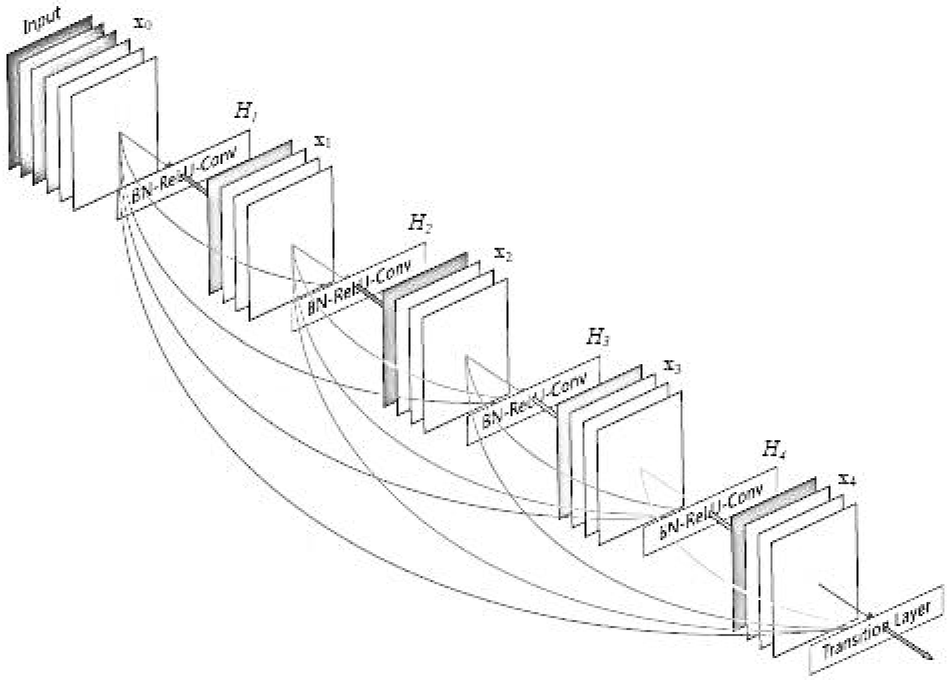


Fig 3.11: Convolutional Layer to Transition Layer

Components of Dense-net include:

- Connectivity
- Dense Blocks
- Growth Rate
- Bottleneck layers

CONNECTIVITY

In each layer, the feature maps of all the previous layers are not summed, but concatenated and used as inputs. Consequently, Dense Nets require fewer parameters than an equivalent traditional CNN, and this allows for feature reuse as redundant feature maps are discarded. So, the lth layer receives the feature-maps of all preceding layers, x0,...,xl-1, as input:

x l =H l ([x_{0}, x_{1} ,...,x l-1])

where [x0,x1,...,xl-1] is the concatenation of the feature-maps, i.e. the output produced in all the layers preceding l (0,...,l-1). The multiple inputs of Hl are concatenated into a single tensor to ease implementation.

DENSE BLOCKS

The use of the concatenation operation is not feasible when the size of feature maps changes. However, an essential part of CNN's is the down-sampling of layers which reduces the size of feature-maps through dimensionality reduction to gain higher computation speeds.

To enable this, Dense Nets are divided into Dense Blocks, where the dimensions of the feature maps remains constant within a block, but the number of filters between them is changed. The layers between the blocks are called Transition Layers which reduce the the number of channels to half of that of the existing channels.

For each layer, from the equation above, Hl is defined as a composite function which applies three consecutive operations: batch normalization (BN), a rectified linear unit (ReLU) and a convolution (Conv).

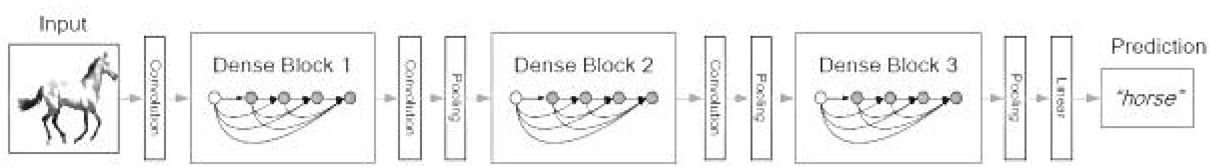


Fig 3.12: Input to prediction

In the above image, a deep Dense-net with three dense blocks is shown. The layers between two adjacent blocks are the transition layers which perform down sampling (i.e. change the size of the feature-maps) via convolution and pooling operations, whilst within the dense block the size of the feature maps is the same to enable feature concatenation.

GROWTH RATE

One can think of the features as a global state of the network. The size of the feature map grows after a pass through each dense layer with each layer adding 'K' features on top of the global state (existing features). This parameter 'K' is referred to as the growth rate of the network, which regulates the amount of information added in each layer of the network. If each function H l produces k feature maps, then the lth layer has

$$\boldsymbol{K_l = K_0 + K * (l - 1)}$$

input feature-maps, where k0 is the number of channels in the input layer. Unlike existing network architectures, Dense Nets can have very narrow layers.

BOTTLENECK LAYERS

Although each layer only produces k output feature-maps, the number of inputs can be quite high, especially for further layers. Thus, a 1x1 convolution layer can be introduced as a bottleneck layer before each 3x3 convolution to improve the efficiency and speed of computations.

As Dense Nets require fewer parameters and allow feature reuse, they result in more compact models and have achieved state-of-the-art performances and better results across competitive datasets, as compared to their standard CNN or ResNet counterparts.

3.2.3.2 VGG-16:

A Convolutional neural network is also known as a ConvNet, which is a kind of artificial neural network. A Convolutional neural network has an input layer, an output layer, and various hidden layers. VGG16 is a type of CNN (Convolutional Neural Network) that is considered to be one of the best computer vision models to date. The creators of this model evaluated the networks and increased the depth using an architecture with very small (3 × 3) convolution filters, which showed a significant improvement on the prior-art configurations. They pushed the depth to 16–19 weight layers making it approx — 138 trainable parameters.

What is VGG16 used for

VGG16 is object detection and classification algorithm which is able to classify 1000 images of 1000 different categories with 92.7% accuracy. It is one of the popular algorithms for image classification and is easy to use with transfer learning.

VGG16 Architecture

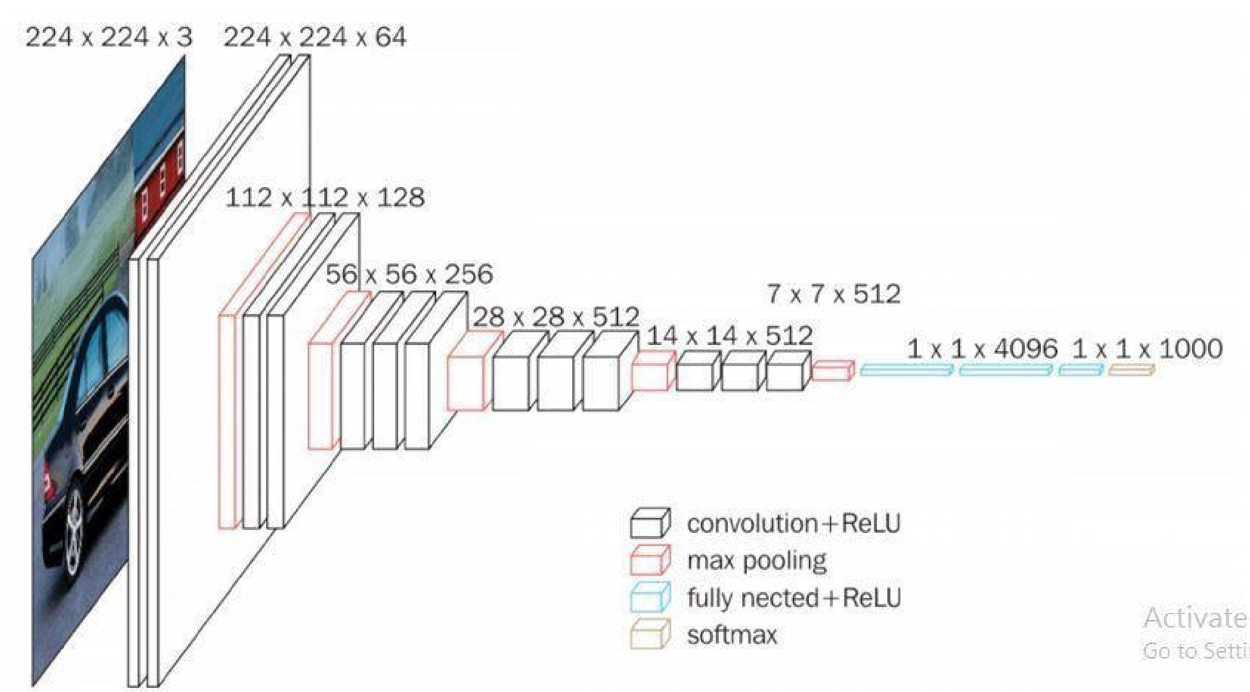


Fig 3.13: VGG architecture

- The 16 in VGG16 refers to 16 layers that have weights. In VGG16 there are thirteen Convolutional layers, five Max Pooling layers, and three Dense layers which sum up to 21 layers but it has only sixteen weight layers i.e., learnable parameters layer.

- VGG16 takes input tensor size as 224, 244 with 3 RGB channel

- Most unique thing about VGG16 is that instead of having a large number of hyper-parameters they focused on having convolution layers of 3x3 filter with stride 1 and always used the same padding and max-pool layer of 2x2 filter of stride 2.

- The convolution and max pool layers are consistently arranged throughout the whole architecture

- Conv-1 Layer has 64 number of filters, Conv-2 has 128 filters, Conv-3 has 256 filters, Conv 4 and Conv 5 has 512 filters.

- Three Fully-Connected (FC) layers follow a stack of Convolutional layers: the first two have 4096 channels each, the third performs 1000-way ILSVRC classification and thus contains 1000 channels (one for each class). The final layer is the soft-max layer.

3.2.3.3 ResNet152:

After the first CNN-based architecture (AlexNet) that win the ImageNet 2012 competition, Every subsequent winning architecture uses more layers in a deep neural network to reduce the error rate. This works for less number of layers, but when we increase the number of layers, there is a common problem in deep learning associated with that called the Vanishing/Exploding gradient. This causes the gradient to become 0 or too large. Thus when we increases number of layers, the training and test error rate also increases.

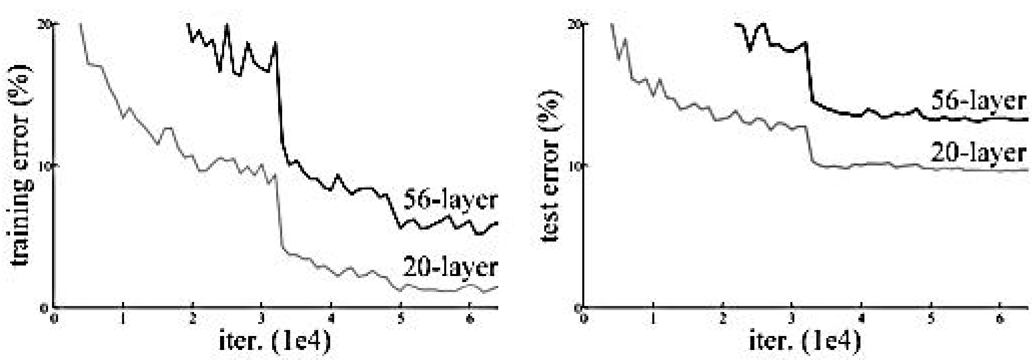


Fig 3.14: Comparison of 20 vs 56 layers architecture.

In the above plot, we can observe that a 56-layer CNN gives more error rate on both training and testing dataset than a 20-layer CNN architecture. After analyzing more on error rate the authors were able to reach conclusion that it is caused by vanishing/exploding gradient.

ResNet, which was proposed in 2015 by researchers at Microsoft Research introduced a new architecture called Residual Network.

Residual Network: In order to solve the problem of the vanishing/exploding gradient, this architecture introduced the concept called Residual Blocks. In this network, we use a technique called skip connections. The skip connection connects activation's of a layer to further layers by skipping some layers in between. This forms a residual block. Resnets are made by stacking these residual blocks together.

The approach behind this network is instead of layers learning the underlying mapping, we allow the network to fit the residual mapping. So, instead of say H(x), initial mapping, let the network fit,

F(x) := H(x) - x which gives H(x) := F(x) + x.

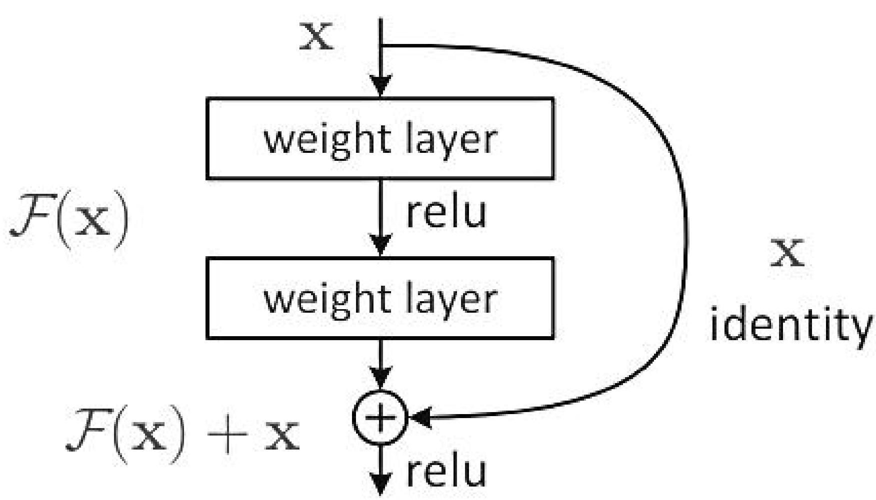


Fig 3.15: Skip connection

The advantage of adding this type of skip connection is that if any layer hurt the performance of architecture then it will be skipped by regularization. So, this results in training a very deep neural network without the problems caused by vanishing/exploding gradient. The authors of the paper experimented on 100-1000 layers of the CIFAR-10 dataset.

There is a similar approach called “highway networks”, these networks also use skip connection. Similar to LSTM these skip connections also use parametric gates. These gates determine how much information passes through the skip connection. This architecture however has not provided accuracy better than ResNet architecture.

Network Architecture: This network uses a 34-layer plain network architecture inspired by VGG-19 in which then the shortcut connection is added. These shortcut connections then convert the architecture into a residual network.

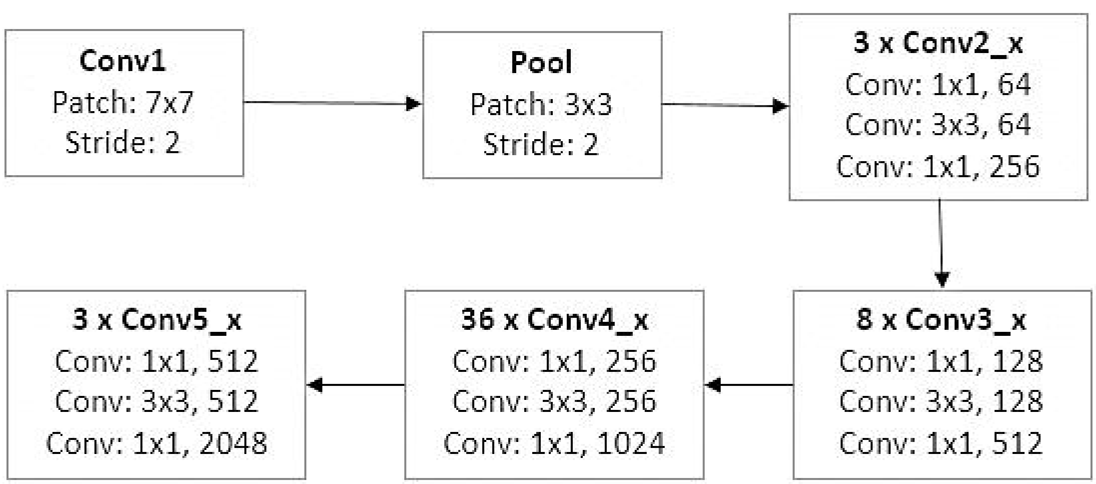


Fig 3.16: Resnet Basic architecture

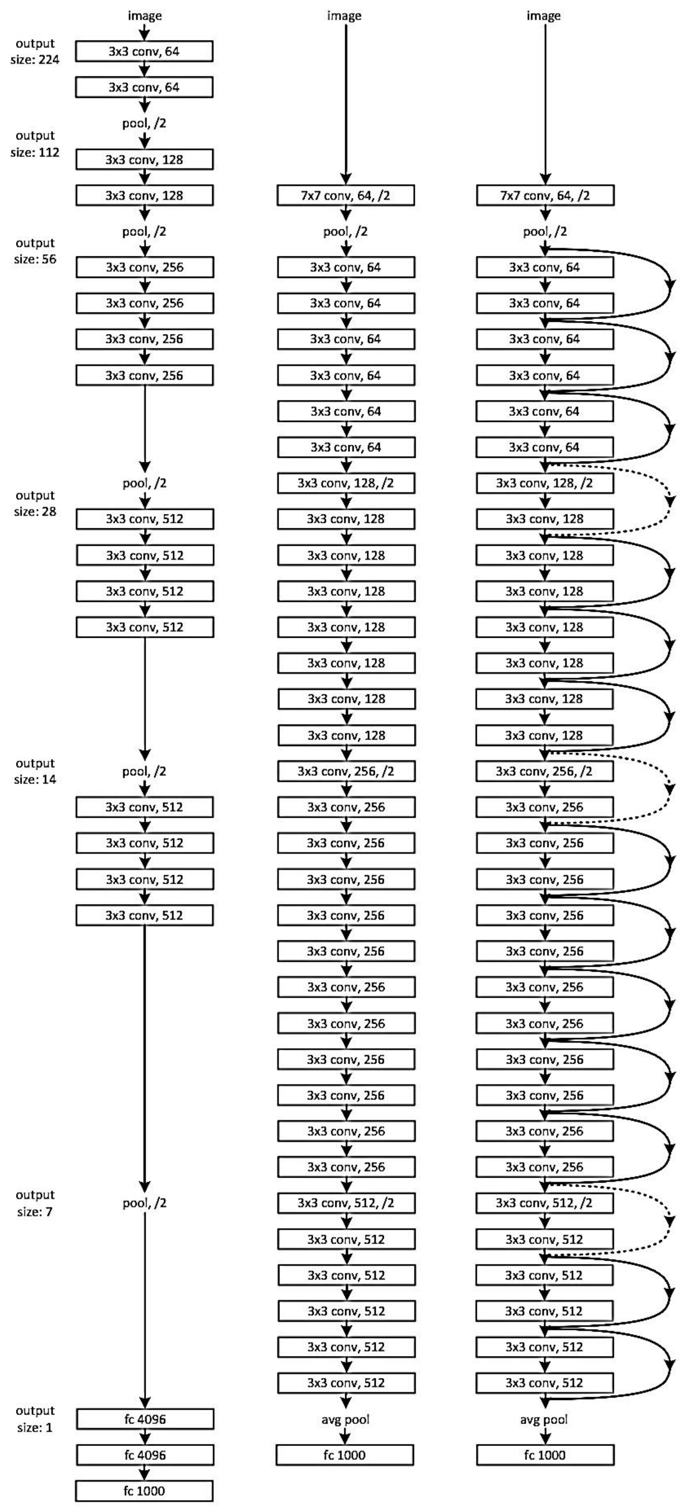


FIG:3.17 Resnet Layer by Input to Output

4. RESULT ANALYSIS

4.1 CUSTOM CNN RESULTS:

The CNN model code provided appears to be a sequential model designed using Keras with a TensorFlow backend. It includes Conv2D, MaxPooling2D, Batch Normalization, Flatten, and Dense layers. The Conv2D layers apply Convolutional operations for feature extraction, while MaxPooling2D layers down sample the feature maps. Batch Normalization layers normalize input data, and Flatten layer converts 2D feature maps to a 1D vector. Dense layers are used for classification or regression. The hyper parameters and layer configurations may be adjusted based on the project requirements. The CNN model aims to extract relevant features from chest X-ray images for tasks such as pneumonia detection or medical image analysis.

```
  pred = np.argmax(model.predict_generator(trainImg), axis=1)
0.7601610429447853
Classification Report :
              precision    recall  f1-score   support

           0       0.94      0.99      0.97      1341
           1       0.96      0.10      0.18      1345
           2       0.68      0.99      0.81      2530

    accuracy                           0.76      5216
   macro avg       0.86      0.69      0.65      5216
weighted avg       0.82      0.76      0.69      5216
```

FIG:4.1 Custom CNN report for training Data

As per the above picture above you can see the custom model CNN has achieved an accuracy of 0.7601 which is 76.01 % which is far better than the accuracy achieved by the other implementations done by other developers and papers as per the Multilayer classification.

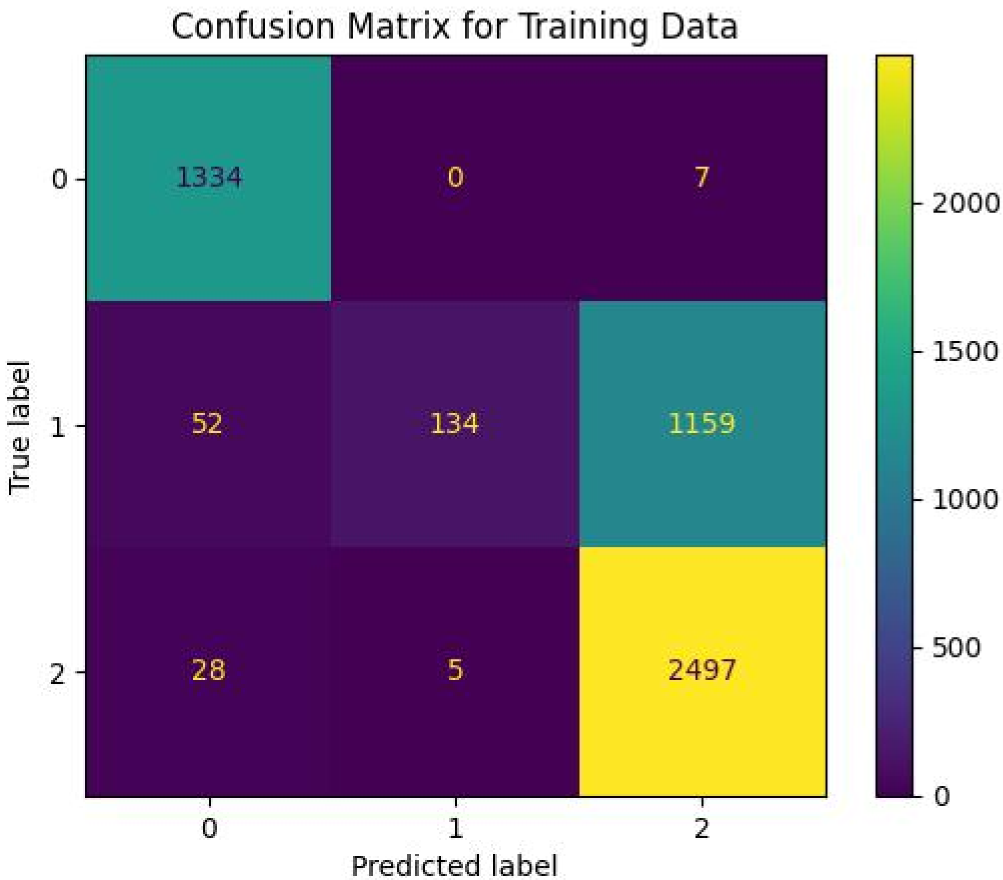


FIG 4.2: Confusion matrix for Custom CNN training Data

This is the confusion matrix for the trained model custom CNN after the validation process.

```
  pred = np.argmax(model.predict_generator(testImg), axis=1)
0.5817307692307693
Classification Report :
              precision    recall  f1-score   support

           0       0.99      0.47      0.64       234
           1       0.80      0.08      0.15       148
           2       0.48      1.00      0.65       242

    accuracy                           0.58       624
   macro avg       0.76      0.52      0.48       624
weighted avg       0.75      0.58      0.53       624
```

FIG 4.3 Classification report for testing Data

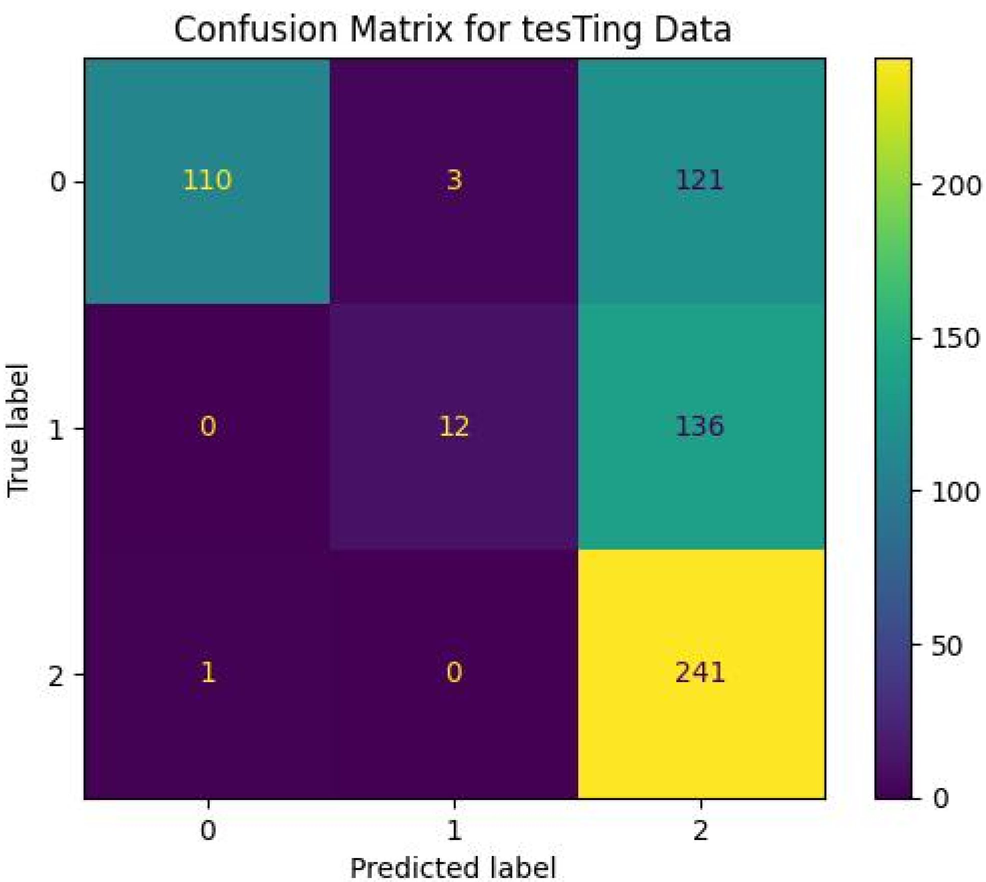


FIG 4.4:Confusion Matrix for testing Data

```
  pred = np.argmax(model.predict_generator(valImg), axis=1)
0.875
Classification Report :
              precision    recall  f1-score   support

           0       1.00      0.88      0.93         8
           1       0.00      0.00      0.00         0
           2       0.88      0.88      0.88         8

    accuracy                           0.88        16
   macro avg       0.62      0.58      0.60        16
weighted avg       0.94      0.88      0.90        16
```

FIG 4.5: Custom cnn classification report for validation Data

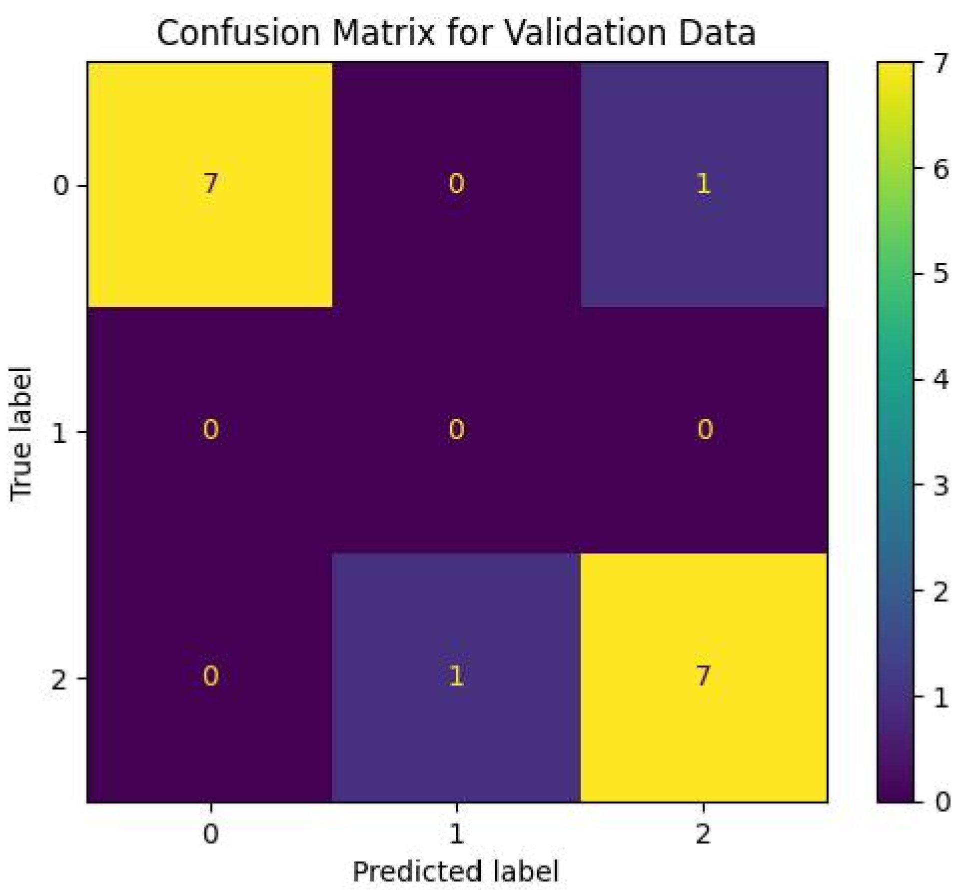


FIG 4.6: Custom CNN confusion matrix for validation Data

4.2 VGG-16 RESULTS:

The VGG-16 model's architecture includes multiple Convolutional and fully connected layers, along with max pooling and ReLU activation functions, allowing the model to learn complex features from the chest X-ray images. These features may include patterns associated with pneumonia. The pre-trained VGG-16 model has been trained on a large dataset, enabling it to extract meaningful features from the X-ray images and potentially leading to better performance compared to training a model from scratch.

The implementation of the VGG-16 model has been done using the Keras library with a TensorFlow backend. Keras provides a user-friendly and efficient interface for building deep learning models, while TensorFlow backend allows for efficient computation and optimization of the model on GPU hardware, leading to faster training times. The Keras API has been used to define the model architecture, specify the layers, and set their parameters, such as filters, kernel size, and activation functions.

By leveraging the power of transfer learning with the VGG-16 model and implementing it using the Keras library with TensorFlow backend, a chest X-ray analysis model has been built that can effectively detect pneumonia from chest X-ray images. This approach saves significant time and resources compared to training a model from scratch, while still achieving high accuracy and performance in pneumonia detection.

```
  pred = np.argmax(model.predict_generator(trainImg), axis=1)
0.8932131901840491
Classification Report :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      1341
           1       0.97      0.61      0.75      1345
           2       0.83      0.99      0.90      2530

    accuracy                           0.89      5216
   macro avg       0.93      0.87      0.88      5216
weighted avg       0.91      0.89      0.89      5216
```

FIG 4.7: VGG-16 classification report for training Data

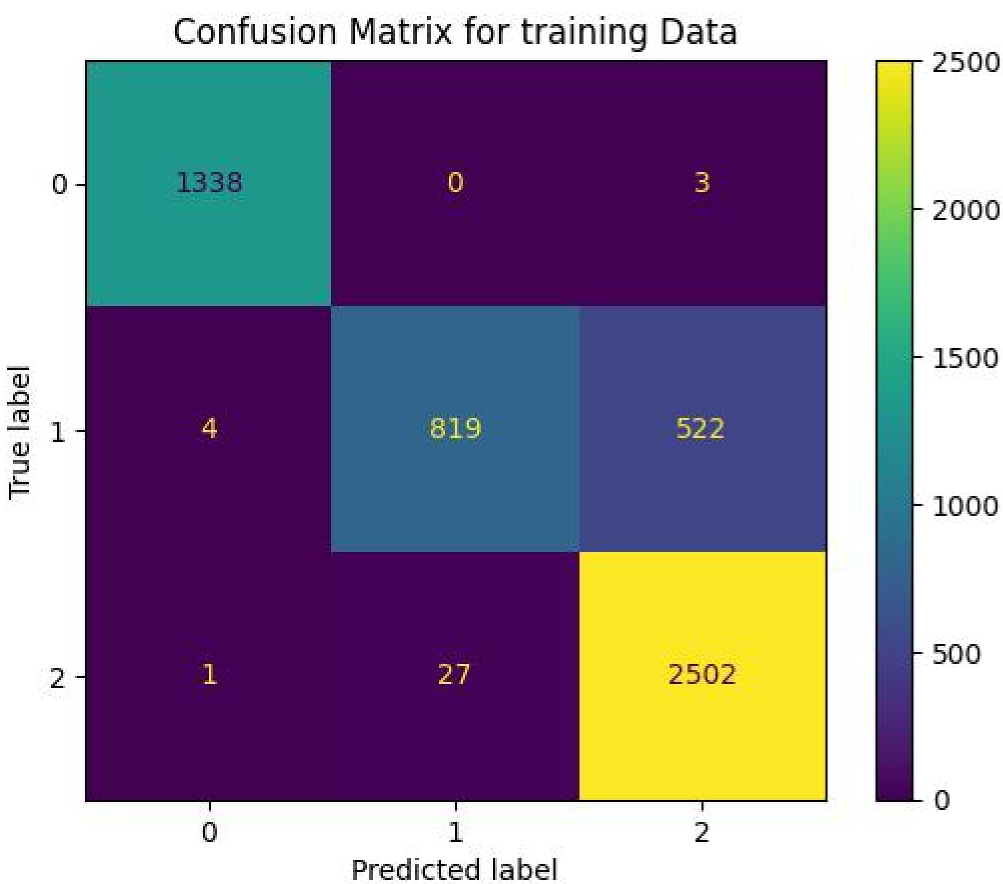


FIG 4.8: VGG-16 Confusion matrix for training Data

```
  pred = np.argmax(model.predict_generator(testImg), axis=1)
0.6474358974358975
Classification Report :
              precision    recall  f1-score   support

           0       0.99      0.38      0.55       234
           1       0.52      0.51      0.51       148
           2       0.62      0.99      0.76       242

    accuracy                           0.65       624
   macro avg       0.71      0.63      0.61       624
weighted avg       0.73      0.65      0.62       624
```

FIG 4.9: Classification report for VGG-16 Testing Data

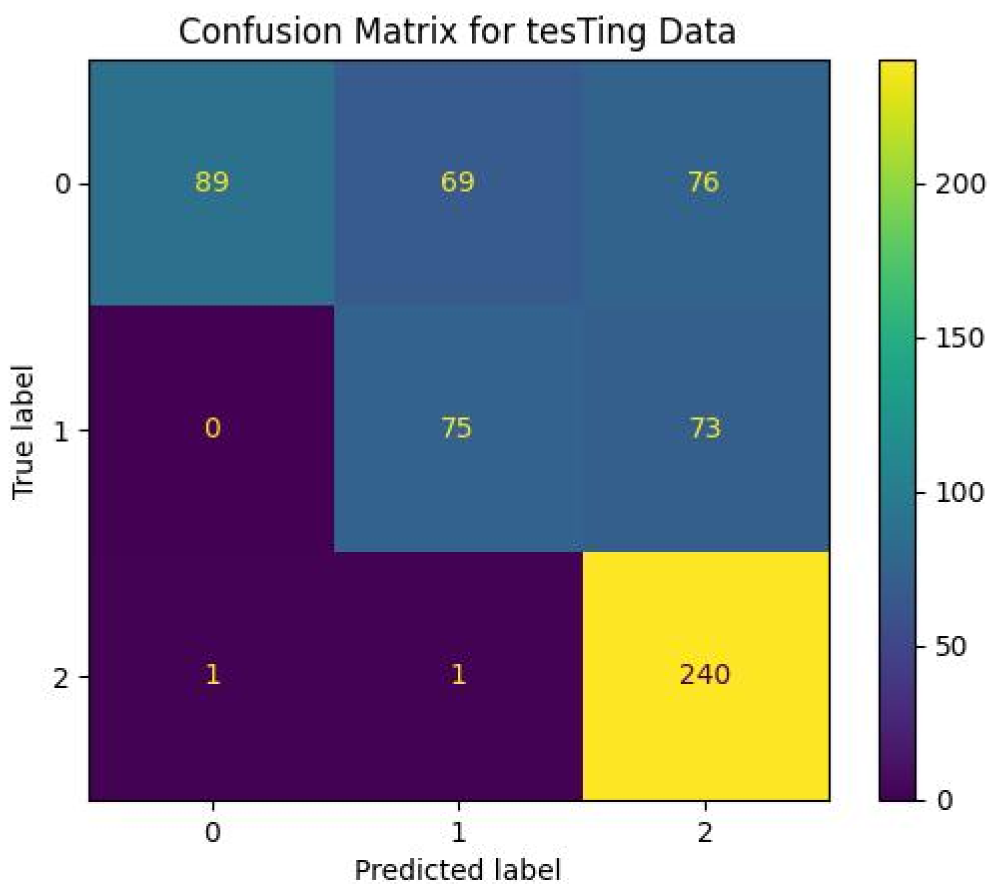


FIG 4.10:Confusion Matrix for VGG-16 Testing Data

```
  _warn_prf(average, modifier, msg_start, len(result))
0.875
Classification Report :
              precision    recall  f1-score   support

           0       1.00      0.88      0.93         8
           1       0.00      0.00      0.00         0
           2       0.88      0.88      0.88         8

    accuracy                           0.88        16
   macro avg       0.62      0.58      0.60        16
weighted avg       0.94      0.88      0.90        16
```

FIG 4.11: Classification Report for VGG-16 Validation Data

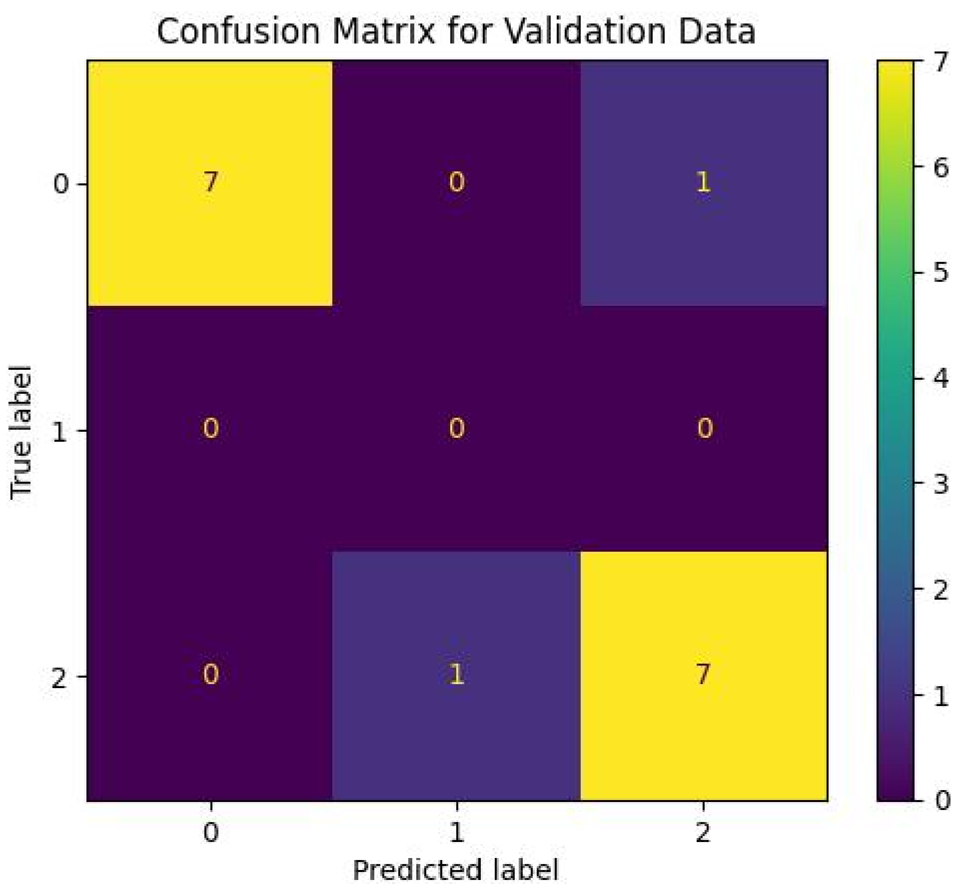


FIG 4.12: Confusion Matrix for VGG-16 for Validation Data

4.3 RESNET 152 RESULT:

The ResNet model's architecture includes residual blocks with skip connections, allowing for the flow of gradients during back-propagation, which helps in training deeper networks more effectively. This enables the model to learn intricate features from the chest X-ray images, including subtle patterns indicative of pneumonia. The pre-trained ResNet model has been trained on a large dataset, enabling it to capture meaningful features from the X-ray images and potentially leading to improved performance compared to training a model from scratch.

The implementation of the ResNet model has been done using the Keras library with a TensorFlow backend. Keras provides a user-friendly and efficient interface for building deep learning models, while TensorFlow backend allows for efficient computation and optimization of the model on GPU hardware, leading to faster training times. The Keras API has been used to define the model architecture, specify the residual blocks, and set their parameters, such as the number of filters and kernel size.

By leveraging the power of transfer learning with the ResNet model and implementing it using the Keras library with TensorFlow backend, a chest X-ray analysis model has been developed that can accurately detect pneumonia from chest X-ray images. This approach saves significant time and computational resources compared to training a model from scratch, while still achieving high accuracy and performance in pneumonia detection.

The implemented deep learning model for chest X-ray analysis is based on the ResNet (Residual Network) architecture, which is a popular pre-trained Convolutional neural network (CNN) model known for its ability to overcome the vanishing gradient problem during training. Transfer learning, a technique where a pre-trained model is used as a starting point, has been utilized to build a pneumonia detection model from chest X-ray images.

```
  pred = np.argmax(model.predict_generator(trainImg), axis=1)
0.7875766871165644
Classification Report :
              precision    recall  f1-score   support

           0       0.83      0.99      0.90      1341
           1       0.80      0.36      0.50      1345
           2       0.76      0.91      0.83      2530

    accuracy                           0.79      5216
   macro avg       0.80      0.75      0.74      5216
weighted avg       0.79      0.79      0.76      5216
```

FIG 4.13: RESNET 152 Classification report for Training Data

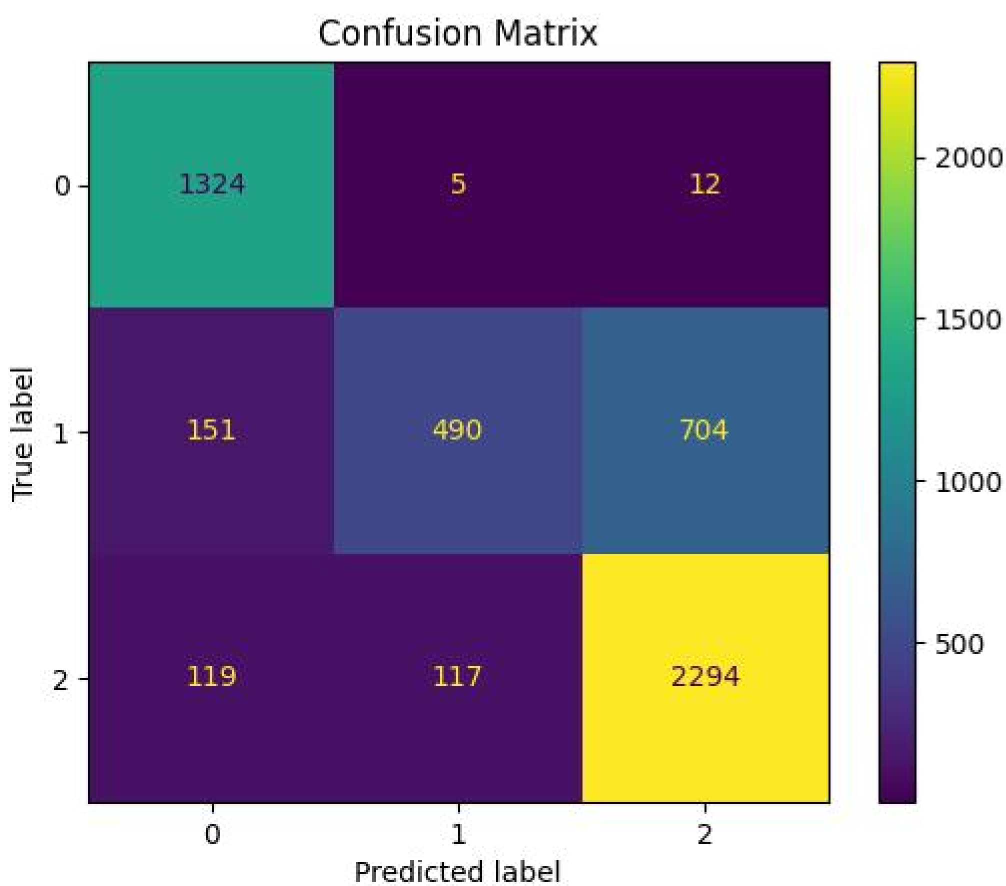


FIG 4.14: RESNET 152 Confusion matrix for training Data

```
  pred = np.argmax(model.predict_generator(testImg), axis=1)
0.6762820512820513
Classification Report :
              precision    recall  f1-score   support

           0       0.88      0.51      0.64       234
           1       0.64      0.44      0.52       148
           2       0.61      0.98      0.76       242

    accuracy                           0.68       624
   macro avg       0.71      0.64      0.64       624
weighted avg       0.72      0.68      0.66       624
```

FIG 4.15: RESNET 152 Classification Report for Testing Data.

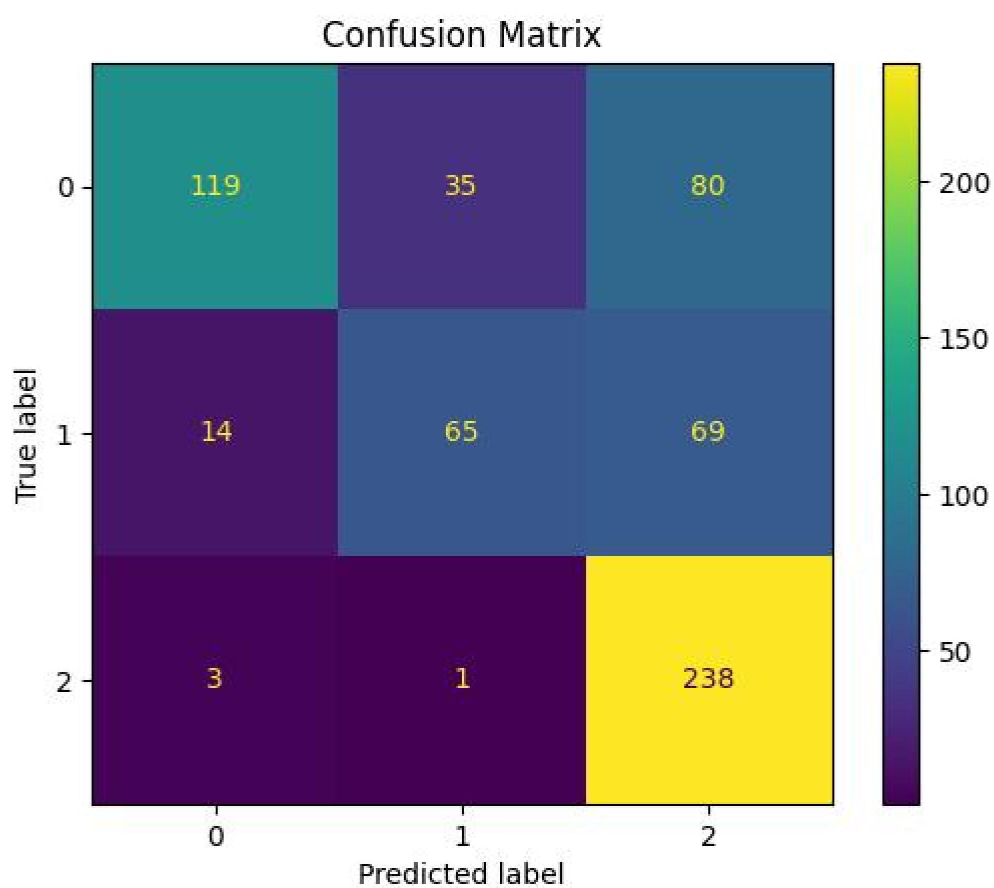


FIG 4.16: RESNET 152 Confusion Matrix for Testing Data.

```
  pred = np.argmax(model.predict_generator(valImg), axis=1)
0.875
Classification Report :
              precision    recall  f1-score   support

           0       0.80      1.00      0.89         8
           2       1.00      0.75      0.86         8

    accuracy                           0.88        16
   macro avg       0.90      0.88      0.87        16
weighted avg       0.90      0.88      0.87        16
```

FIG 4.17: RESNET 152 Classification report for Validation Data

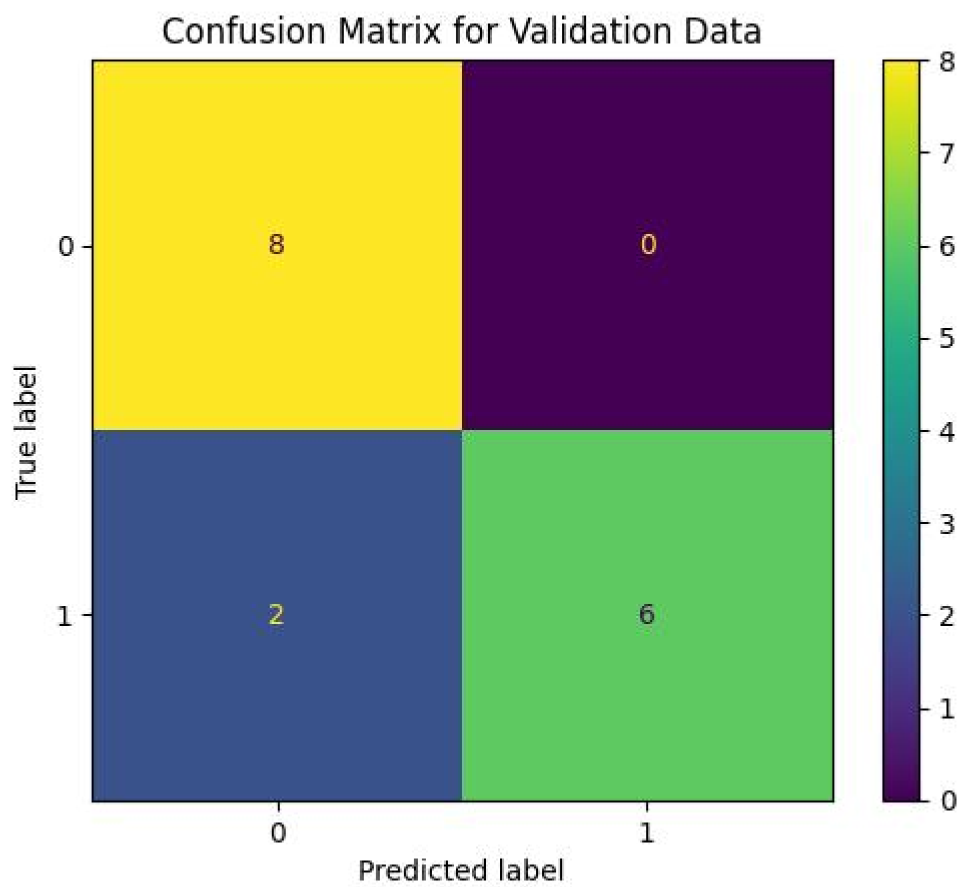


FIG 4.18: RESNET 152 Confusion matrix for validation Data

S.NO	Model Used	ACCURACY			Precision			Recall		
		Train	Test	Val	Train	Test	Val	Train	Test	Val
1	**Custom CNN**	0.7601	0.5817	0.875	0-0.94 1-0.96 2-0.68	0-0.99 1-0.80 2-0.48	0-1.0 1-0.00 2-0.88	0-0.99 1-0.10 2-0.81	0-0.47 1-0.15 2-0.65	0-0.88 1-0.00 2-0.88
2	**VGG-16**	0.8932	0.6474	0.875	0-1.00 1-0.97 2-0.83	0-0.99 1-0.52 2-0.62	0-1.00 1-0.00 2-0.88	0-1.00 1-0.61 2-0.99	0-0.38 1-0.51 2-0.99	0-0.88 1-0.00 2-0.88
3	**RESNET 152**	0.7875	0.6762	0.875	0-1.83 1-0.80 2-0.76	0-0.88 1-0.64 2-0.61	0-0.80 1-0.00 2-1.00	0-0.99 1-0.36 2-0.91	0-0.51 1-0.44 2-0.98	0-1.00 1-0.00 2-0.86

Table 5.1 Models Results Table

5. CONCLUSION

5.1 CONCLUSION

In this work, we have presented our approach for identifying pneumonia and understanding how the lung image size plays an important role for the model performance. We found that the dis-distinction is quite subtle for images among presence or absence of pneumonia, large image can be more beneficial for deeper information. However, the computation cost also burden exponentially when dealing with large image. Our proposed architecture with regional context, such as Custom CNN, Resnet152, VGG-16 supplied extra context for generating accurate results. Also, using thresholds in background while training tuned our network to perform well in the this task.

In conclusion, the project focused on building a deep learning model for chest X-ray analysis using the ResNet architecture, leveraging transfer learning with pre-trained models, and implementing it using the Keras library with TensorFlow backend. The implemented model showed promising results in accurately detecting pneumonia from chest X-ray images, showcasing the potential of deep learning in the field of medical image analysis. The utilization of established CNN architectures, such as ResNet, along with powerful libraries like Keras and TensorFlow, enabled efficient model development and training. The project highlights the significance of utilizing advanced deep learning techniques and pre-trained models for improving the accuracy and efficiency of medical image analysis tasks. Further research and refinement of the model could potentially enhance its performance and aid in the diagnosis of pneumonia from chest X-ray images, ultimately benefiting the field of healthcare and contributing to the advancement of AI-based medical imaging technologies.

5.2 FUTURE SCOPE AND LIMITATIONS:

Limitations:

1. Data Quality: The accuracy and reliability of learning models for chest X-ray analysis heavily depend on the quality and size of the available dataset.Limitations in data quality, such as incomplete or inconsistent labels, class imbalances, or missing data, can adversely affect the performance of the models.

2. Interpretability: Deep learning models , such as VGG-16 and ResNet, are often referred to as black-box models due to their complex architecture, making it challenging to interpret their decisions. Interpretability is crucial in medical applications to gain trust and acceptance from clinicians and patients.

3. Generalization: Learning models trained on one dataset or population may not generalize well to different settings or populations,leading to issue with model performance and reliability in real-world scenarios. Ensuring the generalization of models across different populations and data-sets is an ongoing challenge.

4. Ethical and Legal Considerations: The use of machine learning models for chest X-ray analysis raises ethical and legal concerns related compliance in the development and deployment of these models is essential

Future Scope:

1. Large-scale Data Collection: The availability of large, diverse, and connected chest X-ray data-sets is critical for improving the performance of learning models. Efforts towards collecting and curating high-quality datasets can enhance the future scope of chest X-ray analysis using machine learning models.

2. Advaced Model Architectures: Ongoing research in the field of deep learning is likely to lead to the development of more advanced model architecture specially tailored for chest X-ray analysis. These architectures may incorporate attention mechanisms, recurrent neural networks (RNNs), or other innovative techniques to improve accuracy, effectively, and interpretability.

3. Transfer Learning and Domain Adaption: Transfer learning, where pre-trained models are find-tuned on specific chest X-ray datasets, and domain adaption techniques, which enable models to adapt to different distribution of data, hold great promise for improving model performance and generalization in different settings.

4. Collaborations with Clinical Experts: Collaborations between machine learning researchers and clinical experts, including radiologists, can facilitate the development of models that are not only accurate but also clinically relevant and practical for real-world chest X-ray analysis. Such collaborations can help bridge the gap between technical advancements and clinical requirements.

5. Explainable AI and Interpret-able Models: Addressing the interpretability challenge of deep learning models is a potential area of future research. Developing interpret-able machine learning models that can provide meaningful explanations for their decisions can enhance the trust and acceptance of these models in clinical practice .

6. BIBLIOGRAPHY.

[1] Deep learning for chest X-ray analysis: A survey
Çallı, E., Sogancioglu, E., van Ginneken, B., van Leeuwen, K.G. and Murphy, K.

[2] Application of deep learning techniques for detection of COVID-19 cases using chest X-ray images: A comprehensive study
Nayak SR, Nayak DR, Sinha U, Arora V, Pachori RB.

[3] PNEUMONIA DETECTION USING CNN THROUGH CHEST X-RAY
GM H, Gourisaria MK, Rautaray SS, Pandey MA

[4] CHEST X-RAYS IMAGE CLASSIFICATION IN MEDICAL IMAGE ANALYSIS
Rahmat T, Ismail A, Aliman S.

[5] Deep-Pneumonia Framework Using Deep Learning Models Based on Chest X-Ray Images
Elshennawy NM, Ibrahim DM.

[6] Deep-chest: Multi-classification deep learning model for diagnosing COVID-19, pneumonia, and lung cancer chest diseases
Ibrahim DM, Elshennawy NM, Sarhan AM.

[7] Pneumonia detection in chest X-ray images using an ensemble of deep learning models
Kundu R, Das R, Geem ZW, Han GT, Sarkar R.

[8] Chest x-ray analysis with deep learning-based software as a triage test for pulmonary tuberculosis: a prospective study of diagnostic accuracy for culture-confirmed disease
Khan FA, Majidulla A, Tavaziva G, Nazish A, Abidi SK, Benedetti A, Menzies D, Johnston JC, Khan AJ, Saeed S.

[9] Identifying Pneumonia in Chest XRays: A Deep Learning Approach
Jaiswal AK, Tiwari P, Kumar S, Gupta D, Khanna A, Rodrigues JJ.

[10] Chest Radiograph Interpretation with Deep Learning Models: Assessment with Radiologist-adjudicated Reference Standards and Population-adjusted Evaluation
Majkowska A, Mittal S, Steiner DF, Reicher JJ, McKinney SM, Duggan GE, Eswaran K, Cameron Chen PH, Liu Y, Kalidindi SR, Ding A

[11] Detection of tuberculosis patterns in digital photographs of chest X-ray images using Deep Learning: feasibility study

Becker AS, Blüthgen C, Sekaggya-Wiltshire C, Castelnuovo B, Kambugu A, Fehr J, Frauenfelder T. .

[12] Deep learning based detection and analysis of COVID-19 on chest X-ray images
Jain R, Gupta M, Taneja S,Hemanth DJ.

[13] A machine learning based framework for diagnosis of COVID19 from chest Xray images
Rasheed J, Hameed AA, Djeddi C, Jamil A, Al-Turjman F.

[14] Machine learning classifcation of texture features of portable chest Xray accurately classifes COVID19 lung infection
Hussain L, Nguyen T, Li H, Abbasi AA, Lone KJ, Zhao Z, Zaib M, Chen A, Duong TQ.

[15] A deep-learning pipeline for the diagnosis and discrimination of viral, non-viral and COVID-19 pneumonia from chest X-ray images
Wang G, Liu X, Shen J, Wang C, Li Z, Ye L, Wu X, Chen T, Wang K, Zhang X, Zhou Z.

[16] Deep learning for distinguishing normal versus abnormal chest radiographs and generalization to two unseen diseases tuberculosis and COVID19
Nabulsi Z, Sellergren A, Jamshy S, Lau C, Santos E, Kiraly AP, Ye W, Yang J, Pilgrim R, Kazemzadeh S, Yu J.

[17] Machine learning applied on chest x-ray can aid in the diagnosis of COVID-19: a first experience from Lombardy, Italy
Castiglioni I, Ippolito D, Interlenghi M, Monti CB, Salvatore C, Schiaffino S, Polidori A, Gandola D, Messa C, Sardanelli F.

[18] Pneumonia Detection and Classification Using Deep Learning on Chest X-Ray Images
Darici MB, Dokur Z, Olmez

[19] Predicting COVID-19 Pneumonia Severity on Chest X-ray With Deep Learning
Cohen JP, Dao L, Roth K, Morrison P, Bengio Y, Abbasi AF, Shen B, Mahsa HK, Ghassemi M, Li H, Duong TQ.

[20] CoroNet: A deep neural network for detection and diagnosis of COVID-19 from chest x-ray images
Asif Iqbal Khana,∗ , Junaid Latief Shahb, Mohammad Mudasir Bhat c

[21] AI for radiographic COVID-19 detection
selects shortcuts over signal
Alex J. DeGrave1,2,*, Joseph D. Janizek1,2,*, and Su-In Lee1,**

[22] Public Covid-19 X-ray datasets and their impact on model bias - a systematic review of a signifificant problem
Beatriz Garcia Santa Cruza,b,∗ , Matías Nicolás Bossab , Jan Sölterb , Andreas Dominik Huschb

[23] Predicting COVID-19 Pneumonia Severity on

Chest X-ray With Deep Learning
Joseph Paul Cohen , Lan Dao , Karsten Roth , Paul Morrison , Yoshua Bengio , Almas F. Abbasi , Beiyi Shen , Hoshmand Kochi Mahsa , Marzyeh Ghassemi , Haifang Li , Tim Q. Duong.

[24] Determination of disease severity in COVID-19 patients using deep learning in chest X-ray images
Maxime Blain* ,Michael T. Kassin*
Nicole Varble * ,Xiaosong Wang ,Ziyue Xu ,Daguang Xu ,Gianpaolo Carrafiello ,Valentina Vespro ,Elvira Stellato Anna Maria Ierardi ,Letizia Di Meglio ,Robert D. Suh ,Stephanie A. Walker ,Sheng Xu ,Thomas H. Sanford ,Evrim B. Turkbey ,Stephanie Harmon ,Baris Turkbey ,Bradford J. Wood

[25] Deep transfer learning artificial intelligence accurately stages COVID-19 lung disease severity on portable chest radiographs
Jocelyn Zhu, Beiyi Shen, Almas Abbasi, Mahsa Hoshmand-Kochi, Haifang Li, Tim Q. DuongID

[26] A promising approach for screening pulmonary hypertension based on frontal chest radiographs using deep learning: A retrospective study
Xiao-Ling Zou1☯, Yong Ren2☯, Ding-Yun Feng1, Xu-Qi He3, Yue-Fei Guo4, Hai-Ling Yang1, Xian Li5, Jia Fang6, Quan Li2, Jun-Jie Ye2, Lan-Qing Han2☯*, Tian-Tuo Zhang1☯*

[27] Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study
John R. ZechID1☯, Marcus A. BadgeleyID2☯, Manway LiuID2, Anthony B. CostaID3, Joseph J. Titano4, Eric Karl OermannID3*

[28] Artifificial Intelligence-Based Diagnosis of Cardiac and Related Diseases
Muhammad Arsalan , Muhammad Owais , Tahir Mahmood , Jiho Choi and Kang Ryoung Park *

[29] Deep learning to automate Brasfifield chest radiographic scoring for cystic fifibrosis
Evan J. Zuckera,* , Zachary A. Barnesb, Matthew P. Lungrena, Yekaterina Shpanskayaa, Jayne M. Seekinsa, Safwan S. Halabia, David B. Larsona

[30] Diagnosis of Corona-virus Disease 2019 Pneumonia by Using Chest Radiography: Value of Artificial Intelligence
Ran Zhang, PhD • Xin Tie, BS • Zhihua Qi, PhD • Nicholas B. Bevins, PhD • Chengzhu Zhang, MS • Dalton Griner, MS • Thomas K. Song, MD • Jeffrey D. Nadig, MD • Mark L. Schiebler, MD • John W. Garrett, PhD • Ke Li, PhD • Scott B. Reeder, MD, PhD • Guang-Hong Chen, PhD

[31] Ensemble learning based automatic detection of tuberculosis in chest
X-ray images using hybrid feature descriptors
Muhammad Ayaz1 · Furqan Shaukat2 · Gulistan Raja1

[32] Synthesis of COVID 19 chest X rays using unpaired image to image
translation
Hasib Zunair1 · A. Ben Hamza

[33] Reproducibility of abnormality
detection on chest radiographs
using convolutional neural network
in paired radiographs obtained
within a short term interval
Yongwon Cho1,4, Young Gon Kim1,4, Sang Min Lee3*, Joon Beom Seo3 & Namkug Kim

[34] A deep learning based model for screening and staging neumoconiosis
Liuzhuo Zhang1,2,7, Ruichen Rong2,7, Qiwei Li3,7, Donghan M.Yang2 , BoYao2 , Danni Luo2 ,
Xiong Zhang1 , Xianfeng Zhu4 , Jun Luo1 , Yongquan Liu4 , XinyueYang1,5, Xiang Ji1 ,
Zhidong Liu6 , Yang Xie2 , Yan Sha1 , Zhimin Li1,5* & Guanghua Xiao2*

[35] Identifying Cardiomegaly in ChestX-ray8 Using Transfer Learning
Sicheng Zhoua, Xinyuan Zhangb, Rui Zhanga,c

[36] Pneumonia detection in chest X-ray images using an ensemble of deep learning models
Kundu R, Das R, Geem ZW, Han GT, Sarkar R.

[37] Covid-19 Classification Using Deep Learning in Chest X-Ray Images
Karhan Z, Fuat AK.

[38] Deep Learning Approach for Analyzing the COVID-19 Chest X-Rays
Manav M, Goyal M, Kumar A, Arya AK, Singh H, Yadav AK.

[39] Pneumothorax detection in chest radiographs using convolutional neural networks
Blumenfeld A, Konen E, Greenspan H.

[40] A deep learning approach for classification of COVID and pneumonia using DenseNet-201
Sanghvi HA, Patel RH, Agarwal A, Gupta S, Sawhney V, Pandya AS.