# CHEST X-RAY CLASSIFICATION USING DEEP LEARNING

**A PROJECT REPORT**

***Submitted by***


**SASMITA SAHOO (210301120081)**

**PRANGYA PARAMITA PANI (210301120118)**

**MAMALI SAHOO (210301120142)**

**OMM PRASAD SINGH (210301120138)**


***in partial fulfilment for the award of the***

***degree Of***

**BACHELOR OF TECHNOLOGY**

***in***

**COMPUTER SCIENCE AND ENGNIEERING**

Centurion
UNIVERSITY
*Shaping Lives...*
*Empowering Communities...*


**SCHOOL OF ENGINEERING AND TECHNOLOGY**

**BHUBANESWAR CAMPUS**

**CENTURION UNIVERSITY OF TECHNOLOGY AND MANAGEMENT**

**ODISHA**


**November, 2023**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**SCHOOL OF ENGINEERING AND TECHNOLOGY**

**BHUBANESWAR CAMPUS**

# BONAFDE CERTIFICATE

Certified that this project report "**Chest X-ray Classification using Deep learning**" is the Bonafide work of **"SASMITA SAHOO, MAMALI SAHOO, OMM PRASAD SINGH, PRANGYA PARAMITA PANI"** who carried out the project work under my supervision. This is to further certify to the best of my knowledge that this project has not been carried out earlier in this institute and the university

**SIGNATURE**

**Dr. Chinmayee Dora**

**Professor Of Department of Electronics and Communication Engineering**

*Certified that the above-mentioned project has been duly carried out as per the norms of the college and statutes of the university*

**SIGNATURE**

**Prof. Raj Kumar Mohanta**

**HEAD OF THE DEPARTMENT**

**Professor of Computer Science & Engineering**

DEPARTMENT SEAL

# DECLARATION

We hereby declare that the project entitled **“Chest X-ray Classification using Deep learning”** submitted for the “Minor Project” of 5th semester B. Tech in Computer Science and Engineering is my original work and the project has not formed the basis for the award of any Degree / Diploma or any other similar titles in any other University / Institute.

**Name of the Students:**

**SASMITA SAHOO (210301120081)**

**PRANGYA PARAMITA PANI (210301120118)**

**MAMALI SAHOO (210301120142)**

**OMM PRASAD SINGH (210301120138)**

**Signature of the Student:**

**Place: Jatni, Odisha**

**Date:**

# ACKNOWLEDGEMENTS

I wish to express my profound and sincere gratitude to **Dr. Chinmayee Dora**, Department of Electronics and Communication Engineering SoET, Bhubaneswar Campus, who guided me into the intricacies of this project nonchalantly with matchless magnanimity.

I thank **Prof. Raj Kumar Mohanta**, Head of the Dept. of Department of Computer Science and Engineering, SoET, Bhubaneswar Campus and **Dr. Sujata Chakravarty**, Dean, School of Engineering and Technology, Bhubaneswar Campus for extending their support during Course of this investigation.

I would be failing in my duty if I don't acknowledge the cooperation rendered during various stages of interpretation by team members.

I am highly grateful to the teachers who evinced keen interest and invaluable support in the progress and successful completion of my project work.

I am indebted to my collegues for their constant encouragement, co-operation and help. Words of gratitude are not enough to describe the accommodation and fortitude which they have shown throughout my endeavor.

**Name of the Students:**

**SASMITA SAHOO (210301120081)**

**PRANGYA PARAMITA PANI (210301120118)**

**MAMALI SAHOO (210301120142)**

**OMM PRASAD SINGH (210301120138)**

**Signature of the Student:**

**Place: Jatni, Odisha**

**Date:**

# TABLE OF CONTENTS

# LIST OF FIGURES

:



# LIST OF TABLES

# ABSTRACT

A global health challenge has been created by the widespread growth of diseases like pneumonia, lung cancer, COVID-19, and heart abnormalities. However, the traditional manual analysis of chest X-ray images by medical professionals requires expertise, which is also highly prone to human error and may result in delays in patient care. The rapid and accurate identification of these diseases becomes crucial to stopping their spread and guaranteeing early treatments. So, this research of chest X-ray classification system aims to dramatically increase diagnostic efficiency and accuracy in clinical settings. Our comprehensive approach integrates preprocessing, feature extraction using convolutional neural networks (CNNs) and transfer learning using EfficientNet-B0 and MobileNet-V2 models. The extracted features are optimized with a Bayesian-optimizer and fed to SVM and random forest classifier for classification. An accuracy of 97.3% was achieved by MobileNetV2 whereas 97% was achieved by EfficientNetB1 in the detection of COVID-19 X-ray images. Thorough evaluation and hyperparameter optimization play an essential role in the effective classification of these kinds of scenarios. The study offers encouraging results and enhances healthcare solutions by emphasizing the potential emerging deep learning techniques and their integration to medical image analysis.

Keywords: Deep Learning, Convolutional neural networks(CNN), chest X-ray, Covid-19,

Optimization, MobileNetV2, Feature Extraction, multilabel classification.

# CHAPTER - 1

## INTRODUCTION

### 1.1 Introduction

A global health challenge has been created by the widespread growth of diseases like pneumonia, lung cancer, COVID-19, and heart abnormalities. The rapid and accurate identification of these diseases becomes crucial to stopping their spread and guaranteeing early treatments. Chest X-rays provide a quicker and more affordable understanding of the condition of the chest than complex and time-consuming imaging techniques like MRI. However, there are fundamental challenges in the traditional manual analysis of chest X-ray images by medical professionals requires expertise, which is also highly prone to human error and may result in delays in patient care.

The main goal of an automated chest X-ray classification system aims to dramatically increase diagnostic efficiency and accuracy in clinical settings.

It builds upon previous techniques in medical image analysis, utilizing machine learning methods to extract complex patterns from X-ray images. Data preparation involves preprocessing techniques like image standardization and noise removal, while feature extraction techniques such as texture analysis and edge detection decode intricate patterns. Disease categorization utilizes various approaches, from conventional methods like Support Vector Machines to deep learning models like Convolutional Neural Networks (CNNs). These methods exhibit promise in identifying and classifying chest-related disorders, underscoring their potential for enhancing diagnostic outcomes. These methods have shown potential in the identification and classification of chest-related disorders.

### 1.2 Objective

The objectives of a Chest X-ray classification project typically involves machine learning techniques for automatic analysis of medical images. Here are some key objectives :

- **Develop a system** capable of automatically classifying common chest conditions, such as pneumonia, COVID-19, lung cancer from X-ray images.Aim to improve the accuracy of diagnosis by utilizing machine learning algorithms, reducing the chances of misinterpretation and assisting healthcare professionals in making informed decisions.

- **Early Detection**:Focus on early detection of chest abnormalities, enabling timely intervention and treatment, which is critical for better patient outcomes.
- **Workload Reduction for Radiologists**: Reduce the workload of medical experts and radiologists by providing an automated tool for preliminary analysis.

## 1.3 Chapter summarization

**Chapter 1** highlights about the introduction of the project and objectives to achieve the goal.
**Chapter 2** highlights about the literature survey and summary of the literature survey.
**Chapter 3** highlights about the methodology and proposed system about the project
**Chapter 4** highlights about the software requirement the project
**Chapter 5** highlights about the coding part and results.
**Chapter 6** displays the real world implemenation in website.
**Chapter 7** concludes the project.

# CHAPTER - 2

# BACKGROUND

## 2.1 Literature survey

Chest X-ray (CXR) classification research has evolved with a variety of techniques aimed at improving diagnostic accuracy. One study employed self-supervised deep convolutional neural networks, achieving an impressive AUC of 97.7 % on the CheXpert dataset [1].

In another approach, traditional image processing techniques, including resizing and contrast enhancement [2], yielded accuracy above 99 %  for classifying frontal and lateral CXR images, particularly beneficial for tuberculosis screening in resource-constrained settings.

Deep learning methods were applied to differentiate between pneumonia and COVID-19, with DenseNet-161 achieving 99 % accuracy for the former and ResNet-18 reaching 76 %  for the latter's severity classification [3].

Multi-label CXR classification was addressed, with problem transformation techniques such as Binary Relevance achieving a micro-averaged F1 score of 0.561 on ChestX-ray14[4].

A novel approach to COVID-19 classification achieved 98.23 %  accuracy using VGG19 pre-trained models, emphasizing the effectiveness of class decomposition in transfer learning. Additionally, preprocessing techniques, including histogram equalization and Gaussian blur, resulted in accuracy exceeding 97 % [5].

Lastly, data augmentation and deep learning techniques, particularly EfficientNetB1, achieved a test accuracy of 96.13 % in classifying chest infections from X-ray images [6].

These studies collectively illustrate the diverse range of approaches and significant advancements in CXR classification. As mentioned above, assessing the accuracy of the

classification model is very important. To evaluate the model that we have proposed, we have selected any 50 chest X-ray images (consist of 25 labelled abnormal and 25 labelled normal images) from [7, 8] for model testing which we proposed. The results of the 96%

accuracy evaluation that we will present are detailed in the next section.

chest X-Ray images taken from ChestXRay14 dataset were performed as input data to the

SqueezeNet network using transfer learning technique, using crop, histogram equalization and

CLAHE image processing techniques. The network, which uses 660 images, achieved 90.95% accuracy. SqueezeNet structure, which has been used less than other popular deep learning methods in previous studies, combined with image processing methods, has shown a successful result. In future studies, comparative performance analysis such as performance, network training time, test time, etc. can be done with the network used in different network structures.

Table 1. Comparative analysis of different study

| Study | Method | Accuracy | Recall | Precision | F1-score |
|---|---|---|---|---|---|
| [Ahmed et al.,2021] | CNN | 90.64 | 80 | 92 | 89.8 |
| [Choudhuri& paul,2021] | VGG16 | 98.3 | 88 | - | - |
| [Marques et al,2020] | EfficientNet | 96.70 | 96.69 | 97.59 | 97.11 |
| [Ucar &Kokmaz,2020] | SqueezeNet | 98.3 | - | - | 98.3 |
| [Ozturk et al,.2020] | DarkCovidNet | 87.02 | 85.35 | 89.96 | 87.37 |
| [Torman et al., 2020] | CapsNet | 84.22 | 84.22 | 84.61 | 84.21 |

# CHAPTER – 3

## PROPOSED SYSTEM

### 3.1 Key Functionality Components:

The key functionality components of Chest X-ray classification projects typically involve a combination of data processing, model development, evaluation, and integration into clinical workflows. Here are the key components:

Data Collection and Preprocessing:

**Data Acquisition:** Gather a diverse and representative dataset of Chest X-ray images, categorized by different classes (e.g., normal, pneumonia, COVID-19). Data Cleaning: Address any inconsistencies, anomalies, or artifacts in the dataset.

**Data Augmentation**: Enhance the dataset by applying transformations like rotations, flips, and scaling to improve model robustness.

Model Development:

Architecture Selection: Choose a suitable deep learning architecture (e.g., CNNs) for image classification tasks.

**Transfer Learning:** Utilize pre-trained models on large datasets for feature extraction and fine-tune them for Chest X-ray classification.

**Hyperparameter Tuning:** Optimize model hyperparameters for improved performance.

**Ensemble Learning:** Consider combining predictions from multiple models for enhanced accuracy.

**Training and Validation:**

Training Process: Train the model using labeled data, adjusting weights based on the error calculated during backpropagation. Validation Set: Use a separate dataset for model validation to prevent overfitting and ensure generalization.

**Evaluation Metrics:**

Performance Metrics: Evaluate model performance using metrics such as accuracy, precision, recall, F1 score, and area under the receiver operating characteristic curve (AUC-ROC).

Confusion Matrix: Analyze the distribution of true positives, true negatives, false positives, and false negatives.

## 3.2 **Block Diagram :**

The block diagram of the proposed Chest X-ray classification system begins with the data acquisition module, where x-ray images are collected from Kaggle covid-19 data repository data base. Subsequently, we performed essential data preprocessing tasks, including resizing images and augmenting the dataset to enhance model robustness. The dataset was then meticulously split into training, testing, and validation sets to facilitate effective model training and evaluation. Leveraging state-of-the-art deep learning architectures, namely EfficientNet B0, MobileNet V2, and VGG-16, we extracted meaningful features from the X-ray images. Further refinement ensued through Naive Bayes optimization, refining the model parameters for optimal performance. Following this, a classification ensemble approach was adopted, incorporating Support Vector Machines (SVM) and Random Forest classifiers. The final stage involved meticulous result analysis, employing evaluation metrics such as precision, recall, and confusion matrix. The interpretability of model predictions was enhanced using Grad-CAM, allowing for insightful visualizations of the key regions influencing classification decisions. This holistic approach amalgamates advanced deep learning techniques with traditional classifiers for a thorough Chest X-ray classification framework.

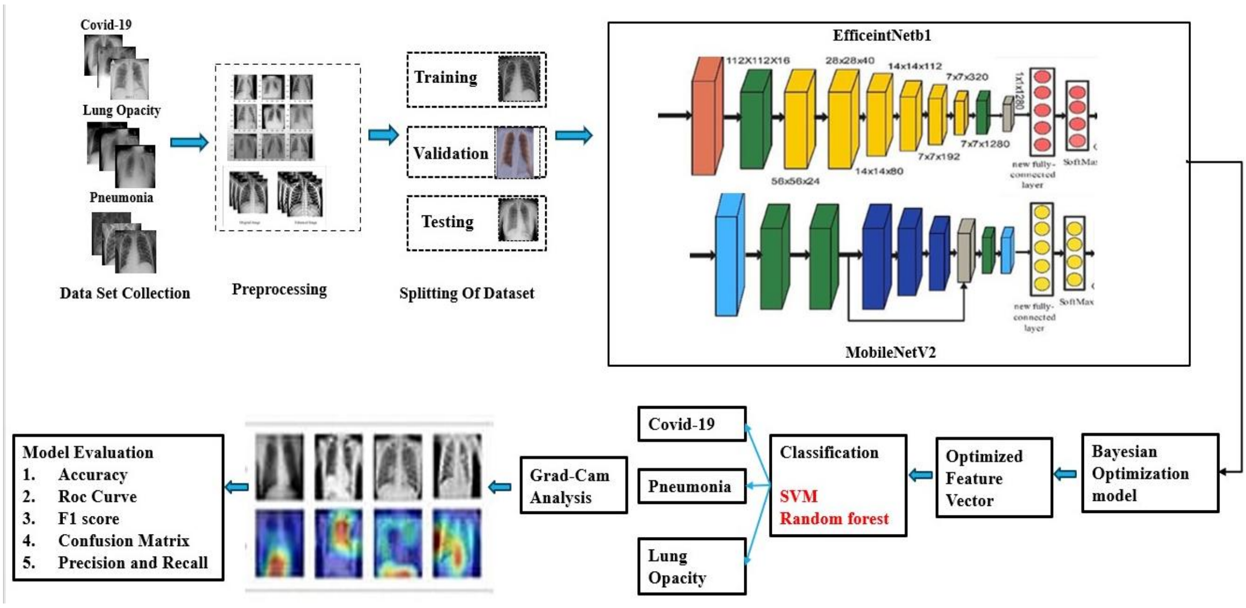



Fig 1. Block diagram of Proposed System

# CHAPTER – 4

## SOFTWARE REQUIREMENTS

### 4.1 SOFTWARE USED

### Programming Language:

**Python:**

Python is a versatile, high-level programming language known for its simplicity and readability. Widely used in machine learning and data science, Python offers extensive libraries and frameworks for various applications.

Version -> 3.10.1 Diverse libraries.

Cross-Platform: Works on various operating systems.

Versatility: Used for web development, data analysis, machine learning, automation, etc.

Fig 2. Python

### 4.2 Integrated Development Environment (IDE):

**Jupyter Notebooks:**

Jupyter provides an interactive and collaborative environment, ideal for data exploration and analysis. Its notebook format supports code, visualizations, and text.

Version -> 7.0.6

Language: Supports multiple languages, with a focus on Python.

Format: Notebooks with code, visualizations, and text.

Use Cases: Ideal for data analysis, visualization, and interactive development.

Fig 3. Jupyter

**Google Colab:**

Colab is a cloud-based platform by Google, offering free access to GPU resources for running Python notebooks, making it suitable for resource-intensive tasks.

Type: Cloud-based, collaborative Jupyter notebook environment.

Pre-installed Libraries: Comes with pre-installed popular Python libraries.

Use Cases: Ideal for machine learning, data analysis, and collaborative coding.

Fig 4. Google Colab

**Anaconda:**

Anaconda simplifies package management and environment setup, providing a comprehensive Python distribution for scientific computing.

Packages: Includes popular data science packages (NumPy, pandas, etc.).

IDE: Can be used with Jupyter Notebooks, Spyder, or other IDEs.

Use Cases: Widely used in data science, machine learning, and scientific computing.

Fig 5. Anaconda

## 4.3 Python Libraries:

**NumPy:**

NumPy is a fundamental library for numerical operations in Python, enabling efficient handling of arrays and matrices.

**Pandas:**

Pandas is a data manipulation library, offering data structures like DataFrames for easy handling and analysis of structured data.

**scikit-learn:**

scikit-learn provides tools for machine learning and statistical modeling, with a user-friendly interface and extensive documentation.

**TensorFlow:**

TensorFlow is a popular open-source machine learning framework developed by Google, known for its flexibility and scalability.

**Keras:**

Keras, now integrated into TensorFlow, is a high-level neural networks API, simplifying the construction and training of deep learning models.

## 4.4 Data Visualization:

Matplotlib:

Matplotlib is a versatile plotting library for creating static, animated, and interactive visualizations in Python.

**Seaborn:**

Seaborn is built on top of Matplotlib and specializes in statistical data visualization, providing a high-level interface.

**Plotly for Python:**

Plotly is a powerful library for interactive plotting, enabling the creation of dynamic charts and dashboards with ease.


Fig 6. Python libraries

# CHAPTER - 5

## CODE IMPLEMENTATION

### 5.1 Dataset Collection:

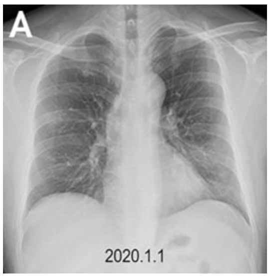
The dataset collection phase is critical for the success of any machine learning project, particularly for chest X-ray classification using deep learning project which is a multi label classification. In The researchers of Qatar University have compiled the COVID-QU-Ex dataset, which consists of 33,920 chest X-ray (CXR) images including: 11,956 COVID-19, 11,263 Non-COVID infections (Viral or Bacterial Pneumonia), and 10,701 Normal Ground-truth lung segmentation masks are provided for the entire dataset. This is the largest ever created lung mask dataset.

Besides, 2,913 COVID-19 Infection Segmentation masks are also provided.

**Dataset Name: Covid-19 Radiography Database**

- **Source of Dataset: Kaggle**
- **Total no. of Images:  21, 165**
- **Covid-19 Images: 3616**
- **Lung Cancer: 6012**
- **Normal : 10,192**
- **Pneumonia: 1345**
- **All the images are in Portable Network Graphics (PNG) file format**
- **The resolution are 299*299 pixels.**





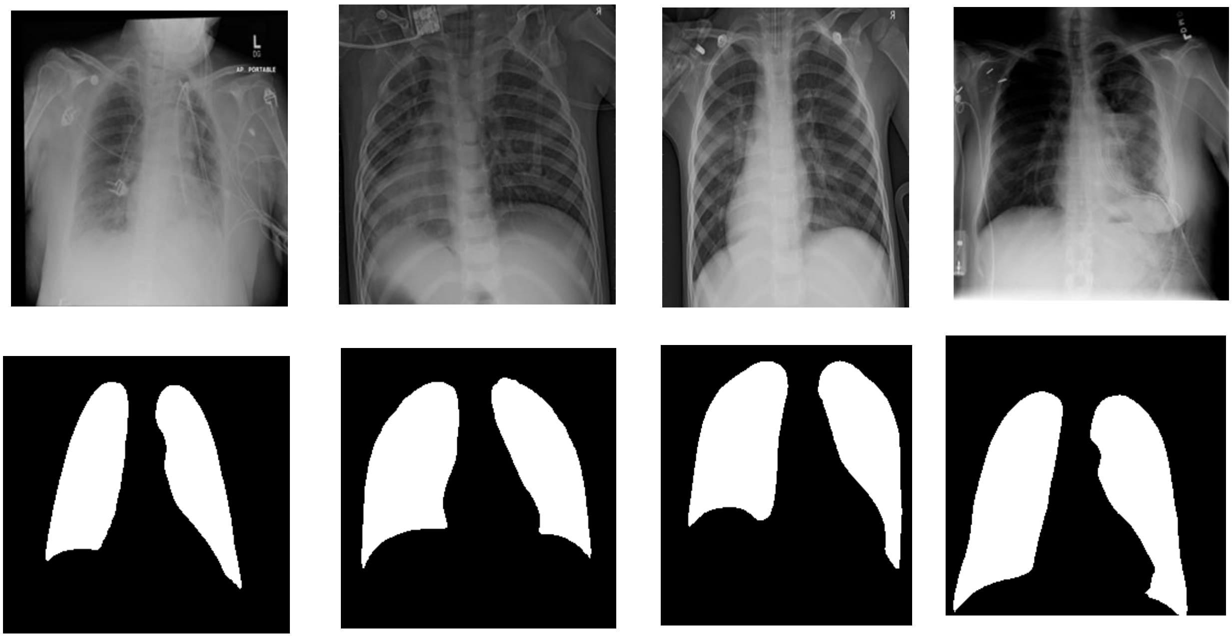
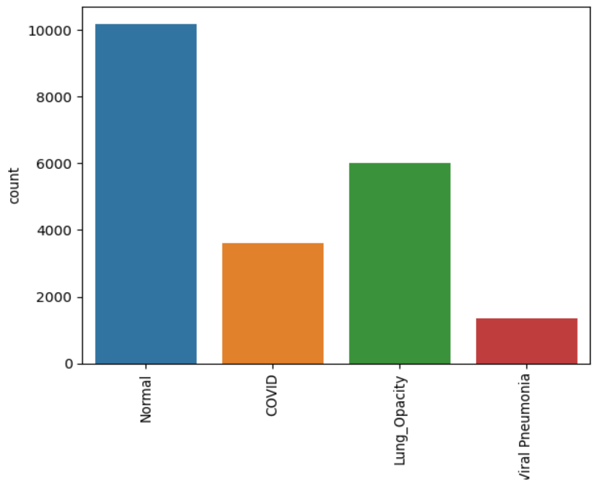


Fig 7. Covid-19 dataset

**5.2 Data Description and Preprocessing:**

In the data description phase of our chest X-ray classification project, we conducted a comprehensive analysis of the dataset, focusing on four key categories: COVID-19, pneumonia, lung opacity, and normal cases. To standardize this information, we converted the count values into a common scale, ranging from 0 to 100. This normalization facilitated a more intuitive understanding of the relative proportions of images across different classes.

Additionally, we delved into the metadata of the chest X-ray images, extracting crucial details such as image format and size. This exploration provided valuable insights into the technical characteristics of the dataset, aiding in preprocessing steps and algorithmic considerations. Furthermore, we implemented code snippets to sample and display specific images, allowing us to visually inspect and verify the quality and diversity of the data.




```
Normal :  10192
Covid :  3616
Opacity :  6012
VIral Pneumonia :  1345
```

Fig 8. Bar plot of count values

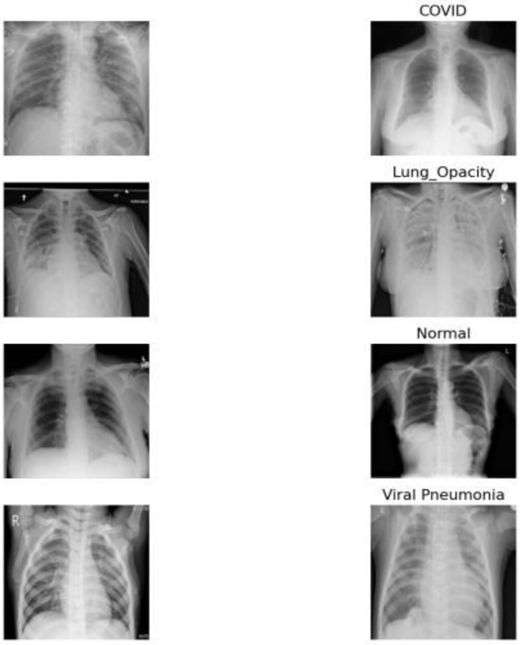



Fig 9. Sample of data

```
Normal              48.15
Lung_Opacity        28.41
COVID               17.08
Viral Pneumonia      6.35
Name: result, dtype: float64
```

Fig 10. Count value range in 100

| | FILE NAME | label |
|---|---|---|
| 98 | COVID-3389 | 0 |
| 1613 | NORMAL-5826 | 1 |
| 1243 | COVID-3303 | 0 |
| 599 | COVID-2121 | 0 |

Fig 11. Labeling class as 0,1,2

| | FILE NAME | FORMAT | SIZE | URL |
|---|---|---|---|---|
| 0 | COVID-1 | PNG | 256*256 | https://sirm.org/category/senza-categoria/covi... |
| 1 | COVID-2 | PNG | 256*256 | https://sirm.org/category/senza-categoria/covi... |
| 2 | COVID-3 | PNG | 256*256 | https://sirm.org/category/senza-categoria/covi... |
| 3 | COVID-4 | PNG | 256*256 | https://sirm.org/category/senza-categoria/covi... |
| 4 | COVID-5 | PNG | 256*256 | https://sirm.org/category/senza-categoria/covi... |

Fig 12. Meta data of COVID

| | FILE NAME | FORMAT | SIZE | URL |
|---|---|---|---|---|
| 0 | NORMAL-1 | PNG | 256*256 | https://www.kaggle.com/c/rsna-pneumonia-detect... |
| 1 | NORMAL-2 | PNG | 256*256 | https://www.kaggle.com/c/rsna-pneumonia-detect... |
| 2 | NORMAL-3 | PNG | 256*256 | https://www.kaggle.com/c/rsna-pneumonia-detect... |
| 3 | NORMAL-4 | PNG | 256*256 | https://www.kaggle.com/c/rsna-pneumonia-detect... |
| 4 | NORMAL-5 | PNG | 256*256 | https://www.kaggle.com/c/rsna-pneumonia-detect... |

Fig 13. Metadata of NORMAL

### 5.3 Exploratory data analysis (EDA) of data:

One possible use of plotting the maximum value distribution on an image dataset is to visualize the range of values present in the dataset. This can be helpful for understanding the characteristics of the dataset . For example, if the maximum value distribution is heavily skewed towards one end of the range, it may indicate that there are a large number of outliers present in the dataset. On the other hand, if the maximum value distribution is more evenly distributed.

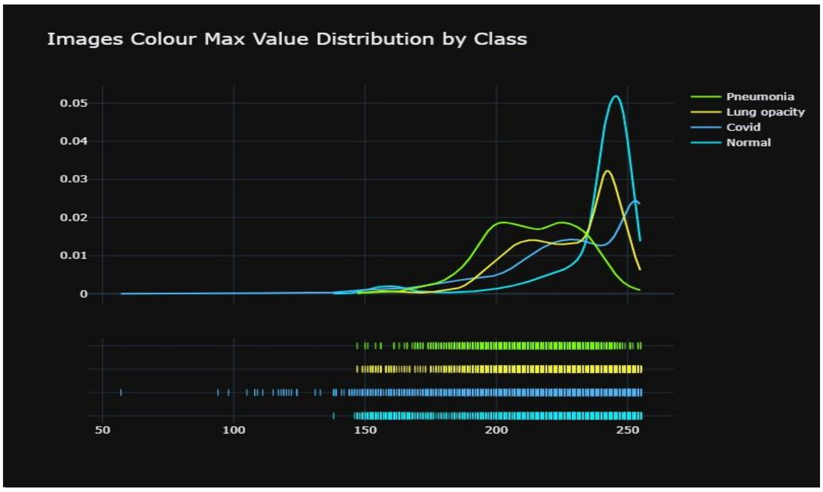


Fig 14. Max Value of Image colour

In addition to understanding the characteristics of the dataset, plotting the maximum value distribution can also be useful for identifying the appropriate scaling or normalization techniques to apply to the data. For example, if the maximum value distribution is heavily skewed. The minimum value distribution on an image dataset can be useful for understanding the characteristics of the dataset and identifying any potential issues or trends that may be present. For example, if the minimum value distribution is heavily skewed towards one end of the range, it may indicate that there are a large number of outliers present in the dataset. Plotting the mean value distribution on an image dataset can be useful for understanding the overall characteristics of the dataset and identifying any potential issues or trends that may be present.

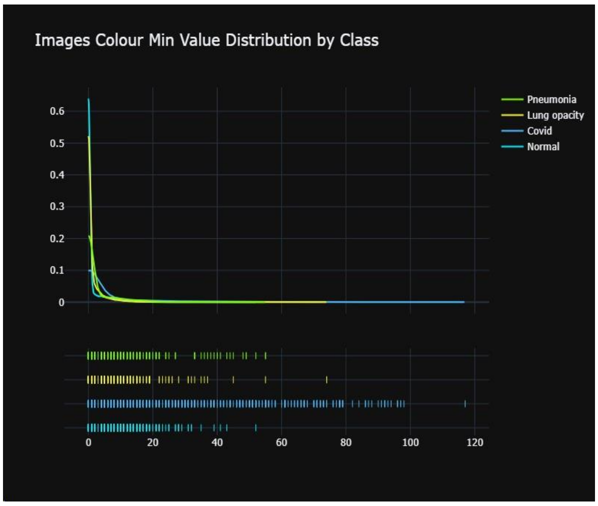


Fig 15. Minimum value of image colour

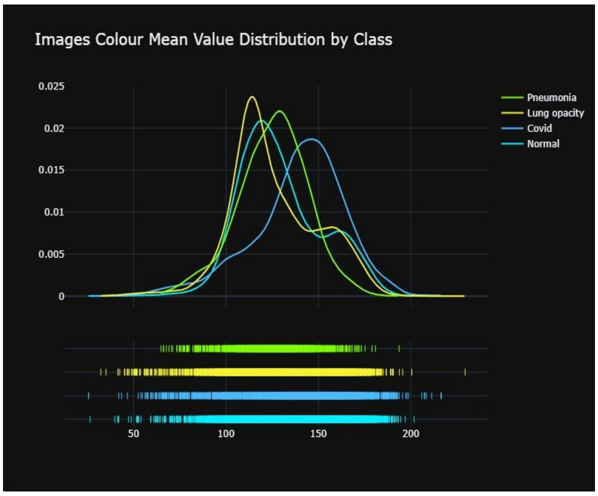


Fig 16. Mean value of image colour

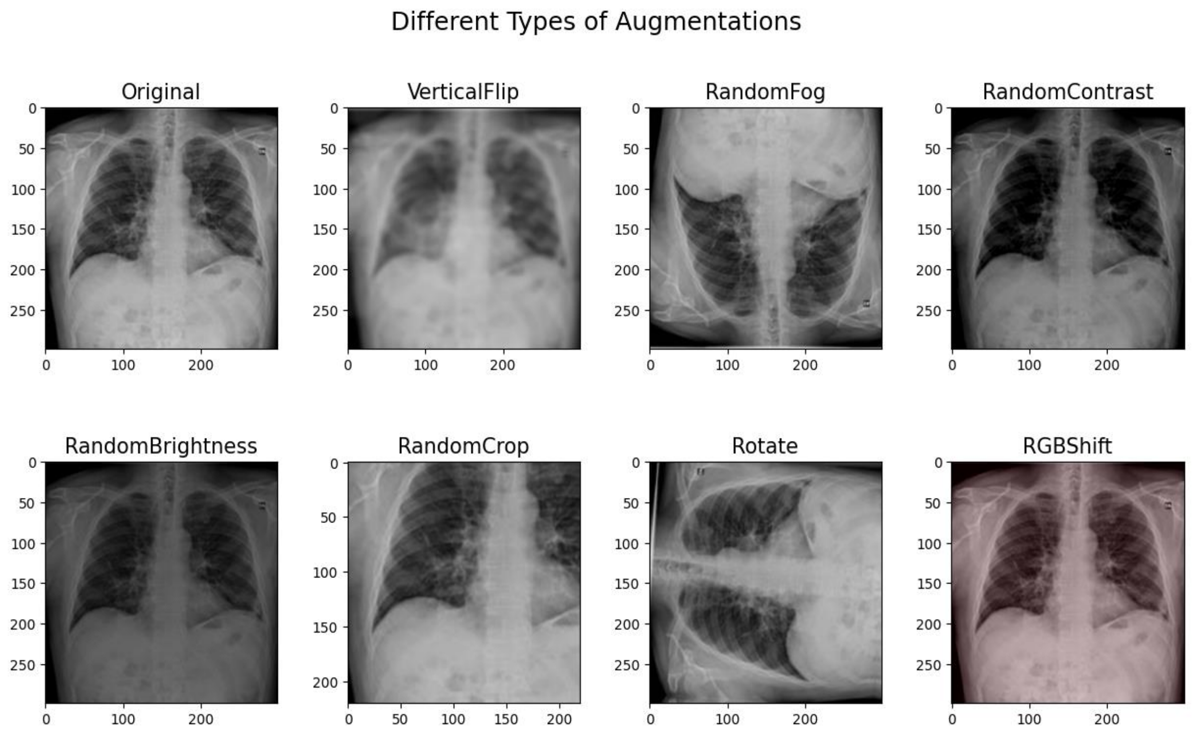



Fig 17. Data Augmentation

**Data augmentation** can be used to create additional training data by applying various transformations to existing images. This can be especially useful when working with small or imbalanced datasets, as it can help to improve the performance of machine learning models by increasing the amount of training data available.

## 5.4 Applying different types of feature extraction techniques:

In the project, three distinct feature extraction techniques were employed: pixel feature extraction, Local Binary Pattern (LBP), and Histogram of Oriented Gradients (HOG). Each technique captures unique aspects of the image, contributing to the diversity of information used for classification.

Pixel Feature Extraction:

Pixel features involve extracting information directly from the pixel values of the image. Each pixel's intensity serves as a feature, and the spatial arrangement of these intensities forms the foundation for image representation. While simple, pixel features provide a fundamental representation of the image's grayscale information.

```
pixel_img = []

for image in tqdm(data['path']):
    img=Image.open(image)
    img=ImageOps.grayscale(img)
    img=img.resize((64,64))
    img=np.asarray(img)
    img=img.reshape((64,64,1))
    pixel_img.append(img)
```



Fig 18.. Pixel image feature extraction

Local Binary Pattern (LBP):

Local Binary Pattern is a texture descriptor that characterizes the local patterns in an image. It operates by comparing each pixel with its neighboring pixels and encoding the result as a binary pattern. LBP is effective in capturing textural details and patterns, making it suitable for applications where texture information is crucial, such as in medical image analysis.

```
def extract_lbp_features(image):
    gray = np.array(image)
    lbp = local_binary_pattern(gray, 8, 1, method='uniform')
    hist, _ = np.histogram(lbp.ravel(), bins=np.arange(0, 60), range=(0, 59))
    return hist
```

Fig 19. lbp feature extraction

Histogram of Oriented Gradients (HOG):

HOG focuses on the distribution of gradient information in an image. It divides the image into small cells, calculates the gradient magnitude and orientation within each cell, and constructs a histogram of orientations. This technique excels in capturing edge and shape information, making it particularly valuable for object detection and recognition tasks.

```
from skimage.feature import hog
def extract_hog_features(image):
    # Convert image to grayscale
    gray = np.array(image)

    # Compute HOG features
    features = hog(gray, orientations=9, pixels_per_cell=(8, 8),
                   cells_per_block=(2, 2), transform_sqrt=True,
                   block_norm='L2-Hys')

    return features
```

Fig 20 Hog image feature extraction

## 5.5 Applying Classification Algorithm:

In the Applying Classification Algorithm phase, diverse machine learning algorithms are implemented to discern the most effective approach for chest X-ray classification. Popular algorithms such as Support Vector Machines, Random Forests, and Convolutional Neural Networks are considered. Hyperparameter tuning and model selection play a crucial role in optimizing classification accuracy.

**Support Vector Machine (SVM):**

Achieved high accuracy with the RBF kernel.

Robust performance in distinguishing between sonar signals from rocks and mines.

Effective in handling complex relationships within the dataset.

```
[ ]   ▸ LogisticRegression
      LogisticRegression()

[ ]  decision_tree.fit(X_train, y_train)

      ▸ DecisionTreeClassifier
      DecisionTreeClassifier()

     random_forest.fit(X_train, y_train)

      ▸ RandomForestClassifier
      RandomForestClassifier()

[ ]  RandomForestClassifier()

      ▸ RandomForestClassifier
      RandomForestClassifier()

[ ]  xgb_model.fit(X_train, y_train)

      ▸ XGBClassifier
      XGBClassifier(base_score=None, booster=None, callbacks=None,
                    colsample_bylevel=None, colsample_bynode=None,
                    colsample_bytree=None, device=None, early_stopping_rounds=None,
                    enable_categorical=False, eval_metric=None, feature_types=None,
                    gamma=None, grow_policy=None, importance_type=None,
                    interaction_constraints=None, learning_rate=None, max_bin=None,
                    max_cat_threshold=None, max_cat_to_onehot=None,
                    max_delta_step=None, max_depth=None, max_leaves=None,
                    min_child_weight=None, missing=nan, monotone_constraints=None,
                    multi_strategy=None, n_estimators=None, n_jobs=None,
                    num_parallel_tree=None, objective='multi:softprob', ...)

[ ]  kn_clas.fit(X_train, y_train)

      ▸ KNeighborsClassifier
      KNeighborsClassifier()

[ ]  nb_model.fit(X_train, y_train)

      ▸ MultinomialNB
      MultinomialNB()

[ ]  svm_model.fit(X_train, y_train)

      ▸ SVC
      SVC()
```

Fig 21. Classification techniques

**Decision Tree and Random Forest:**

Encountered overfitting issues, especially in scenarios with intricate decision boundaries. Decision Tree and Random Forest models tend to capture noise in the training data, impacting generalization on unseen data. While decision trees offer interpretability, the ensemble nature of Random Forest often addresses overfitting but requires careful tuning.

**Logistic Regression (LR) :**

LR showed competitive performance. Effective in scenarios with a linear decision boundary.Logistic Regression provides simplicity and interpretability.

**XGBoost**:

It stands out as a high-performance machine learning algorithm widely employed for classification tasks. Belonging to the gradient boosting family, XGBoost enhances predictive models by sequentially incorporating weak learners, typically decision trees. Notable features include built-in regularization to counter overfitting, support for parallel and distributed computing for efficiency, and an intuitive mechanism for assessing feature importance. It effectively handles missing data, offers cross-validation support, and presents a plethora of tunable hyperparameters for optimization. With compatibility across multiple programming languages, including Python and R, XGBoost has become a go-to choice for researchers and practitioners seeking accuracy, speed, and adaptability in classification tasks.

**Naive Bayes (NB):**

Provided a baseline performance, especially in scenarios with strong independence assumptions. Simplicity and efficiency make Naive Bayes suitable for certain datasets. GDM and ET displayed competitive accuracy. GDM's ability to optimize complex functions complements its performance.

**k-Nearest Neighbors (kNN):**

Dependent on the choice of k and the distance metric. KNN may face challenges in datasets with varying densities and noise.

**5.5 Evaluation Parameters:**

The Evaluation Parameters section involves a meticulous analysis of the model's performance. Metrics such as accuracy, precision, recall, and F1-score are computed to quantitatively assess how well the model distinguishes between rocks and mines. A comprehensive evaluation provides insights into potential areas for improvement and fine-tuning.

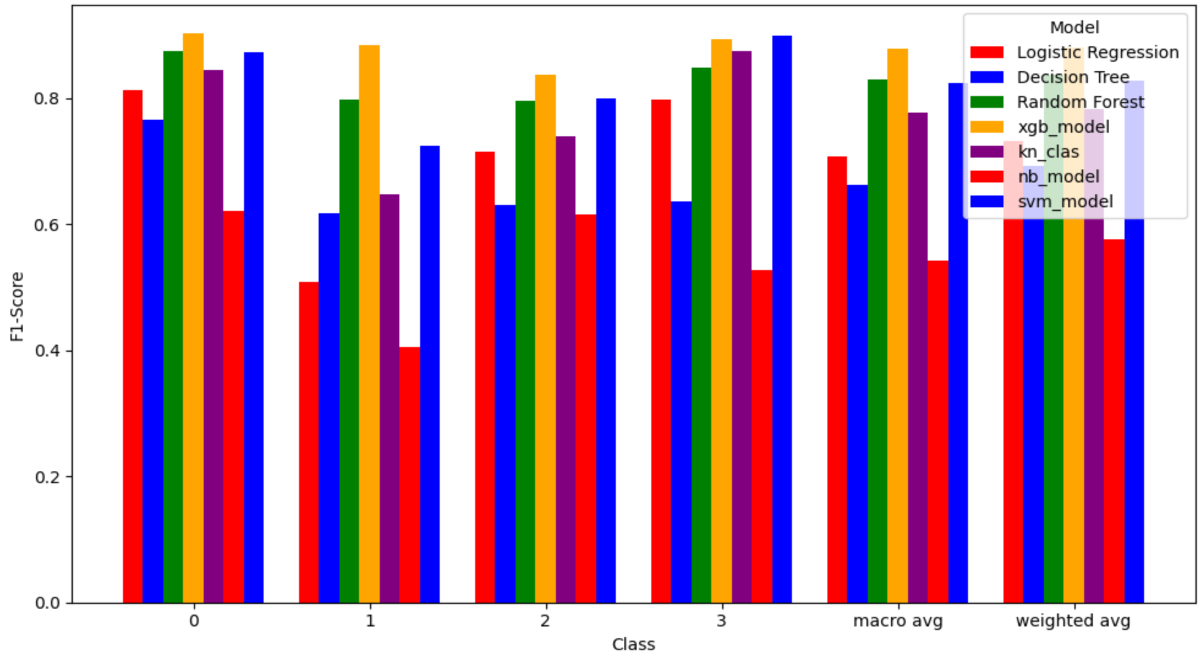



Fig 22. Comparison bar plot of F1 score of different models with pixel feature extraction

SVM exhibits the highest accuracy among the classifiers, emphasizing its effectiveness in discerning patterns within pixel-level features. Decision Tree and Random Forest perform moderately well, suggesting reasonable capability in leveraging basic pixel information. LR, while competitive, and XGB showcase competitive but slightly lower accuracies, indicating sensitivity to nuanced patterns within the pixel features.

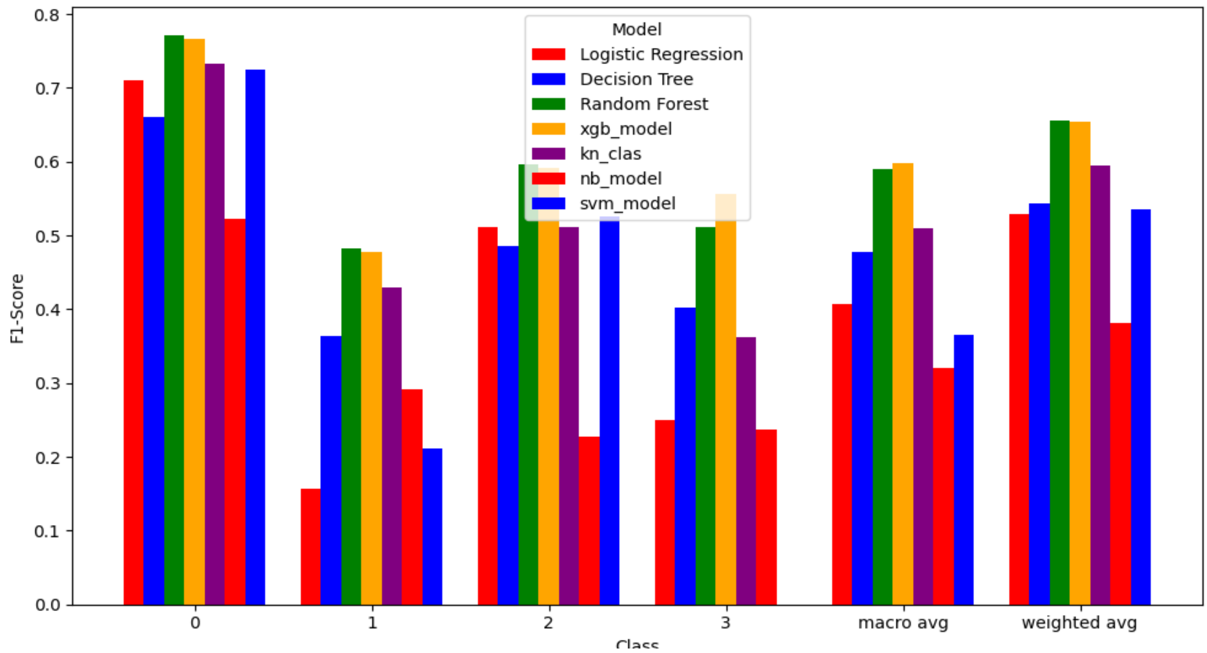



Fig 23. Comparison bar plot of F1 score of different models with lbp feature extraction

SVM remains the top performer, emphasizing its robustness in capturing the texture-based features encoded by LBP. Decision Tree and Random Forest show improved accuracy compared to pixel features, benefitting from the richer information provided by LBP. LR and XGB, although effective, exhibit lower accuracy, suggesting potential challenges in handling the intricacies of LBP-derived features.

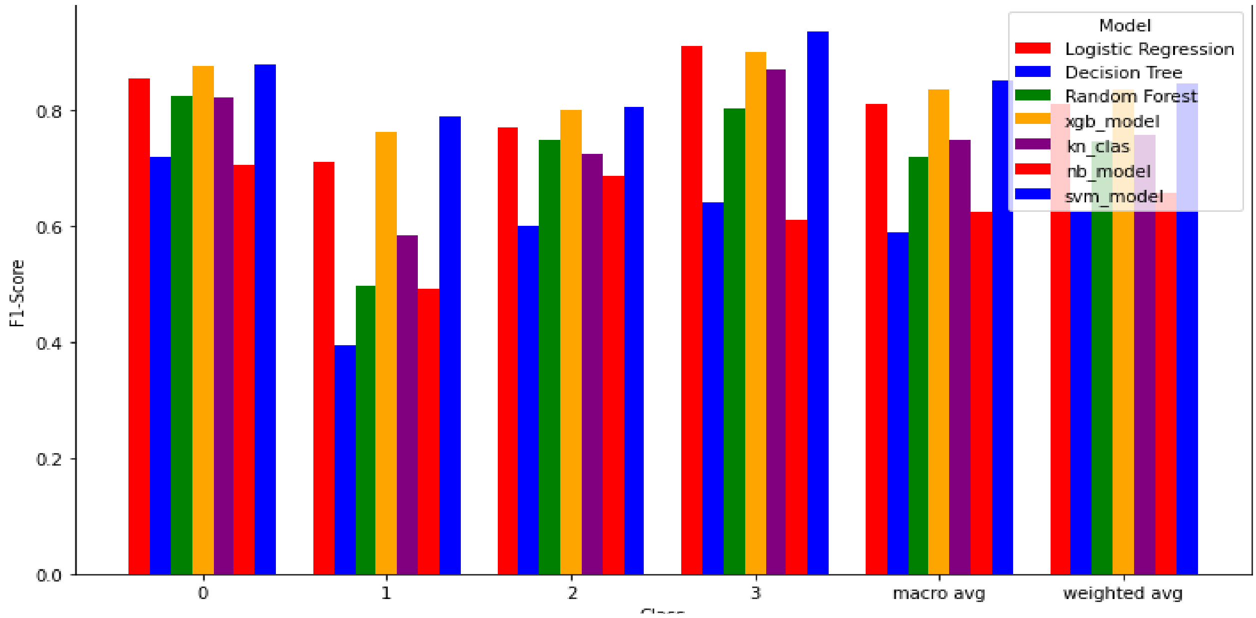



Fig 24. Comparison bar plot of F1 score of different models with hog feature extraction

SVM once again excels, underlining its prowess in recognizing shape and gradient-based information from HOG features. Decision Tree and Random Forest witness substantial accuracy improvements over pixel and LBP features, showcasing the strength of HOG in capturing shape-related patterns. LR and XGB maintain competitive accuracies, signaling adaptability but with slight performance trade-offs compared to SVM.

### 5.6 Applying CNN:

In our project, we strategically used three distinct convolutional neural network (CNN) architectures: VGG-16, EfficientNet B1, and MobileNetV2. Each of these architectures brings unique characteristics to the table, contributing to the overall effectiveness of our model.

**VGG-16:**

The VGG-16 architecture, developed by the Visual Geometry Group at the University of Oxford, is renowned for its simplicity and powerful feature extraction capabilities. Comprising 16 layers, including convolutional and fully connected layers, VGG-16 excels in discerning intricate patterns within images, making it a robust choice for tasks requiring nuanced visual recognition.

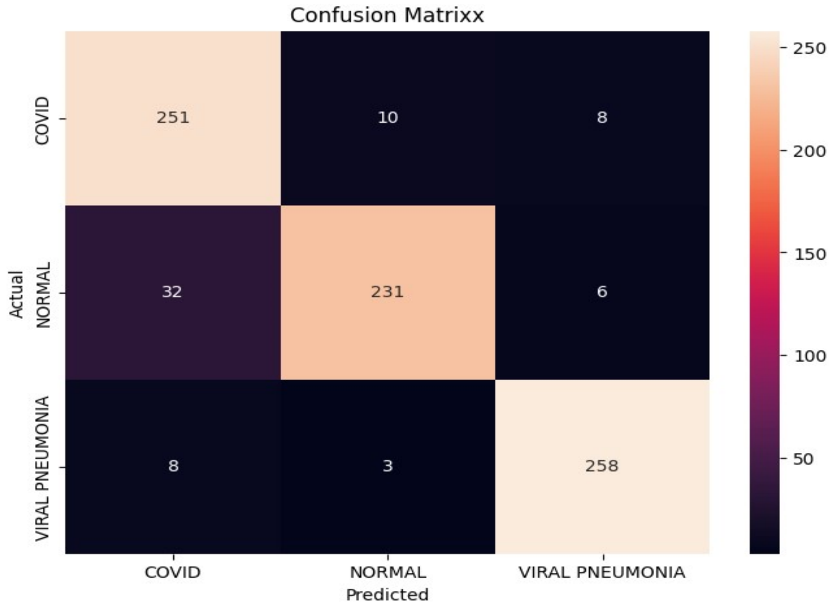



Fig 25. Confusion matrix of VGG-16

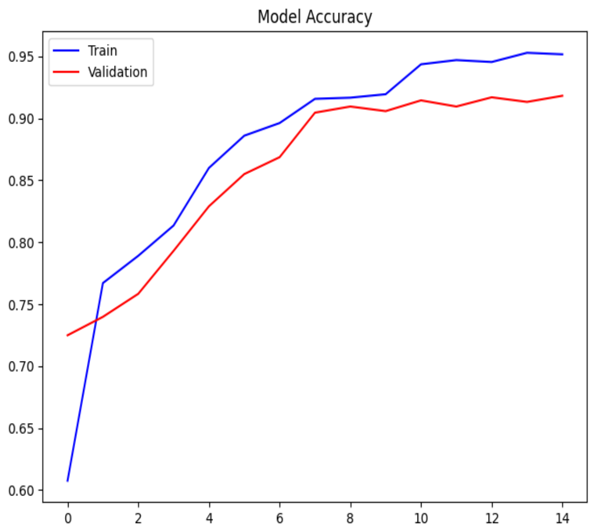



Fig 26. Model accuracy

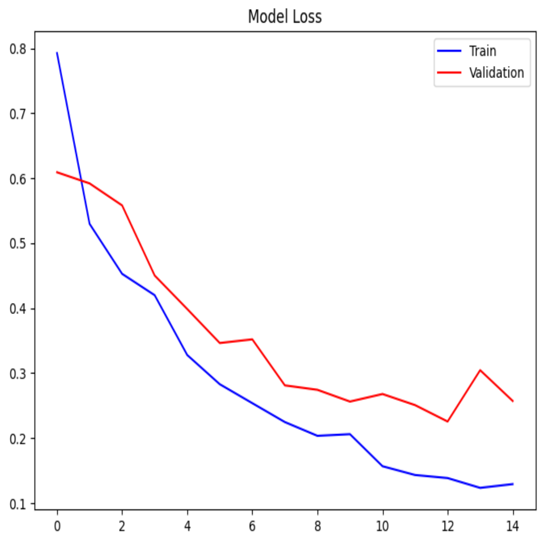



Fig 27. Model loss

**EfficientNet B0:**

EfficientNet B1 represents a member of the EfficientNet family, designed to optimize the balance between network depth, width, and resolution. This architecture introduces a novel compound scaling method, achieving superior performance with fewer parameters. EfficientNet B1's efficiency makes it particularly well-suited for resource-constrained environments without compromising accuracy.

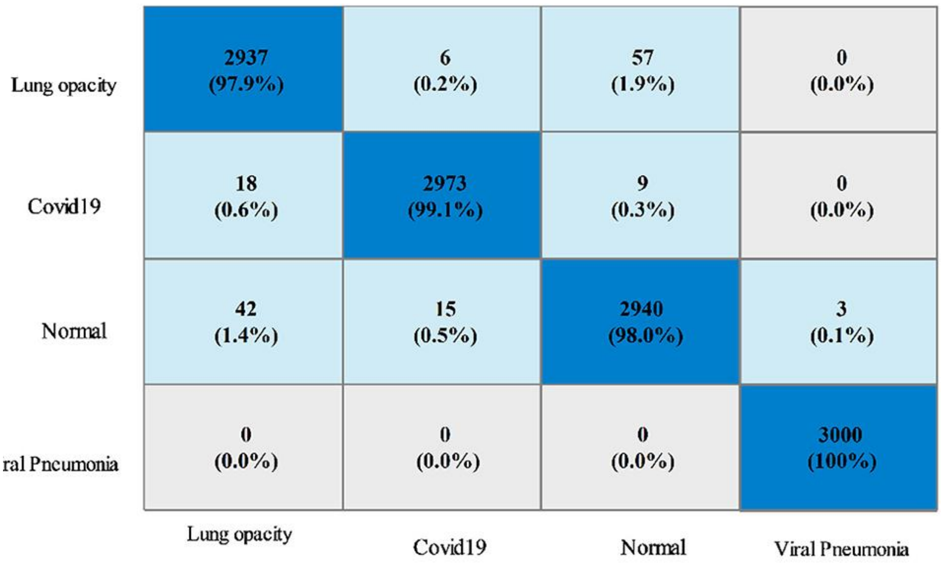
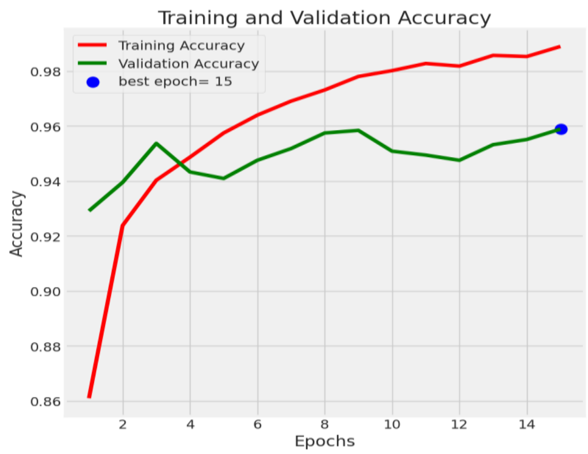



Fig 28. Confusion matrix of Efficient Net B0



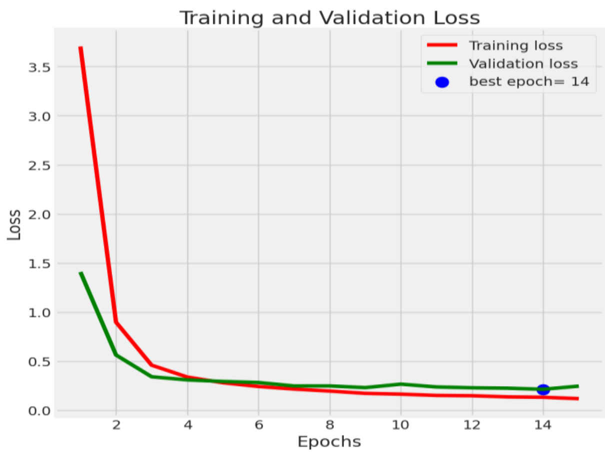




Fig 29. Model Accuracy

Fig 30. Model Loss

**MobileNetV2:**

MobileNetV2, tailored for mobile and edge devices, focuses on achieving a harmonious blend of accuracy and efficiency. This lightweight architecture incorporates inverted residuals and linear bottlenecks, making it ideal for real-time applications on devices with limited computational resources. MobileNetV2's ability to maintain high performance while accommodating hardware constraints enhances its utility in diverse deployment scenarios.

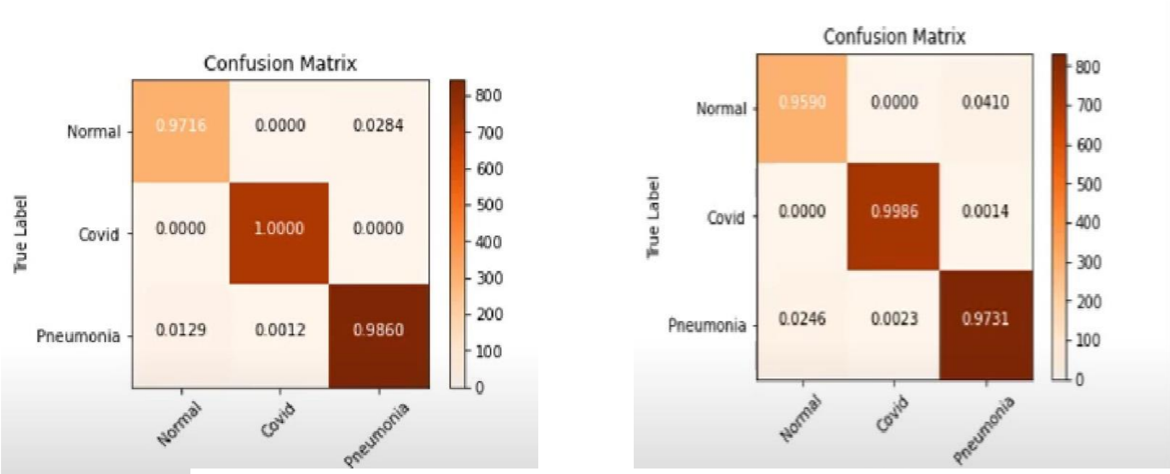



Fig 31. Confusion matrix with and without transfer learning

## 5.7 Applying Grad CAM:

Grad-CAM (Gradient-weighted Class Activation Mapping) is a visualization technique that is used to understand which parts of an image are most important for a deep learning model's predictions. The technique works by generating a heatmap that highlights the regions of the image that the model is most sensitive to when making a prediction.

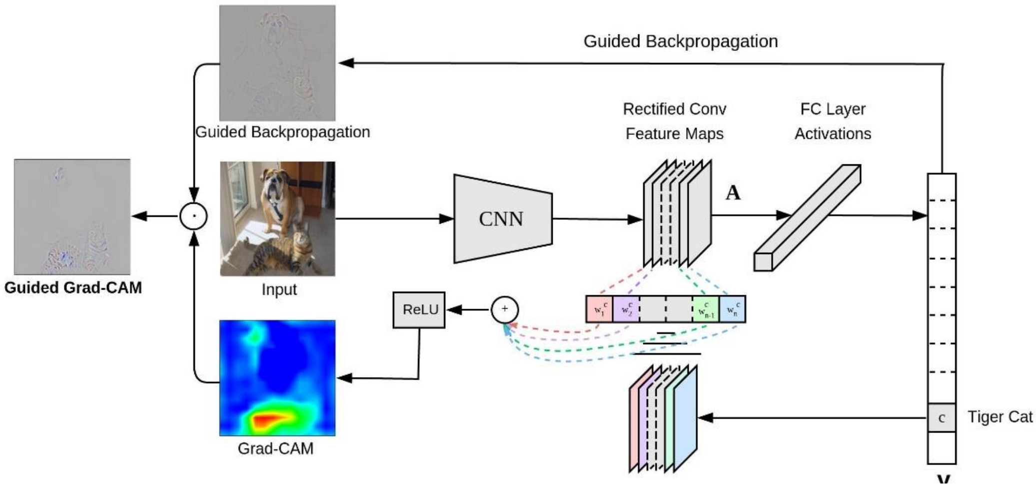



Fig 32. Explanation of Grad CAM

There are several reasons why Grad-CAM visualization can be important:

1. **Model understanding**: Grad-CAM visualization can be useful for understanding how a deep learning model is making its predictions. By highlighting the regions of the image that are most important for the model's predictions, Grad-CAM can help to provide insight into the decision-making process of the model.

2. **Model explanation**: Grad-CAM visualization can be useful for explaining the predictions of a deep learning model to non-technical stakeholders. By highlighting the regions of the image that are most important for the model's predictions, Grad-CAM can provide a more intuitive understanding of the model's decision-making process.

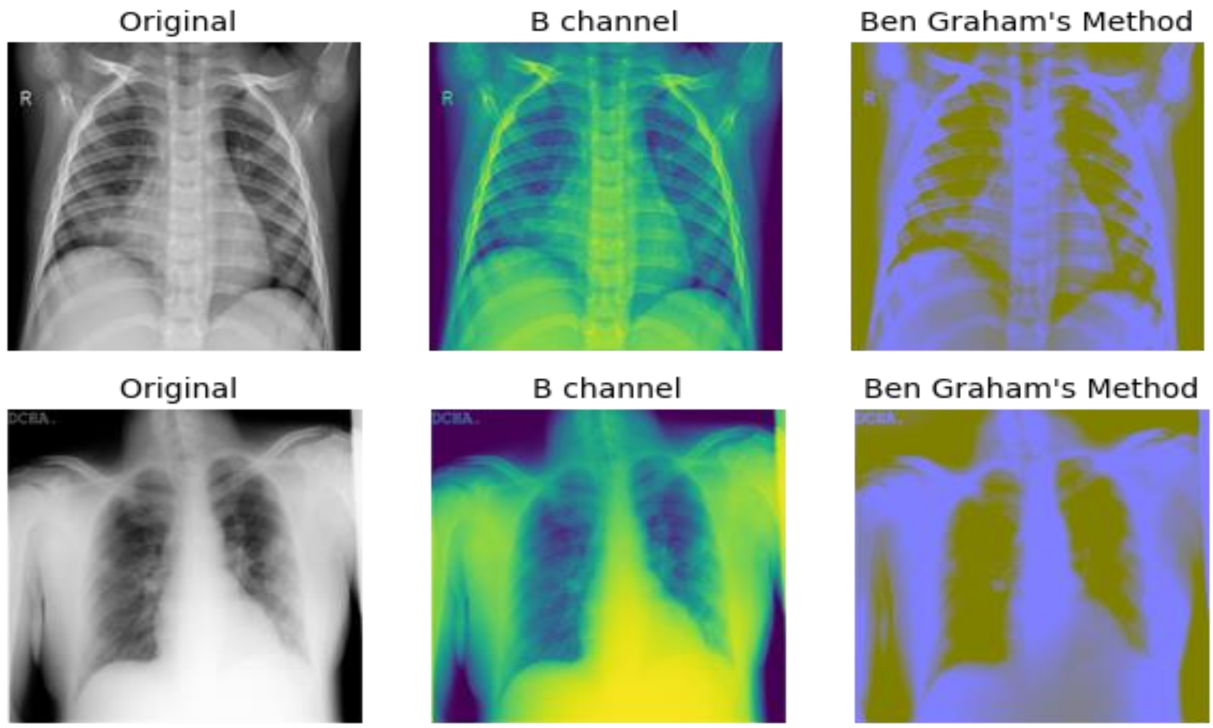



Fig 33. Sample of grad cam

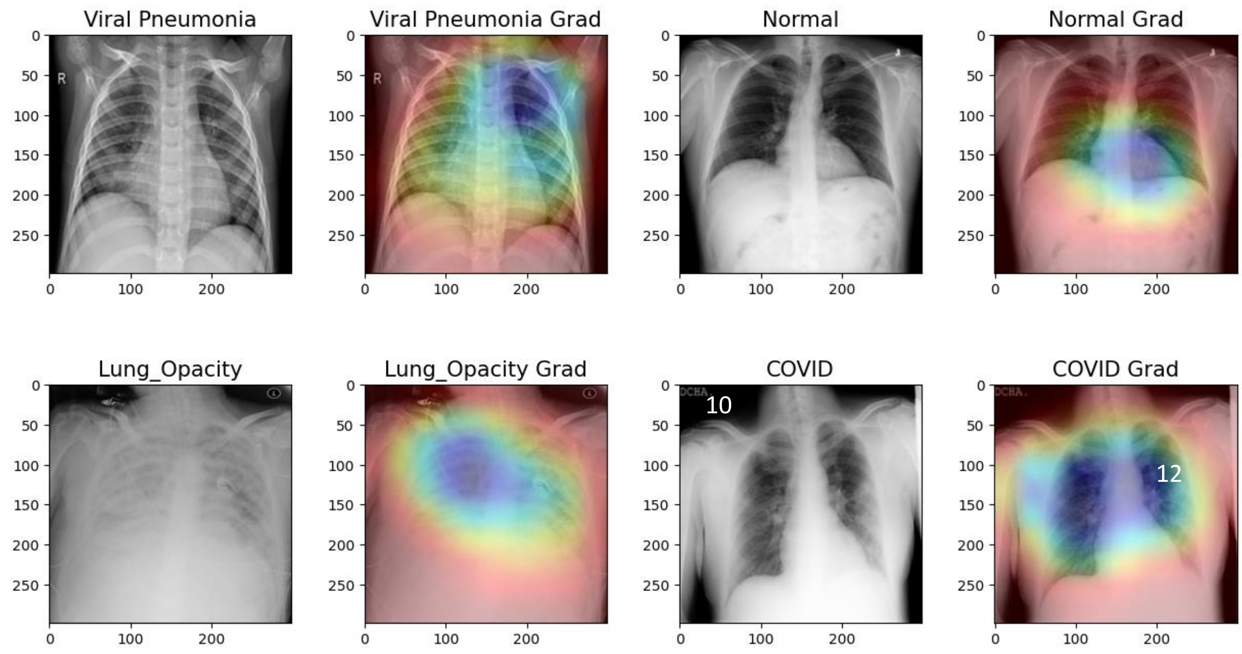



Fig 34. Grad CAM analysis of images

# CHAPTER - 6

## RESULT AND DISCUSSION

In this section the result has highlighted.

We got good accuracy in classifying chest X-ray images into three categories, including COVID-19, lung opacity and pneumonia, using advanced deep learning and classification techniques such as EfficientNetB0 and MobileNetV2 , VGG-16 with SVM.

Table 2. Accuracy comparison

| **Techniques** | **Accuracy** |
|---|---|
| Pixel feature extraction+Xgb classifier | 87% |
| Lbp feature extraction+Random forest | 85% |
| HOG feature extraction+SVM | 91% |
| EfficientNet BO | 97% |
| MobilenetV2 | 96.3% |
| VGG-16 | 92.8% |
| Grad CAM | 96% |

# CONCLUSION

In conclusion, using deep learning algorithms for the classification of COVID-19 using X-ray images can be a powerful tool for identifying and diagnosing the disease. By training a model on a large dataset of X-ray images labeled with COVID-19 diagnosis, it is possible to achieve high levels of accuracy and robustness in the model's predictions. Through the use of techniques such as image augmentation and model optimization, it is possible to improve the performance of the model and increase its ability to generalize to new data. Overall, deep learning algorithms can be a valuable tool for the classification of COVID-19 using X-ray images, and their use in this application has the potential to significantly impact the diagnosis and treatment of the disease.

# FUTURE SCOPE

There are some further work we can do in this project are:

- Multi-Modality Fusion
- Usinng modern AI Models
- Automated Reporting system
- Interactive Visualization Tools:
- User-Friendly Interfaces:.
- Patient-Centric Applications:

# REFERENCES

[1] M.E.H. Chowdhury, T. Rahman, A. Khandakar, R. Mazhar, M.A. Kadir, Z.B. Mahbub, K.R. Islam, M.S. Khan, A. Iqbal, N. Al-Emadi, M.B.I. Reaz, M. T. Islam, "Can AI help in screening Viral and COVID-19 pneumonia?" IEEE Access, Vol. 8, 2020, pp. 132665 - 132676.

[2] Rahman, T., Khandakar, A., Qiblawey, Y., Tahir, A., Kiranyaz, S., Kashem, S.B.A., Islam, M.T., Maadeed, S.A., Zughaier, S.M., Khan, M.S. and Chowdhury, M.E., 2020. Exploring the Effect of Image Enhancement Techniques on COVID-19 Detection using Chest X-ray Images.

[3] J. Read, ''A pruned problem transformation method for multi-label classification,'' in Proc. New Zealand Comput. Sci. Res. Student Conf., 2008, pp. 143–150.

[4] ong W, Agarwal PP. Chest imaging appearance of COVID19 infection. Radiol: Cardiothorac Imaging. 2020 Feb 13;2(1):e200028.

[5] Li Y, Xia L. Coronavirus disease 2019 (COVID-19): role of chest CT in diagnosis and management. American J Roentgenol. 2020 214(6):1280–6.

[6] Abbas A, Abdelsamea MM, Gaber MM. Classifcation of COVID19 in chest X-ray images using DeTraC deep convolutional neural network. Appl Intell. 2021 Feb;51(2):854–64.

[7] Minaee S, Kafeh R, Sonka M, Yazdani S, Souf GJ. Deep-covid: Predicting covid-19 from chest x-ray images using deep transfer learning. Med image Anal. 2020;65:101794.

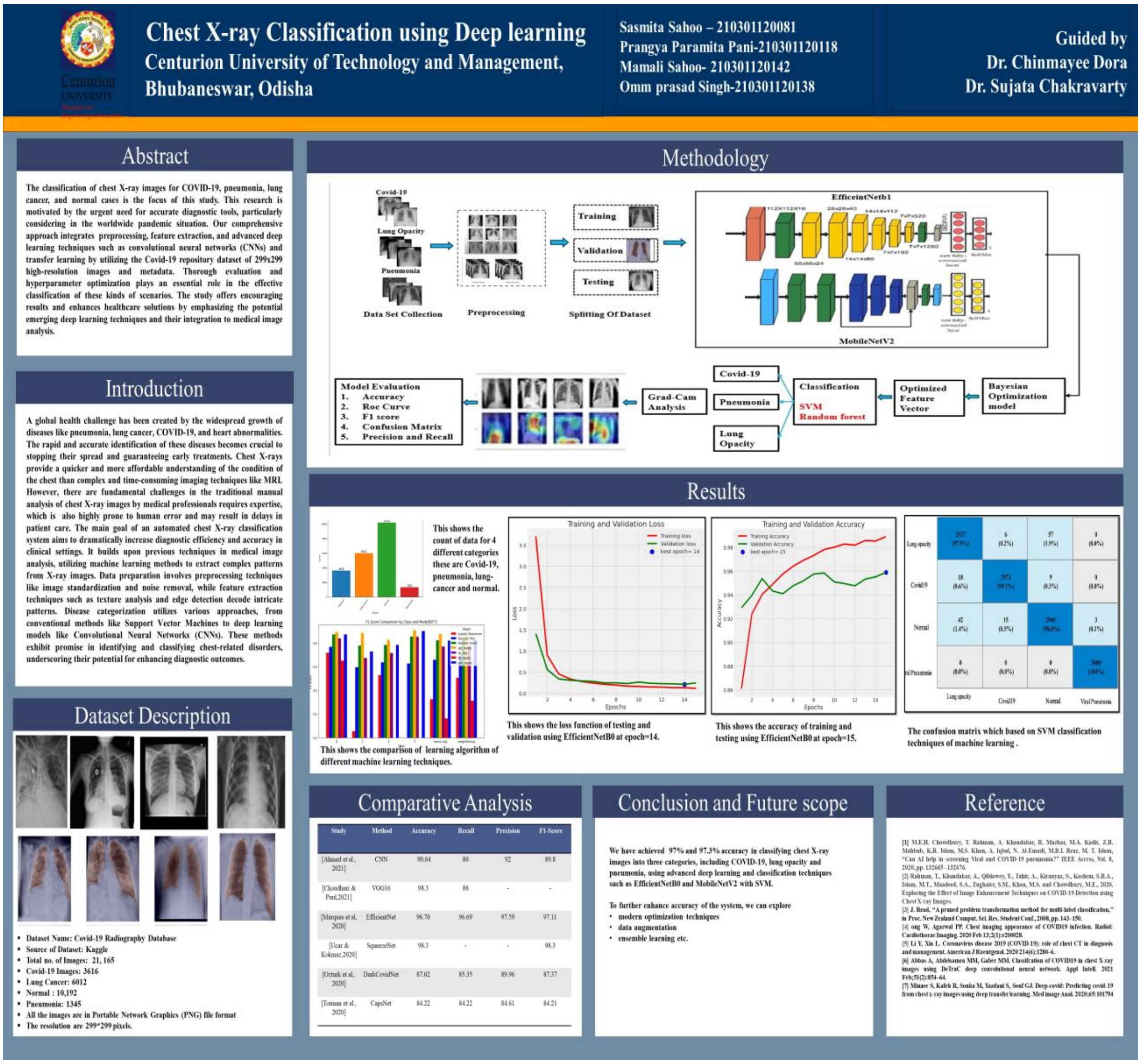
## APPENDICES

```
import os
import pandas as pd
import numpy as np
from PIL import Image, ImageOps
import plotly.express as px
import matplotlib.pyplot as plt
import seaborn as sns
import cv2
from tqdm import tqdm
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, precision_score, recall_score,
f1_score
import time

import tensorflow as tf
import keras
from keras import layers
from keras.models import Sequential
from keras.layers import Dense, Activation, Dropout
from sklearn.metrics import classification_report
from tensorflow.keras.callbacks import EarlyStopping
from keras.preprocessing.image import ImageDataGenerator
from sklearn.naive_bayes import MultinomialNB

import xgboost as xgb
from sklearn import svm
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
import os
os.environ['KAGGLE_USERNAME'] = "mansajyoti"
os.environ['KAGGLE_KEY'] = "e4bd7b9457db86f62466c55ce4a827a0"
!kaggle datasets download tawsifurrahman/covid19-radiography-database
```

```
levels = ['Normal', 'COVID', 'Lung_Opacity', 'Viral Pneumonia']
path = "COVID-19_Radiography_Dataset"
data_dir = os.path.join(path)

data = []
for id, level in enumerate(levels):
    for file in os.listdir(os.path.join(data_dir, level + '/images')):
        data.append(['{}/images/{}'.format(level, file), level])

data = pd.DataFrame(data, columns = ['image_file', 'result'])

data['path'] = path + '/' + data['image_file']

data.head()
print("Numbers of X-ray images: {}".format(data.shape[0]))
import seaborn as sns
import matplotlib.pyplot as plt

# Convert 'result' column to categorical
data['result'] = pd.Categorical(data['result'], categories=['Normal',
'COVID', 'Lung_Opacity', 'Viral Pneumonia'], ordered=True)

# Create count plot with specified order
sx = sns.countplot(x='result', data=data,
order=data['result'].cat.categories)
sx.set_xticklabels(labels=sx.get_xticklabels(), rotation=90)
plt.show()
print('Normal : ', list(data['result']).count('Normal'))
print('Covid : ', list(data['result']).count('COVID'))
print('Opacity : ', list(data['result']).count('Lung_Opacity'))
print('VIral Pneumonia : ', list(data['result']).count('Viral Pneumonia'))
round(data['result'].value_counts() / data.shape[0] * 100,2)
pixel_img = np.array(pixel_img)
label_img = data['result'].map({'Normal': 0, 'COVID': 1, 'Lung_Opacity' :
2,
                               'Viral Pneumonia' : 3})

print(pixel_img.shape, label_img.shape)
round(y_train.value_counts() / y_train.shape[0] * 100,2)
```

# APPENDIX B.

## a.Poster presentation:



**ASSESSMENT**

**Internal:**

| SL NO | RUBRICS | FULL MARK | MARKS OBTAINED | REMARKS |
|---|---|---|---|---|
| 1 | Understanding the relevance, scope and dimension of the project | 10 | | |
| 2 | Methodology | 10 | | |
| 3 | Quality of Analysis and Results | 10 | | |
| 4 | Interpretations and Conclusions | 10 | | |
| 5 | Report | 10 | | |
| | **Total** | **50** | | |

**Date:** **Signature of the Faculty**

**COURSE OUTCOME (COs) ATTAINMENT**

➢ **Expected Course Outcomes (COs):**

**(Refer to COs Statement in the Syllabus)**

________________________________________

________________________________________

________________________________________

________________________________________

➢ **Course Outcome Attained:**

**How would you rate your learning of the subject based on the specified COs?**

| ▢ | ▢ | ▢ | ▢ | ▢ | ▢ | ▢ | ▢ | ▢ | ▢ |
|---|---|---|---|---|---|---|---|---|---|
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |

**LOW** **HIGH**

➢ **Learning Gap (if any):**

________________________________________

________________________________________

________________________________________

________________________________________

➢ **Books / Manuals Referred:**

________________________________________

________________________________________

________________________________________

________________________________________

**Date:** **Signature of the Student**

➢ **Suggestions / Recommendations:**

**(By the Course Faculty)**

________________________________________

________________________________________

________________________________________

________________________________________

**Date:** **Signature of the Faculty**