TUBERCULOSIS PROGNOSIS THROUGH RADIOGRAPHIC PREDICTIVE MODELING

A Thesis Submitted
In Partial Fulfillment of the
Requirementsfor the Degree of

MASTER OF TECHNOLOGY

in
Artificial Intelligence
by

RITIK JAIN
(Roll No. 2K22/AFI/16)

Under the Supervision of
Prof. RAJNI JINDAL
(Prof, Dept of Computer Science & Engineering)

To the
Department of Computer Science and Engineering

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of
Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-110042.
India

May, 2024

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

ACKNOWLEDGEMENTS

I have taken efforts in this survey paper. However, it would not have been possible without the kind support and help of many individuals and organizations. I would liketo extend my sincere thanks to all of them.

I am highly indebted to **Prof. Rajni Jindal** for her guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing this review paper. I would like to express my gratitude towards the **Head of the Department (Computer Science and Engineering, Delhi Technological University)** for their kind cooperation and encouragement which helped me in the completion of this research survey. I would like to express my special gratitude and thanks to all the Computer Science and Engineering staff for giving me such attention and time.

My thanks and appreciation also go to my colleague in writing the survey paper and the people who have willingly helped me out with their abilities.

Ritik Jain

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, Ritik Jain, Roll No. 2K22/AFI/16 student of M.Tech (Artificial Intelligence), hereby certify that the work which is being presented in the thesis entitled "**Tuberculosis prognosis through radio-graphic predictive modeling**" in partial fulfillment of the requirements for the award of the Degree of Master of Technology in Artificial Intelligence in the Department of Computer Science and Engineering, Delhi Technological University is an authentic record of my own work carried out during the period from August 2022 to Jun 2024 under the supervision of Rajni Jindal, Prof, Dept of Computer Science and Engineering. The matter presented in the thesis has not been submitted by me for the award of any other degree of this or any other Institute.

Place: Delhi **Candidate's Signature**

This is to certify that the student has incorporated all the corrections suggested by the examiners in the thesis and the statement made by the candidate is correct to the best of our knowledge.

Signature of Supervisor **Signature of External Examiner**

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CERTIFICATE

Certified that **Ritik Jain** (Roll No. 2K22/AFI/16) has carried out the research work presented in the thesis titled "**Tuberculosis prognosis through radio-graphic predictive modeling**", for the award of Degree of Master of Technology from Department of Computer Science and Engineering, Delhi Technological University, Delhi under my supervision. The thesis embodies result of original work and studies are carried out by the student himself and the contents of the thesis do not form the basis for the award of any other degree for the candidate or submit else from the any other University /Institution.

Prof. Rajni Jindal
Professor Department of CSE
Delhi Technological University

Date:

Tuberculosis Prognosis Through Radio-Graphic Predictive Modeling
RITIK JAIN

ABSTRACT

In the field of lung disease classification, the fusion of diverse datasets shows significant challenges and opportunities for improving predictive accuracy. This study showcases the efficacy of various deep-learning models in classifying pulmonary tuberculosis using a combined dataset of CX-Rays sourced from diverse resources. Leveraging datasets from multiple origins and combining them, we performed classification on state-of-the-art ConvNets, including but not limited to, ResNet50, DenseNet121, EfficientNetB0 to B7. Through rigorous experimentation and analysis, we identified the model with the highest performance on the combined dataset. We identify the model with the best performance on the pre-processed dataset with the help of precise experimentation and analysis.

Additionally, to strengthen the classification results, additional layers and optimization techniques were applied into the best model. These enhancements ranged from feature extraction with transfer learning to fine-tuning, aimed at improving the model's predictive capabilities. By thoroughly comparing the performance of different convolutional models and techniques, we emphasize insights into the most effective approaches for pulmonary tuberculosis classification.

Our research illustrated the addition of traditional ConvNet Classification models with extra layers and techniques yields notable improvements in model evaluations. The outcomes of this research have shown how different models work on chest X-ray images and how they can be used in decision-making in healthcare and for the diagnosis of lung diseases

LIST OF PUBLICATIONS

1. Paper title: ‘Tuberculosis Prognosis Through Radiographic Predictive Modeling’ , 2nd International Conferences on Optimization Techniques in Engineering and Technology (ICOTET 2024) in May 2024.
2. Paper title: ‘A Review Of Advances In Pulmonary Tuberculosis Detection Using Deep Learning’ 1st International Conference on Applied Artificial Intelligence And Machine Learning (ICAAIML 2024) in May 2024

TABLE OF CONTENTS

List of Tables

List of Figures

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Tuberculosis (TB) is still a daunting health disease, targeting the large number of people every year. While significant work has been done in the detection and treatment of TB, early detection and prediction of the disease still present substantial hurdles. In this era of rapid technological advancement, leveraging the power of Machine Learning and Deep Learning [31], [24] techniques holds great promise in revolutionizing TB diagnosis and management.

According to the World Health Organization [58] in India 2,590,000 which is approximately 2.6 million cases in the year 2021 were projected in every 1,00,0000 population there are around 188 persons who are affected. Tuberculosis is a transmissible disease, therefore it is very important to detect it early as the impact of the delayed diagnosis on the individual can have huge concern to family and public health.

Machine Learning and Deep Learning [31] excel at analyzing complex medical data and making accurate forecasts. ML is skilled at recognizing patterns, automatically extracting features, and classifying or predicting outcomes. DL, with its deep neural networks, is particularly useful for tasks like analyzing medical images (e.g., X-rays) and handling high dimensional data, offering precision and the potential to revolutionize Tuberculosis prediction by efficiently interpreting intricate medical data and automating disease identification.

Radiographic predictive modeling [9] utilizes clinical imaging information, such as chest X-rays, Ultrasounds, MRIs, etc. to forecast or predict health-related outcomes, particularly in diagnosing and monitoring various medical conditions. It involves applying AI and machine learning techniques to interpret and analyze radiographic images for early disease detection, treatment planning, and patient prognosis.

1.2 OBJECTIVE

The project's aim is to explore how advanced deep learning technologies can enhance the accuracy, speed, and cost-efficiency of detecting pulmonary tuberculosis through X-rays (CXR) and clinical reports. By doing so, it seeks to improve patient outcomes and make public health interventions more efficient.

- Comparing Deep Learning Model how they perform with the same data, assessing their performance and their potential impact on TB diagnosis and management.
- Applying various computer vision techniques like histogram equalization[1], feature extraction,geometric transformation,augmentation and segmentation[16] for image enhancement and better prediction.
- Studying the role of AI in the automation of diagnosis of tuberculosis according to recent studies done on this.
- Studying how medical history can have effect on results of Tuberculosis Prognosis

CHAPTER 2

RELATED WORK

2.1 SYSTEMATIC LITERATURE SURVEY

The WHO Global Tuberculosis (TB) Report 2020 indicates a persistent global burden of pulmonary TB, with 10 million new cases, 465,000 drug-resistant cases, and 14 Lakh deaths reported in the year 2019. Despite a decrease in overall numbers, TB remains a significant infectious threat, claiming nearly 4,000 lives daily. Early screening and treatment are crucial, with two common tests: IGRA that is an Interferon Gamma Release assay and MTBs test (Monteux TB Skin). However, challenges arise due to limitations in sensitivity, cost, and implementation in low and middle-income regions (LMIRs).

CXRs are widely used for TB screening, especially in LMIRs, owing to their cheapness, low radiation, and ease of carry. While CXRs offer a two-dimensional view of three-dimensional anatomy, their effectiveness is hampered by modest specificity[4], [2]. Advanced ways of imaging like magnetic resonance imaging (MRI), CT(Computed Tomography) provide greater accuracy but are costly and require expertise[5].

In the last 10 years, there has been enormous research in automated CXR data analysis, focusing on deep neural networks and the neural networks using convolution i.e. CNN's.DL methods aim to address the expertise and access gaps in global TB screening efforts. Unlike traditional machine learning, DL extracts features automatically, eliminating the need for expert-based feature extraction[9], [11]. AI-guided solutions, such as DL methods, have the potential to assist clinicians in triaging cases, particularly in resource-limited regions[46], [38].

This review[22] follows a systematic workflow, encompassing identification, screening, eligibility, and inclusion criteria. The search focused on the areas like ‘Pulmonary TB’, 'CXRs,' and 'Models of Deep Learning’ in repositories such as PubMed and Web of Science. The review includes research articles published between 2016-2021, excluding reprints to ensure peer-reviewed content. The meta-analysis considers dataset characteristics, DL models, and corresponding performance scores.

The paper's structure involves sections on data collection, availability, and sources, followed by a comprehensive review of TB screening tools using CXRs from 2016-

2021[22]. The discussion encompasses performance comparison and analysis of DL-based medical imaging tools, considering factors like transfer learning, data augmentation, and disease localization. Recurrent neural networks[12] (RNNs) are special NN that process data in continuous form or the data where there is a relationship between each token like text, speech, or video. RNNs consist of the hidden state which is changed at every timestamp on the basis of the previous state and on the input. RNNs are used for various tasks, such as modeling of language, recognition of speech, translation of machines, and analysis of videos. RNNs have also been applied to clinical image dataset analysis, such as the diagnosis of TB from chest X-rays. RNNs can capture the temporal and spatial dependencies in the images and generate features that are useful for classification. However, RNNs also have some limitations, such as difficulty in learning long-term dependencies, gradient vanishing or exploding problems, and high computational cost.

Attention[10] is a method that gives the special vectors that give dynamic contextual semantics for a word and have better performance than word2Vec and traditional embedding methods and RNNs and LSTMs. Also unlike traditional textual models, they do not use sequential learning they work on parallel processing to find relationships between words and give the best vector embeddings. They use Neural Networks or weighted similarity to get the best results. Further, they were extended to self-attention and multi-head self-attention in the paper attention is all you all need. They have show good results for Tuberculosis diagnosis on chest x-ray's images from some convolutional neural networks [18].

Transformers[10] are the type of NN that use attention to process the data, that is in sequential form without using recurrence or convolution. Transformers have decoders to generate new sets of data by taking input from encoders which extract more informed descriptors from the data, they contain multiple layers of self-attention and FFN. Transformers can learn the dependencies and the representations of the input and the output sequences, without relying on the order or the length of the sequences. Transformers have achieved SOTA results in various NLP tasks, like machine translation, summarizing of text, and natural language understanding. Transformers have also been extended to CV tasks, like classification in images, segmentation in semantic form, and detection of objects. Transformers can also be used for clinical data analysis, such as diagnosis of pulmonary TB from chest X-rays[25]. Transformers can learn the global and local features of the images and generate high-level representations that are useful for classification.

Vision transformers[13] (ViTs) are special transformers that are inspired by ConvNet for image classification they divide images into 9 patches of equal sizes and convert them to embedding with position embedding as we do in the case of textual data also inspired from BERT has extra embedding then we pass to the transformer encoder model to get meaningful vector's and then we pass it to MLP head and get the final results. They have shown great results with images as they tell about the relationship between regions of the image. Recently many papers have used this to showcase how transformers can work on Tuberculosis Prognosis and additionally improve results with efficientNet as a base model to extract feature maps and then passing to ViT's to get S.O.T.A. solutions [39].

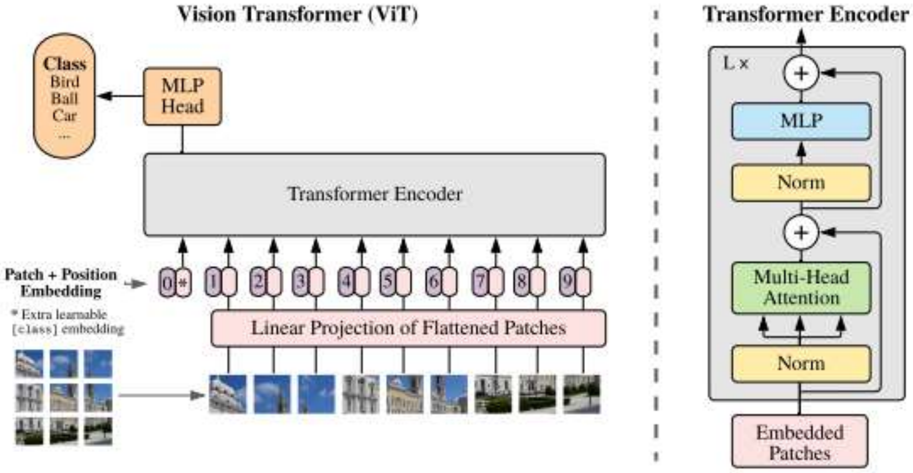


Fig. 1: Vision Transformer and Transformer Encoder model Architecture

2.2 PAPER AND DATASET STUDIED

Table 1: Literature Survey of latest paper related to Pulmonary Tuberculosis

Paper	Database	Techniques Used	Performance	Year
[21]	AMC dataset and public CXR dataset containing total of 35,985 CXRs (5893 TB & 30,092 normal)	Distillation for self-supervision and self train learning (DISTL), ViT	Average Accuracy of 93%	2022
[20]	The combined four data sets of CXR images	Input Enhanced ViT model	F1-score 96.39% - 100%, sensitivity 93.50% - 100% , precision 97.96% - 100%	2022

[23]	CheXpert and pediatric pneumonia dataset	VGGNet and ResNet as baseline models, ViT	Accuracy - 88.46%	2022
[26]	2233 NIAID TB dataset	Drug-resistant TB classifcation (2-class) (ML classifer)	Accuracy — 72%	2022
[27]	80,000 total images, where 3615 CXR images Covid -19, 5856 CXR images of pneumonia from RSNA, SIRM, and Radiopaedia, 20000 Lung cancer , 1400 CXR TB images etc.	VGG19 + CNN	96.48 % accuracy, 93.75 % recall, 97.56 % precision, 95.62 % F1 score, and 99.82 % area under the curve (AUC).	2022
[37]	3500 chest X-ray images of COVID-19, pneumonia, pneumothorax, tuberculosis, and normal,	DenseNet-201	Accuracy of 97.2%, 94.28% of sensitivity, and 97.92% of specificity	2022
[53]	Montgomery, Shenzhen, Belarus dataset	EfficientNetB3 with U-Net	Accuracy of 98.7%	2022
[34]	Shenzhen Hospital (SH) and Montgomery County (MC), Belarus and Kaggle CXR Lung Disease Dataset	DenseNet-169, U-Net and Attention	Accuracy 97.7%	2023

[41]	TBX, 11000 images	SymFormer	Accuracy - 95.1%	2023
[51]	718 (test only 82 images) **Private dataset**	TB classification (ResNet fused External Attention Network)	Accuracy — 95%	2023
[40]	1500 Private dataset	TB classifcation (deep denseNet)	AUC — 84%	2023
[33]	7000 images-, NLM dataset,the Montgomery and Shenzhen datasets, Belarus dataset: Belarus Set NIAID TB dataset and RSNA CXR dataset	ViT and Distillation model with ensemble technique	Clean Accuracy - 94.86%	2023
[44]	4200 images, Kaggle dataset	Voting ensemble classifier, GCN , autoencoder (AE), and extreme learning machine (ELM), DenseNet-121, Pelican Optimization	Accuracy of 98.83%.	2023

[47]	Chest X-ray images from 3314 patients infected with Mycobacterium tuberculosis (MTB) or non-tuberculosis mycobacterium (NTM)	Ensemble model combining EffcientNet B4 and ResNet 50	Accuracy of the ensemble model was 0.85 for MTB-LD and 0.78 for NTM-LD	2023
[32]	5000 image dataset of Montgomery, Shenzen and Tuberculosis (TB) chest X-ray dataset	Convolutional Block Attention Module (CBAM) and the Wide Dense Net (WDnet)	Accuracy (98.80%), sensitivity (94.28%), precision (98.50%), specificity (95.7%) and F1 score (96.35%)	2023
[35]	Montgomery and Belarus TB CXR dataset and 6432 CXR images from Kaggle of various Lung Diseases	TB-UNet, which is based on dilated fusion block (DF) and Attention block (AB) block for accurate segmentation and DenseNet	Accuracy (0.9510)	2023
[28]	8000 images of pneumonia, CoronaVirus Disease, Normal, and Tuberculosis.	VGG-19, Inception Net V3, and ResNet 50 with Grad-CAM	Accuracy of 95%, 87%, and 98%	2023
[29]	4200 images, Kaggle dataset (Tuberculosis (TB) Chest X-ray Database)	WSODTL-TBC, Water Strider Optimization Algorithm with LSTM	Accuracy - 98.9	2023
[43]	447 CXR segmented images	VGG-19 model with a softmax layer in conjunction with the ORB(Oriented FAST and Rotated BRIEF)	Accuracy- 98.2	2023

[36]	Collaborative CXR Dataset Of Various Lung Diseases	Custom VGG	Accuracy of 95-97% on the train split and 90-95% on the test split	2023
[45]	4200 images, Kaggle dataset (Tuberculosis (TB) Chest X-ray Database)	Median filtering based noise removal, U-Net segmentation, MobileNetv2 feature extraction, HHO based hyper parameter tuning, and gated recurrent unit (GRU) classifier . HHODL-TBC model	Accuracy 99.33%	2023
[42]	1400 images,4200 images subset of Kaggle dataset (Tuberculosis (TB) Chest X-ray Database)	OrthoSNet model using Discrete Ortho-Normal Stockwell transforms	accuracy of 99 %	2023
[48]	266 posteroanterior CXR images	U-Net segmentation, Pyradiomics for feature extraction, Pearson correlation for dimension Reduction and Random Forest	SD(AUC=0.9444 (95% CI, 0.8762; 0.9814)), variance (AUC=0.9288 (95% CI, 0.9046; 0.9843))	2023
[50]	NIH Dataset	CNN is combined with VGG, data augmentation, and the spatial transformer network (STN)	Accuracy - 97.56%	2023

[30]	posterior-to-anterior (AP)/anterior-to-posterior (PA) image of chest X-ray (citation below)	TL-AttSharpNet, a U-net-like architecture, VGG16 and attension for segmentation	0.9786 dice score and 0.9503 Jaccard	2023
[49]	2000 CXR image Dataset (IEEE)	EfficientNetB2	Accuracy - 96%	2023
[54]	QU-MLG-TB dataset 40,000 (CXR and 3037 for drug-resistant TB	SVM, KNN, Logistic Regression, Random Forest, AdaBoost, XG-Boost, LDA (linear discriminant analysis) and CheXNet Encoder	TB classifcation (accuracy: 93.2%), Drug-resistant TB (2-class classifcation) accuracy: ~87%, Drug-resistant TB (3-class classifcation) accuracy: ~80%	2024
[52]	1000 images IEEE Dataport-Tuberculosis (TB) chest X-ray database	Shallow-CNN, DenseNet	Accuracy — 95%	2024
[56]	5863 X-ray images, Women and Children’s Medical Center in Guangzhou from children aged one to five. Pneumonia kaggle dataset	ViT	Accuracy of 97.61%, sensitivity of 95%, and specifcity of 98%	2024
[55]	4200 images, Kaggle dataset (Tuberculosis (TB) Chest X-ray Database)	CLBO_DenseNet, integration of the chef-based optimization algorithm (CBOA) and hybrid leader-based optimization (HLBO) with DenseNet	Accuracy, TPR, TNR, PPV, NPV, F1-score, and dice coefficient with the values of 98.90%, 97.60%, 94.60%, 94.60%, 93.30%, 96.10%, and 96.50%	2024

The result and concluding section give the final insights of the literature survey and outcome of the model compared and optimization is done on the best model.

2.3 MODEL USED

2.3.1 Distillation for Self – Supervised Learning

Distillation for self-supervised learning (SSL) and training leverages two powerful techniques in machine learning: self-supervised learning and knowledge distillation, to achieve better model performance. Here's a breakdown of the concepts involved:

A) Self-Supervised Learning (SSL):

- Traditional supervised learning requires a lot of labeled data, which can be expensive and time-consuming to obtain.
- SSL tackles this challenge by training models on unlabeled data. The model learns meaningful representations from the data itself through tasks designed to extract useful features.
- For instance, an SSL model for image recognition might predict rotations of the same image or perform image inpainting (filling in missing parts of an image).

B) Knowledge Distillation:

- Knowledge distillation involves transferring knowledge from a complex, powerful model (teacher) to a smaller, faster model (student).
- The teacher model, trained on labeled data, has a good understanding of the task. The student model learns from the teacher's predictions, mimicking its behavior and improving its own performance.

C) Combining SSL and Distillation**:**

- Distillation for SSL combines these techniques by leveraging a pre-trained teacher model obtained through SSL.
- The teacher model's knowledge, extracted from unlabeled data, guides the training of the student model. This can be particularly useful in scenarios where labeled data is scarce.

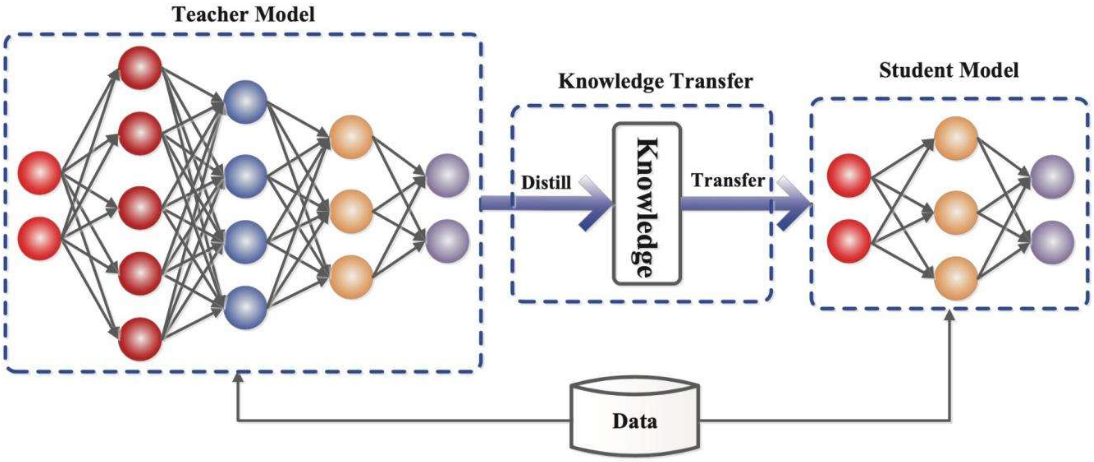


Fig 2: Model Architecture of Knowledge Distillation Model

2.3.2 SymFormer

Symmetric Search Attention (SymAttention) in SymFormer likely functions as follows:

- SymAttention exploits the inherent bilateral symmetry, the mirror-like property, present in most chest X-ray (CXR) images.
- During the feature extraction process, SymAttention concentrates the model's attention on corresponding regions on the left and right sides of the CXR image.
- By comparing features from these corresponding symmetrical regions, SymAttention aims to help identify subtle differences that might indicate abnormalities or pathology.

Symmetric Positional Encoding (SPE):

- Challenge**:** CXR images might not always be perfectly symmetrical due to various factors like patient positioning or pathology.
- Function**:** SPE addresses this by potentially recalibrating the features extracted from the image.
- Goal**:** This recalibration aims to make the features from corresponding left and right regions more comparable, even if the image isn't perfectly symmetrical.
- Analogy**:** Imagine adjusting the brightness or contrast of two sides of an image to make them more aligned for easier comparison.

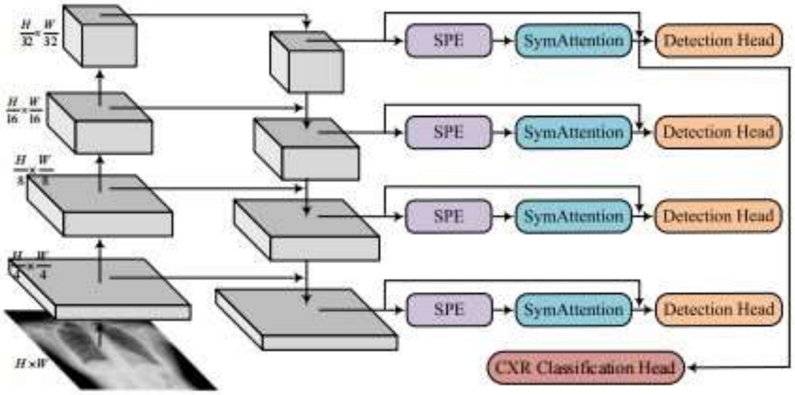


Fig 3: Model Architecture of SymFormer using FPN(Feature Pyramid Network)

2.3.3 VGGNet:

- Convolutional Core: VGGNet is a Convolutional Neural Network (CNN) that relies extensively on convolutional layers to extract features from image data.
- Depth: A defining feature of VGGNet is its depth. VGG16 includes 16 weight layers, comprising 13 convolutional layers, while VGG19 has 19 weight layers, including 16 convolutional layers.
- Small Filters: VGGNet employs small 3x3 convolutional filters. Despite their small size, stacking these filters enables a gradual increase in the complexity of the extracted features.
- Pooling Layers: The network incorporates pooling layers between the convolutional layers to downsample the image data, reducing its spatial resolution while preserving essential features.
- Fully Connected Layers: Following the convolutional layers, VGGNet includes fully connected layers that transform the extracted features into probabilities for different image classes.
- Strengths and Limitations: VGGNet is renowned for its high accuracy and simplicity, making it an excellent baseline for further research. However, its deep architecture demands substantial computational power and has a large number of parameters, which can lead to overfitting if not trained with ample data.

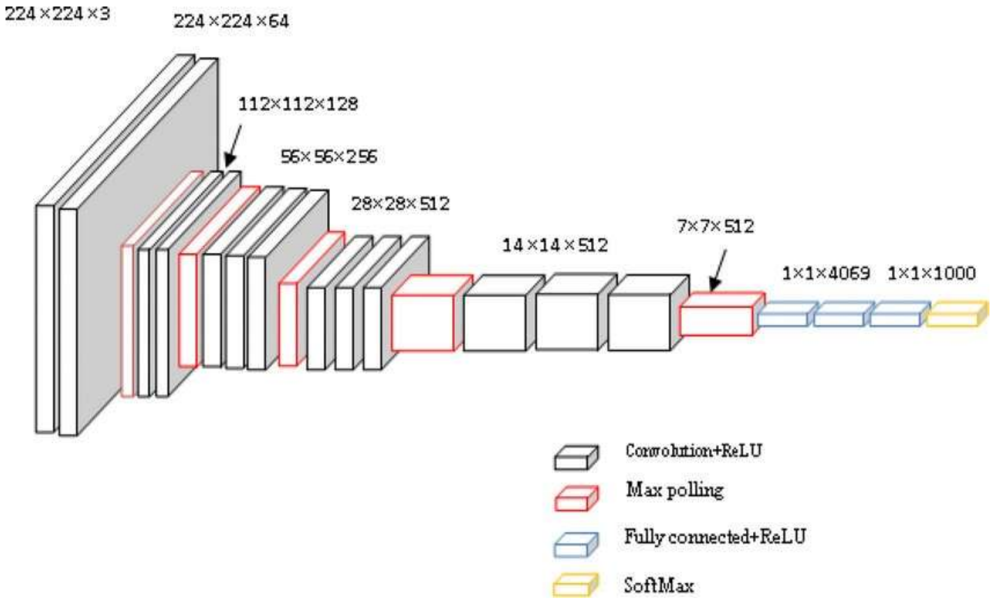


Fig 4: Model Architecture of VGG-16 Model

2.3.4 Stacking ML Classification Models:

These models are used to predict the category (class) that a new data point belongs to, based on a training dataset with labeled examples.

- Support Vector Machine (SVM):
    - Creates a hyperplane that separates data points of different classes in high-dimensional space.
    - Aims to maximize the margin between the hyperplane and the closest data points (support vectors).
    - Effective for high-dimensional data and works well with smaller datasets.
    - Can be computationally expensive for very large datasets.
- K-Nearest Neighbors (KNN):
    - Classifies a data point based on the labels of its k nearest neighbors in the training data.
    - Simple to understand and implement.
    - Performance can be sensitive to the choice of k and the distance metric used.
    - Can suffer from the curse of dimensionality in high-dimensional spaces.
- Logistic Regression:

  - Models the relationship between features and a binary class label (0 or 1) using a sigmoid function.
  - Interpretable - Coefficients of the model can provide insights into feature importance.
  - Not suitable for multi-class classification problems without additional modifications.
  - Assumes a linear relationship between features and the class label.
- <u>Random Forest:</u>
  - Ensemble method that combines multiple decision trees, trained on random subsets of features and data points.
  - Robust to overfitting and can handle a mix of data types (numerical and categorical).
  - Can be computationally expensive to train for large datasets.
  - Interpretation of individual tree decisions might be complex.

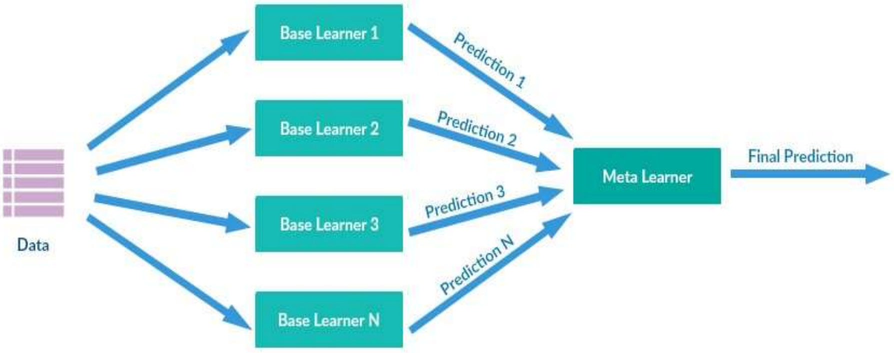


Fig 5: Representation of Stack Based Ensemble Learning

Ensemble Boosting Models:

These models combine multiple weak learners (models) to create a stronger final model.

- <u>AdaBoost (Adaptive Boosting):</u>
  - Sequentially trains weak learners, focusing on data points that were misclassified by previous learners in the ensemble.
  - Improves performance iteratively.
  - Can be sensitive to noisy data.

- XGBoost (Extreme Gradient Boosting):
    - Powerful boosting algorithm that uses decision trees as weak learners.
    - Employs regularization techniques to prevent overfitting.
    - Highly tunable with various parameters.
    - Can be more complex to set up compared to simpler algorithms.

Dimensionality Reduction Model:

- Linear Discriminant Analysis (LDA):
    - Finds a linear transformation that projects data points onto a lower-dimensional space while maximizing class separation.
    - Useful for visualization and reducing computational cost in high-dimensional settings.
    - Assumes a Gaussian distribution of data within each class.
    - Not suitable for non-linear relationships between features and the class label.

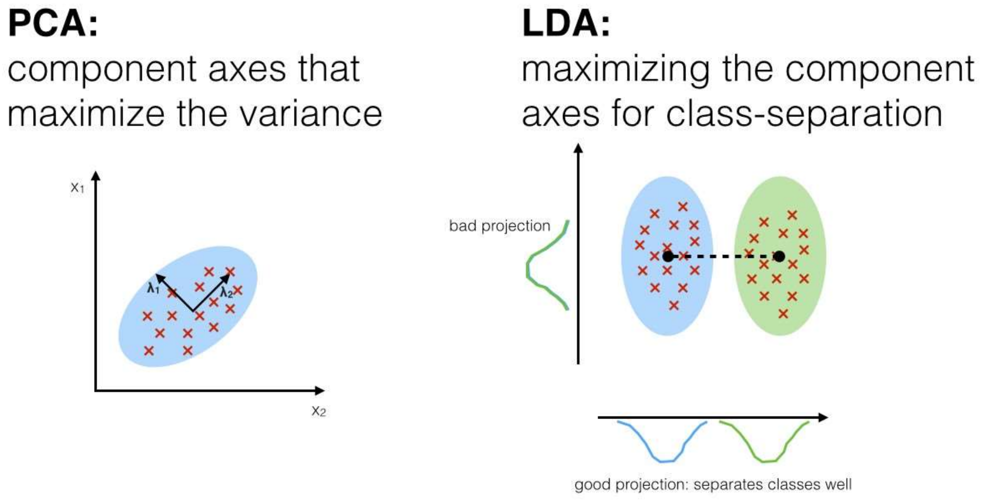

PCA:
component axes that maximize the variance
x_1
x_2
λ_1
λ_2
LDA:
maximizing the component axes for class-separation
bad projection
good projection: separates classes well

Fig 6: Comparison of PCA(Principle Component Analysis) and LDA

2.3.5 Shallow CNN for Tuberculosis (TB) Detection

- Deep learning models excel in image classification but lack interpretability, which is crucial for medical professionals.
- Shallow Convolutional Neural Networks (S-CNNs) have fewer layers, enhancing interpretability and making them suitable for medical diagnosis.
- S-CNNs are composed of input, convolution, pooling, dense, and

classification layers designed for efficiency and feature extraction relevant to TB.
- S-CNNs require less training time and computational resources compared to deeper models.
- They achieve similar accuracy to deep models like AlexNet but are more interpretable for medical use.
- S-CNNs balance accuracy and interpretability, aiding doctors in making informed decisions for TB detection in chest X-rays.

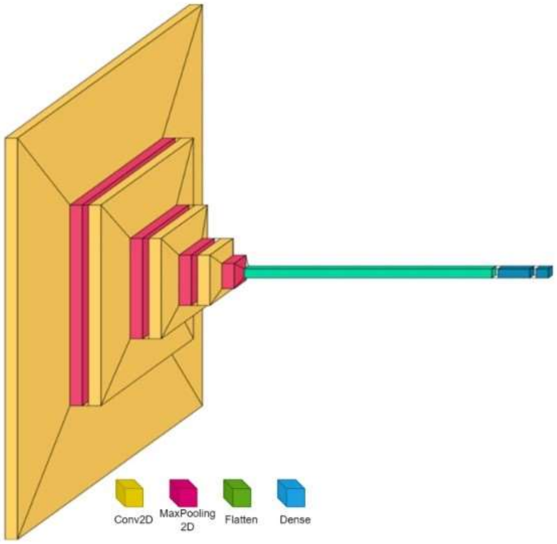


Fig 7: Model Architecture of Shallow- CNN

2.3.6 Voting Ensemble Classifier with GCN, AE, and ELM for TB Detection

This ensemble classifier combines three powerful models to potentially improve TB detection accuracy in chest X-rays while offering some interpretability benefits. Here's a deeper dive into each model and how they might work together:

1. Graph Convolutional Network (GCN):

- <u>Focus</u>: To improve feature extraction, GCNs take advantage of spatial relationships between image parts.
- <u>Graph construction</u>: Create a graph with nodes representing pixels and edges representing their connections (adjacency or similarity) using the chest X-ray image.

- Feature Learning: GCNs enhance the identification of TB-related patterns by performing convolutions on graphs that take into account the properties of a pixel as well as those of nearby pixels.

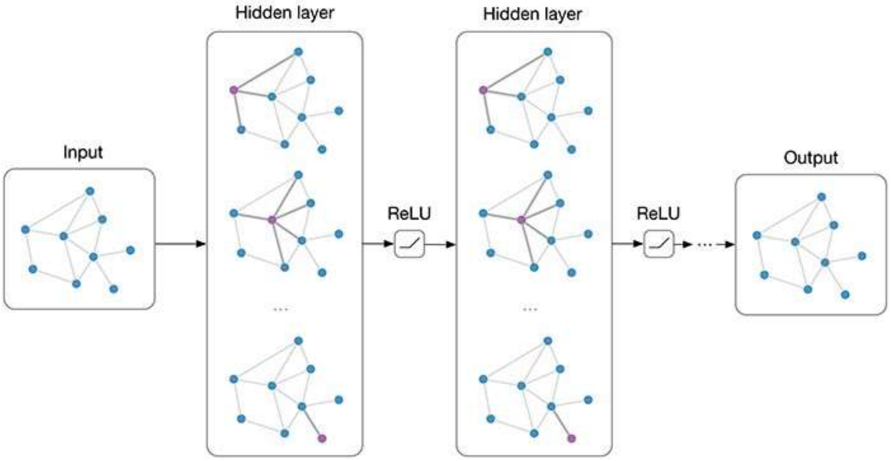


Fig 8: Representation of Graph Neural Network

2. Autoencoder (AE):

- Focus: Acquire an abridged depiction that accentuates salient characteristics associated with tuberculosis detection.
- Encoding: Using a lower-dimensional latent representation, compress the chest X-ray image while retaining its most important properties.
- Decoding (Optional): To make sure the encoded features are relevant and instructive, recreate the original image from the latent representation.

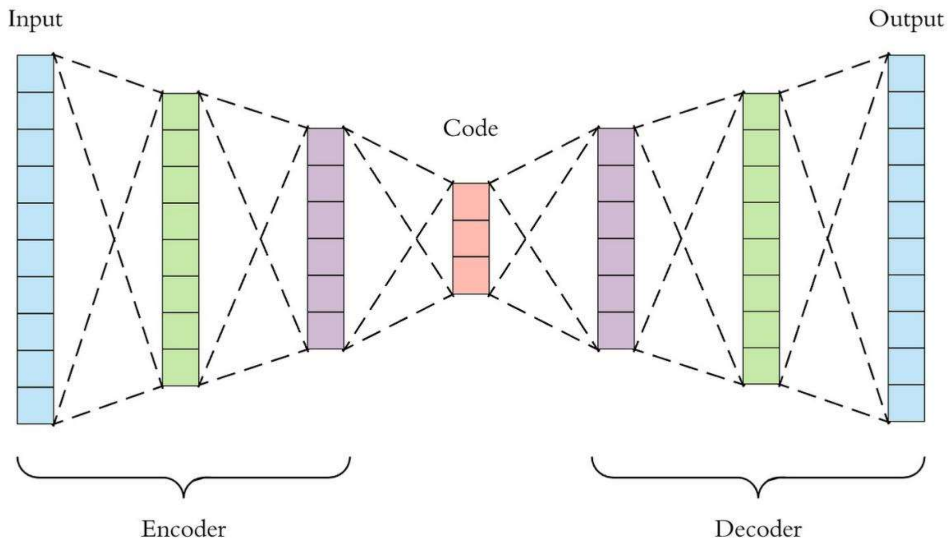


Fig 9: Model Architecture of Autoencoder Model

Voting Ensemble Mechanism:

- Each model (GCN, AE, ELM) independently analyzes the chest X-ray and outputs a classification (normal or abnormal).
- Prediction of each model is then passed to Voting Model and The one with majority is set as a class.

Benefits of Ensemble Approach:

- Greater Accuracy: The ensemble may be able to attain greater overall accuracy than any one model by integrating the advantages of each model. While AE concentrates on important features, GCN records spatial relationships, while ELM offers a quick categorization layer.
- Less Variance: The forecasts' variance is lessened with the use of ensemble voting. The majority vote can help to reduce error and provide a more reliable categorization even in cases when one model makes a mistake.
- Partial Interpretability: Although not as interpretable as more straightforward models, knowing the functions of each component (GCN, AE) might shed light on the characteristics the ensemble uses to classify data.

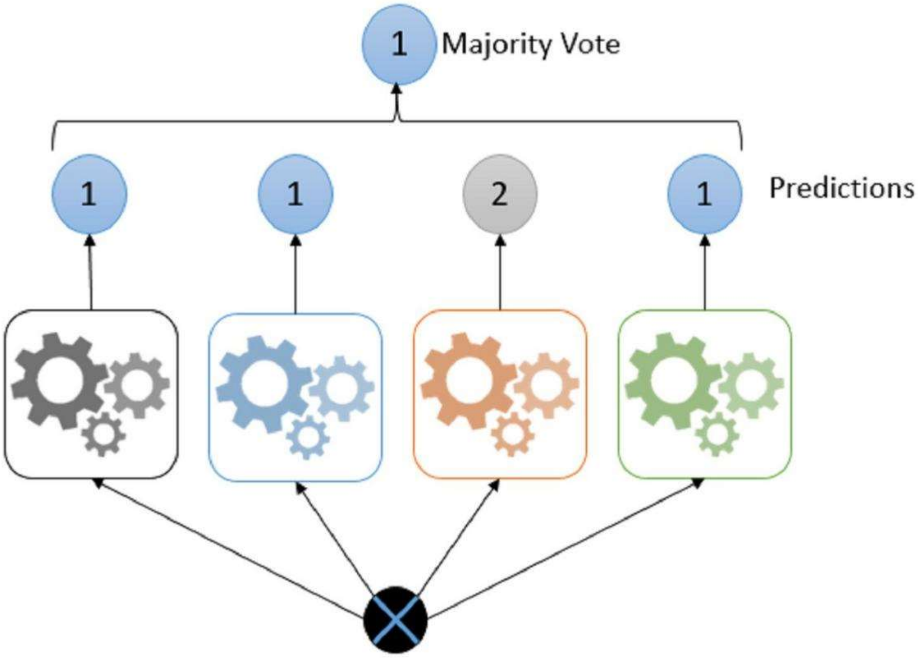


Fig 10: Representation of Majority Voting Ensemble Technique

Overall, this voting ensemble classifier offers a data-driven approach to TB detection in chest X-rays. By combining the feature extraction capabilities of GCN and AE with the fast classification of ELM, the ensemble aims to achieve accurate and robust results.

2.3.7 MobileNetV2.

- Goal: MobileNetv2 is intended to efficiently classify images on embedded and mobile devices.
- Depthwise Separable Convolutions:
    - Depthwise Convolution: Accelerates computation by applying a single filter to each input channel.
    - Pointwise Convolution: 1x1 convolutions are used to combine feature maps from the depthwise step.
- Linear bottlenecks: Prior to applying non-linear activations, feature maps are made less dimensional, which reduces computation while maintaining learning capacity.

- Efficacy: MobileNetv2 is more memory-efficient and faster than conventional CNNs, making it the perfect choice for handsets with limited resources.

- Accuracy: Despite its lightweight architecture, it achieves competitive accuracy on picture classification tasks.

- Transfer Learning: A variety of image identification tasks can be made easier with the availability of pre-trained models.

- Applications: Beneficial for embedded vision systems in devices with constrained computational power, real-time image recognition, and mobile image categorization.

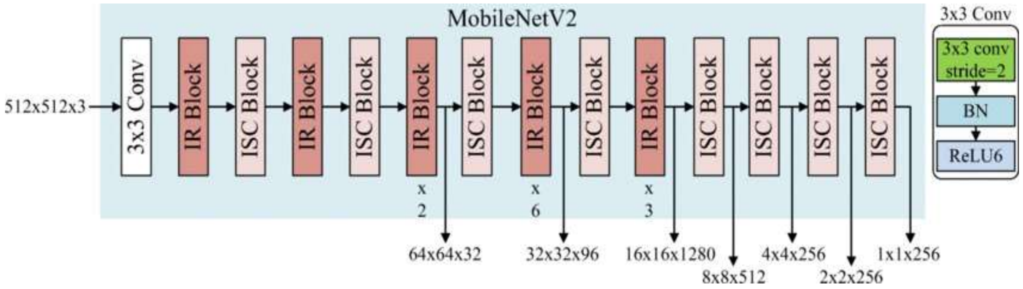


Fig 11: Model Architecture of MobileNetV2

2.3.8 GRU

- Goal: Developed to solve the RNN vanishing gradient issue, allowing for long-term dependency learning.
- Sequential Data Processing: Excel spreadsheets containing time series data, tasks involving sequential picture classification, and natural language processing.
- Mechanism: Update and reset gates are used to regulate the flow of information and save pertinent data.
- Efficiency: In comparison to LSTMs, simpler and more computationally efficient.
- Applications: Machine translation, anomaly detection, and speech recognition.
- Overall Utility: GRUs balance computational efficiency and dependency learning when processing sequential input.

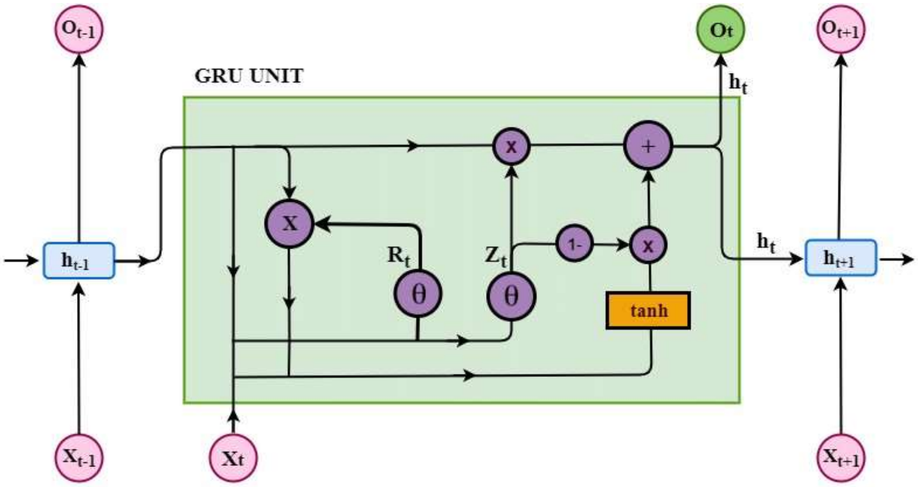


Fig 12: Model Architecture of LSTM model.

2.4 SEGMENTATION

2.4.1 U-NET

U-Net is a CNN architecture designed for image segmentation, especially in medical imaging. It classifies each pixel in an image to separate objects from the background. The architecture features an encoder to capture contextual information and a decoder to upsample and combine features, ensuring accurate pixel-wise classification. U-Net excels with limited labeled data and addresses vanishing gradients, providing accurate segmentation results.

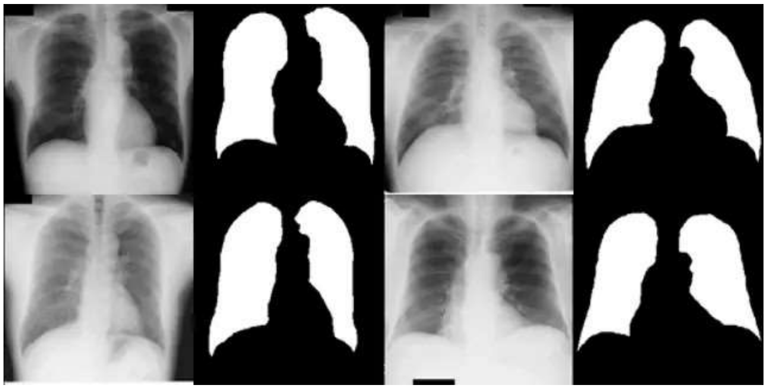


Fig 13: CXR images after UNet Segmentation

2.5 VISUALIZATION

2.5.1 Grad-CAM

Grad-CAM visualizes the regions a CNN focuses on for classification, aiding in understanding model decisions. It is particularly useful for diagnosing lung diseases by highlighting influential areas in chest X-rays or CT scans. Grad-CAM uses a pre-trained CNN to calculate gradients and create a weighted heatmap over the original image. This technique enhances interpretability, helps in early disease detection, and guides radiologists to areas needing closer examination, improving diagnostic workflow. However, CNNs remain complex, and Grad-CAM focuses on the most prominent features and the final convolutional layer, which may miss some relevant information. Despite these limitations, Grad-CAM is valuable for visualizing CNN decision-making in lung disease classification.

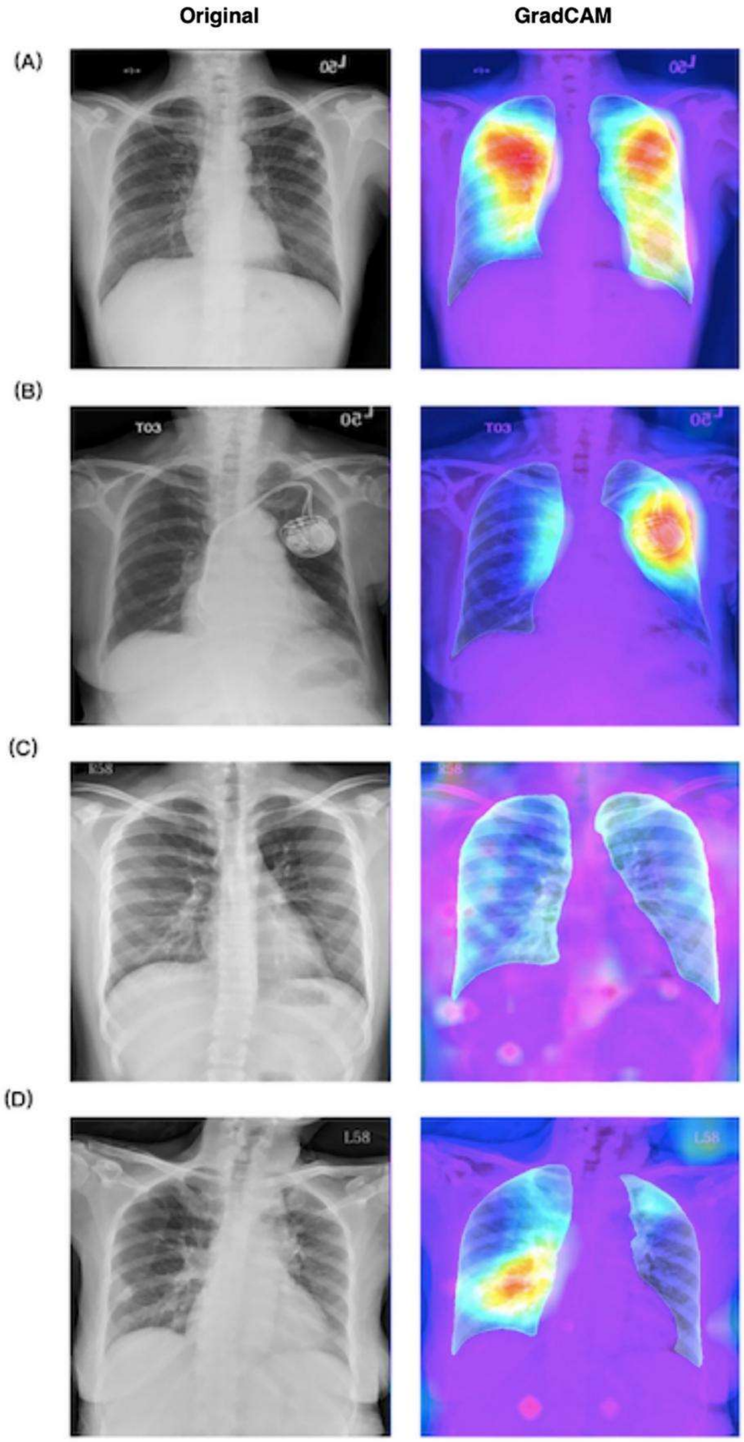


Fig 14: Tuberculosis Affected Areas in CXR images after Grad-CAM

CHAPTER 3

MATERIAL AND METHODOLOGIES

3.1 DATASET

- Shenzhen Dataset: Total of 662 cases – where 50.75 percent pulmonary TB cases and 49.25 percent normal case. This dataset is from one of the hospitals of China located in Guangdong, the city of Shenzhen. The images are chest X-rays of people containing both types of categories [3].
- Montgomery Dataset: Total of 138 cases – where 58 percent pulmonary TB cases and 42 percent normal cases. This dataset is from health services of USA located in MD collected to study pulmonary tuberculosis in the initiative to control this disease. The images are chest X-rays of people before and after treatment of tuberculosis [3].
- NIAID Pulmonary Tuberculosis dataset: Total of 3087 cases- where ~97percent pulmonary TB cases and 3percent normal cases. These are CXRs collected as part of TB program.
- Qatar and Bangladesh Dataset: This dataset is from Malaysia Hamad Medical Corporation and Bangladesh with the research team from Qatar University, Doha, and Dhaka University. Total of 4200 cases - where 17 percent pulmonary TB cases and 83 percent Normal case [57].

3.2 IMAGE PROCESSING

3.2.1 Resizing

The original Image size is (~3000, ~3000,4) I.e. height and width are approximately 3000 pixels and number of channels are 4. So, images are resized based on model requirements like for Resnet 50 the image is resized to (224,224,3).

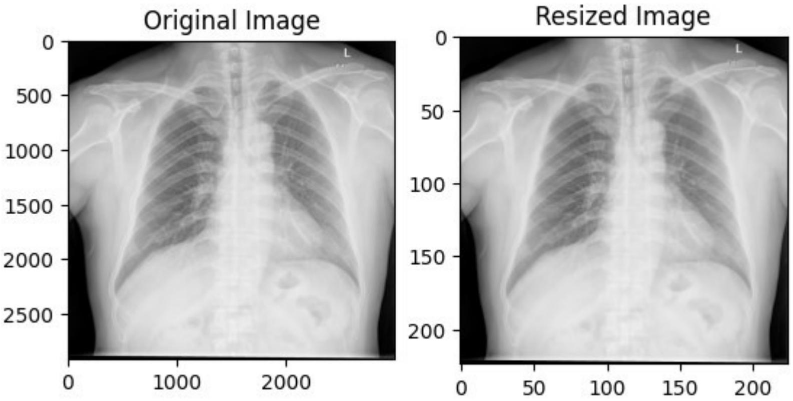
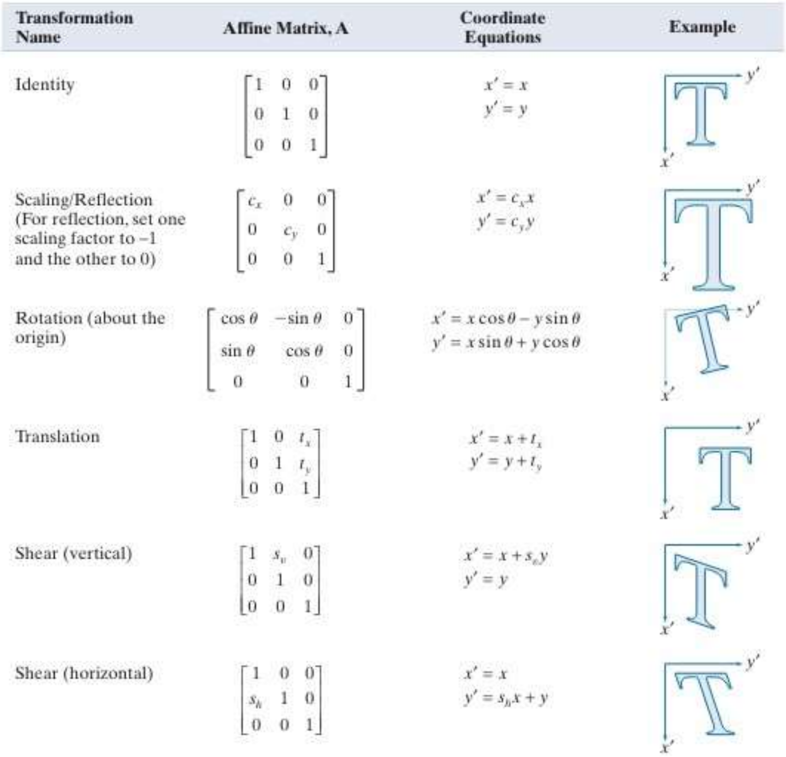


Fig 15: Re-sized image from (3000,3000) to (256,256)

3.2.2 Image Shearing, Zooming and Horizontal Flipping

Various Data Transformation techniques are applied like:



Transformation Name	Affine Matrix, A	Coordinate Equations	Example
Identity	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x$ $y' = y$	
Scaling/Reflection (For reflection, set one scaling factor to −1 and the other to 0)	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = c_x x$ $y' = c_y y$	
Rotation (about the origin)	$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x\cos\theta - y\sin\theta$ $y' = x\sin\theta + y\cos\theta$	
Translation	$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x + t_x$ $y' = y + t_y$	
Shear (vertical)	$\begin{bmatrix} 1 & s_v & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x + s_v y$ $y' = y$	
Shear (horizontal)	$\begin{bmatrix} 1 & 0 & 0 \\ s_h & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x$ $y' = s_h x + y$	

Fig 16: Image-Digital Image Processing by Rafael and Richard from Unit 2 affine transformations

3.2.3 Image Segmentation

In this digital image processing task, entails partitioning an image into various segmentsor regions according to specific criteria. The target of it is to simplify the representation of an image or make it more meaningful for further analysis. Each segment or region in the imageusually corresponds to objects or meaningful areas, making it easier to extract informationfrom the image.

We used the cv2 library K-Means algorithm for Image segmentation[6] as it gives a clear view of portions of tb-affected areas on CXR image by taking k value equal to 3.

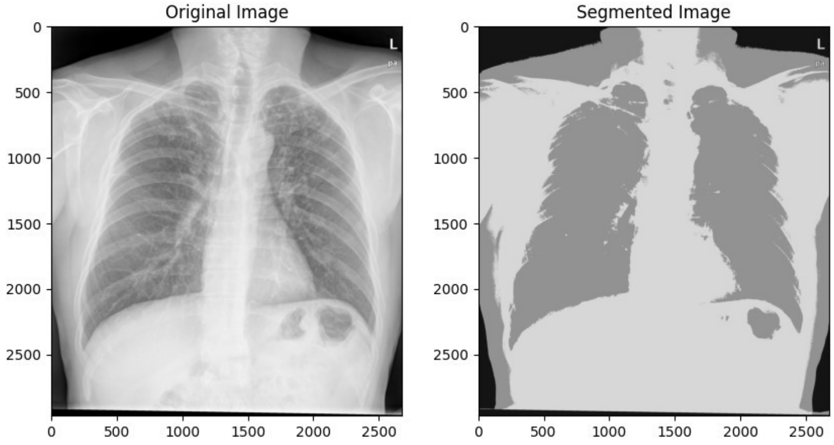


Fig 17: CX-Ray Image Segmentation

3.2.4 Histogram Equalization[1]

It is a technique used in digital image processing and advanced computer vision to sharpen the image by fixating it contracted pixel intensities value by spreading, i.e. goal is to obtain a more uniform distribution of pixel intensities, making details more visible, especially in images with poor contrast.

- Compute the Histogram: Calculate the frequency of each pixel value of the input image i.e. there are intensities that can be [0-9] or [0-255] level depending on the scale. Being continuous it is represented in the form of a histogram.
- Calculate the Probability Distribution Function (PDF): For the given intensity histogram calculate the probability of individual intensity level frequency with respect to the total number of pixels in a image.

  PDF(Pi) = Fi/sum(Fi)(i =0 to 255) or scale in which we are representing

No index entries found.

- Calculate the Cumulative Distribution Function (CDF): Compute the cumulative distribution function from the PDF calculated. The CDF represents the cumulative probabilities of the level of intensities.
- Normalize the CDF: Normalize the CDF to map the intensity values to the full range (0 to 255 in the case of 8-bit images).
  NormalizedCDF(i)= (CDF(i)MinCDF) * 255 / (MaxCDF – MinCDF)
- Create the Mapping Function: Now map the old intensities value with normalized CDF values or equalized ones.
  EqualizedIntensity(i) = NormalizedCDF(i)
- Apply Equalization: Apply the equalized intensities to each pixel value.

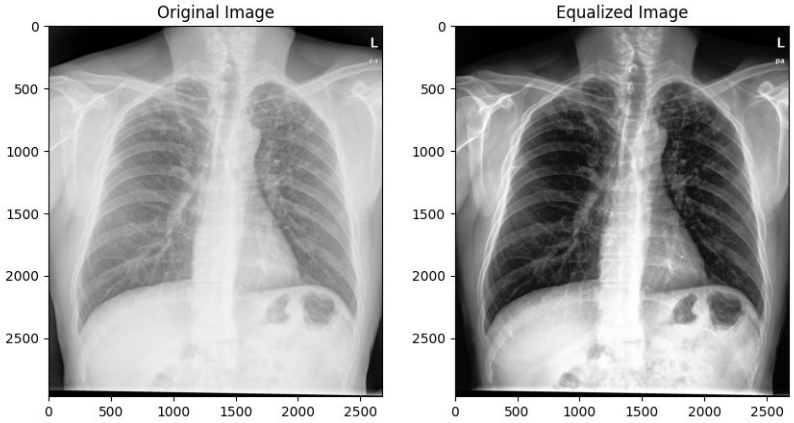


Fig 18: Chest X-rays with Histogram Equalization

3.3 MODEL SELECTION AND PREDICTION

3.3.1 ResNet50[7]

Residual Network with 50 layers, is a powerful neural network architecture that uses convolutions that are CNNs known for its remarkable efficiency and accuracy in image recognition, categorization, and object detection. It was introduced in the 2015 paper "Deep Residual Learning for Image Recognition," and its innovations revolutionized the field of deep learning.

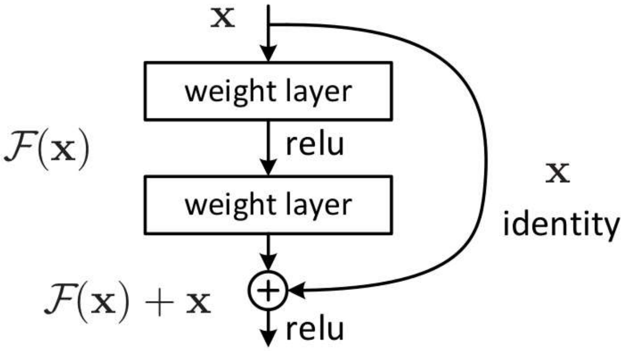

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Fig 19: Residual block of ResNet-50 Model

Key Features:

- Residual connections: The core concept involves bypassing layers in the network with identity connections, allowing gradients to flow directly, and addressing the drawback of deep neural networks which is the problem during backpropagation in the learning process of gradient loss or vanishing gradients.
- Pre-trained models: These models are hugely available on various image datasets and allow for transfer learning, facilitating quick adaptation to new tasks.
- Depth: With 50 layers, ResNet-50 can extract complex features and achieve high accuracy.
- Bottleneck blocks: These blocks also improve efficiency as in traditional architectures, the count of trainable parameters is more but by skipping the connection, the number of parameters is reduced.

Benefits:

- High Accuracy: ResNet-50 consistently performs well on various image tasks, surpassing previous architectures.
- Efficient Training: Compared to other deep networks, ResNet-50 can be trained more efficiently with a lower risk of vanishing gradients.
- Transfer Learning: Pre-trained models accelerate development, reducing training time and computation resources.

3.3.2 DenseNet121

DenseNet121 [8] is a neural network architecture that uses convolution that is CNN known for its efficiency and efficiency in image recognition tasks. It is characterized by its dense connections, here every layer gets the feature map as input of all previous layers by concatenating them.

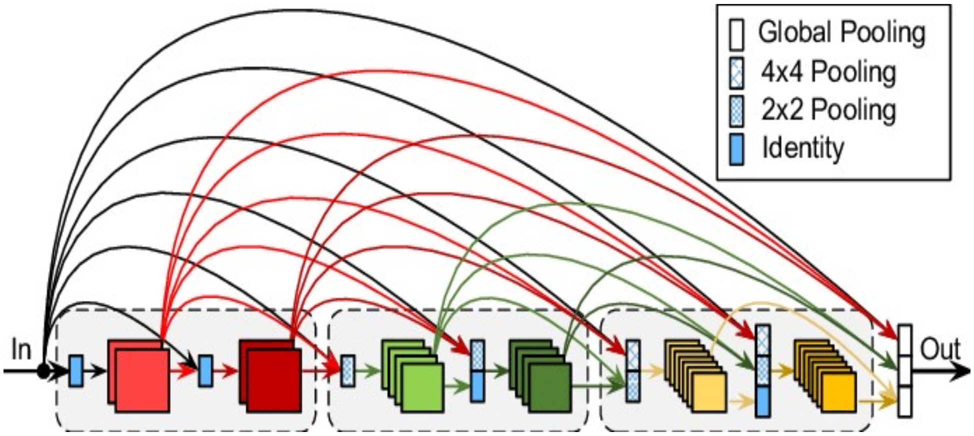


Fig 20: Model architecture of DenseNet-121

Key Features:

- Dense Connectivity: Each layer in the network gets the feature map as input of all previous layers, reusing features and information flow.
- Growth rate: Each layer adds a fixed number of filters, allowing for a gradual increase in complexity and feature representation.
- Bottleneck Design: Bottleneck layers help in decreasing the learnable parameters as they are already trained.
- 121 layers: The "121" are the number of layers in the network, offering a balance between depth and efficiency.
- Pre-Trained models: These models are hugely available on various image datasets to facilitate transfer learning.

Benefits:

- High Accuracy: They achieve S.O.T.A. Performance on different image classification benchmarks.
- Efficient Training: Dense connectivity promotes feature reuse, leading to faster training and lower computation costs.
- Reduced parameter Count: The bottleneck design helps in decreasing the number of parameters, resulting the network efficiency in deployment on limited resources.

3.3.3 EfficientNet [14]

A CNN model that achieves high accuracy on image recognition tasks while being efficient and having balanced accuracy. They were introduced in the year 2019

Key Features:

- Compound Scaling: It scales network width, depth, and resolution simultaneously using fixed coefficients so that the model can process higher-resolution images while maintaining efficiency. For example, by increasing the network depth by α^N width by β^N and image size by γ^N we can use 2^N more computation resources. Here α, β, γ are constant coefficients determined by grid search on the model. The model uses compound coefficient ϕ to uniformly scale network these parameters.
- MobileNetV2 Inspired Blocks: They utilize MobileNetV2's inverted bottleneck blocks, which are efficient and have good accuracy.
- Squeeze-and-Excitation Optimization: Results in the network to focus on important features within a channel.

Benefits:

- State-of-the-Art Accuracy: On the image classification task they have shown state-of-the-art results in the ImageNet dataset.
- Efficiency: They have offered significant reductions in model size and FLOPs (floating-point operations), making it ideal for deployment on devices with limited resources.
- Transfer Learning Capabilities: These models excel in transfer learning tasks. They deliver impressive results even when fine-tuned on smaller datasets.
- Scalability: They come with sizes from B0 to B7, smaller models working better with less resource devices like mobile and large models with powerful hardware.

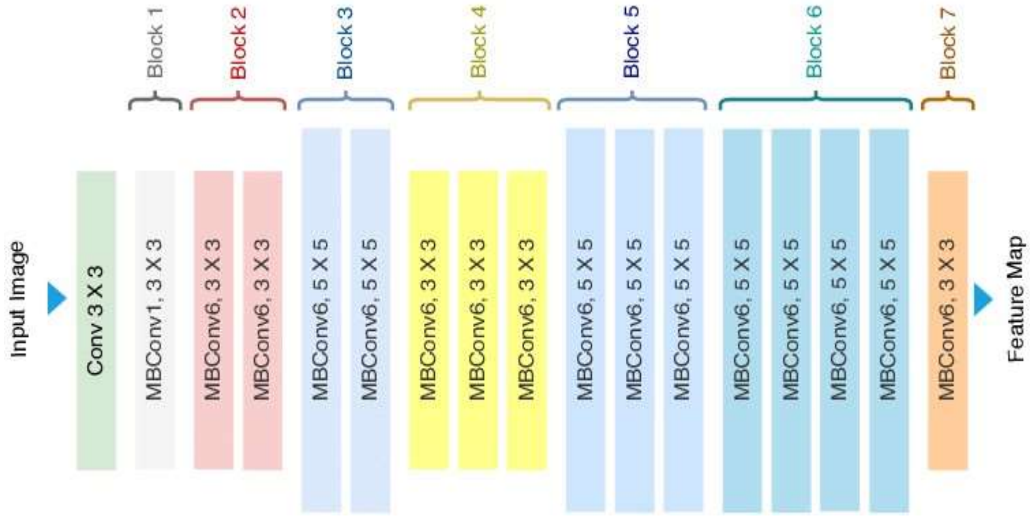


Fig 21: Model architecture of EfficientNet

3.3.4 ConvNeXT [19]

These are convNet inspired by transformers by modifying the standard ResNet to resemble Vision Transformer, identifying key components that improved performance. They were first introduced in 2020 and published in 2022.

Key Features:

- Depth-wise Separable convolutions: Similar to self-attention in transformers, they efficiently process information within a channel.
- Increased Channel Width: More channels allow for richer feature extraction.
- Simplified Normalization : Replacing batch normalization with layer normalization that is used in transformers
- New Activation Function: Replacing ReLU with GeLU (Gaussian Error Linear Units) slightly improves performance
- Down-Sampling Layer: Better handling of resolution change in images.

Benefits:

- Achieves S.O.T.A. accuracy on benchmarks like ImageNet, competing with ViT's.
- Efficiency: Better for image segmentation and object detection, faster training and less resources requirement for deployment
- Scalability: Various sizes in ConvNeXT family, for better accuracy and efficiency based on needs.

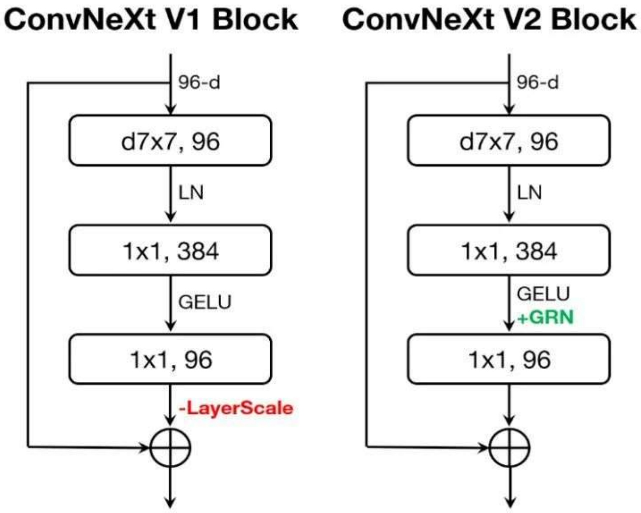


Fig 22: Representation of residual block of ConvNextV1 and ConvNextV2

3.4 OPTIMIZATION TECHNIQUES

3.4.1 Batch normalization

It is a method for deep neural network training. It assists in resolving an internal covariate shift problem, which can cause training to lag. Here are a few advantages of it:

- Quicker learning: Batch normalization speeds up the learning process by stabilizing the network.
- Lessening of the requirement for meticulous initialization: batch normalization makes training more forgiving by lessening the network's sensitivity to the initial weight settings.
- Higher learning rates: Using higher learning rates during training with batch normalization frequently enables you to train more quickly.
- Regularization: By lowering overfitting and enhancing the model's capacity for generalization, batch normalization can serve as a type of regularization. You could even be able to rely less on other strategies like dropout thanks to this.

3.4.2 Dropout

It alludes to a method for dealing with overfitting in artificial neural networks during training. Here's a brief summary:

- What it does: A predetermined percentage of neurons (nodes) from a layer are randomly removed by dropout during training. For the duration of that training cycle, these fallen neurons are ignored.
- The reasons it's beneficial as a result, the network is kept from growing unduly dependent on any particular neuronal connections. It compels the network to pick up stronger features and lessen its sensitivity to the particular training set.
- Advantage: Better generalization, the ability of the model to function well on unobserved data—is the result of less overfitting.

A straightforward yet powerful technique for enhancing neural network performance is dropout.

3.4.3 Reduce Learning Rate on Plateau (RLRP):

- What it does**:** This technique monitors a specific metric (usually loss) on the validation set during training. If the metric stops improving for a certain number of epochs (patience), the learning rate is reduced by a predefined factor.

- Why it helps**:** When the model reaches a plateau in learning, a high learning rate can cause it to oscillate around the minimum or even jump past it. Reducing the learning rate allows for finer adjustments and helps the model converge to a better minimum. This can lead to improved performance and avoid getting stuck in local minima.

Early Stopping:

- What it does**:** This technique also monitors the validation performance during training. However, instead of focusing on short-term plateaus, it looks for a more significant trend. If the validation performance stops improving for a certain number of epochs (patience), training is completely stopped.
- Why it helps**:** Early stopping prevents the model from overfitting to the training data. As training progresses, the model might start memorizing noise in the data, leading to poor performance on unseen data (generalization). Early stopping avoids this by stopping training before this memorization starts.

Working Together:

- RLRP and early stopping are complementary techniques. RLRP helps the model find the best minimum within the landscape, while early stopping prevents it from spending time exploring irrelevant areas after reaching a good minimum.
- You can use them together to achieve better training results. RLRP can give early stopping a better starting point by allowing the model to refine its weights further before stopping training completely.

Key Differences:

- Focus**:** RLRP focuses on short-term plateaus in the validation metric, while early stopping looks for a longer-term trend of stagnation.
- Action**:** RLRP adjusts the learning rate, while early stopping terminates training altogether.

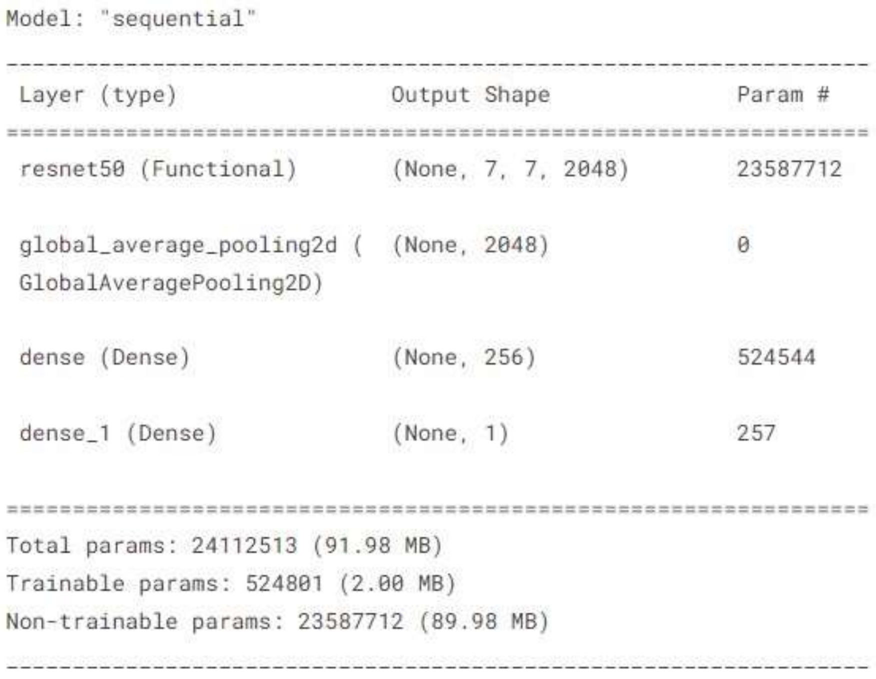
In essence, RLRP gives the model smaller steps to take when stuck, while early stopping recognizes when it's time to stop walking altogether**.**

CHAPTER 4

RESULT AND EVALUATION

4.1 MODELS AND ARCHITECTURE

4.1.1 Resnet 50



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 resnet50 (Functional)       (None, 7, 7, 2048)        23587712

 global_average_pooling2d (  (None, 2048)              0
 GlobalAveragePooling2D)

 dense (Dense)               (None, 256)               524544

 dense_1 (Dense)             (None, 1)                 257

=================================================================
Total params: 24112513 (91.98 MB)
Trainable params: 524801 (2.00 MB)
Non-trainable params: 23587712 (89.98 MB)
_________________________________________________________________
```

Fig 23: TB prognosis Model structure using Resnet-50

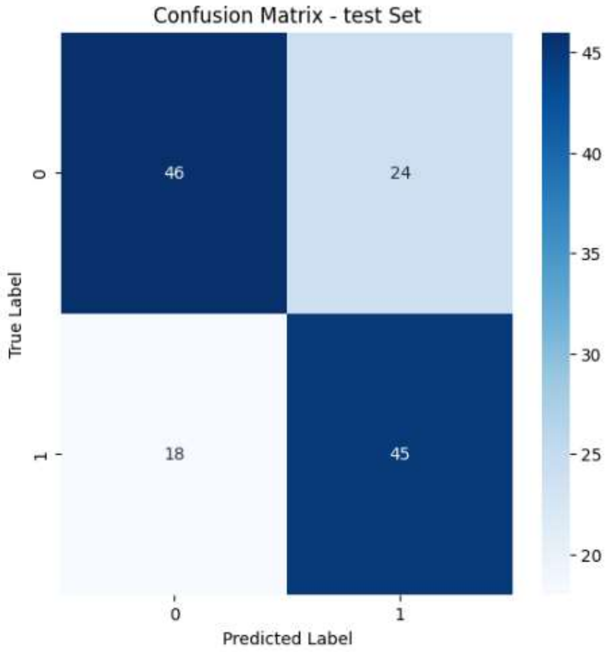
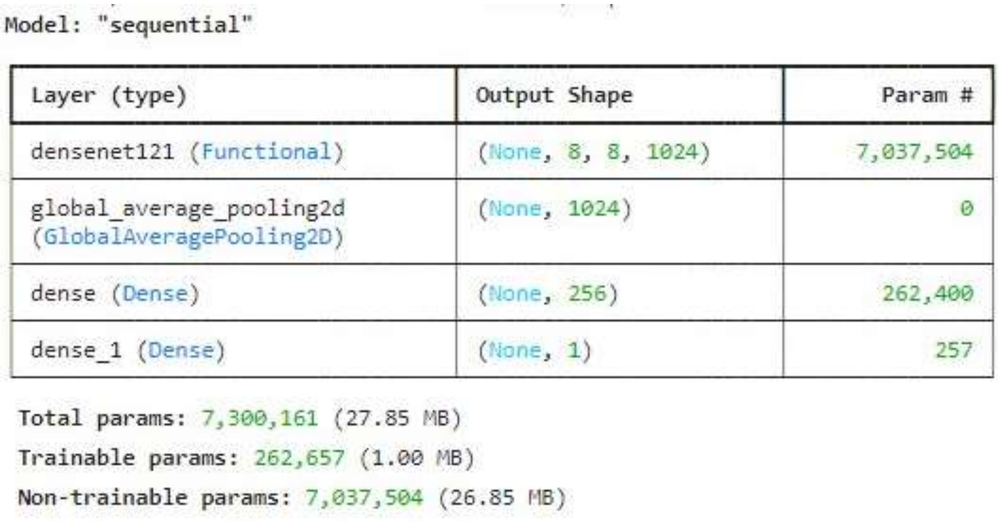


Fig 24: Resultant Confusion Matrix of Resnet-50

4.1.2 DenseNet-121



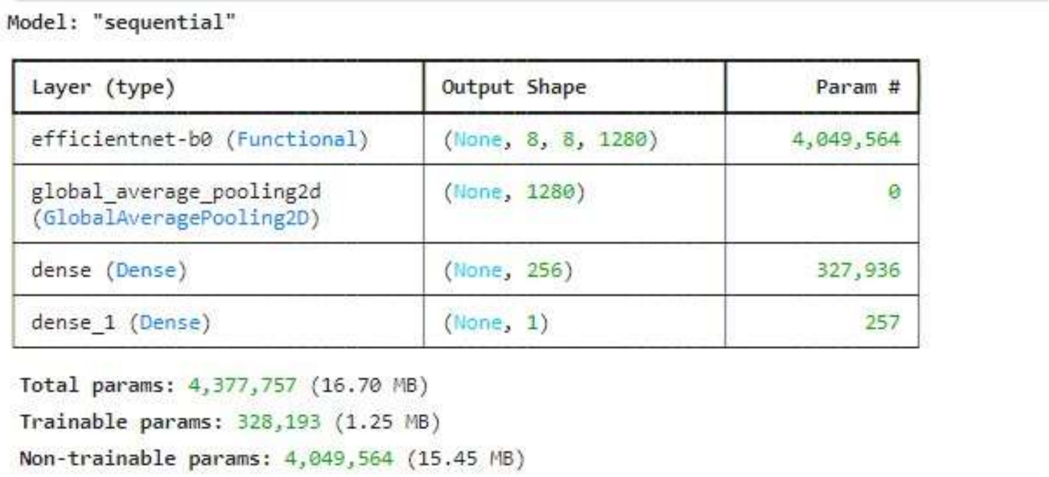
```
Model: "sequential"

┌──────────────────────────────────┬──────────────────────────┬──────────────┐
│ Layer (type)                     │ Output Shape             │      Param # │
├──────────────────────────────────┼──────────────────────────┼──────────────┤
│ densenet121 (Functional)         │ (None, 8, 8, 1024)       │    7,037,504 │
├──────────────────────────────────┼──────────────────────────┼──────────────┤
│ global_average_pooling2d         │ (None, 1024)             │            0 │
│ (GlobalAveragePooling2D)         │                          │              │
├──────────────────────────────────┼──────────────────────────┼──────────────┤
│ dense (Dense)                    │ (None, 256)              │      262,400 │
├──────────────────────────────────┼──────────────────────────┼──────────────┤
│ dense_1 (Dense)                  │ (None, 1)                │          257 │
└──────────────────────────────────┴──────────────────────────┴──────────────┘

 Total params: 7,300,161 (27.85 MB)
 Trainable params: 262,657 (1.00 MB)
 Non-trainable params: 7,037,504 (26.85 MB)
```

Fig 25: TB prognosis Model structure using DenseNet-121

4.1.3 EfficientNetB0



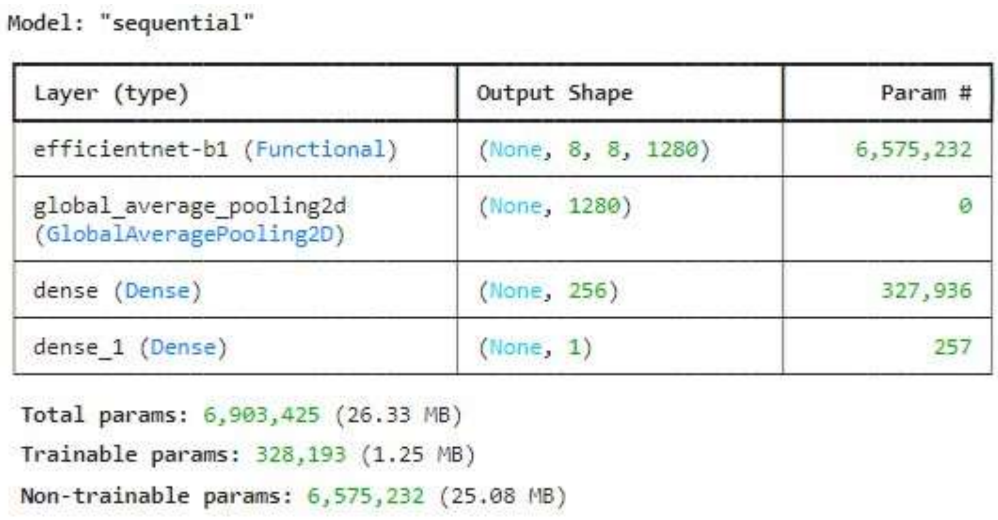
```
Model: "sequential"

┌──────────────────────────────────────┬───────────────────────────┬─────────────┐
│ Layer (type)                         │ Output Shape              │     Param # │
├──────────────────────────────────────┼───────────────────────────┼─────────────┤
│ efficientnet-b0 (Functional)         │ (None, 8, 8, 1280)        │   4,049,564 │
├──────────────────────────────────────┼───────────────────────────┼─────────────┤
│ global_average_pooling2d             │ (None, 1280)              │           0 │
│ (GlobalAveragePooling2D)             │                           │             │
├──────────────────────────────────────┼───────────────────────────┼─────────────┤
│ dense (Dense)                        │ (None, 256)               │     327,936 │
├──────────────────────────────────────┼───────────────────────────┼─────────────┤
│ dense_1 (Dense)                      │ (None, 1)                 │         257 │
└──────────────────────────────────────┴───────────────────────────┴─────────────┘

 Total params: 4,377,757 (16.70 MB)
 Trainable params: 328,193 (1.25 MB)
 Non-trainable params: 4,049,564 (15.45 MB)
```

Fig 26: TB prognosis Model structure using EfficientNetB0

4.1.4 EfficientNetB1



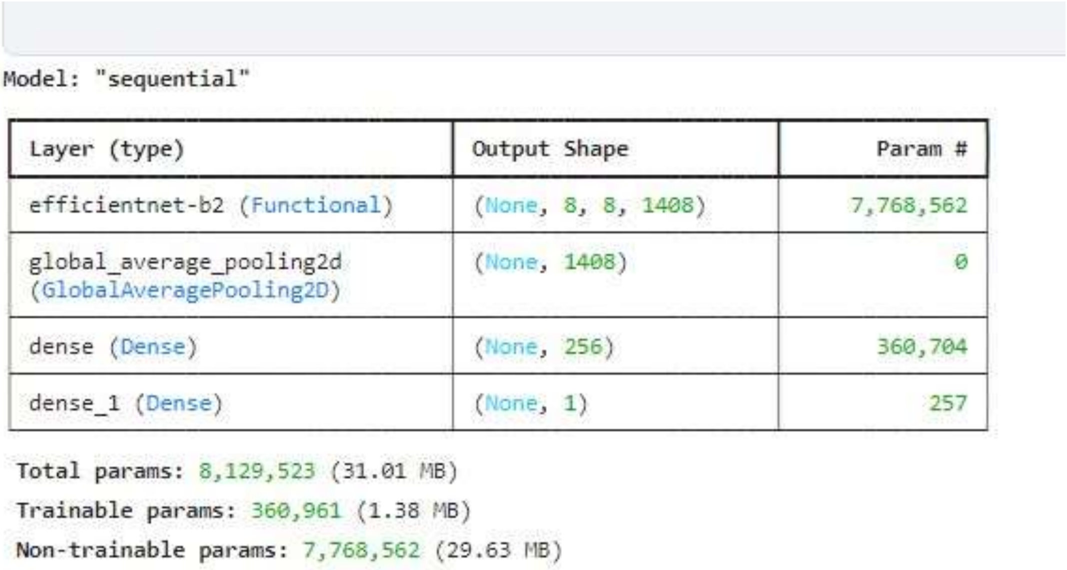
```
Model: "sequential"

┌──────────────────────────────────────┬───────────────────────────┬─────────────┐
│ Layer (type)                         │ Output Shape              │     Param # │
├──────────────────────────────────────┼───────────────────────────┼─────────────┤
│ efficientnet-b1 (Functional)         │ (None, 8, 8, 1280)        │   6,575,232 │
├──────────────────────────────────────┼───────────────────────────┼─────────────┤
│ global_average_pooling2d             │ (None, 1280)              │           0 │
│ (GlobalAveragePooling2D)             │                           │             │
├──────────────────────────────────────┼───────────────────────────┼─────────────┤
│ dense (Dense)                        │ (None, 256)               │     327,936 │
├──────────────────────────────────────┼───────────────────────────┼─────────────┤
│ dense_1 (Dense)                      │ (None, 1)                 │         257 │
└──────────────────────────────────────┴───────────────────────────┴─────────────┘

 Total params: 6,903,425 (26.33 MB)
 Trainable params: 328,193 (1.25 MB)
 Non-trainable params: 6,575,232 (25.08 MB)
```

Fig 27: TB prognosis Model structure using EfficientNetB1

4.1.4 EfficientNetB2



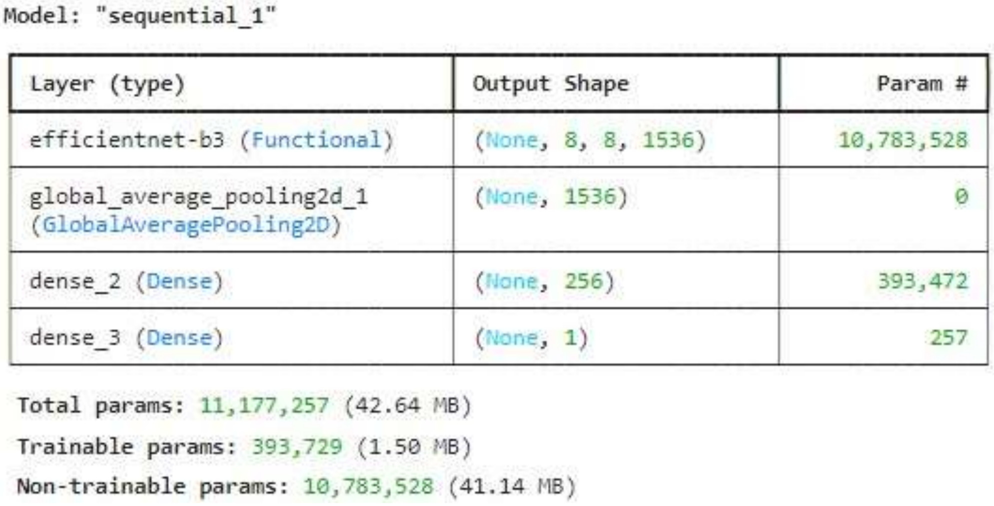
Model: "sequential"

Layer (type)	Output Shape	Param #
efficientnet-b2 (Functional)	(None, 8, 8, 1408)	7,768,562
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1408)	0
dense (Dense)	(None, 256)	360,704
dense_1 (Dense)	(None, 1)	257

```
Total params: 8,129,523 (31.01 MB)
Trainable params: 360,961 (1.38 MB)
Non-trainable params: 7,768,562 (29.63 MB)
```

Fig 28: TB prognosis Model structure using EfficientNetB2

4.1.5 EfficientNetB3



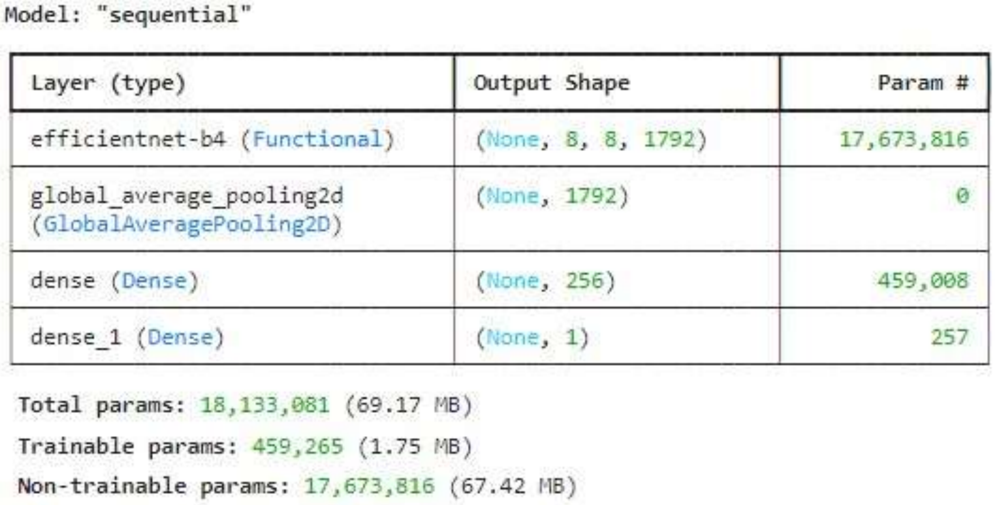
Model: "sequential_1"

Layer (type)	Output Shape	Param #
efficientnet-b3 (Functional)	(None, 8, 8, 1536)	10,783,528
global_average_pooling2d_1 (GlobalAveragePooling2D)	(None, 1536)	0
dense_2 (Dense)	(None, 256)	393,472
dense_3 (Dense)	(None, 1)	257

```
Total params: 11,177,257 (42.64 MB)
Trainable params: 393,729 (1.50 MB)
Non-trainable params: 10,783,528 (41.14 MB)
```

Fig 29: TB prognosis Model structure using EfficientNetB3

4.1.6 EfficientNetB4



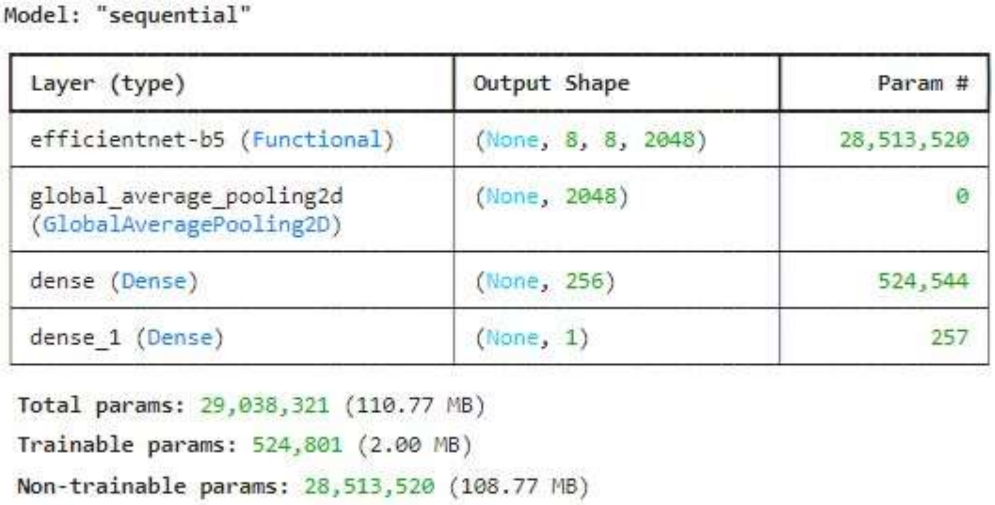
Model: "sequential"

Layer (type)	Output Shape	Param #
efficientnet-b4 (Functional)	(None, 8, 8, 1792)	17,673,816
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1792)	0
dense (Dense)	(None, 256)	459,008
dense_1 (Dense)	(None, 1)	257

```
Total params: 18,133,081 (69.17 MB)
Trainable params: 459,265 (1.75 MB)
Non-trainable params: 17,673,816 (67.42 MB)
```

Fig 30: TB prognosis Model structure using EfficientNetB4

4.1.7 EfficientNetB5



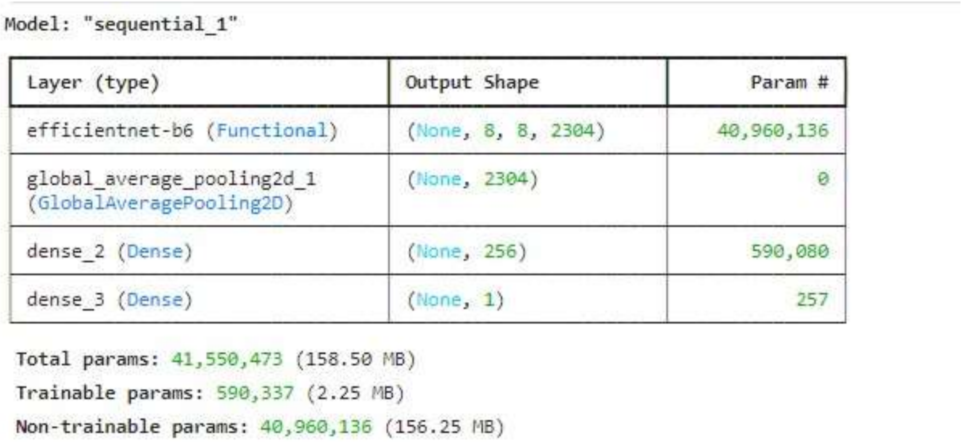
Model: "sequential"

Layer (type)	Output Shape	Param #
efficientnet-b5 (Functional)	(None, 8, 8, 2048)	28,513,520
global_average_pooling2d (GlobalAveragePooling2D)	(None, 2048)	0
dense (Dense)	(None, 256)	524,544
dense_1 (Dense)	(None, 1)	257

```
Total params: 29,038,321 (110.77 MB)
Trainable params: 524,801 (2.00 MB)
Non-trainable params: 28,513,520 (108.77 MB)
```

Fig 31: TB prognosis Model structure using EfficientNetB5

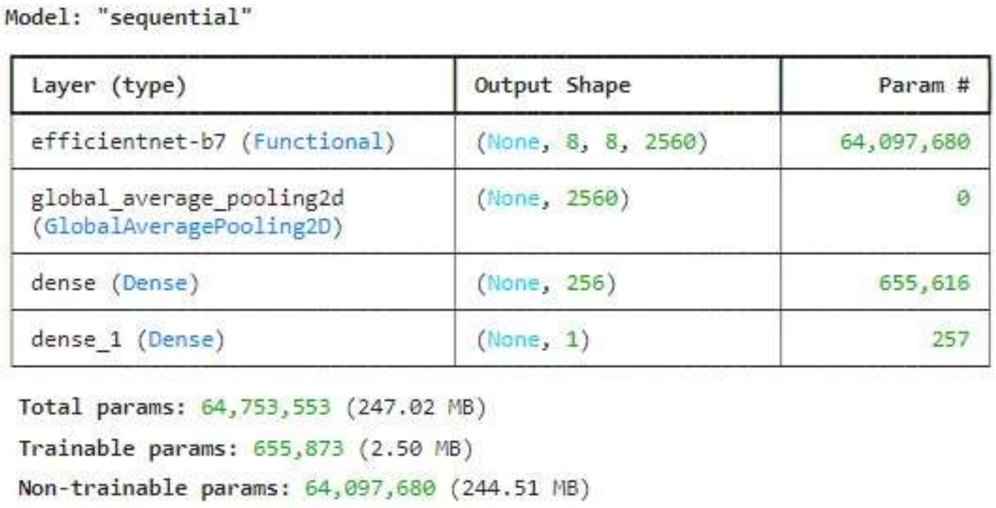
4.1.8 EfficientNetB6



```
Model: "sequential_1"
┌─────────────────────────────────┬────────────────────────┬───────────────┐
│ Layer (type)                    │ Output Shape           │       Param # │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ efficientnet-b6 (Functional)    │ (None, 8, 8, 2304)     │    40,960,136 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ global_average_pooling2d_1      │ (None, 2304)           │             0 │
│ (GlobalAveragePooling2D)        │                        │               │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dense_2 (Dense)                 │ (None, 256)            │       590,080 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dense_3 (Dense)                 │ (None, 1)              │           257 │
└─────────────────────────────────┴────────────────────────┴───────────────┘
 Total params: 41,550,473 (158.50 MB)
 Trainable params: 590,337 (2.25 MB)
 Non-trainable params: 40,960,136 (156.25 MB)
```

Fig 32: TB prognosis Model structure using EfficientNetB6

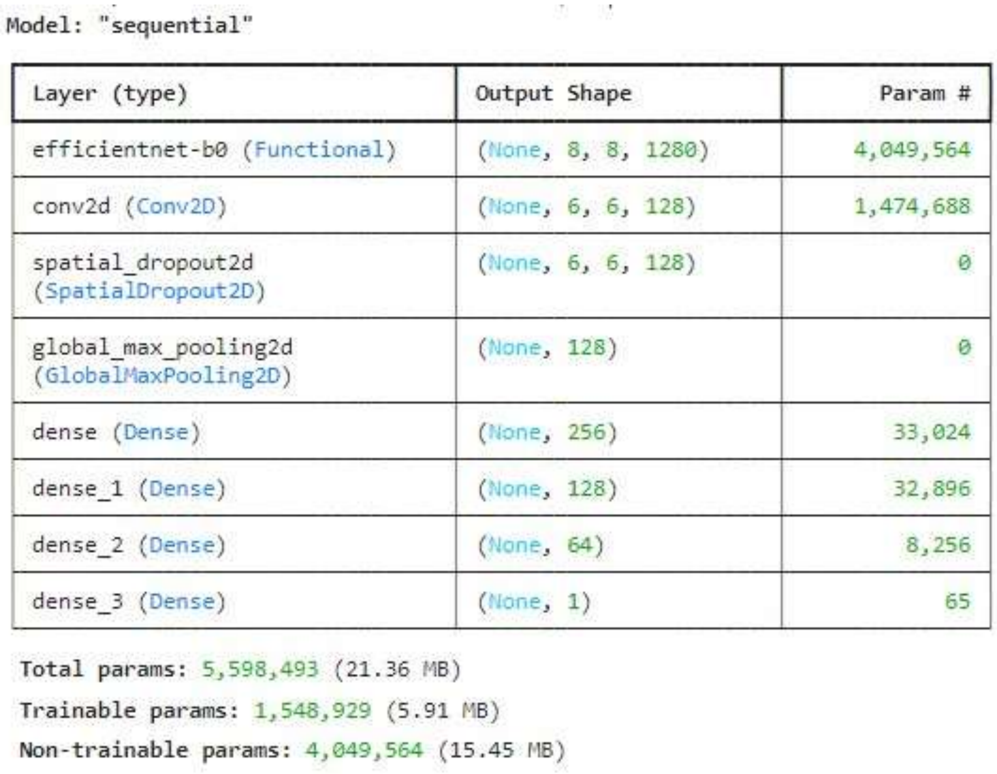
4.1.9 EfficientNetB7



```
Model: "sequential"
┌─────────────────────────────────┬────────────────────────┬───────────────┐
│ Layer (type)                    │ Output Shape           │       Param # │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ efficientnet-b7 (Functional)    │ (None, 8, 8, 2560)     │    64,097,680 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ global_average_pooling2d        │ (None, 2560)           │             0 │
│ (GlobalAveragePooling2D)        │                        │               │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dense (Dense)                   │ (None, 256)            │       655,616 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dense_1 (Dense)                 │ (None, 1)              │           257 │
└─────────────────────────────────┴────────────────────────┴───────────────┘
 Total params: 64,753,553 (247.02 MB)
 Trainable params: 655,873 (2.50 MB)
 Non-trainable params: 64,097,680 (244.51 MB)
```

Fig 33: TB prognosis Model structure using EfficientNetB7

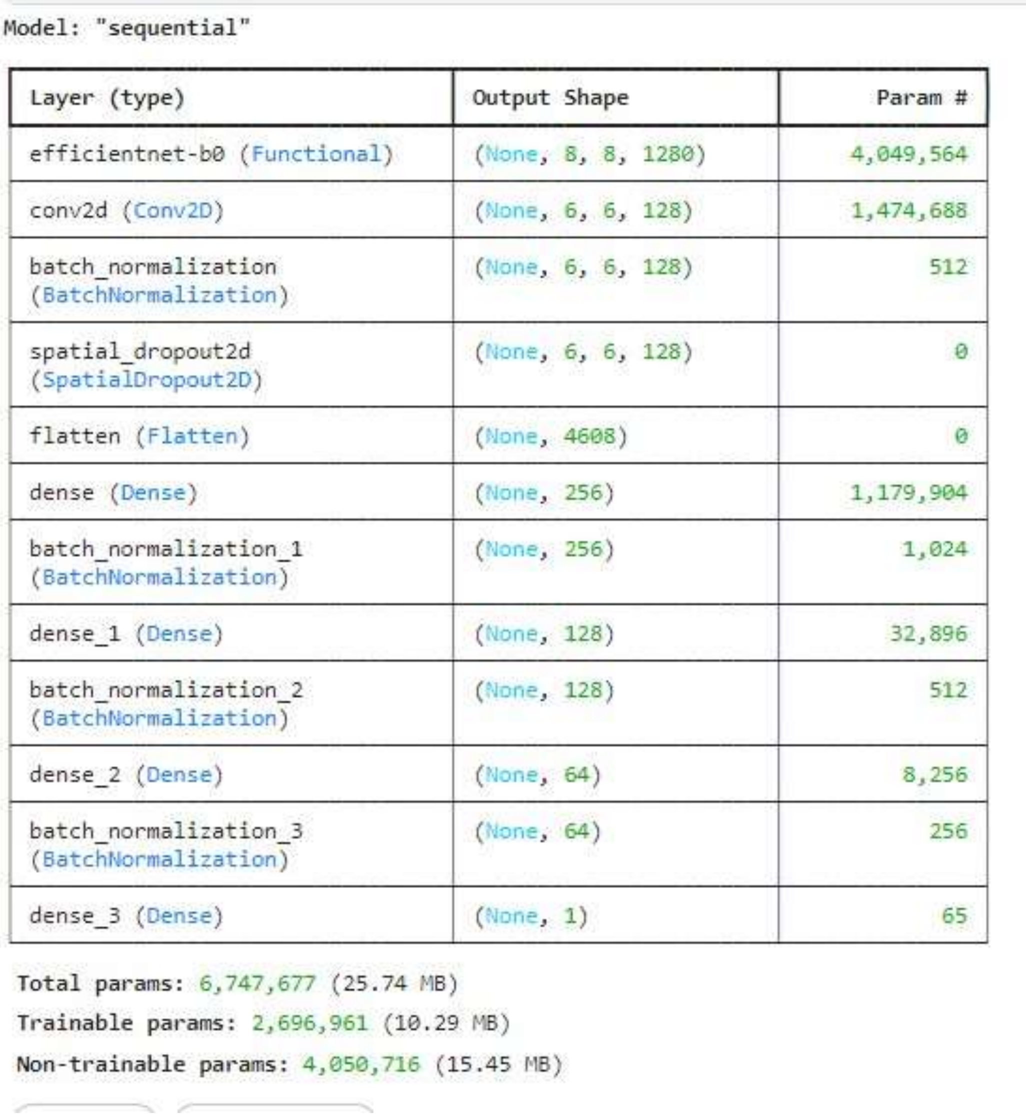
4.1.10 EfficientNetB0 With SpatialDropout



```
Model: "sequential"
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━┓
┃ Layer (type)                     ┃ Output Shape           ┃       Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━┩
│ efficientnet-b0 (Functional)     │ (None, 8, 8, 1280)     │     4,049,564 │
├──────────────────────────────────┼────────────────────────┼───────────────┤
│ conv2d (Conv2D)                  │ (None, 6, 6, 128)      │     1,474,688 │
├──────────────────────────────────┼────────────────────────┼───────────────┤
│ spatial_dropout2d                │ (None, 6, 6, 128)      │             0 │
│ (SpatialDropout2D)               │                        │               │
├──────────────────────────────────┼────────────────────────┼───────────────┤
│ global_max_pooling2d             │ (None, 128)            │             0 │
│ (GlobalMaxPooling2D)             │                        │               │
├──────────────────────────────────┼────────────────────────┼───────────────┤
│ dense (Dense)                    │ (None, 256)            │        33,024 │
├──────────────────────────────────┼────────────────────────┼───────────────┤
│ dense_1 (Dense)                  │ (None, 128)            │        32,896 │
├──────────────────────────────────┼────────────────────────┼───────────────┤
│ dense_2 (Dense)                  │ (None, 64)             │         8,256 │
├──────────────────────────────────┼────────────────────────┼───────────────┤
│ dense_3 (Dense)                  │ (None, 1)              │            65 │
└──────────────────────────────────┴────────────────────────┴───────────────┘
 Total params: 5,598,493 (21.36 MB)
 Trainable params: 1,548,929 (5.91 MB)
 Non-trainable params: 4,049,564 (15.45 MB)
```

Fig 34: TB prognosis Model structure using EfficientNetB0 with SpatialDropout

4.1.11 EfficientNetB0 With BatchNormalization



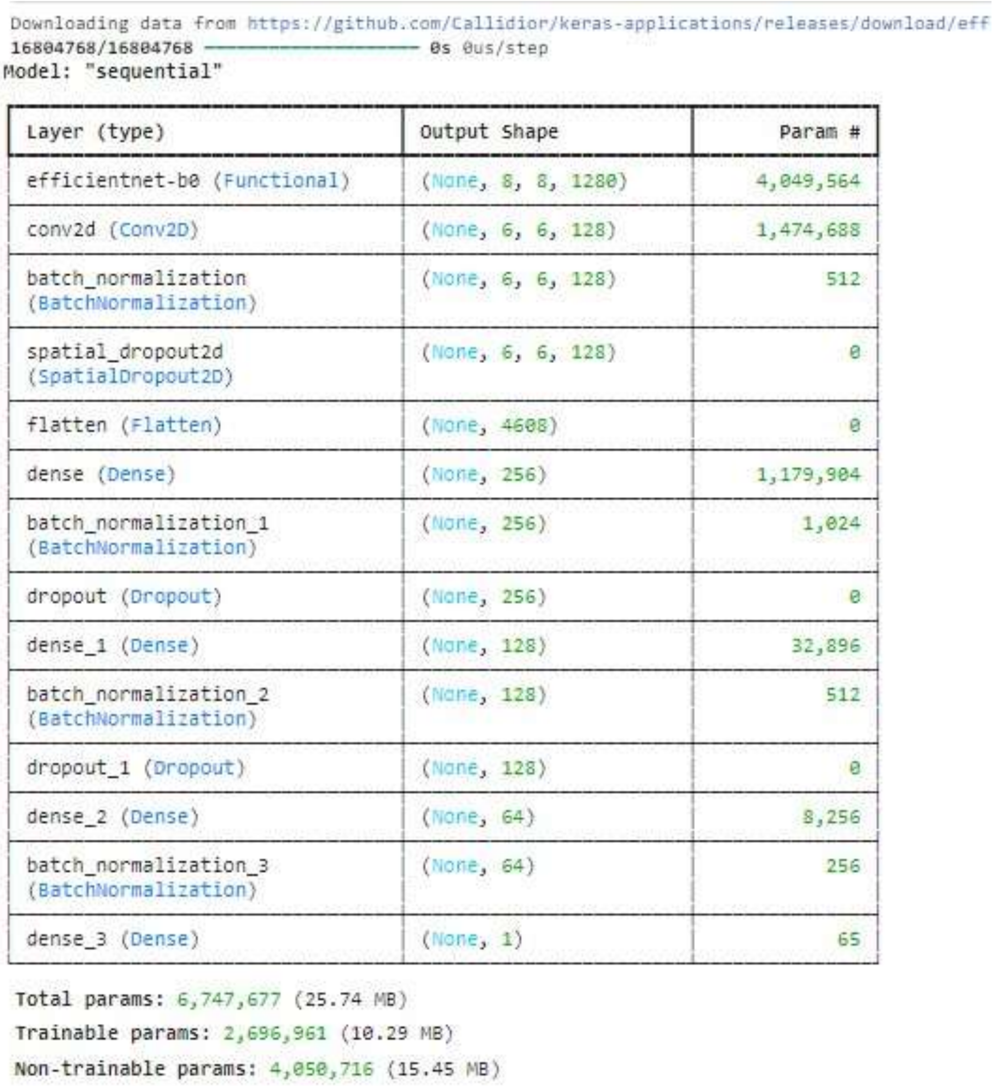
Model: "sequential"

Layer (type)	Output Shape	Param #
efficientnet-b0 (Functional)	(None, 8, 8, 1280)	4,049,564
conv2d (Conv2D)	(None, 6, 6, 128)	1,474,688
batch_normalization (BatchNormalization)	(None, 6, 6, 128)	512
spatial_dropout2d (SpatialDropout2D)	(None, 6, 6, 128)	0
flatten (Flatten)	(None, 4608)	0
dense (Dense)	(None, 256)	1,179,904
batch_normalization_1 (BatchNormalization)	(None, 256)	1,024
dense_1 (Dense)	(None, 128)	32,896
batch_normalization_2 (BatchNormalization)	(None, 128)	512
dense_2 (Dense)	(None, 64)	8,256
batch_normalization_3 (BatchNormalization)	(None, 64)	256
dense_3 (Dense)	(None, 1)	65

```
Total params: 6,747,677 (25.74 MB)
Trainable params: 2,696,961 (10.29 MB)
Non-trainable params: 4,050,716 (15.45 MB)
```

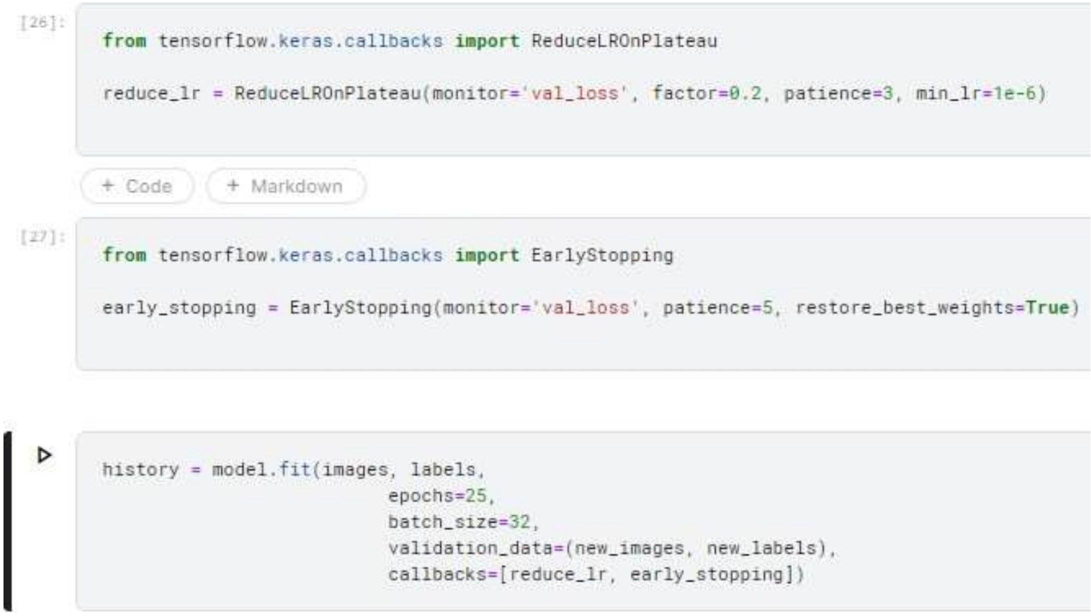
Fig 35: TB prognosis Model structure using EfficientNetB0 with Batch-Normalization

4.1.12 EfficientNetB0 with DropOut, BatchNorm, EarlyStopping and Variable LearningRate



```
Downloading data from https://github.com/Callidior/keras-applications/releases/download/eff
16804768/16804768 ━━━━━━━━━━━━━━━━━━━━ 0s 0us/step
Model: "sequential"
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━┓
┃ Layer (type)                     ┃ Output Shape           ┃       Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━┩
│ efficientnet-b0 (Functional)     │ (None, 8, 8, 1280)     │     4,049,564 │
├──────────────────────────────────┼────────────────────────┼───────────────┤
│ conv2d (Conv2D)                  │ (None, 6, 6, 128)      │     1,474,688 │
├──────────────────────────────────┼────────────────────────┼───────────────┤
│ batch_normalization              │ (None, 6, 6, 128)      │           512 │
│ (BatchNormalization)             │                        │               │
├──────────────────────────────────┼────────────────────────┼───────────────┤
│ spatial_dropout2d                │ (None, 6, 6, 128)      │             0 │
│ (SpatialDropout2D)               │                        │               │
├──────────────────────────────────┼────────────────────────┼───────────────┤
│ flatten (Flatten)                │ (None, 4608)           │             0 │
├──────────────────────────────────┼────────────────────────┼───────────────┤
│ dense (Dense)                    │ (None, 256)            │     1,179,904 │
├──────────────────────────────────┼────────────────────────┼───────────────┤
│ batch_normalization_1            │ (None, 256)            │         1,024 │
│ (BatchNormalization)             │                        │               │
├──────────────────────────────────┼────────────────────────┼───────────────┤
│ dropout (Dropout)                │ (None, 256)            │             0 │
├──────────────────────────────────┼────────────────────────┼───────────────┤
│ dense_1 (Dense)                  │ (None, 128)            │        32,896 │
├──────────────────────────────────┼────────────────────────┼───────────────┤
│ batch_normalization_2            │ (None, 128)            │           512 │
│ (BatchNormalization)             │                        │               │
├──────────────────────────────────┼────────────────────────┼───────────────┤
│ dropout_1 (Dropout)              │ (None, 128)            │             0 │
├──────────────────────────────────┼────────────────────────┼───────────────┤
│ dense_2 (Dense)                  │ (None, 64)             │         8,256 │
├──────────────────────────────────┼────────────────────────┼───────────────┤
│ batch_normalization_3            │ (None, 64)             │           256 │
│ (BatchNormalization)             │                        │               │
├──────────────────────────────────┼────────────────────────┼───────────────┤
│ dense_3 (Dense)                  │ (None, 1)              │            65 │
└──────────────────────────────────┴────────────────────────┴───────────────┘
 Total params: 6,747,677 (25.74 MB)
 Trainable params: 2,696,961 (10.29 MB)
 Non-trainable params: 4,050,716 (15.45 MB)
```

Fig 36: TB prognosis Model structure using EfficientNetB0 with Batch-Normalization and DropOut



```
from tensorflow.keras.callbacks import ReduceLROnPlateau

reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=3, min_lr=1e-6)
```

```
from tensorflow.keras.callbacks import EarlyStopping

early_stopping = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)
```

```
history = model.fit(images, labels,
                          epochs=25,
                          batch_size=32,
                          validation_data=(new_images, new_labels),
                          callbacks=[reduce_lr, early_stopping])
```

Fig 37: TB prognosis Model layer using EfficientNetB0 with Variable Learning Rate and Early-Stopping

4.2 RESULTS

Table 2: Model Comparison Based on Accuracy and Loss

Model Name	Epochs	Train Accuracy	Train Loss	Test Accuracy	Test Loss
DenseNet121	10	0.9737	0.0767	0.963	0.0992
EfficientNetB0	10	0.9744	0.0684	0.967	0.0932
EfficientNetB1	10	0.9627	0.0806	0.953	0.1077
EfficientNetB2	10	0.9677	0.0856	0.976	0.0693
EfficientNetB3	15	0.9607	0.091	0.961	0.0857
EfficientNetB4	10	0.9538	0.1151	0.961	0.0998
EfficientNetB5	10	0.9511	0.1217	0.959	0.1062
EfficientNetB6	10	0.9496	0.1369	0.948	0.1266
EfficientNetB7	10	0.9416	0.1556	0.942	0.1297
EfficientNetB7	20	0.9416	0.1455	0.961	0.1003
ConvNextTiny	10	0.7859	0.5015	0.78	0.4925
DenseNet121_HistEqual	10	0.968	0.0825	0.954	0.1162
EfficientNetB0_HistEqual	10	0.9675	0.0715	0.972	0.0713
EfficientNetB2_HstEqual	10	0.9657	0.0844	0.963	0.098
EfficientNetB0_SpatialDropout	25	0.9887	0.0320	0.971	0.1046
EfficientNetB0_SD_BatchNorm	25	0.9894	0.0308	0.972	0.0987
Final_Model_Variable_lr_EarlyStopping	16	0.9826	0.0440	0.982	0.0622

Table 3: Model Comparison Based on Evaluation Metrics

Model Name	Precision (1)	Recall (1)	F1-score (1)	Precision (0)	Recall (0)	F1-score (0)
DenseNet121	0.9	0.94	0.92	0.98	0.97	0.98
EfficientNetB0	0.91	0.95	0.93	0.98	0.97	0.98
EfficientNetB1	0.85	0.95	0.9	0.99	0.95	0.97
EfficientNetB2	0.95	0.94	0.94	0.98	0.99	0.99
EfficientNetB3	0.9	0.93	0.91	0.98	0.97	0.97
EfficientNetB4	0.97	0.85	0.91	0.96	0.99	0.98
EfficientNetB5	0.9	0.91	0.91	0.97	0.97	0.97
EfficientNetB6	0.87	0.90	0.88	0.97	0.96	0.97
EfficientNetB7	0.85	0.89	0.87	0.97	0.96	0.96
EfficientNetB7	0.91	0.92	0.91	0.98	0.97	0.97
ConvNextTiny	0.4	0.01	0.02	0.78	1	0.88
DenseNet121_HistEqual	0.89	0.9	0.9	0.97	0.97	0.97
EfficientNetB0_HistEqual	0.96	0.91	0.93	0.97	0.99	0.98
EfficientNetB2_HstEqual	0.93	0.89	0.91	0.97	0.98	0.98
EfficientNetB0_SpatialDropout	0.94	0.92	0.93	0.98	0.98	0.98
EfficientNetB0_SD_BatchNorm	0.92	0.96	0.95	0.99	0.98	0.98
Final_Model_Variable_lr_EarlyStopping	0.96	0.95	0.95	0.98	0.99	0.99

Table 4: Model Comparison Based on Confusion Metrics

Model Name	TN	TP	FN	FP
DenseNet121	757	206	13	24
EfficientNetB0	760	207	12	21
EfficientNetB1	744	209	10	37
EfficientNetB2	771	205	14	10
EfficientNetB3	758	203	16	23
EfficientNetB4	775	186	33	6
EfficientNetB5	760	199	20	21
EfficientNetB6	751	197	22	30
EfficientNetB7	748	194	25	33
EfficientNetB7	760	201	18	21
ConvNextTiny	778	2	217	3
DenseNet121_HistEqual	757	197	22	24
EfficientNetB0_HistEqual	773	199	20	8
EfficientNetB2_HstEqual	767	196	23	14
EfficientNetB0_SpatialDropout	769	202	17	12
EfficientNetB0_SD_BatchNorm	762	210	9	19
Final_Model_Variable_lr_EarlyStopping	773	207	12	8

Table 2 , Table 3 and Table 4 represent model comparison of ConvNet class with their results on various factors showcasing the behavior of different models in context to chest X-rays of pulmonary tuberculosis images of the combined dataset of Shenzen, Montgomery, Qatar and Bangladesh using transfer learning.

- Models like DenseNet-121, EfficientNetB0-B7 and their variants show good results but some later versions of EfficientNet family show some decline in the performance of complex models i.e. larger number of Non-trainable parameters where whereas EfficientNetB0-B2 work very well.

- Models like ConvNextTiny and ResNet50 don't work i.e. learning is very minute on base models because of architectural differences like:-
  - o Residual Connections: They heavily rely on residual connections to solve vanishing gradient problems. As our classification is not very dependable on it and more lean towards very fine features, therefore the benefits of residual connection are less highlighted.
  - o Depth-wise Separable Convolutions: ConvNext model utilizes it, so they might be not effective on tuberculosis prognosis using chest x-rays
- There are significant changes in results when spatial dropout, dropout and Batch Normalization layers are added with conv2d and dense layers on best-performing model using pre-processed images.
- Final Model with Variable Learning Rate and Early Stopping is added on a modified model of EfficientNetB0. We got the best results with a test accuracy of 98% and the overall performance is shown in Fig. 38 and 39.

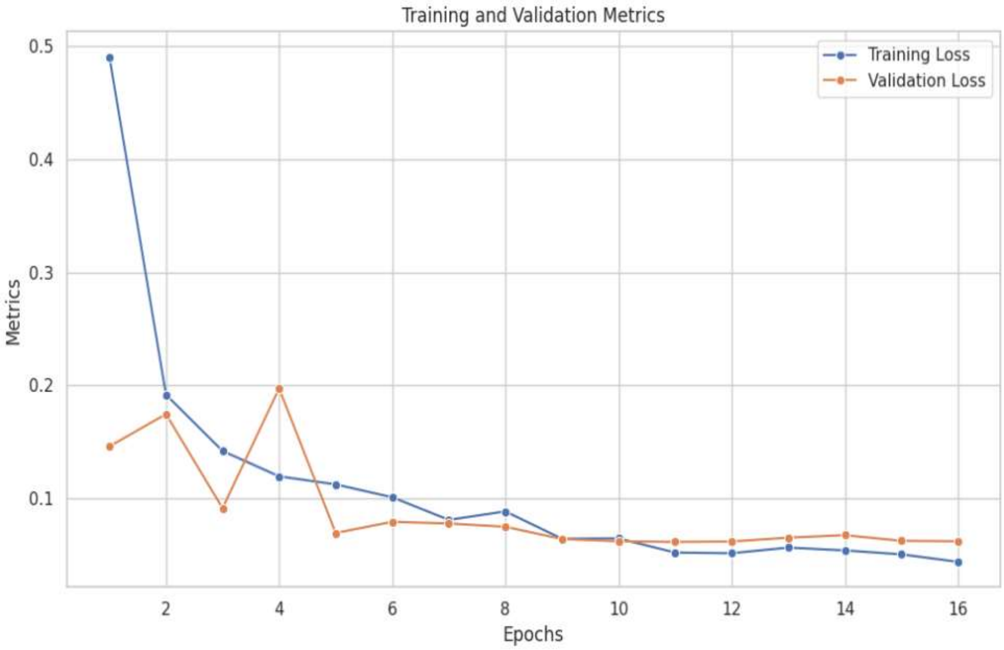


Fig 38: Train Loss Vs Validation Loss of Final Proposed Model

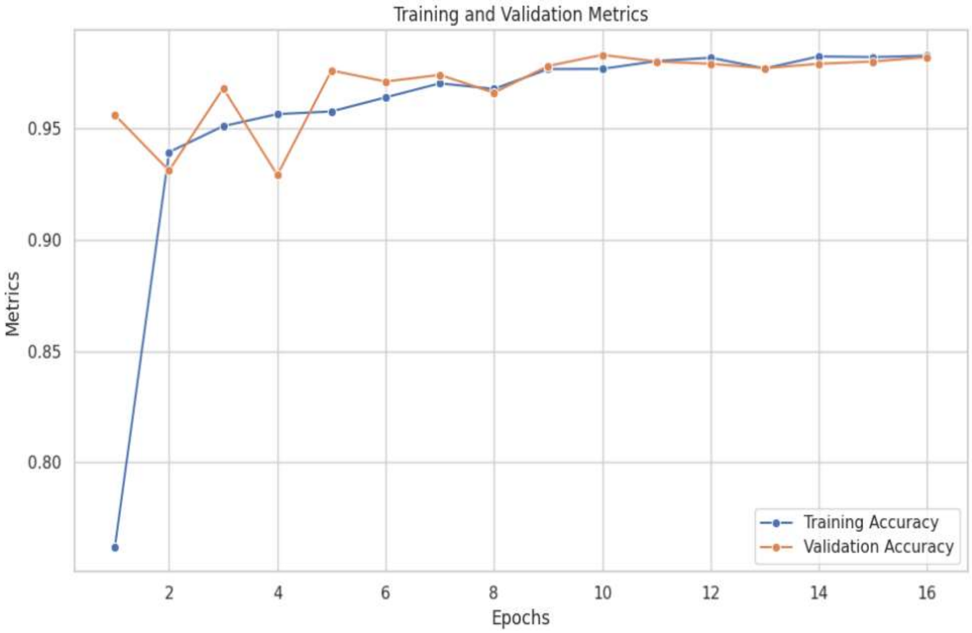


Fig 39 Train Accuracy Vs Validation Accuracy of Final Proposed Model

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 CONCLUSION

The application of Neural Networks Uses Convolutions that is CNNs in clinical image analysis, particularly for tuberculosis prognosis using chest X-rays (CXR), presents a formidable hardship due to the limited availability of clinical images and the sensitivity of predictions in medical contexts. While CNNs excel in image-related tasks, the scarcity of public medical image datasets necessitates extensive practical research to assess model performance accurately.

It's evident that the effectiveness of CNNs is contingent upon the availability of the clinical data, with larger datasets contributing to exceptional performance. However, the intricate nature of medical images, especially in disease diagnosis, underscores the importance of meticulous model selection and hyperparameter tuning. Larger models generally yield more accurate results, but striking a balance is crucial to avoid over-fitting. The implementation of batch training, alongside techniques like hyper-parameter tuning and the utilization of callbacks such as early stopping and variable learning rates, becomes imperative for optimizing model performance. Evaluation metrics like accuracy, precision, recall, F1-score, and confusion matrix provide a detailed overview of the work efficacy on the available tuberculosis prognosis dataset.

5.2 FUTURE WORK

Looking ahead, the future of tuberculosis prognosis through AI techniques is promising.

- The advent of transformer models for images, particularly Vision Transformers and family DeepVit [17], CrossVit [15], holds great potential for image classification in medical diagnostics.
- The ongoing research on transformers, coupled with advanced segmentation techniques, is anticipated to enhance the clarity of disease-affected areas in medical images, thereby improving the accuracy of tuberculosis diagnosis.
- Graph neural networks can be the new possible breakthrough in disease classification and area of research.

In essence, the convergence of AI and medical imaging is paving the way for automated diagnosis, and the evolution of transformer models is poised to play a pivotal role in revolutionizing tuberculosis prognosis through chest X-ray analysis. As research progresses, we can expect increasingly sophisticated models that contribute to more accurate and efficient medical image analysis, ultimately benefiting both clinicians and patients in the quest for improved healthcare outcomes.

REFERENCES

[1] O. Patel, Y. P. S. Maravi, and S. Sharma. "*A Comparative Study of Histogram Equalization Based Image Enhancement Techniques for Brightness Preservation and Contrast Enhancement*". In: Signal Image Process 4.5 (Nov. 2013), pp. 11–25. doi: 10.5121/sipij.2013.4502.

[2] L. M. Pinto et al. "*Scoring systems using chest radiographic features for the diagnosis of pulmonary tuberculosis in adults: a systematic review*". In: European Respiratory Journal 42.2 (Aug. 2013), pp. 480–494. doi: 10 . 1183 / 09031936 .00107412.

[3] S. Jaeger et al. "*Two public chest X-ray datasets for computer-aided screening of pulmonary diseases*". In: Quant Imaging Med Surg 4.6 (Dec. 2014), pp. 475–477. doi: 10.3978/j.issn.2223-4292.2014.11.20.

[4] R. Piccazzo, F. Paparo, and G. Garlaschi. "*Diagnostic Accuracy of Chest Radio-graphy for the Diagnosis of Tuberculosis (TB) and Its Role in the Detection ofLatent TB Infection: a Systematic Review*". In: J Rheumatol Suppl 91.0 (May2014), pp. 32–40. doi: 10.3899/jrheum.140100.

[5] J. B. Bomanji et al. "*Imaging in Tuberculosis*". In: Cold Spring Harb Perspect Med 5.6 (June 2015), a017814–a017814. doi: 10.1101/cshperspect.a017814.

[6] N. Dhanachandra, K. Manglem, and Y. J. Chanu. "*Image Segmentation Using K-means Clustering Algorithm and Subtractive Clustering Algorithm*". In:(2015), pp. 764–771. doi: 10.1016/j.procs.2015.06.090.

[7] K. He et al. "*Deep Residual Learning for Image Recognition*". In: (Dec. 2015).

[8] G. Huang et al. "*Densely Connected Convolutional Networks*". In: (Aug. 2016).

[9] P. Lakhani and B. Sundaram. "*Deep Learning at Chest Radiography: Automated Classification of Pulmonary Tuberculosis by Using Convolutional Neural Networks*". In: Radiology 284.2 (Aug. 2017), pp. 574–582. doi: 10.1148/radiol.2017162326.

[10] Vaswani and et al. "*Attention Is All You Need*". In: (June 2017).

[11] T. K. Kim and et al. "*Deep Learning Method for Automated Classification of Anteroposterior and Posteroanterior Chest Radiographs*". In: J Digit Imaging 32.6 (Dec. 2019), pp. 925–930. doi: 10.1007/s10278-019-00208-0.

[12] R. M. Schmidt. Recurrent Neural Networks (RNNs): A gentle Introduction and Overview. http://arxiv.org/abs/1912.05911. Nov. 2019.

[13] Dosovitskiy and et al. "*An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*". In: (Oct. 2020).

[14] Mingxing Tan and Quoc V. Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. 2020. arXiv: 1905.11946 [cs.LG].

[15] C.-F. Chen, Q. Fan, and R. Panda. "*CrossViT: Cross-Attention Multi-Scale Vision Transformer for Image Classification*". In: (Mar. 2021).

[16] S. Rajaraman et al. "*Improved Semantic Segmentation of Tuberculosis—Consistent Findings in Chest X-rays Using Augmented Training of Modality-Specific U-Net Models with Weak Localizations*". In: Diagnostics 11.4 (Mar. 2021), p. 616. doi: 10.3390/diagnostics11040616.

[17] D. Zhou and et al. "*DeepViT: Towards Deeper Vision Transformer*". In: (Mar.2021).

[18] J. R. Brown et al. "*Wide Attention Is The Way Forward For Transformers*?" In:(Oct. 2022).

[19] Zhuang Liu et al. A ConvNet for the 2020s. 2022. arXiv: 2201.03545 [cs.CV].

[20] Gabriel Iluebe Okolo, Stamos Katsigiannis, and Naeem Ramzan. "*IEViT: An enhanced vision transformer architecture for chest X-ray image classification*". In: Computer Methods and Programs in Biomedicine 226 (2022), p. 107141.

[21] Sangjoon Park et al. "*Self-evolving vision transformer for chest X-ray diagnosis through knowledge distillation*". In: Nature communications 13.1 (2022), p. 3848.

[22] K. Santosh et al. "*Advances in Deep Learning for Tuberculosis Screening using Chest X-rays: The Last 5 Years Review*". In: J Med Syst 46.11 (Nov. 2022). doi:10.1007/s10916-022-01870-8.

[23] Mohammad Usman, Tehseen Zia, and Ali Tariq. "*Analyzing transfer learning of vision transformers for interpreting chest radiography*". In: Journal of digital imaging 35.6 (2022), pp. 1445–1462.

[24] G. Varoquaux and V. Cheplygina. "*Machine learning for medical imaging: methodological failures and recommendations for the future*". In: NPJ Digit Med 5.1 (Apr. 2022), p. 48. doi: 10.1038/s41746-022-00592-y.

[25] J. Wensel, H. Ullah, and A. Munir. "*ViT-ReT: Vision and Recurrent Transformer Neural Networks for Human Activity Recognition in Videos*". In: (Aug. 2022).

[26] Feng Yang et al. "*Differentiating between drug-sensitive and drug-resistant tuberculosis with machine learning for clinical and radiological features*". In: Quantitative Imaging in Medicine and Surgery 12.1 (2022), p. 675.

[27] Goram Mufarah M Alshmrani et al. "*A deep learning architecture for multiclass lung diseases classification using chest X-ray (CXR) images*". In: Alexandria Engineering Journal 64 (2023), pp. 923–935.

[28] Padmaja B et al. "*Chest X-Ray Image Analysis for Respiratory Disease Prediction using Grad-CAM*". In: 2023 2nd Edition of IEEE Delhi Section Flagship Conference (DELCON). 2023, pp. 1–7. doi: 10 . 1109 / DELCON57910 . 2023 .10127464.

[29] Jos´e Escorcia-Gutierrez et al. "*Computer-aided diagnosis for tuberculosis classification with water strider optimization algorithm*". In: Computer Systems Science and Engineering 46.2 (2023), pp. 1337–1353.

[30] Mohammad Raihan Goni, Ayokunle Olalekan Ige, and Nur Intan Raihana Ruhaiyem. "*TL-AttSharpNet: Automated Lung Image Segmentation using Transfer Learning with Depthwise Convolution and Attention*". In: 2023 IEEE2nd National Biomedical Engineering Conference (NBEC). IEEE. 2023, pp. 133–137.

[31] S. Hansun et al. "*Machine and Deep Learning for Tuberculosis Detection on Chest X-Rays: Systematic Literature Review*". In: Journal of Medical Internet Research 25 (2023). doi: 10.2196/43154.

[32] Vo Trong Quang Huy and Chih-Min Lin. "*An improved densenet deep neural network model for tuberculosis detection using chest X-Ray images*". In: IEEEAccess (2023).

[33] Raza Imam et al. "*SEDA: Self-ensembling ViT with Defensive Distillation andAdversarial Training for Robust Chest X-Rays Classification*". In: MICCAIWorkshop on Domain Adaptation and Representation Transfer. Springer. 2023,pp. 126–135.

[34] Ahmed Iqbal, Muhammad Usman, and Zohair Ahmed. "*Tuberculosis chest X-ray detection using CNN-based hybrid segmentation and classification approach*". In: Biomedical Signal Processing and Control 84 (2023), p. 104667.

[35] Ahmed Iqbal, Muhammad Usman, and Zohair Ahmed. "*Tuberculosis chest X-ray detection using CNN-based hybrid segmentation and classification approach*". In: Biomedical Signal Processing and Control 84 (2023), p. 104667.

[36] Vamsi Kakani et al. "*Post-COVID Chest Disease Monitoring using self adaptive Convolutional Neural Network*". In: 2023 IEEE 8th International Conference for Convergence in Technology (I2CT). 2023, pp. 1–6. doi: 10.1109/I2CT57861.2023.10126288.

[37] Sahebgoud Hanamantray Karaddi and Lakhan Dev Sharma. "*Automated multi-class classification of lung diseases from CXR-images using pre-trained convolutional neural networks*". In: Expert Systems with Applications 211 (2023),p. 118650.

[38] E. Kotei and R. Thirunavukarasu. "*A Comprehensive Review on Advancement in Deep Learning Techniques for Automatic Detection of Tuberculosis from Chest X-ray Images*". In: Archives of Computational Methods in Engineering (2023). doi: 10.1007/s11831-023-09987-w.

[39] T. Lei et al. "*CiT-Net: Convolutional Neural Networks Hand in Hand with Vision Transformers for Medical Image Segmentation*". In: (June 2023).

[40] Chia-Jung Liu et al. “*A deep learning model using chest X-ray for identifying TB and NTM-LD patients: a cross-sectional study*”. In: Insights into Imaging 14.1 (2023), p. 67.

[41] Yun Liu et al. “Revisiting *Computer-Aided Tuberculosis Diagnosis*”. In: IEEE Transactions on Pattern Analysis and Machine Intelligence (2023).

[42] Rajesh T M and Anvik Kumar Achar. “*OrthoSNet: A framework for Identifying and Verifying Tuberculosis in Chest Radio graph images using Discrete Orthonormal Stockwell transform*”. In: 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT). 2023, pp. 1–9. doi: 10.1109/ICCCNT56998.2023.10308237.

[43] Hassaan Malik et al. “*A novel fusion model of hand-crafted features with deep convolutional neural networks for classification of several chest diseases using x-ray images*”. In: IEEE Access (2023).

[44] K Manivannan and S Sathiamoorthy. “*Pelican Optimization with Majority Voting Ensemble Model for Tuberculosis Detection and Classification on ChestX-Ray Images*.” In: International Journal of Intelligent Engineering & Systems16.5 (2023).

[45] K. Manivannan and S. Sathiamoorthy. “*Robust Tuberculosis Detection using Optimal Deep Learning Model using Chest X-Rays*”. In: 2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC). 2023, pp. 259–264. doi: 10.1109/ICAAIC56838.2023.10140661.

[46] Z. Mustafa and H. Nsour. “*Using Computer Vision Techniques to Automatically Detect Abnormalities in Chest X-rays*”. In: Diagnostics 13.18 (Sept. 2023). doi:10.3390/diagnostics13182979.

[47] Minwoo Park et al. “*Distinguishing nontuberculous mycobacterial lung disease and Mycobacterium tuberculosis lung disease on X-ray images using deep transfer learning*”. In: BMC Infectious Diseases 23.1 (2023), p. 32.

[48] Tamarisk du Plessis et al. “*Introducing a secondary segmentation to construct a radiomics model for pulmonary tuberculosis cavities*”. In: La radiologia medica 128.9 (2023), pp. 1093–1102.

[49] A Rama et al. “*Detection of TB from Chest X-ray: A Study with EfficientNet*”. In: 2023 International Conference on System, Computation, Automation and Networking (ICSCAN). IEEE. 2023, pp. 1–4.

[50] Aneesha Sharma, Anshika Yadav, and Manoj Kumar. “*Detection of Lung Disorders from X-Ray Images Using Hybrid Deep Learning*”. In: 2023 3rd Asian Conference on Innovation in Technology (ASIANCON). IEEE. 2023, pp. 1–8.

[51] Chukwuebuka Joseph Ejiyi et al. “*ResfEANet: ResNet-fused external attention network for tuberculosis diagnosis using chest X-ray images*”. In: Computer Methods and Programs in Biomedicine Update 5 (2024), p. 100133.

[52] B Uma Maheswari et al. “*Explainable deep-neural-network supported scheme for tuberculosis detection from chest radiographs*”. In: BMC Medical Imaging 24.1 (2024), p. 32.

[53] Saad I Nafisah and Ghulam Muhammad. “*Tuberculosis detection in chest radiograph using convolutional neural network architecture and explainable artificial intelligence*”. In: Neural Computing and Applications 36.1 (2024), pp. 111–131.

[54] Tawsifur Rahman et al. “*TB-CXRNet: Tuberculosis and Drug-Resistant Tuberculosis Detection Technique Using Chest X-ray Images*”. In: Cognitive Computation (2024), pp. 1–20.

[55] NK Roopa and GS Mamatha. “*CLBO: chef leader-based optimization enabled deep learning for tuberculosis detection using x-ray images*”. In: Signal, Image and Video Processing 18.1 (2024), pp. 877–887.

[56] Sukhendra Singh et al. “*Efficient pneumonia detection using Vision Transformers on chest X-rays*”. In: Scientific Reports 14.1 (2024), p. 2487.

[57] NIAID TB portal program dataset. Available online: https://tbportals.niaid.nih.gov/download-data.

[58] WHO TB Stats. https://tbfacts.org/tb-statistics-india/.

Ritik Jain <ritik09052000@gmail.com>

Acceptance : ICOTET 2024

1 message

ICOTET2024 <icotetdgi2024@gmail.com> Fri, May 10, 2024 at 2:58 PM
To: Ritik Jain <ritik09052000@gmail.com>

Greetings from ICOTET 2024**!**

Dear Author (s)

We are pleased to inform you that **Paper ID 2508** entitled " Tuberculosis Prognosis Through Radiographic Predictive Modeling" submitted by you has been accepted by the 2nd International Conference on Optimization Techniques in Engineering and Technology Engineering (ICOTET 2024).

You are advised to register for the conference by 16th of May, 2024! Payment details for registration can be found at the bottom of this email.

You are requested to fill out the following Google form for the registration and payment information etc.:

https://forms.gle/mqyRFhx45hqkJ6cx7

All the registered and presented papers for the 2nd ICOTET 2024 will be published in the AIP Conference Proceedings (Scopus Index) and Springer Nature Conference Proceedings (Scopus Index). Please note that the plagiarism level of the paper should not exceed 15%.

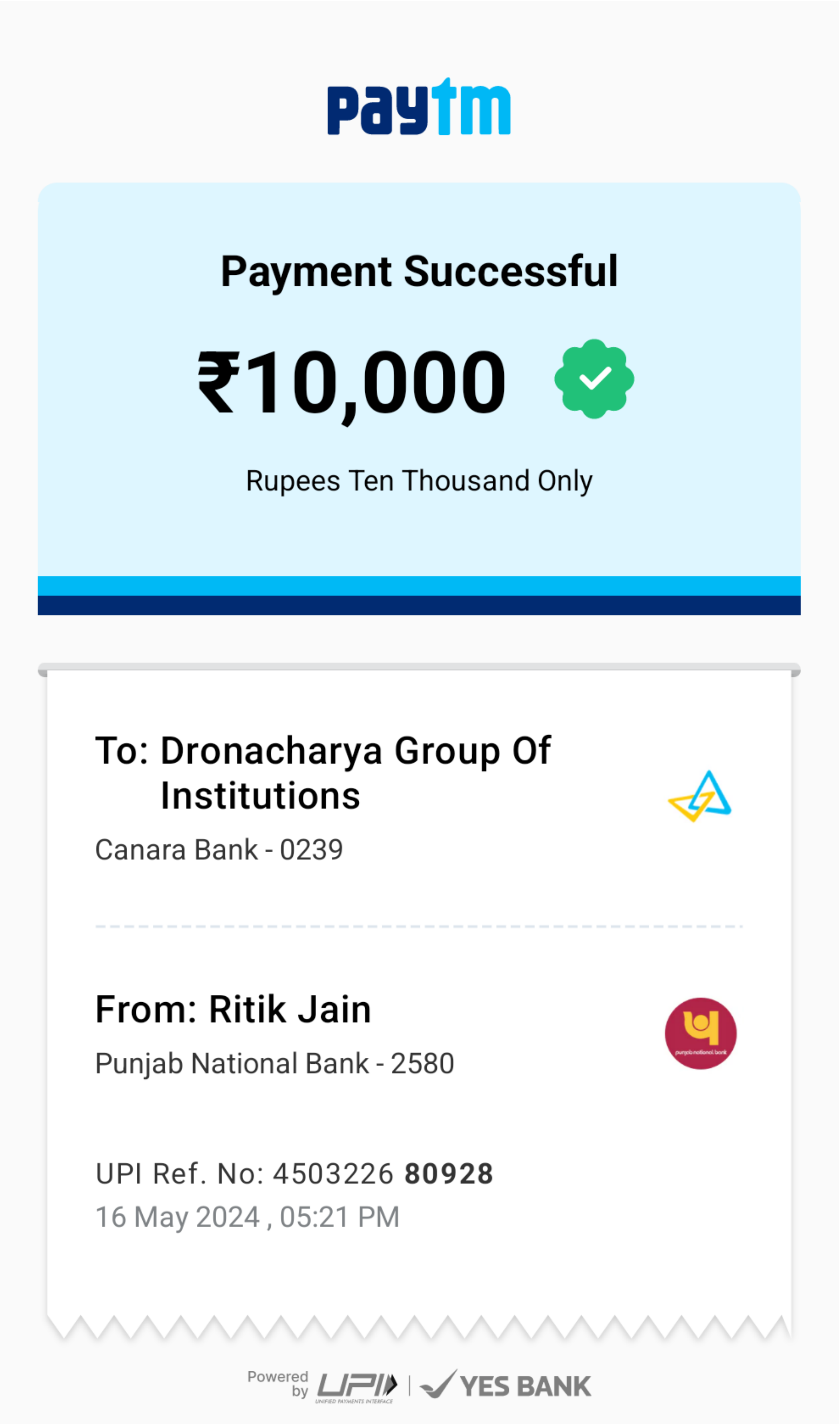
For further details, please visit the official website: https://www.icotet.in/registration

Thanks & Regards

Organizing Committee

ICOTET 2024.

Bank Transfer Details:	
Bank Name	Canara Bank, Jagat Farm, Greater Noida
Account Name	Dronacharya Group of Institutions
Account Number	88951010000239
IFSC Code	CNRB0002807
MICR Code	110015485



Paytm

Payment Successful

₹10,000 ✓

Rupees Ten Thousand Only

To: Dronacharya Group Of Institutions

Canara Bank - 0239

From: Ritik Jain

Punjab National Bank - 2580

UPI Ref. No: 4503226 **80928**

16 May 2024 , 05:21 PM

Powered by UPI UNIFIED PAYMENTS INTERFACE | YES BANK

Gmail **Ritik Jain <ritik09052000@gmail.com>**

Acceptance mail -- ICAAIML 2024

2 messages

iccse VGNT <iccse@vignanits.ac.in> Wed, May 29, 2024 at 10:21 AM
To: Ritik Jain <ritik09052000@gmail.com>

Dear Ritik

It is our pleasure to inform you that your papers entitled **A Review Of Advances In Pulmonary Tuberculosis Detection Using Deep Learning** (Paper Id: ICAAIML -25) has been provisionally accepted for Virtual oral paper presentation at ICAAIML-2024 on 30th and 31st August 2024, and also your paper has been accepted to publish in **AIP conference proceeding (SCOPUS)**

We request you to complete the early bird conference registration fee and publication charges i,e Rs 3000+ publication charges Rs 8500= 11,500 **If you don't want AIP publication then just pay Rs 3000 only**, After payment send the payment proof along with full manuscript.

Pay the registration fee through

Bank A/C No 00421140047879

Account Name : B Sridhar Babu

Bank Name: HDFC

IFSC Code: HDFC0000042

For conference updates please join our telegram channel : https://t.me/+VioSEzuF3b5NSzJ4

Thank you

with regards

ICAAIML

iccse VGNT <iccse@vignanits.ac.in> Thu, May 30, 2024 at 10:47 AM
To: Ritik Jain <ritik09052000@gmail.com>

Dear Author
https://icaaiml.com/registration/
Please fill the registration form after payment of Registration fee

Payment Details
Pay the registration fee through

Bank A/C No 00421140047879

Account Name : B Sridhar Babu

Bank Name: HDFC

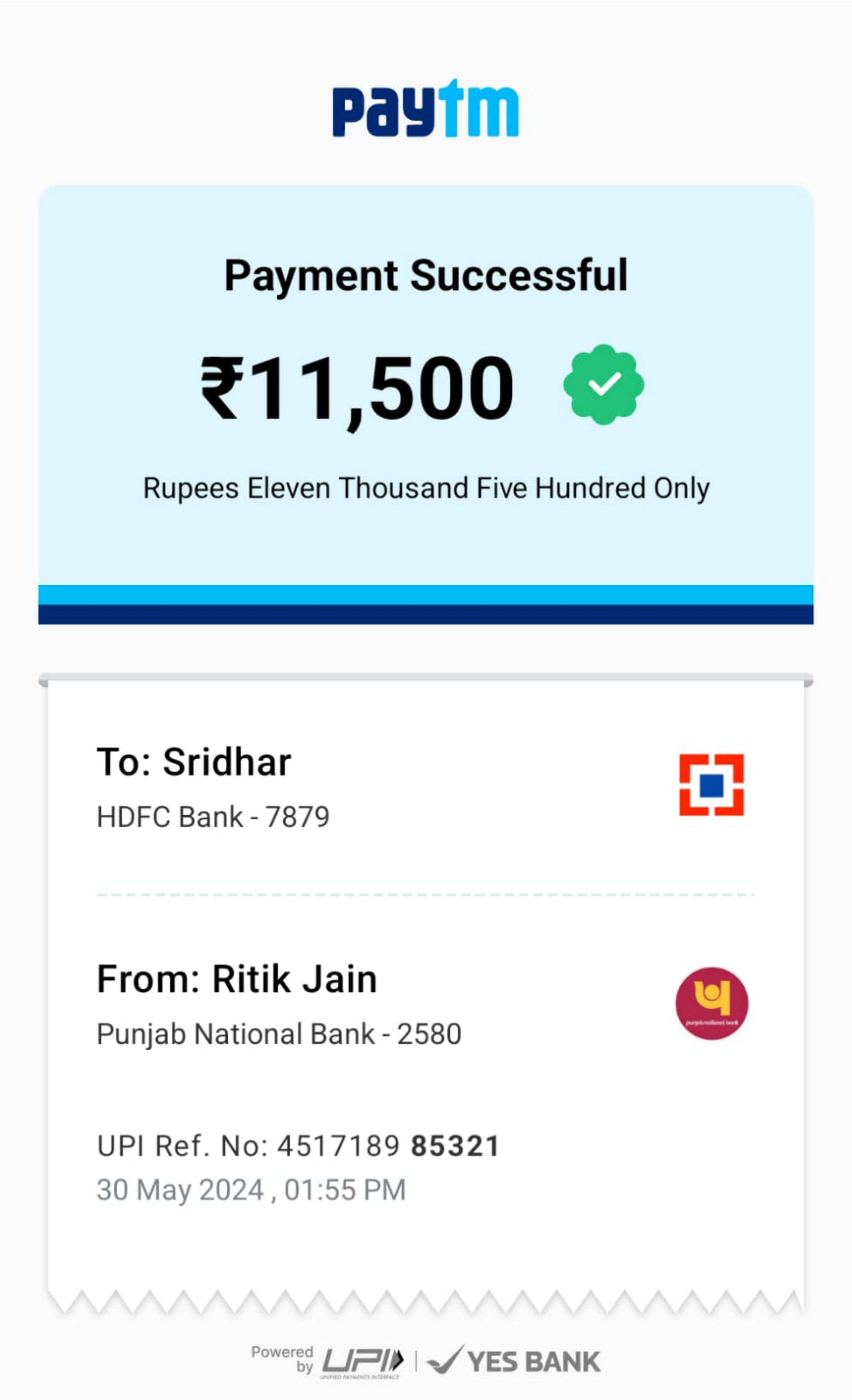
IFSC Code: HDFC0000042

For conference updates please join our telegram channel : https://t.me/+Y2wyeC96EHwwZTI1

with regards

ICMMSE

[Quoted text hidden]



paytm

Payment Successful

₹11,500 ✓

Rupees Eleven Thousand Five Hundred Only

To: Sridhar

HDFC Bank - 7879

From: Ritik Jain

Punjab National Bank - 2580

UPI Ref. No: 4517189 **85321**

30 May 2024 , 01:55 PM

Powered by UPI | YES BANK

PAPER NAME	AUTHOR
ThesisPlag.docx	Ritik Jain
WORD COUNT	CHARACTER COUNT
7294 Words	43603 Characters
PAGE COUNT	FILE SIZE
55 Pages	2.1MB
SUBMISSION DATE	REPORT DATE
May 30, 2024 2:07 PM GMT+5:30	May 30, 2024 2:08 PM GMT+5:30

● 11% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 6% Internet database
- Crossref database
- 6% Submitted Works database
- 7% Publications database
- Crossref Posted Content database

● Excluded from Similarity Report

- Small Matches (Less then 8 words)

● 11% Overall Similarity

Top sources found in the following databases:

- 6% Internet database
- 7% Publications database
- Crossref database
- Crossref Posted Content database
- 6% Submitted Works database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	louisdl.louislibraries.org Internet	1%
2	researchgate.net Internet	<1%
3	N. K. Roopa, G. S. Mamatha. "CLBO: chef leader-based optimization en... Crossref	<1%
4	geeksforgeeks.org Internet	<1%
5	Tawsifur Rahman, Amith Khandakar, Ashiqur Rahman, Susu M. Zughai... Crossref	<1%
6	Savita K. Shetty, Annapurna P. Patil. "Duck Pack Optimization With Dee... Crossref	<1%
7	actaneurocomms.biomedcentral.com Internet	<1%
8	doaj.org Internet	<1%

9	researchid.co Internet	<1%
10	KC Santosh, Siva Allu, Sivaramakrishnan Rajaraman, Sameer Antani. "... Crossref	<1%
11	Indian Institute of Technology on 2024-03-01 Submitted works	<1%
12	University of Glasgow on 2023-04-27 Submitted works	<1%
13	eprints.lancs.ac.uk Internet	<1%
14	Liverpool John Moores University on 2023-07-15 Submitted works	<1%
15	Aberystwyth University on 2023-07-13 Submitted works	<1%
16	Vamsi Kakani, B. Varun, Jyostna Devi Bodapati, Konda Raja Sekhar. "P... Crossref	<1%
17	Indiana University on 2024-05-01 Submitted works	<1%
18	Sukhendra Singh, Manoj Kumar, Abhay Kumar, Birendra Kumar Verma, ... Crossref	<1%
19	Aneesha Sharma, Anshika Yadav, Manoj Kumar. "Detection of Lung Dis... Crossref	<1%
20	Liverpool John Moores University on 2023-02-27 Submitted works	<1%

21	R. Swathi Sri, A. Menaka Pushpa. "Systematic Study on Diagnosis of L... Crossref	<1%
22	Sahebgoud Hanamantray Karaddi, Lakhan Dev Sharma. "Automated m... Crossref	<1%
23	Vo Trong Quang Huy, Chih-Min Lin. "An Improved Densenet Deep Neur... Crossref	<1%
24	journal.esrgroups.org Internet	<1%
25	medium.com Internet	<1%
26	recerc.eu Internet	<1%
27	fastercapital.com Internet	<1%
28	University College London on 2018-03-23 Submitted works	<1%
29	University of Cincinnati on 2023-04-09 Submitted works	<1%
30	idoc.tips Internet	<1%
31	khazna.ku.ac.ae Internet	<1%
32	Curtin University of Technology on 2023-02-01 Submitted works	<1%

33	Hyeonwoo Noh, Seunghoon Hong, Bohyung Han. "Learning Deconvolut... Crossref	<1%
34	repository.up.ac.za Internet	<1%
35	diva-portal.org Internet	<1%
36	Brunel University on 2024-04-19 Submitted works	<1%
37	Padmaja B, Madhubala M, Nagaraju M, Nandhan Varma Somalaraju, M... Crossref	<1%
38	South Bank University on 2023-05-26 Submitted works	<1%
39	Yue, Xiangyu. "Learning Transferable Representations Across Domains... Publication	<1%
40	Débora N. Diniz, Mariana T. Rezende, Andrea G. C. Bianchi, Claudia M. ... Crossref	<1%
41	Gouranga Charan, Abinash Mohanty, Xiaocong Du, Gokul Krishnan, Raji... Crossref	<1%
42	Mohammad Raihan Goni, Ayokunle Olalekan Ige, Nur Intan Raihana Ru... Crossref	<1%
43	Sydney Polytechnic Institute on 2024-05-26 Submitted works	<1%
44	United International University on 2024-01-26 Submitted works	<1%

45	University of Portsmouth on 2023-05-18 Submitted works	<1%
46	University of Sheffield on 2023-11-28 Submitted works	<1%
47	University of Sunderland on 2023-07-24 Submitted works	<1%
48	etd.uwc.ac.za Internet	<1%
49	journalofcloudcomputing.springeropen.com Internet	<1%
50	publications.eai.eu Internet	<1%
51	researchoutput.csu.edu.au Internet	<1%

RITIK JAIN

+919582410394 ritik09052000@gmail.com
2K22/AFI/016 Github| LinkedIn

EDUCATION

M.TECH(Artificial Intelligence)	2022-present	Delhi Technological University, New Delhi	9.17
B.TECH(Computer Engineering)	2018-2022	Krishna Engineering College AKTU, Lucknow	8.94
CBSE (Class XII)	2018	DAV Public School	93.12 %
CBSE (Class X)	2016	DAV Public School	9.2

EXPERIENCE

Associate Software Engineer, Nagarro, Gurugram January 2022-August 2022

- **Developed** and managed **CRM solutions** on **Adobe Experience Manager** (AEM) platform, with **comprehensive training** in web development [**Node.js**, Postman, Express, MongoDB].
- Led the conceptualization, design, and implementation of reusable components that enhanced the overall user experience and reduced development time.

Python Intern, HCL Technologies, Noida June 2019-July 2019

- **Developed** a project using **Python GUI** (Graphical User Interface) to create an intuitive and user-friendly application. **Worked closely** with senior developer for project discussions and to incorporate changes as per project requirement.

ACADEMIC PROJECT

ML Based Text Analyzer—Python Notebook**|** WebLink **|** Python**|** Data Analysis | NLP| ML

- Developed and optimized classifiers for spam detection, movie reviews, and restaurant reviews using various ML algorithms and **vectorization** techniques.
- Achieved high accuracy through advanced text processing and model tuning.

Olympics EDA— Web App | WebLink | Python | Streamlit | Data Analysis

- Conducted in-depth exploratory data analysis on a comprehensive data-set of Olympic Games from 1896-2016, with **interactive dashboard** analyzing various aspects.

Skin Lesion Detection — *Python Notebook* | Link | Python | Deep Learning

- Explored and implemented a skin detection system using DenseNet, achieving [**Accuracy**:90.3, **Precision**:89.07, **Recall**:89.67], with optimization and Fine Tunning.

ACADEMIC ACHIEVEMENTS AND AWARDS

- 1200+ score on GFG and 250+ problem solved on Leetcode , 5 star in python on HackerRank.
- Won various coding competitions in Fests and Events and 2nd in college hackathon (B.Tech.)

TECHNICAL SKILLS AND COURSEWORK

Programming : C | C++ | Python | Nodejs | SQL | HTML | CSS | GIT	**Familiar With :** Java | Streamlit | ReactJS	**Software**: Postman | Jupyter Notebook | VsCode | Pycharm
Subjects: Data Analysis, Machine Learning, Natural Language Processing, Deep Learning, Object OrientedProgramming, Database Management, Data Structures and Algorithm, Operating System.		

POSITIONS OF RESPONSIBILITY

- **Role: Team Leader -** *College Fest Coding Contest*
  - Led a team in a competitive coding contest during [KEC/2k20 Revamp], with managing time constraint and assigned problem based on teammates' strengths.
- **Role: Project Coordinator/Team Member-** *24 hr College Hackathon*
  - Worked with a team of 6 members on Text To Speech project '**Aur Sunao**' [KEC/ 2k19].
  - Specifically tasked with managing a **backend (Python)** part of the project along with twoother teammates. **Facilitated discussions** and done decision-making responsibilities.