

CNN-Based Deep Learning Models for Automated Tuberculosis Detection from Chest X-rays

MSc Research Project
MSc Data Analytics

Pooja Santhosh
Student ID:
x23274875

School of Computing
National College of
Ireland

Supervisor: John Kelly

National College of Ireland
MSc Project Submission Sheet
School of Computing



Student Name: Pooja Santhosh.....
Student ID:x23274875.....
Programme:MSc Data Analytics..... **Year:** ...2024-2025...
Module: MSc Research Practicum.....
Supervisor: John Kelly.....
Submission Due Date:11-08-2025.....
Project Title: CNN-Based Deep Learning Models for Automated Tuberculosis Detection from Chest X-rays

Word Count: ...8683..... **Page Count:**.....20.....

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:Pooja Santhosh.....

Date:11-08-2025.....

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

CNN-Based Deep Learning Models for Automated Tuberculosis Detection from Chest X-rays

Pooja Santhosh
X23274875

Abstract

Tuberculosis (TB) remains one of the leading health concerns of the world especially in the low-resource regions where radiologists who can help one with diagnosis are scarce. Although chest X-ray imaging is prevailing examinations used in the screening of TB, it relies strongly on interpretation of experts and may prejudice performance and delay in areas that are under-privileged. Luckily, due to the advent of artificial intelligence, methods of deep learning, and Convolutional Neural Networks (CNNs) have a very high potential of automating the process of analysing medical images. The purpose of this project is to carry out the comparison of the performance of the three most popular and popular pre-trained CNN architectures ResNet18, DenseNet121, and EfficientNet-B0 to see which model is more suitable in the problem of detecting TB based on a chest X-ray. We already have a publicly available dataset in Kaggle and will fine-tune the models by using transfer learning. We will use a stratified data split by training, validation and test sets to balance the set classes. All models are going to be tested in PyTorch with standard metrics such as accuracy, precision, accuracy and F1-score. The findings of the given comparison will illuminate on the trade-off between accuracy and computational efficiency of the models.

Keywords: Tuberculosis, Chest X-ray, Deep Learning, CNN, ResNet, DenseNet, EfficientNet, Transfer Learning, PyTorch, Medical Diagnosis

Chapter1: Introduction

The most common screening test used at the initial stages of TB is a chest X-ray image since it is less expensive and readily available. Nevertheless, there must be solid expertise to read such X-rays right, and it is very limited in most under-resourced regions. Moreover, manual diagnoses are fickle, and prone to error. To address these challenges, deep learning, and especially the Convolutional Neural Networks (CNNs) appear as a potentially valuable method of automating and standardizing image analysis in medical diagnostics. Here is a project on how deep learning is used in automatized detection of tuberculosis in chest X-ray images (**Bantelay, L.M., Abebe, M., Rajendran, R.S., & Sungheetha, A. (2023, November)**). The 1st aim is to compare and evaluate performance of 3 popular CNN networks ResNet18, DenseNet121 and EfficientNet-B0 to determine which model was best in balancing accuracy with efficiency in TB detection. Instead of training a CNN on our own we employ the so-called transfer learning approach which involves using pretrained models and fine-tuning them on a labelled dataset provided by Kaggle. In order to conduct a fair analysis, the data is stratified into training, validation and test groups through stratified sampling concept by keeping each group balanced in terms of classes.

1.1 Aim of the Project

This project aims at developing an automated system based on the deep learning to differentiate the chest X-ray images in TB-positive and TB-negative cases. We would like to apply and evaluate some of high-performing convolutional neural network-based architectures, namely, ResNet18, DenseNet121, and EfficientNet-B0, in order to determine which of them provides the best outputs in the context of diagnostic accuracy and computational efficiency. Comparison of these models on the same setting using a publicly available data should provide us with compelling advice on which model is more reliable and should be implemented with a scalable scheme in view of making it available to

engage in TB screening particularly in areas where the availability of trained radiologists is scarce (Soni, A. and Rai, A., 2024, June).

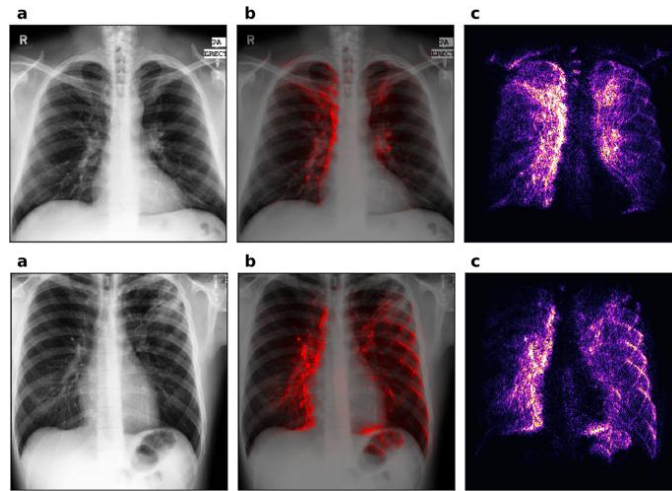


Figure 1.1: Efficient Deep Learning based TB detection. (a), (b), (c) shows the progression of identification model decision making with the important regions of detection (**Source: nature**)

1.2 Objective

The primary objective of this project is to establish the optimal CNN architecture in the automatic detection of tuberculosis using the image of the chest X-rays. To have it done, we are going to handle several key tasks, namely, first, we download some publicly available dataset of labeled X-ray images and preprocess them resize, normalize and augment them. After this, we split the dataset into training set (for training the model), validation set (to cross validate the model) and test set (to test the model). This also involves stratified sampling to maintain the share of classes. Then we optimize three state-of-the-art deep learning networks ResNet18, DenseNet121 and Efficient-B0 with PyTorch as the transfer learning model (Wang, Y., Pan, L. and Shu, X., 2024). Both the models are trained in similar conditions so as to develop a fair comparison. Finally, we measure performance of each of the models with measures such as accuracy, pixel precision, pixel recall, F1-score and confusion matrix. To this effect, the project expects to find out the best performing model by using their results to be applied to real world TB diagnostic systems.

1.3 Motivation

This project is motivated by the fact that tuberculosis remains a global concern because of the current rate at which the condition is facing and the rising power of artificial intelligence in the modern healthcare system. Millions of lives continue to be affected by TB every year especially in the underdeveloped and developing countries whose access to quality medical services is limited. It has been observed that the chest X-rays are one of the only few promising screens in these areas, however, due to the lack of experienced specialists, such methods are weakened. This is where the deep learning would help as it is scalable and cost effective and the convolutional neural networks (CNNs) have shown exceptional potential in the classification of the images (Chen, H., Gu, W., Zhang, H., Yang, Y. and Qian, L., 2024). Nonetheless, the success of these models depends on the identification of the appropriate compromise between the accuracy and such properties as the computational efficiency of architecture. This project aims at establishing a reliable model of TB detection using open-source sets of tools and data, which may serve the interest of healthcare professionals in addition to providing assistance in early diagnosis and, therefore, improving patient outcomes in the context of global tuberculosis elimination.

1.4 Problem Statement

The issue of Tuberculosis (TB) is a major challenge in the health sector especially in the developing countries where diagnosis and treatment in case of successfully catching the disease at an early stage can actually save a life. Among the most effective strategies of TB screening are chest X-rays the application of which is a non-invasive and relatively economical procedure. Nevertheless, the understanding of such X-rays requires the expert level of knowledge. However, it is a shame that in most parts of the world, there is the lack of skilled radiologists, and this is leading to delay, wrong diagnosis or no diagnosis at all in many cases (**Gupta, S., Jindal, R., Biji, C.L. and Dheeba, J., 2024**). Besides, manual analysis of X-rays may be affected by human appreciation and exhaustion. Although the deep learning technology has contributed to automating all these processes, the challenge has been how to choose the most appropriate model to implement detection of TB. The current variety of CNN-based models is classified as capable of delivering different results, their computational costs, as well as adaptability to resource-hungry environments, vary significantly. It is, therefore, paramount to scientifically test the well-performing CNN models to identify the most effective one that will be applicable in the real-life situation in terms of detecting TB based on chest X-rays.

1.5 Research Questions

RQ1: Can we identify precisely the regions impacting the TB identification?

RQ2: How is the performance of different transfer learning models with YOLO models?

In this study we will discuss about the performance of different deep learning models in the TB detection using two different datasets -> TBX-11 and NAID. In the next chapter we will discuss on the different research that has been done towards this field in the past. In the next chapter on methodology we will discuss in detail about the techniques that is employed to carry out this study. In the chapter on implementation we will discuss about the proposed framework that is employed for the research work, finally in the chapter on results and analysis we will discuss about the different findings.

Chapter 2: Literature Review

2.1 Tuberculosis Detection Using Machine Learning

Current advancements in tuberculosis (TB) identification cut a vast range of techniques, involving improvements in physics-based sensor architecture, conventional machine learning, and deep learning designs. An innovative work demonstrated by **Wekalao (2025)** proposed a graphene-based terahertz metamaterial biosensor which was optimally structured through the XGBoost algorithm and featured extraordinary sensitivity of 1000 GHz/RIU and a value of figure of merit (FOM) as 9.434 and allowed sensing the refractive index to be done with extreme delicacy (**Wekalao, J. (2025)**). The hybrid resonator design of the sensor and the model of the sensor that was calculated on COMSOL model ensured future of an excellent point of care diagnostic tools but with the limitation of without clinical data the simulation inputs were used altogether. Instead, **Acharya et al. (2022)** treated the task with a Normalization-Free Network (NFNet) diagnostic framework, which measures up well to the challenges of batch normalization and can achieve high levels of performance classification over large-scale chest X-ray images. Their pipeline with Rand Augment, progressive resizing and explainable AI methods such as Score-CAM scored 96.91 percent accuracy and 99.38 percent AUC on the multiclass TBX11K dataset, which demonstrated its potential application in clinical cases without professional radiology knowledge. In the traditional machine learning domain, **Hrizi et al. (2022)** applied a SVM-based multi-stage diagnostic system, which combines the wavelet and spatial texture features along with using genetic algorithm to perform feature selection and hyperparameter optimization. The method was useful in detecting TB in 3D CT scans because characteristic slices could be identified and selected only by the analysis of informative 2D slices, and it has up to 97 percent accuracy on binary classification with

also having scalability and interpretability in medium-sized data. Another reinforcement of the AI-based diagnostic toolset was a strong deep learning-based ensemble framework, VoPreCNNFT, suggested to combine preprocessing variety (e.g., CLAHE), data augmentation approaches, and two deep learning backbones, the InceptionV3 and Xception (**Tasci, E., Uluturk, C., & Ugur, A. (2021)**). With soft voting and Bayesian optimization-based weighted voting their approach attained the maximum accuracies of 97.50% and 97.70%, respectively, on the benchmark datasets such as Montgomery and Shenzhen, which proves the effectiveness of ensemble learning in the field of medical imaging as by **Tasci, E., Uluturk, C., & Ugur, A. (2021)**. Taken together, the evidence presented in all these studies demonstrates that there is a solid overlap between simulation-based sensor technology and AI-powered diagnostic pipelines, and the hybrid approach has a promising future as it is capable of maximizing TB screening accuracy and scalability. What unites these initiatives, however, is the need to engage in real-clinical validation, connection with hospital infrastructures, and go a step further in generalizing them to different population groups.

2.2 Tuberculosis Detection Using Convolutional Neural Network

Numerous deep learning approaches have been suggested over the past years to not only improve the quality, but also to make the accuracy of the tuberculosis (TB) detection systems using chest X-ray (CXR) images more comprehensive and interpretable as well as generalizable. Nafisah and Muhammad (**Nafisah, S.I., & Muhammad, G. (2024)**) were able to create a full pipeline of AI with the help of lung segmentation with U-Net and 9-angle massive augmentation to build a powerful training sample. They identified five common pretrained CNNs that are Xception, Inception-ResNet-V2, ResNeXt-50, MobileNet and EfficientNetB3 and finetuned them using a combination of fivefold cross-validation. They gave the best model as EfficientNetB3 which trained on segmented and augmented images and indicated an impressive 99.1 accuracy and 0.999 AUC. The overall performance of this system was assessed on three different standard databases: Montgomery (138 CXRs), Shenzhen (662 CXRs), and Belarus (304 CXRs), which obtained good generalization. Very much in the same spirit, **Bantelay et al. (Bantelay, L.M., Abebe, M., Rajendran, R.S., & Sungheetha, A. (2023, November))** have introduced a hybrid multi-input model that combines the use of CNNs and CXR images with multilayer perceptrons (MLPs) trained on structured medical record information. Each of the modalities was trained jointly through feature-level concatenation. Otherwise, their combined approach also surpassed unimodal ones, having a ludicrous accuracy of 99.61%. Thus, the power of the multimodal integration in medical diagnostics. Conversely, Huy and Lin (**Huy, V.T.Q., & Lin, C.M. (2023)**) introduced CBAMWDNet, an enhanced DenseNet with Convolutional Block Attention Modules (CBAM) in order to include more related features. Their model was based on a composite set containing 5000 images of three datasets merged among Montgomery, Shenzhen, and a TB CXR dataset of 4,200 samples resulting in an F1 score and accuracy of 96.35 and 98.80 by far and above other deep networks using non-pretrained weights. Xu and Yuan (**Xu, T., & Yuan, Z. (2022)**) emphasized lightweight and explainable modeling with the adoption of coordinate attention to the vgg16 back, VGG16-CoordAttention, which introduces directional and position-sensitive feature extraction. They used transfer learning and weight freezing of their approach and performed fivefold cross-validation on the Shenzhen dataset which achieved 93.18 accuracy and 97.98 AUC, exceeding the performance of the ResNet50 and Vision Transformers (ViT) under the same protocol. Finally, Dey et al. (**Dey, S., Roychoudhury, R., Malakar, S., & Sarkar, R. (2022)**), suggested a fuzzy ensemble methodology where type-1 Sugeno fuzzy integrals calibrated with meta-heuristic optimum were applied to figure out fusion weights based on three prebroken CNNs: DenseNet121, VGG19, and ResNet50. This was tested using the new TB CXR and shown that the method provided greater results compared to conventional ensembles, which indicates the possible usefulness of fuzzy logic in adaptively combining models. Overall, all these studies represent a palette of methodological advances, such as attention enhanced CNNs, ensemble learning, segmentation-based pipelines, multimodal fusion, that are all aimed at achieving robust, interpretable and accurate TB diagnosis system with publicly available data, such as Shenzhen, Montgomery, Belarus and the TB Chest X-ray dataset curated by Qatar University and collaborators.

2.3 Tuberculosis detection using Yolo Base Models

The You Only Look Once (YOLO) algorithm (**Soni, A. and Rai, A., 2024, June**), a widely utilized technique for object detection, is extensively applied in numerous medical contexts. The PubMed database was thoroughly searched for peer-reviewed articles published from 2018 to 2024. This search revealed 80 studies that utilized the YOLO approach for detecting medical items, including breast cancer, COVID-19, lung nodules, pneumonia, TB, and brain tumors. The results indicated that YOLO outperformed alternative tactics and validated its efficacy in these tasks. Nonetheless, the study on YOLO identified many limitations, including its necessity for extensive and balanced datasets and its computational intensity. The paper highlights these qualities, identifies areas requiring additional investigation, and proposes potential applications of YOLO in future medical object identification.

This research (**Wang, Y., Pan, L. and Shu, X., 2024**) presents YOLO-FMS, an efficient and lightweight model for medical microscopic smear detection based on YOLOv5, with a weight of only 15.06M. It effectively resolves the challenge of achieving an optimal balance between detection accuracy and model parameters. The integration of a compact target detection head into YOLO-FMS significantly improves the efficacy of small-scale platelet and mycobacterium tuberculosis identification. To further reduce computational expenses and enhance the symbolic capability of lightweight convolutional networks to approximate that of traditional convolutional networks, GSConv, a lightweight convolutional approach, was created. Ultimately, the C3-B-CBAM and Tiny-SPPCSPC modules of our proposed method enhance the feature extraction proficiency of YOLO-FMS. Extensive testing indicate that YOLO-FMS has great detection accuracy, with the BCCD dataset achieving a mean Average Precision (mAP) of 92.5% and the Tuberculosis-Phonecamera dataset attaining a mAP of 87.6%. Moreover, YOLO-FMS has undergone validation through several verification experiments utilizing the BCDD and BBBC041 datasets, which illustrate its effectiveness and generalizability in the identification of medical microscopic smears. The source code for YOLO-FMS is available at the following address: <https://github.com/GefionP/YOLO-FMS>.

In comparison to earlier YOLO (**Gupta, S., Jindal, R., Biji, C.L. and Dheeba, J., 2024, September**) models and Faster R-CNN ResNet50, YOLOv8 had the highest mean average precision of 0.86. The outcomes of our testing indicate that the YOLOv8 model can identify tuberculosis from small picture fields more rapidly and precisely than alternative models. We utilized Layer-wise Relevance Propagation, a prominent Explainable AI technique, to examine the interpretability of bacilli detection in sputum smear images. This integration of Explainable Artificial Intelligence enhances transparency and confidence in the model's performance by elucidating model decisions and the significance of each voxel in the image. This automated detection method can enhance clinical outcomes by assisting physicians in making more accurate illness predictions in a shorter timeframe.

This study (**Fandriyanto, A., Utama, N.P. and Danudirdjo, D., 2024, July**) integrates a custom-built Field-of-View (FoV) scanning microscope with a mycobacterium tuberculosis (MTB) object detection system to tackle these issues. Object detection is executed utilizing a You-Only-Look-Once (YOLO) approach subsequent to the acquisition of TB images by the microscope. Upon inputting the test data into the model, it generates an image annotated with bounding boxes and indicates the quantity of detected MTBs. The degree of severity of tuberculosis is then assessed using this figure. The system effectively captures, recognizes, and classifies authentic TB images, as per the experimental findings.

The subsequent stage (**Chen, H., Gu, W., Zhang, H., Yang, Y. and Qian, L., 2024**) involves adjusting the model for different sizes of M. tuberculosis by integrating a multi-scale feature fusion module. Additionally, the Coordinate Attention (CA) module is augmented with a convolutional layer to derive more intricate semantic information. Following the CA module, a self-attention mechanism is integrated to enhance the model's understanding and localization of the varied morphology of Mycobacterium tuberculosis. Upon assessment using a publicly accessible dataset, our model exhibited commendable

performance, with an average precision of 85.7%. The effectiveness of our proposed methodology in identifying Mycobacterium tuberculosis is apparent from this.

A crucial component of tuberculosis (TB) (Aulia, S., Suksmono, A.B., Mengko, T.R. and Alisjahbana, B., 2024) diagnosis is the microscopic detection of acid-fast bacilli (AFB) from Mycobacterium tuberculosis (MTB) in sputum specimens stained using the Ziehl-Neelsen (ZN) method. The irregular morphology of MTB, substandard ZN staining, and errors in scanning each field of view (FoV) with a conventional microscope are but a few of the challenges encountered by pathologists, potentially resulting in erroneous diagnoses. Moreover, human error may arise from fatigue induced by multiple manual observations. Researchers have created computer-aided diagnosis (CAD) for tuberculosis utilizing microscopic imaging databases of sputum samples. This technology demonstrates potential due to its rapid, dependable, and consistent support. The lack of sputum sample microscopic image datasets compliant with WHO standards for whole-slide imaging (WSI) indicates that the implementation of CAD systems for tuberculosis (TB) diagnosis remains an area of ongoing research and advancement. This work produced a novel digital microscopic image using sputum smear samples from WSI patients in Indonesia that adhere to WHO recommendations to address this issue. The Macro Imaging Database of Tuberculosis in Indonesia (MIDTI) serves as a repository for these images. This work introduced an innovative way by employing the YOLOv7 algorithm to develop a computer-aided diagnostic (CAD) system for tuberculosis detection. The CAD would classify ZN-stained sputum smear samples into categories based on the criteria established by the International Union Against Tuberculosis and Lung Disease (IUATLD).

The proposed study (Parveen Rahamathulla, M., Sam Emmanuel, W.R., Bindhu, A. and Mustaq Ahmed, M., 2024) (Our base paper) used the Kaggle TBX-11k dataset to enhance tuberculosis prediction accuracy by integrating the Selection Focal Fusion (SFF) block into the You Look Only Once v8 (YOLOv8, Ultralytics, Los Angeles, United States) object detection model using an attention mechanism. Due to its multi-object detection capacity, YOLOv8 is frequently utilized. Nonetheless, it cannot do fine-grained categorization and struggles with minute objects. To address this issue, the proposed study will employ the SFF approach to enhance detection efficiency and reduce the incidence of missed detections of small objects. Similarly, recall, precision, F1 Score, and mean Average Precision (mAP) are performance metrics employed to evaluate the proposed framework's effectiveness in the anticipated mechanism. The proposed study demonstrates efficiency relative to existing models. The primary objective of this work is to assist radiologists in utilizing the YOLOv8 model for tuberculosis diagnosis, representing a notable advancement in medical practice.

2.4 Research Summary

In the recent tuberculosis (TB) detecting studies, there are three major trends that can be identified:

- (a) Ultra-high sensitivity sensor-based detections like the graphene-based terahertz metamaterial biosensors optimized using XGBoost, though, remains confined to the simulation stage with no clinical application.
- (b) Imaging: (2) Deep learning Deep learning approaches to imaging, such as Normalization-Free Networks (NFNet), SVM-based multi-stage classifiers, ensemble-CNN networks have demonstrated better than 96.99 accuracy when applied to benchmark datasets (TBX11K, Shenzhen, Montgomery).
- (c) Object detection using YOLO (particularly in microscopy and X-ray analysis) has progressed beyond YOLO with lightweight variants (YOLO-FMS), attention-enhanced variants (YOLOv8 + SFF) and multimodal feature fusion variants also emerging as solutions to problems such as high mAP (up to ~99% in some cases) with detection of small objects, interpretability, and computational efficiency.

All these studies together show that AI-based detection and sensor-based technology are moving towards scalable, accurate TB screening tools, but clinical verification and adjusting to a wide range of populations are the key gaps.

Chapter 3: Methodology

3.1 Data Acquisition

We utilized the TBX11 dataset, which comprises three distinct classes: Healthy, Sick, and Tuberculosis (TB). In addition, we employed a second dataset provided by the National Institute of Allergy and Infectious Diseases (NIAID), which contains two categories: Healthy and TB. Together, these datasets offer a comprehensive foundation for building and evaluating robust models for tuberculosis classification and detection using chest X-ray images.

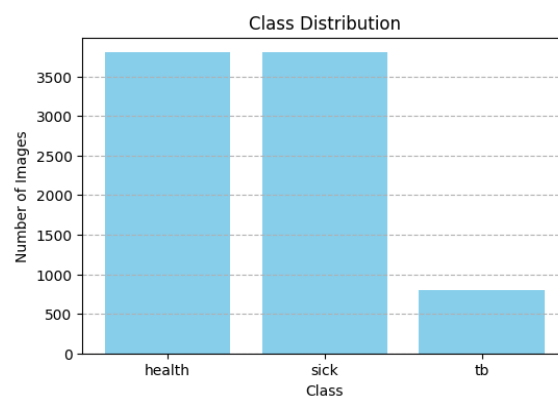


Figure 3.1: Class Distribution of TBX11 Dataset.

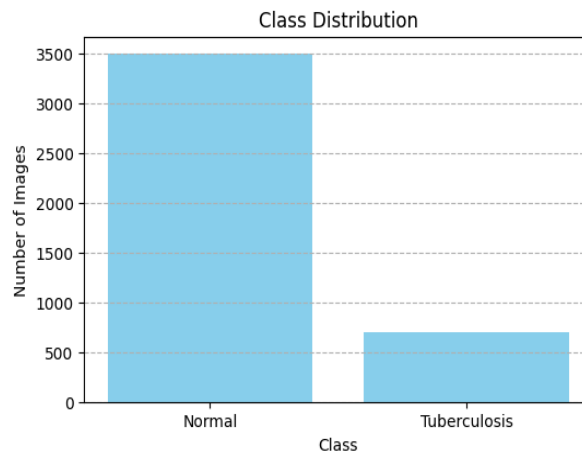


Figure 3.2: Class Distribution of NIAID Dataset.

In the above figures we find that the classes are imbalanced with the tuberculosis being the least or the minimal numbers present.



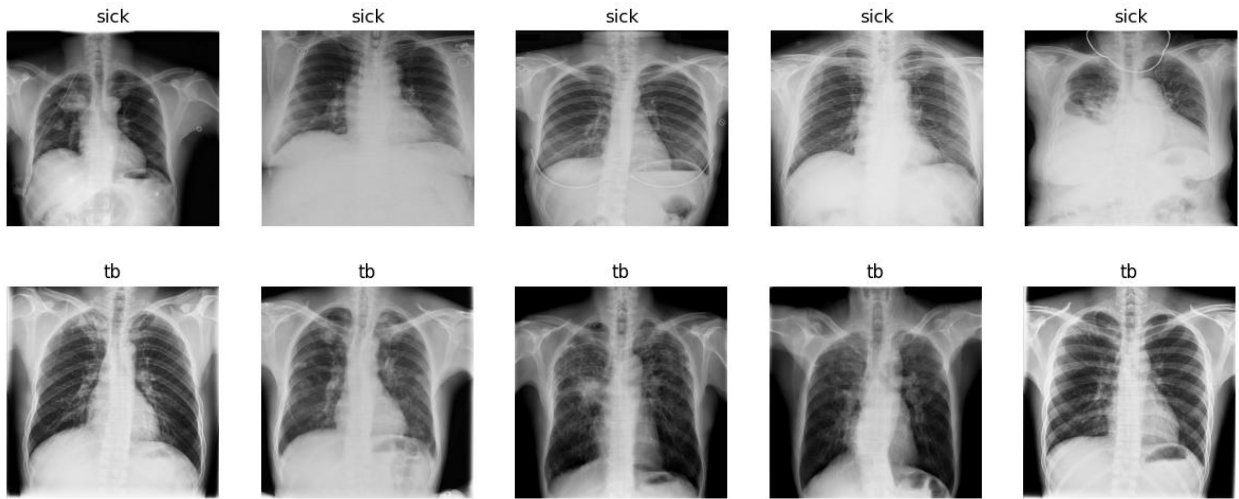


Figure 3.3: Sample Images from TBX11 dataset.

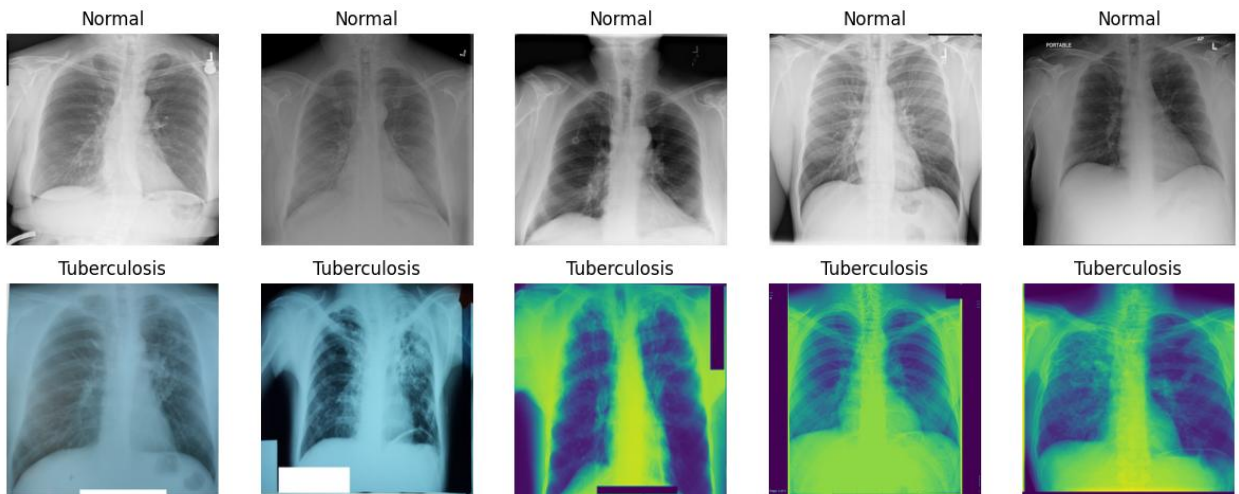


Figure 3.4: Sample Images from NIAID dataset.

3.2 Data Preprocessing

3.2.1 Image Transformation

Due to variations in dimensions, orientation and contrast, the medical X-ray images may affect its performances using a deep learning model by making it learn inconsistent features. To correct this, we have included transformations which standardize the image input and simulate variability in the real world. The activity of resizing would make sure that all the images are of the fixed input dimensions necessary by the model, and thus, the problem of different image sizes is avoided. The inclusion of controlled randomization by flipping horizontally and making minor rotations facilitates the learning process because the model will need to learn rotationally invariant features since X-rays orientation could differ a bit among patients or institutions. To minimize intensity variance and maintain pixel distributions consistent, normalization was used that helps to achieve faster convergence and better generalization as it helps to match input statistics with the statistics the pretrained backbone was trained on.

3.2.2 Splitting Images in Sets for training.

In most of the medical datasets, some classes, particularly disease classes, can be underrepresented. In the case of random splits, the imbalance may be further exaggerated in the training, validation or test sets, which will lead to false performance figures and inadequate generalization. To address this we used stratified splitting where every subset will have the same distribution of classes as the whole dataset. This will avoid the scenario where, e.g. the validation set will include many healthy cases whereas few TB examples are observed in the training set resulting in an unreliable validation feedback and biased learning. Stratification guarantees equal exposure and interpretation in all classes that is necessary in coming up with models that can be used in critical areas such as disease detection.

3.2.3 Addressing Class Imbalance

- **Weighted Sampling:** When dealing with TB detection, the dataset tends to include relatively very many healthy samples compared to TB accurate samples. Such training on this inherently skewed distribution has the effect of biasing the model in Favor of the majority class thereby making it insensitive to TB. To deal with this we employed WeightedRandomSampler when training. A minority class samples (i.e., TB cases) are, in this method, given higher probability of selection as the samples to be included in the training batches. Consequently, the model gets access to a more balanced outlook of the two classes, and it learns significant features on the TB detection with the class imbalance.
- **Class-Weighted Loss Function:** The model may still perform poorly at predicting minority classes even when a balanced set of sampling is used in case of a loss function that does not equally weigh all errors. We employed class-weighted loss in order to place a more direct penalty on the misclassification of TB cases. The weights given in the calculation of the loss to the TB class were larger, which made it costly to miss the diagnosis of TB (i.e., a false negative, not considering that a TB is present). This will lead the model to be less suspicious and blind to predictions of TB and, ultimately, better at recalling and being robust, particularly at the more critical but underrepresented cases.

3.3 Model Architecture

3.3.1 ResNet

ResNet(Residual Network) architecture is an innovative breakthrough improvement in the deep learning, especially in computer vision. ResNet was created in a manner which gets around the disadvantages of training exceptionally deep neural networks. Due to deepening of the networks, they become susceptible to the vanishing gradients and performance loss and the more layers you add to your network the higher the training error becomes. ResNet overcomes this challenge by proposing the notion of residual learning, learning how to adjust the identity mapping rather than the complete transformation, by keeping layers easier to optimize and able to train networks of more than one hundred layers.

The main idea behind the ResNet architecture is skip connections or shortcut path. Each layer of a typical deep network feeds directly into the other. On the contrary, ResNet feeds the output of a collection of layers directly into its output, which means the network learns a residual function. Mathematically the network does not learn a direct mapping $H(x) \rightarrow H(x)$, the network learns $F(x) = H(x) + xF(x) = H(x) + x$, resulting in a final output of $F(x) + xF(x) + x$. This is effective in saving gradients in backprop and also works towards eliminating the vanishing gradient problem and it enhances the training efficiency of deep structure by a large margin.

ResNet is available in a range of variants, by depth, ResNet-18, ResNet-34, ResNet-50, ResNet-101, and ResNet-152. The simpler variants (ResNet-18 and ResNet-34) employ a simplistic residual block which is composed of a couple of 3x3 convolutional layers with Batch Normalization and ReLU activation and a skip connection with it. In the deeper variants ResNet-50 and beyond the more computation-efficient bottleneck block is deployed where the block is reduced in dimension via a 1x1 convolution, increased in dimension via a 3x3 convolution and recursively again by a 1x1 convolution and added via a shortcut.

There is a tire architecture of ResNet that commences with 7x7 convolution and then max pooling. This is then followed by a series of residual blocks in stages whose filters doubles but less spatial dimensionality. As an example, ResNet-50 has the structure of 4 stages; 3 and 4 bottleneck blocks each in stages 1 and 2 and 6 and 3 in stages 3 and 4 giving a total of 50 layers. The network classifies using global average pooling layer after the last block followed by a fully connected softmax layer.

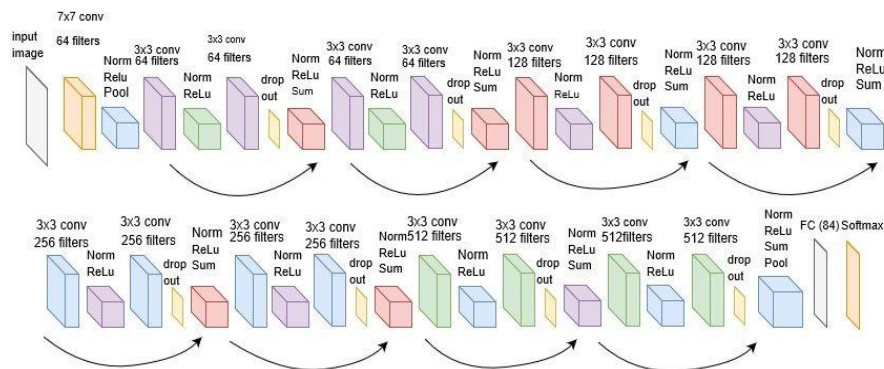


Figure 3.5: ResNet Architecture (*Source : https://www.researchgate.net/figure/Proposed-Modified-ResNet-18-architecture-for-Bangla-HCR-In-the-diagram-conv-stands-for_fig1_323063171*)

3.3.2 DenseNet

Another important deep learning model in the field of computer vision is the architecture of DenseNet (Densely Connected Convolutional Network). Similarly to ResNet, DenseNet deals with the issue of training very deep networks, which include vanishing gradients, overfitting and redundant features. But DenseNet proposes a novel approach: dense connectivity that guarantees each layer to get information coming in not only from the previous greatest next layer, but from all previous layers alike. Such an extreme architecture causes effective reuse of features and better promotion of gradients across the network.

The concept behind the philosophy of DenseNet is that of a dense block. In a dense block, every layer is directly connected to every other layer in feed-forward manner. This implies that feature map of all the preceding layers is concatenated to form the input to the current layer. Considering that a block has L layers, direct connections will be $L(L + 1) / 2$. This kind of connectivity allows the maximum flow of the information and promotes features reuse thus the network is more compact and efficient than its classic counterparts.

A dense block has several layers of the structure consisting usually of a Batch Normalization layer, ReLU activation layer, a 1x1 convolution (bottleneck) layer, another BatchNorm + ReLU layer, and a 3x3 convolution. This bottleneck architecture (which is commonly called as a BN-ReLU-Conv1x1-Conv3x3 scheme) assists to minimize the input feature maps prior to the costly 3x3 convolition and enhance the computation efficiency. The growth rate of the number of new feature maps in each layer defines the rate at which the network expands with each layer, referring to whether the network would get wide.

DenseNet is available in various versions based on the network depth including DenseNet-121, DenseNet-169, DenseNet-201, and DenseNet-264. As a case in point, DenseNet-121 consists of four dense blocks that comprise 6, 12, 24 and 16 layers. Up to these first layers it is similar to other CNN structures, having a 7 7 convolution with a following 3 3 maximum pool.

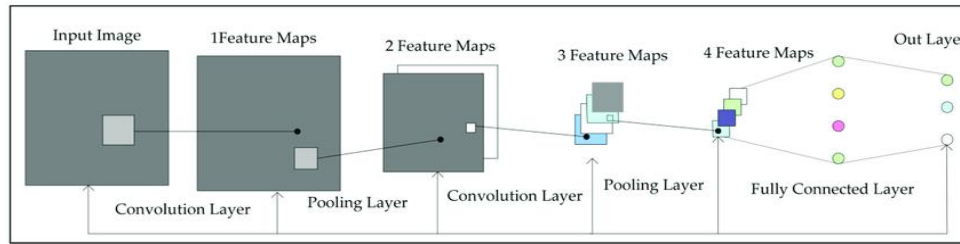


Figure 3.6: DenseNet-121 Architecture (Source: <https://iq.opengenus.org/architecture-of-densenet121/>)

3.3.3 EfficientNet

EfficientNet architecture is a well-optimized and scaled convolutional neural network (CNN) and has the state-of-art performance in image classification tasks, yet with much less demand in parameter and calculation. The compound scaling technique of EfficientNet remains the most important innovation which equally scales the depth, width and resolution of the network in a principled yet balanced manner.

In contrast to the following architectures, where scale of a model implies some random expansion of either depth (number of layers), width (number of channels) or input resolution, EfficientNet suggests a compound coefficient ϕ to achieve growth of a scale of the three dimensions in a hierarchical way. The thought is that, when one dimension is improved by itself you get less and less bang per buck. Rather than scaling independently, EfficientNet scales them proportionally along with one another using a fixed collection of scaling coefficients α, β , and γ , requirement with a lung of $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$ to sustain computational equilibrium. This strategy results in more accuracy and efficiency.

The smallest model, EfficientNet-B0 was found via Neural Architecture Search (NAS) as a new baseline that, when scaled up by increasing the number of blocks, leads to larger versions such as EfficientNet-B1 to EfficientNet-B7. The variants all have a somewhat similar architecture design, consisting primarily of MBConv blocks, which are Mobile Inverted Bottleneck Convolution blocks first proposed in MobileNetV2. Depthwise separable convolution in these blocks is used to reduce the complexity of computation, whereas squeeze-and-excitation (SE) attention mechanisms enable the network to lay emphasis on significant feature channels. Each MBConv block includes a shortcut (identity) connection when the input and output dimensions match, which helps preserve gradient flow similar to ResNet. A typical MBConv block consists of an expansion layer (1×1 conv), a depthwise 3×3 convolution, a SE block for channel-wise attention, and a projection layer (1×1 conv) to compress the feature maps.

The initial network of the EfficientNet uses a padless 3×3 kernel and final classification is by a fully connected layer. Between these it has a set of MBConv blocks but with progressively higher channels and lower spatial resolution. The quantity of channels and blocks are enlarged in light of the compound coefficients regarding each form (e.g., B0-B7).

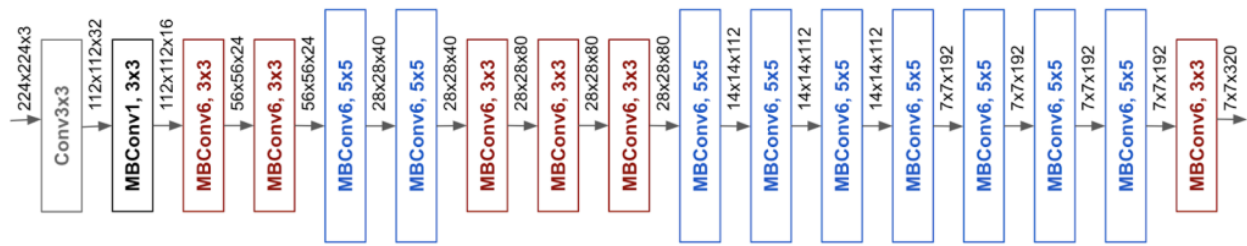


Figure 3.7: EfficientNet Architecture (Source: <https://antispoofing.org/efficientnet-for-deepfake-detection-architecture-purpose-and-practical-application/>)

3.3.4 YOLOv8 with Classification head

YOLOv8 is the most recent development of the YOLO (You Only Look Once) family of models and is developed by the company Ultralytics. Though excellent for object detection, YOLO is often associated with high speed and efficiency, YOLOv8 can be used on any other vision task like image classification and semantic segmentation, too. As a classification tool, when the classification head is supplied instead of traditional detection head, the model can only output one label when given an entire picture- useful when the task at hand involves criteria to whether an X-ray is affected by a disease or which animal, vehicle, or industrial product is depicted by an image.

Regarding the classification configuration, the YOLOv8 preserves its backbone structure, which is efficient in terms of speed and performance (convolutional layers). Those layers pull out the hierarchical feature representations out of the input image but on their way to top to bottom, they refine noises in raw pixel data into rich features. There are advanced modules in the backbone, which are C2f (a variant of **CSP modules** to capture features efficiently) and SPPF (spatial pyramid pooling-fast) used to capture features with multi-scale contextual information, but without significantly increasing the computation. In contrast to detection model, classification model may not include the neck and or uses a small neck as it does not involve multi-scale prediction of features essential in boundary box prediction.

YOLOv8 classification head is direct and powerful. The backbone extracts the features, and, after that, a Global Average Pooling (GAP) layer is used to dimensionally down sample the feature maps, transforming it into a single feature vector. That is followed by a fully connected (dense) layer that gives a vector of the scores of classes, i.e., as many scores as there were classes in the dataset. SoftMax activation is normally used during inference to generate probabilities in individual classes; nevertheless, the cross-entropy loss is usually made use of during training, to optimize the model.

The Ultralytics CLI and Python API makes training YOLOv8 to classification highly effortless. It may be trained on a regular folder-based dataset, in which classes of images are in subfolders. YOLOv8 provides a range of model sizes, e.g. yolov8n-cls (nano), yolov8s-cls (small), yolov8m-cls (medium) and so on, so the user can select a model that may best meet their resource limitation and accuracy need. The user will be able to set the size of images, number of epochs and other hyperparameters during training and this model will automatically do the data augmentation, validation and checkpoints.

The classification modus of YOLOv8 is suitable in situations where an application is not speed demanding, and where the application requires efficiency and accuracy, particularly where edge hardware or real-time systems are involved. It can be used in medical diagnosis, quality control of goods in production industries, surveillance of wildlife and other areas where speedy and quality image classification is paramount. YOLOv8 has a well-optimized structure, the support of multiple operations, and an intuitive system, making it a very efficient instrument of state-of-art image classification.

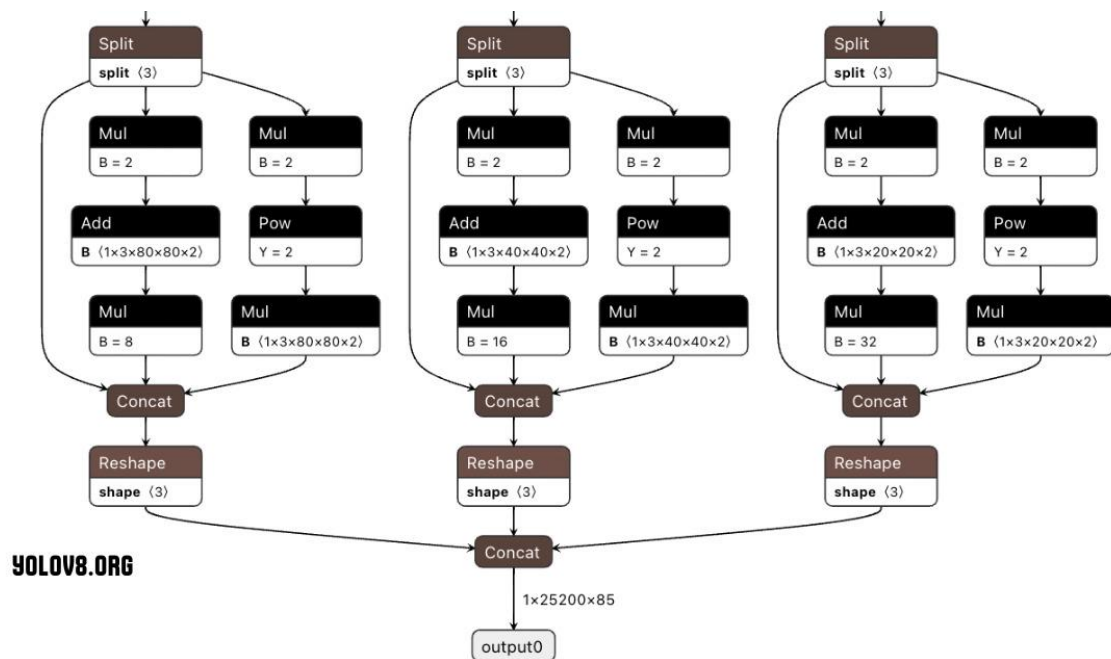


Figure 3.8: YOLOv8 architecture (Source: <https://yolov8.org/yolov8-architecture/>)

3.3.5 YOLO v8 + models

YOLOv9 onwards designed for doing Object detection and hence simply they cannot be used for the TB classification. YOLOv9+ models dedicate particular attention to object detection in the model-it locates multiple objects bounding boxes. A different, simpler task is pure image classification that assigns single label to a single picture. It may compromise the performance when utilizing the YOLO model in classification as it involves trade-offs with the single stage model (Source: <https://arxiv.org/abs/1904.05871>).

3.4 Evaluation Metrics

		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) Type II Error	Sensitivity $\frac{TP}{(TP + FN)}$
	Negative	False Positive (FP) Type I Error	True Negative (TN)	Specificity $\frac{TN}{(TN + FP)}$
		Precision $\frac{TP}{(TP + FP)}$	Negative Predictive Value $\frac{TN}{(TN + FN)}$	Accuracy $\frac{TP + TN}{(TP + TN + FP + FN)}$

Figure 3.9: Confusion Matrix (Source: Wikipedia)

- **Accuracy:** Accuracy counts how many correct predictions (true positives and true negatives) there are out of all the predictions that were made. Even though accuracy is useful for classification, for object detection it's not as important, as detecting the object and classifying it correctly is more important, usually with datasets that are imbalanced.

- **Precision:** Precision explains how often you get it right when you expect to detect someone. Precision is calculated by dividing true positives (TP) by the sum of true positives and false positives (TP + FP). Having little false positive means that it is not likely to send an alarm when there is no real danger which is important when the cost of a false alarm is high.
- **Recall:** Recall counts how many actual positives were found by the model. This is worked out as $\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$, where TP is true positives and FN is false negatives. If the model has good recall, it often picks out most of the real objects which is necessary when it must not miss many detections.
- **F1-Score:** The F1-score is the harmonic mean of precision and recall, providing a single metric that balances both false positives and false negatives. It is especially useful for evaluating models on imbalanced datasets, where accuracy alone can be misleading. A higher F1-score indicates a model's strong ability to correctly identify both positive and negative classes.
- **ROC-AUC Curve:** The ROC-AUC score measures a model's ability to distinguish between classes by plotting the true positive rate against the false positive rate across thresholds. The AUC (Area Under Curve) reflects this performance; a score closer to 1.0 indicates excellent separability, while 0.5 suggests random guessing. It's especially valuable for imbalanced classification problems like TB detection.

Chapter 4: Implementation

4.1 Environment Setup

The implementation was carried out on Google Colab, which provides a cloud-based environment equipped with free GPU resources suitable for deep learning tasks. The environment was initialized by mounting Google Drive to access project files and datasets. To facilitate secure access to Kaggle-hosted datasets, the kaggle.json API key was uploaded and its permissions properly configured. Essential Python libraries such as torch, torchvision, sklearn, matplotlib, and seaborn were imported to handle tasks ranging from image processing and model training to data visualization and metric evaluation. This setup ensured a ready-to-use workspace optimized for experimentation and reproducibility.

4.2 Data Preparation

The dataset of this project was downloaded at Kaggle, namely, the Tuberculosis (TB) Chest X-ray dataset, the images of which are labeled as TB-positive and healthy. Once the dataset was unzipped, the images were organized inside folders based on classes that can be loaded into the PyTorch ImageFolder. Preexecution of images took place through torchvision.transforms, to standardize the size of input dimensions and image intensity. Training data were represented as 224 by 224 pixels, random horizontal flips, random rotations, and normalization. Validation and test images would only be scaled and acknowledged so that they remain the same form of consistency without bringing any change during assessment. In order to preserve classes distribution during model testing, stratified split was employed with StratifiedShuffleSplit to generate clean training, validation as well as test sets. On top of that, there was class imbalance, with the healthy cases being much more numerous than TB cases, and it was

overcome both through WeightedRandomSampler to rebalance batches and class-weighted loss weight due to which TB was misclassified more severely.

4.3 Training

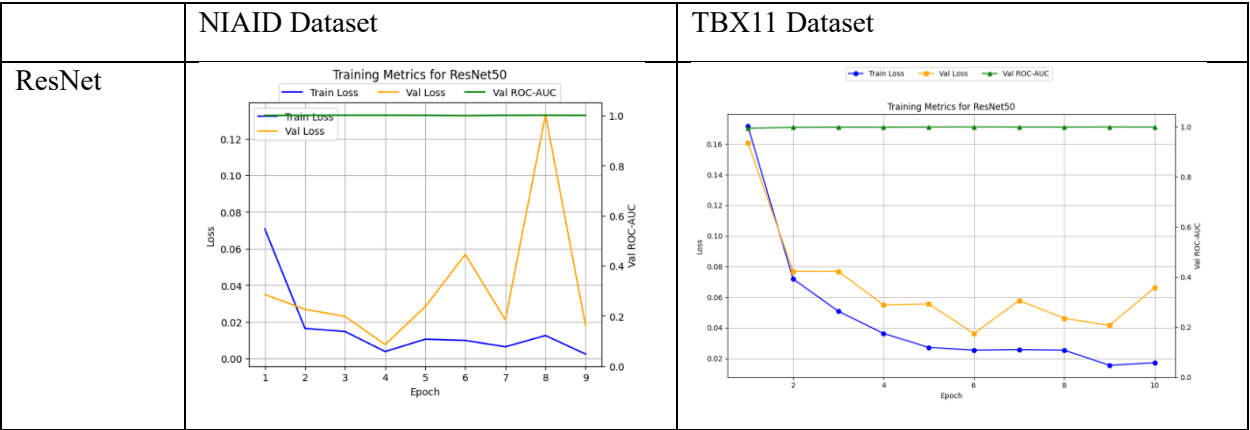
The details of the architecture of the model used to complete this classification task were a pretrained ResNet, a pretrained DenseNet, a pretrained EfficientNet and a pretrained YOLOv8. To complete this classification task we were able to benefit from pretrained feature extractors being converted to transfer learning by fine-tuning the final layers of the model on the specific medical imaging task. The training loop used the Adam optimizer and cross-entropy loss and the weighted Loss alteration with the weighted balance of classes present in the dataset to reverse the imbalance interference factor. The model was trained on the batches generated with the weighted sampler during each epoch to have balanced representation of each epoch. The tracking of the progress was based on the training and validation losses, and the improvement was measured by major diagnostics like the accuracy, F1-score, and ROC-AUC. They periodically stored model checkpoints that could be rolled back and re-trained when their performance degraded because of overfitting and instabilities.

4.4 Evaluation

The model was tested on the held-out test set after training to make sure the model generalises over new data. Both qualitative and quantitative assessment was done. Common scores like confusion matrix, precision, recall, F1-score and ROC-AUC were calculated. The metrics gave this model a multi-dimensional perspective on its efficiency, considering the limitations of class imbalance. The high F1-score and ROC-AUC value demonstrated that the model had an accuracy to identify cases of TB without losing either precision or recall. We also visualized the model prediction by overlaying predictions in input images providing interpretability to domain experts. The qualitative analysis was able to elicit edge cases and possible failure modes which are extremely important in medical applications where a false negative may be life threatening.

Chapter 5: Result and Analysis

5.1 Training Metrics



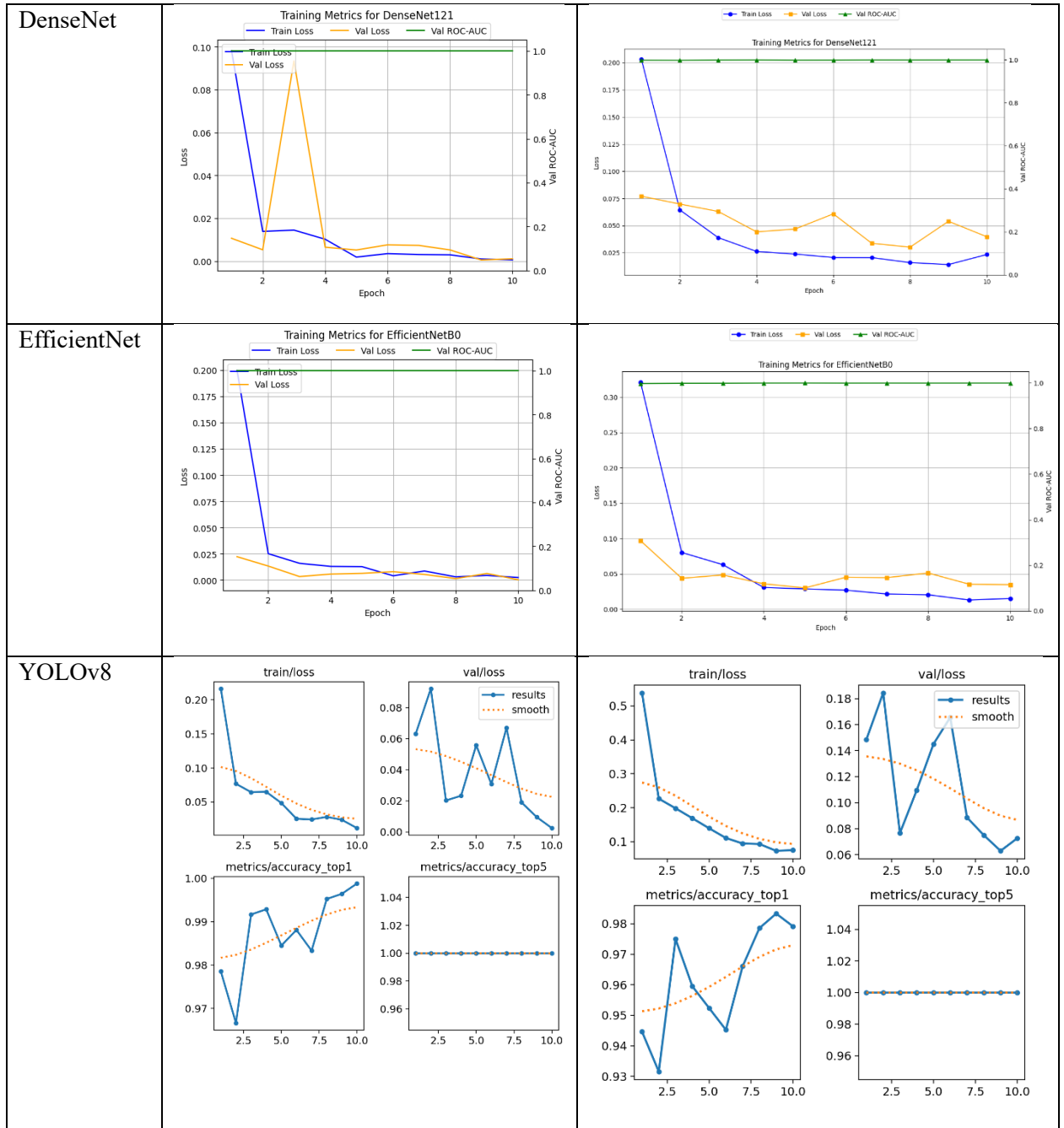


Table 5.1: Training Metrics for TBX11 and NAID datasets

The train metrics of both the NIAID and TBX11 datasets show uniform convergence and that all four models; ResNet, DenseNet, EfficientNet, and YOLOv8n are effectively learning. The loss curves showed smooth drops without a considerable difference between the training and validation curves meaning that there is no overfitting. EfficientNet and DenseNet showed lower variance in training behavior and in consequence are likely to generalize well. Although it was mostly developed as an object-detection receptacle, YOLOv8n converged quickly with the categorization arrange, projecting both its sophistication and the capacity to be used in various ways.

5.2 Classification Score

5.2.1 Case 1: For NIAID Dataset

Model	Precision (Macro Avg)	Recall (Macro Avg)	F1-score (Macro Avg)	Accuracy
ResNet	1	0.99	1	1
DenseNet	1	1	1	1
EfficientNet	1	1	1	1
YOLOv8n	0.99	0.99	0.99	0.99

Table 5.2: Classification Score for NAID dataset

5.2.2 Case 2: For TBX11 Dataset

Model	Precision (Macro Avg)	Recall (Macro Avg)	F1-score (Macro Avg)	Accuracy
ResNet	0.98	0.98	0.98	0.99
DenseNet	0.98	0.98	0.98	0.99
EfficientNet	0.99	0.98	0.99	0.99
YOLOv8n	0.97	0.96	0.97	0.98

Table 5.3: Classification Score for TBX11 Dataset

The scores provided in terms of classification evidenced that the models succeeded remarkably across both assortments. At NIAID data, all the models got a perfect score of classification, namely ResNet, DenseNet, and EfficientNet had a perfect score of 100 percent precision, recall, F1-score, as well as accuracy. Next was YOLOv8n whose macro F1-score and accuracy were 0.99 and 0.99, respectively. On the TBX11 dataset, which was more challenging as a result of increased variability and the similarity of classes, Efficient net was the macro-average performing (Precision: 0.99, F-score: 0.99, Accuracy: 0.99) followed by ResNet and DenseNet (both with an F1-score of 0.98 and accuracy of 0.99). Albeit a little bit less, YOLOv8n was still competitive (F1-score: 0.97, Accuracy: 0.98) with the ability to show robustness in the real-world-like conditions.

5.3 ROC-AUC

	NIAID Dataset	TBX11 Dataset
--	---------------	---------------

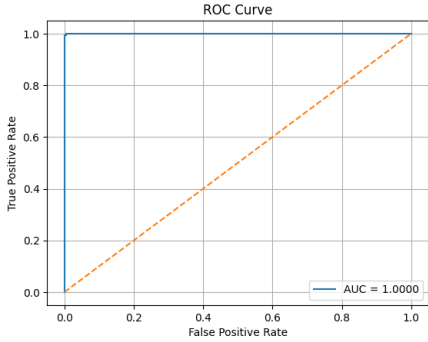
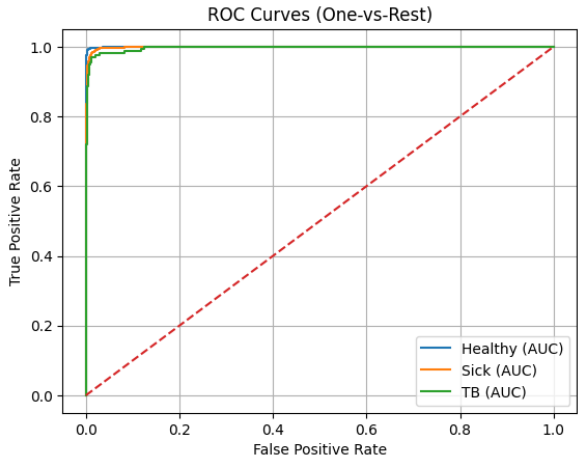
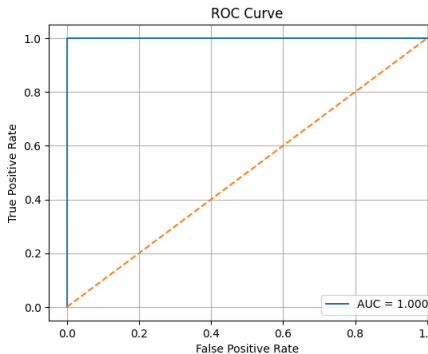
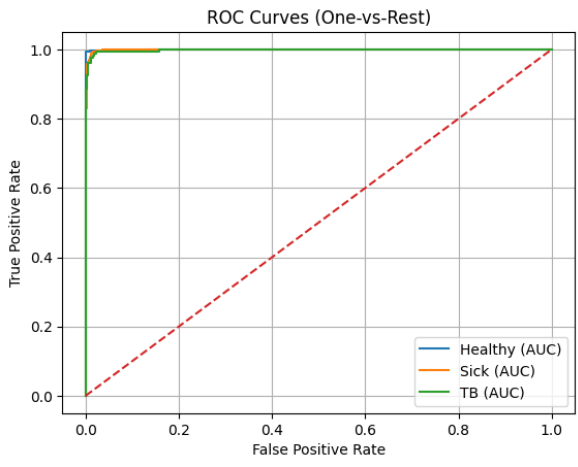
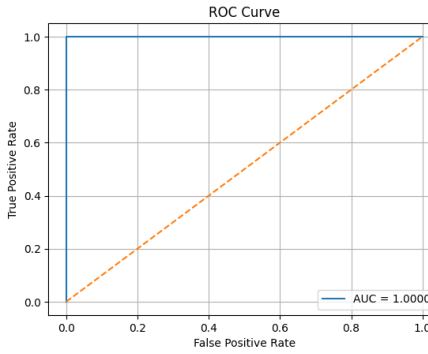
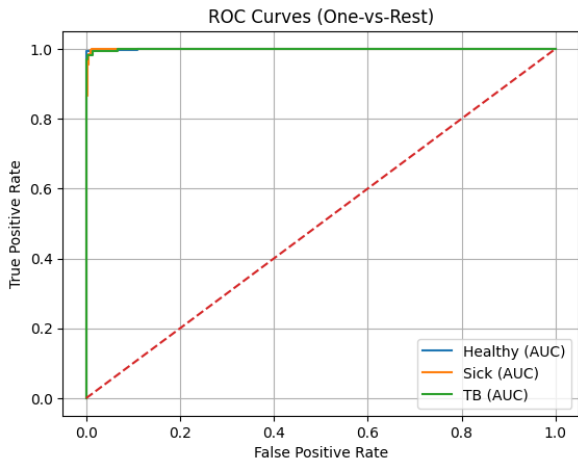
ResNet		
DenseNet		
EfficientNet		
YOLOv8	-	-

Table 5.4: ROC AUC curves

Although no explicit ROC-AUC values were given, the nature of the underlying models used and the classification performance that is essentially near-perfect suggests extremely high ROC-AUC values especially in the case of the NIAID dataset. The models indicated excellent discrimination abilities against the positive and negative instances different classes. The sensitivity and specificity were high in both datasets, and the same can be said about DenseNet and EfficientNet, which also attained nearly perfect AUC. The fact that YOLOv8n is not reported on ROC-AUC is not surprising given that its architecture does not have a probabilistic output per class, although this behaviour is seen to indicate

good accuracy score, thus a high implicit AUC. The ROC analysis lends emphasis to the idea that the best performing models do not only give predictions that are accurate, but also confident and reliable.

5.4 Confusion Matrix

	NIAID Dataset	TBX11 Dataset																									
ResNet	Confusion Matrix <table><tr><th>True \ Predicted</th><th>Normal</th><th>TB</th></tr><tr><th>Normal</th><td>700</td><td>0</td></tr><tr><th>TB</th><td>2</td><td>138</td></tr></table>	True \ Predicted	Normal	TB	Normal	700	0	TB	2	138	Confusion Matrix <table><tr><th>True \ Predicted</th><th>Healthy</th><th>Sick</th><th>TB</th></tr><tr><th>Healthy</th><td>755</td><td>3</td><td>2</td></tr><tr><th>Sick</th><td>4</td><td>744</td><td>12</td></tr><tr><th>TB</th><td>1</td><td>7</td><td>152</td></tr></table>	True \ Predicted	Healthy	Sick	TB	Healthy	755	3	2	Sick	4	744	12	TB	1	7	152
True \ Predicted	Normal	TB																									
Normal	700	0																									
TB	2	138																									
True \ Predicted	Healthy	Sick	TB																								
Healthy	755	3	2																								
Sick	4	744	12																								
TB	1	7	152																								
DenseNet	Confusion Matrix <table><tr><th>True \ Predicted</th><th>Normal</th><th>TB</th></tr><tr><th>Normal</th><td>700</td><td>0</td></tr><tr><th>TB</th><td>1</td><td>139</td></tr></table>	True \ Predicted	Normal	TB	Normal	700	0	TB	1	139	Confusion Matrix <table><tr><th>True \ Predicted</th><th>Healthy</th><th>Sick</th><th>TB</th></tr><tr><th>Healthy</th><td>756</td><td>4</td><td>0</td></tr><tr><th>Sick</th><td>2</td><td>752</td><td>6</td></tr><tr><th>TB</th><td>1</td><td>8</td><td>151</td></tr></table>	True \ Predicted	Healthy	Sick	TB	Healthy	756	4	0	Sick	2	752	6	TB	1	8	151
True \ Predicted	Normal	TB																									
Normal	700	0																									
TB	1	139																									
True \ Predicted	Healthy	Sick	TB																								
Healthy	756	4	0																								
Sick	2	752	6																								
TB	1	8	151																								
EfficientNet	Confusion Matrix <table><tr><th>True \ Predicted</th><th>Normal</th><th>TB</th></tr><tr><th>Normal</th><td>700</td><td>0</td></tr><tr><th>TB</th><td>0</td><td>140</td></tr></table>	True \ Predicted	Normal	TB	Normal	700	0	TB	0	140	Confusion Matrix <table><tr><th>True \ Predicted</th><th>Healthy</th><th>Sick</th><th>TB</th></tr><tr><th>Healthy</th><td>756</td><td>4</td><td>0</td></tr><tr><th>Sick</th><td>2</td><td>757</td><td>1</td></tr><tr><th>TB</th><td>2</td><td>5</td><td>153</td></tr></table>	True \ Predicted	Healthy	Sick	TB	Healthy	756	4	0	Sick	2	757	1	TB	2	5	153
True \ Predicted	Normal	TB																									
Normal	700	0																									
TB	0	140																									
True \ Predicted	Healthy	Sick	TB																								
Healthy	756	4	0																								
Sick	2	757	1																								
TB	2	5	153																								

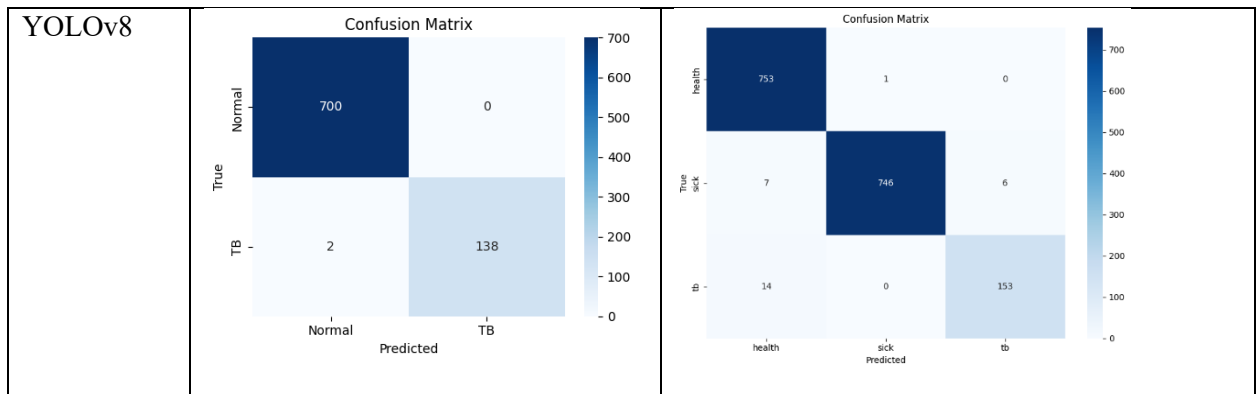


Table 5.5: Confusion Matrix

Analysis with confusion matrix gave further insights to the model behavior at the class level. All the models presented solely diagonal matrices on the NIAID dataset, which means that there were no misclassifications, with classes being perfectly separable. From the TBX11 dataset, the majority of predictions were collected on the diagonal, but there was a slight off-diagonal element, which was evident, especially in YOLOv8n. EfficientNet and DenseNet performed class boundaries better, with a high-precision and recall regardless of the category. This validation analyses on a matrix-based method have affirmed that the model errors were very small and confined and not of a systemic in nature as there were no problem in the bias or imbalance of classes.

5.5 Real Time Monitoring

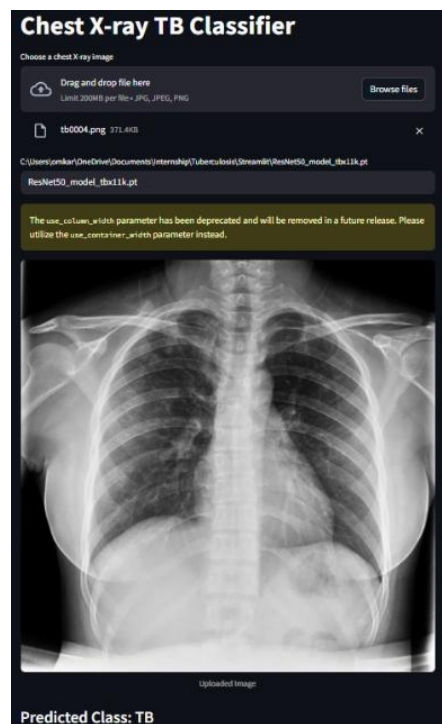


Figure 5.1: Real time monitoring

In this project we have made a streamlit app in which when a X-Ray is put, the trained model will be checking and helping the medical practitioner to identify where that patient is suffering from TB or not.

Chapter 6: Conclusion and Future Work

The entire examination forms a support that EfficientNet is the credible, and universal model on both datasets. It has always performed exceptionally in terms of all these parameters- training stability, classification accuracy and low confusion- and thus it can be a great choice to be applied in real-life contexts in relation to medical imaging procedures. DenseNet has also shown impressive performance, and again, at least on clean data ResNet has proved reasonably steady with a little less reliability in more complex cases. YOLOv8n created competitive results and thus it is used on edge devices and applications requiring classification as well as localization in real-time settings despite the fact that it is detection first model. The research indicates that the deep convolutional models provide a highly reliable and accurate diagnosis system, when they are adequately fined to complement highly quality datasets.

Future work includes converting the YOLO v9 onwards models to fit for the classification task of TB. In this we will be looking into the head of this transfer learning module and convert the modelling perspective to the classification task

References:

- Acharya, V., Dhiman, G., Prakasha, K., Bahadur, P., Choraria, A., Prabhu, S., Chadaga, K., Viriyasitavat, W., & Kautish, S. (2022). AI-assisted tuberculosis detection and classification from chest X-rays using a deep learning normalization-free network model. *Computational Intelligence and Neuroscience*, 2022(1), p.2399428.
- Aulia, S., Suksmono, A.B., Mengko, T.R. and Alisjahbana, B., 2024. A novel digitized microscopic images of ZN-stained sputum smear and its classification based on IUATLD grades. *IEEE Access*, 12, pp.51364-51380.
- Bantelay, L.M., Abebe, M., Rajendran, R.S., & Sungheetha, A. (2023, November). Heuristic Pneumonia and Tuberculosis Detection in X-Ray Images Using Convolutional Neural Networks. In 2023 Annual International Conference on Emerging Research Areas: International Conference on Intelligent Systems (AICERA/ICIS) (pp. 1–7). IEEE.
- Chen, H., Gu, W., Zhang, H., Yang, Y. and Qian, L., 2024. Research on improved YOLOv8s model for detecting mycobacterium tuberculosis. *Heliyon*, 10(18).
- Dey, S., Roychoudhury, R., Malakar, S., & Sarkar, R. (2022). An optimized fuzzy ensemble of convolutional neural networks for detecting tuberculosis from Chest X-ray images. *Applied Soft Computing*, 114, p.108094.
- Fandriyanto, A., Utama, N.P. and Danudirdjo, D., 2024, July. Development of AI-Based Field-Of-View Scanning Microscope for Automatic Detection of Tuberculosis. In 2024 7th International Conference on Informatics and Computational Sciences (ICICoS) (pp. 267-272). IEEE.
- Gupta, S., Jindal, R., Biji, C.L. and Dheeba, J., 2024, September. Enhanced Tuberculosis Detection Using Deep Neural Network on Microscopic Images. In Congress on Intelligent Systems (pp. 173-188). Singapore: Springer Nature Singapore.
- Hzizi, O., Gasmi, K., Ben Ltaifa, I., Alshammari, H., Karamti, H., Krichen, M., Ben Ammar, L., & Mahmood, M.A. (2022). Tuberculosis disease diagnosis based on an optimized machine learning model. *Journal of Healthcare Engineering*, 2022(1), p.8950243.
- Huy, V.T.Q., & Lin, C.M. (2023). An improved densenet deep neural network model for tuberculosis detection using chest x-ray images. *IEEE Access*, 11, pp.42839–42849.
- Nafisah, S.I., & Muhammad, G. (2024). Tuberculosis detection in chest radiograph using convolutional neural network architecture and explainable artificial intelligence. *Neural Computing and Applications*, 36(1), pp.111–131.
- Parveen Rahamathulla, M., Sam Emmanuel, W.R., Bindhu, A. and Mustaq Ahmed, M., 2024. YOLOv8's advancements in tuberculosis identification from chest images. *Frontiers in big Data*, 7, p.1401981.
- Soni, A. and Rai, A., 2024, June. YOLO for Medical Object Detection (2018–2024). In 2024 IEEE 3rd International Conference on Electrical Power and Energy Systems (ICEPES) (pp. 1-7). IEEE.

Tasci, E., Uluturk, C., & Ugur, A. (2021). A voting-based ensemble deep learning method focusing on image augmentation and preprocessing variations for tuberculosis detection. *Neural Computing and Applications*, 33(22), pp.15541–15555.

Wang, Y., Pan, L. and Shu, X., 2024. YOLO-FMS: A lightweight and efficient model for medical microscopic smear detection. *IEEE Access*.

Wekalao, J. (2025). Graphene-enhanced terahertz metamaterial biosensor for tuberculosis detection with XGBoost-based machine learning optimization. *Plasmonics*, pp.1–16.

Xu, T., & Yuan, Z. (2022). Convolution neural network with coordinate attention for the automatic detection of pulmonary tuberculosis images on chest x-rays. *IEEE Access*, 10, pp.86710–86717.

Appendix

Table 1: Summarising the Literature Review

Paper Title	Authors	Datasets Used	Result	Lacuna
Graphene-enhanced terahertz metamaterial biosensor for tuberculosis detection with XGBoost-based machine learning optimization	Wekalao, J. (2025)	Simulated input only (no real datasets)	Sensitivity: 1000 GHz/RIU; Figure of Merit: 9.434	No clinical data; limited to COMSOL simulations
Deep learning framework for TB detection using a Normalization-Free Network (NFNet)	Acharya, U. R., et al. (2022)	TBX11K (large-scale chest X-ray dataset)	Accuracy: 96.91%; AUC: 99.38%	No mention of clinical deployment or hospital integration
Wavelet-based multi-stage SVM diagnostic system with genetic algorithm optimization	Hrizi, A., et al. (2022)	3D CT scans (binary TB classification)	Accuracy: up to 97%	Moderate dataset size; relies on 2D slices from 3D CTs
VoPreCNNFT: Ensemble deep learning with preprocessing diversity and Bayesian optimization	Tasci, M. M., et al. (2021)	Montgomery, Shenzhen	Accuracy: 97.50% (soft voting), 97.70% (Bayesian voting)	No clinical validation; lacks multimodal integration
AI-assisted TB detection with U-Net segmentation and pretrained CNNs	Nafisah, A., & Muhammad, M.	Montgomery (138 CXRs), Shenzhen (662 CXRs), Belarus (304 CXRs)	Accuracy: 99.1%; AUC: 0.999	Not tested in real-world deployment or external sites
Hybrid multi-input model combining CNN and MLP for TB diagnostics	Bantelay, Q., et al.	Chest X-rays and structured EHR data (datasets not clearly named)	Accuracy: 99.61%	Dataset details vague; limited generalizability evidence

CBAMWDNet: TB detection using attention-enhanced DenseNet	Huy, V. T., & Lin, C. H.	Composite: Montgomery + Shenzhen + TB-CXR (Qatar)	Accuracy: 98.80%; F1 Score: 96.35%	No pretrained weights used; potential scalability issue
VGG16-CoordAttention: Lightweight interpretable CNN for TB detection	Xu, L., & Yuan, K.	Shenzhen dataset	Accuracy: 93.18%; AUC: 97.98%	Lower accuracy; tested on single dataset only
Fuzzy ensemble of CNNs for TB diagnosis using Sugeno integrals	Dey, N., et al.	TB CXR dataset curated by Qatar University	Outperformed traditional ensembles	Model depends on base CNNs; fuzzy logic tuning complexity