

City, University of London

MSc Data Science

Project Report

2024

**Comparative Analysis of YOLO Models and
Faster R-CNN for Tuberculosis Detection in
Chest X-rays**

Shreenika Aldur Krishnegowda

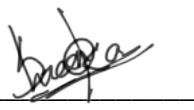
Supervised by: Kevin Ryan

Date of Submission: 9/12/2024

Declaration

By submitting this work, I declare that this work is entirely my own except those parts duly identified and referenced in my submission. It complies with any specified word limits and the requirements and regulations detailed in the assessment instructions and any other relevant programme and module documentation. In submitting this work I acknowledge that I have read and understood the regulations and code regarding academic misconduct, including that relating to plagiarism, as specified in the Programme Handbook. I also acknowledge that this work will be subject to a variety of checks for academic misconduct.

Signed:

A handwritten signature in black ink is written over a solid horizontal line. The signature appears to begin with a capital letter 'B' and end with a lowercase 'a'. The handwriting is cursive and fluid.

Abstract

Tuberculosis has remained a serious global health challenge, demanding for accurate and efficient diagnostic tools for improving patient outcomes. This project focuses on object detection models, specifically YOLO versions(YOLOv5 to YOLOv11) and Faster R-CNN, to detect TB abnormalities in chest X rays using the TBX11K dataset. This dissertation aimed to evaluate the model's performance, explore the evolution of YOLO architectures, and address challenges like class imbalance through data augmentation techniques. YOLOv8 and YOLOv11 performed better than other models achieving the highest precision, recall, and mAP scores, while Faster R-CNN underperformed due to its two-stage architecture's limitations in handling small abnormalities. Hyperparameter tuning further refined YOLOv8 and YOLOv11, demonstrating their potential for robust and accurate TB detection. but, challenges such as computational constraints and false positives highlight the need for further optimization before clinical use.

Keywords: <YOLO Models, Faster R-CNN, Object Detection, Model Comparison, Neural Networks in Healthcare>

Contents

1	Introduction	5
1.1	Project Background	5
1.2	Motivation	6
1.3	Objectives	7
2	Context	8
2.1	Theoretical Background	9
2.1.1	Object Detection in Healthcare	10
2.1.2	YOLO models	11
2.1.3	Faster RCNN and Two-Stage Detectors	13
2.2	Relevance to the Project	15
3	Methods	17
3.1	Dataset	17
3.2	preprocessing	18
3.2.1	scaling of bounding boxes	18
3.3	YOLO	18
3.3.1	preprocessing	20
3.3.2	Training the model	22
3.3.3	Data augmentation for YOLO	23
3.4	Baseline model faster RCNN	23
3.4.1	data augmentation for faster RCNN	26
3.4.2	Model training	27
3.5	Hyperparameter tuning	28
3.6	Testing and Evaluation	28

4 Results	29
4.1 Dataset	29
4.2 Model Testing Results	30
4.3 Testing results after applying Data Augmentation	32
4.3.1 Data Augmentation	32
4.4 Hyperparameter tuning	38
5 Discussion	39
5.1 Relation to objectives	39
5.2 Limitations in Hyperparameter Tuning Approach	40
5.2.1 Future Work	41
6 Evaluation, Reflections, and Conclusions	42
6.1 Evaluation	42
6.2 Reflection	43
6.2.1 Acknowledgment of Existing Work	43
6.2.2 Challenges faced	44
6.2.3 Future work	44
6.2.4 Note for Healthcare Professionals	45
6.3 Conclusion	45
7 Glossary	47
A Graphs	52

Chapter 1

Introduction

1.1 Project Background

Tuberculosis (TB) is a serious global health concern, causing millions of deaths annually. This disease can exist in various forms, primarily categorized as Pulmonary Tuberculosis, which affects the lungs, and Extra pulmonary Tuberculosis, which occurs in other parts of the body such as lymph nodes, bones, and the brain. Among pulmonary TB cases there are different types based on disease progression and appearance in medical imaging.

In this project, the focus is specifically on Active Tuberculosis and Obsolete Pulmonary Tuberculosis (Inactive/Healed TB), as categorized in the TBX11K dataset. chest X-rays are the most important diagnostic tool for detecting tuberculosis (TB) by revealing radiological markers indicative of the disease.

Active Tuberculosis :is characterized by infiltrates, consolidation, cavitary lesions, and nodules, commonly observed in the upper lobes of the lungs. Dynamic progression, such as increasing lesion size or new cavitation formation, is indicative of active disease, which is infectious and requires immediate intervention [Bhalla et al. \(2015\)](#)

Obsolete Pulmonary Tuberculosis on the other hand, represents healed or inactive disease. It appears as fibrotic scars, calcified nodules (Ghon lesions), pleural thickening, and architectural distortion on chest X-rays. These findings are static and non-infectious, reflecting the body's immune response to prior TB infection [Bhalla et al. \(2015\)](#)

Recent advancements in artificial intelligence (AI) and deep learning have transformed medical imaging by enabling automated and precise detection of diseases like TB. Among these technologies, object detection models have contributed more in recognizing abnormalities di-

rectly from medical images. By eliminating the need for manual feature engineering, these models are robust, scalable, and adaptable to complex datasets. The ability of AI to process large volumes of data and identify subtle patterns that might be missed by human experts represents a great difference in medical diagnostics.

One of the most recent and famous object detection framework is YOLO (You Only Look Once), a deep learning model which was designed for real time object detection. YOLO processes an image in a single pass through its network, and simultaneously predicting the bounding boxes and class probabilities for objects in the image. This single stage detection pipeline is a major advantage over traditional two-stage models in which first region proposals are generated and then are classified . where as YOLO's unified approach significantly reduces inference time which makes it suitable for applications where speed is important.

In this project, YOLO models (YOLOv5 to YOLOv11) are examined for their ability to detect TB-related abnormalities in chest X-rays with faster RCNN as a base model. Recent iterations of YOLO, such as YOLOv8 and YOLOv11, shows advanced features like cross scale attention mechanisms, multi scale detection, and enhanced feature fusion. These innovations improve YOLO's capacity to detect small and subtle abnormalities, including early stage TB lesions detection.

1.2 Motivation

The decision to choose YOLO for my project comes from my interests in Neural Networks in my master's studies, where I developed a strong interest in neural computing and its applications in solving real-world problems and the ability of neural networks to learn and adapt to complex patterns fascinated me, especially in the field of medical imaging, where such advancements have the potential to change the diagnostics. This academic passion inspired me to explore new discovery of object detection models, such as YOLO which uses single neural network and Faster R-CNN which can be applied on healthcare challenges.

And this project was also motivated by the observation that modern YOLO architectures, such as YOLOv9 through YOLOv11, remain under explored for TB detection. Where as earlier versions like YOLOv4 and YOLOv5 have been studied, and the latest models introduce advanced features such as cross scale attention mechanisms and neural architecture search and Addressing this research gap offers an opportunity to contribute to the insights into the appli-

cation of AI in healthcare. There are also so many challenges present in TB diagnosis till now which includes class imbalance and small object detection. These complexities requires AI models that are both accurate and efficient. This combination of my academic passion and the ability of YOLO networks to make real world impact motivated me to take this project.

1.3 Objectives

This project's main goal is to compare and contrast the effectiveness of object detection models for TB detection utilizing chest X-rays from the TBX11K dataset.

- **To know which aspects make YOLO more desirable than traditional methods:** Explore specific features and architectural aspects that makes YOLO models more desirable than traditional object detection methods like Faster R-CNN, particularly in the context of medical imaging.
- **What aspects changed the evolution of yolo:** Examine the architectural and functional changes introduced across YOLO versions (YOLOv5 to YOLOv11) and analyze how these improvements effect their performance in detecting TB abnormalities.
- **Does class imbalance effect YOLO models?if so how to rectify it:** Explore how class imbalance in TBX11K dataset affects YOLO models performance and evaluate methods, to remove its impact on minority class detection.
- **Does recent models perform better than past models? in medical imaging:** To check whether recent YOLO architectures outperform traditional models like Faster R-CNN in terms of accuracy, speed, and robustness when applied to medical imaging tasks.

Chapter 2

Context

Tuberculosis (TB) has remained a significant global health challenge, particularly in regions with limited access to healthcare resources. This highly contagious disease, caused by *Mycobacterium tuberculosis*, predominantly affects the lungs but can also target other organs, leading to severe complications if left untreated. According to the World Health Organization, (Bista et al. 2023) states that TB continues to account for millions of new cases and deaths annually, despite substantial advancements in medical science .The critical need for early detection and intervention has brought focus to radiological imaging, especially chest X-rays, as a cost-effective and widely accessible diagnostic tool (M et al. 2024).

relying on manual examination of X-rays often results in diagnostic inaccuracies due to the subtle and diffuse nature of TB lesions. Even experienced radiologists face challenges in distinguishing these features, with reported diagnostic accuracy rates as low as 68.7% (Bista et al. 2023). Such limitations requires the urgent need for automated solutions to improve TB detection accuracy and efficiency. Deep learning, particularly object detection models like YOLO (You Only Look Once) and Faster RCNN, offers a promising avenue to address these issues by improving the advancements in computer vision to analyse medical images more effectively (Diwan et al. 2023).

This dissertation focuses on deep learning for TB detection using the TBX11K dataset which contains annotated chest X-rays. The study aims to conduct a comparative evaluation of YOLO models from YOLOv5 to YOLOv11, alongside Faster RCNN as a baseline, to identify the most effective approach for detecting active TB and absolute pulmonary TB. Furthermore, pre-processing techniques, including data augmentation for minority classes, and hyperparameter tuning are implemented to address challenges such as class imbalance and small object detec-

tion.

2.1 Theoretical Background

Object detection, a cornerstone of modern computer vision, has changed numerous fields, including healthcare. It involves identifying and localizing objects within images, offering precise annotations in the form of bounding boxes. In the context of medical imaging, object detection automates the identification of pathological features, significantly reducing diagnostic errors and improving scalability (M et al. 2024). For diseases like TB, which often present with subtle radiological signs, this capability is especially very important.

Deep learning has emerged as a very important criterian in object detection, with models like YOLO and Faster RCNN leading the field. YOLO, introduced by (Redmon et al. 2015), is a single-stage detector that processes images in real time, making it highly suitable for applications requiring speed and efficiency. Subsequent iterations, such as YOLOv7 and YOLOv8, have introduced architectural innovations like efficient layer aggregation networks and Selection Focal Fusion blocks, enhancing both speed and accuracy (Diwan et al. 2023). On the other hand, Faster RCNN operates as a two-stage detector, generating region proposals in the first stage and classifying objects in the second. While slower than YOLO (Girshick 2015).

The application of these models in healthcare, particularly for TB detection, poses unique challenges. One of the most significant issues is the imbalance in datasets. This imbalance can effect models performance, leading to poor detection rates for minority classes. To mitigate this, techniques such as data augmentation and class weighting have been employed, resulting in improved generalization and detection accuracy (Bista et al. 2023). Additionally, the small size and diffuse nature of TB lesions make them difficult to detect, necessitating advanced techniques like attention mechanisms and feature aggregation, as implemented in YOLOv8 (M et al. 2024).

The TBX11K dataset, used in this study, represents a valuable resource for evaluating object detection models in medical imaging. It includes annotated chest X-rays that enable precise localization of TB lesions. Previous studies utilizing this dataset have demonstrated the effectiveness of YOLO models, particularly in addressing data imbalance through augmentation and hyperparameter tuning (Bista et al. 2023). These insights form the foundation for this dissertation's comparative analysis of YOLO versions and Faster RCNN, aiming to identify the most

effective approach for automated TB detection.

2.1.1 Object Detection in Healthcare

The application of object detection in healthcare has evolved medical imaging by enabling automated, precise, and efficient analysis of complex medical data. This technology relies on computer vision techniques to identify and localize objects of interest within medical images, such as lesions, tumors, or organs, and is impacted in diagnostic radiology. ([Ganatra 2021](#)) stated that recent advancements in deep learning, especially convolutional neural networks (CNNs), have accelerated the development and deployment of object detection models in healthcare, offering solutions to long-standing challenges in manual diagnostics .

Object detection models broadly fall into two categories, two stage detectors like Faster RCNN and single stage detectors like YOLO. Both types have been applied to medical imaging tasks, including tumor segmentation, lesion detection, and organ localization. Two stage detectors, such as Faster RCNN, are known for their high accuracy due to their region proposal mechanism, which isolates regions of interest before classification. As demonstrated by ([Girshick 2015](#)), Faster RCNN has significantly advanced medical imaging tasks, particularly in detecting small and subtle features. but, the computational cost and slower inference speed of these models limit their utility in time sensitive scenarios([Ganatra 2021](#)). Single stage detectors like YOLO have been designed for real time applications. YOLO models process an image in a single forward pass, predicting bounding boxes and class probabilities simultaneously. This approach enhances detection speed and simplifies object detection systems deployment . ([Redmon et al. 2015](#)) first introduced YOLO as a pioneering single-stage detector, and subsequent versions have significantly improved its performance in healthcare applications.

Despite these advancements, there have been many challenges applying object detection to medical imaging. One of the issues is the class imbalance often observed in medical datasets, where certain abnormalities are underrepresented. This imbalance favours predictions on dominant classes, reducing the reliability of the models. Techniques like data augmentation, class weighting, and transfer learning have been employed to mitigate this issue, with promising results reported in TB lesion detection and breast cancer diagnostics ([Ganatra 2021](#)). Additionally, small-object detection continues to be a challenge, as abnormalities in medical images, such as early-stage tumors or subtle lesions, often occupy minimal space and can be overlooked by traditional algorithms.

2.1.2 YOLO models

YOLO (You Only Look Once) is a very good framework for real-time object detection, developed to simplify and accelerate the detection process, YOLO models are known for their outstanding ability to perform object detection in real time while maintaining high accuracy, making them a cornerstone in computer vision applications, including medical imaging.

The journey began with yolov1, which was developed in 2015 , Developed by Joseph Redmon and Ali Farhadi (Redmon et al. 2015), YOLOv1 introduced the concept of treating object detection as a single regression problem. This unified approach allowed YOLO to process images in real-time, achieving speeds of up to 45 frames per second (FPS). but YOLOv1 struggled with detecting small objects and localizing dense clusters due to its grid-based structure. Building on to this, YOLO9000 (YOLOv2) was also developed by Joseph Redmon and Ali Farhadi in 2006 (Redmon & Farhadi 2016)

improved upon YOLOv1 by incorporating anchor boxes, high-resolution classifiers, and batch normalisation. It introduced a novel hierarchical classification system, enabling the detection of over 9,000 object categories, a milestone in scaling object detection, and in 2018 , YOLOv3 was developed and marked a significant advancement by employing multi-scale predictions and a deeper backbone network. These enhancements improved performance, particularly for smaller objects, and introduced logistic regression for class prediction. but, YOLOv3 still had limitations in handling overlapping objects and edge cases, and later YOLOv4 was developed in 2020 which was developed by Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao (Bochkovskiy et al. 2020). With this development object detection entered a new era where yolov4 optimized the training pipeline with features like Cross-Stage Partial (CSP) connections, the Path Aggregation Network (PAN), and "bag of freebies," improving both speed and accuracy. These innovations made YOLOv4 the benchmark for real-time object detection at that time.

The development of YOLOv5 was developed by Glenn Jocher and the Ultralytics team introduced PyTorch supportJocher et al. (2020), making it more accessible to developers. YOLOv5 featured auto-anchor learning, mosaic augmentation, and a lighter architecture, highlighting on user-friendly deployment and rapid training capabilities, even though it was not a direct continuation of Redmon's work, YOLOv5 became widely popular due to its simplicity and efficiency. Following this in 2021 YOLOv6 was created by Meituan Vision AI Department (Ultralytics 2020), focusing on industrial applications. It introduced further optimizations for training effi-

ciency and detection accuracy. And later in 2022, yolov7 was developed by Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao, YOLOv7 was presented in "YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors." (Wang et al. 2022) focused on further optimizing the architecture and introducing trainable "bag-of-freebies" YOLOv7 set new benchmarks in speed and accuracy, particularly excelling in scenarios demanding real-time object detection. yolov8 was developed in 2023 by the ultralytics team. The model's accuracy is assessed using evaluations on both COCO and Roboflow 100 datasets (Solloway & Francesco 2024). YOLOv8 introduced neural architecture search (NAS) techniques for more efficient model design. This model addressed performance bottlenecks in earlier versions and pushed the limits of both accuracy and speed. yolov9, yolov10, yolo11 all these models were developed in 2023. Where YOLOv9 further refined these concepts by combining cutting-edge architecture with broader applications, including domain-specific tasks like medical imaging. YOLOv10 introduced end-to-end training pipelines, and YOLOv11 explored enhanced feature fusion and cross-scale attention mechanisms, cementing YOLO's position as a leading object detection framework.

These yolo models have proven highly effective for detecting TB-related abnormalities in chest X-rays and other medical imaging modalities. The lightweight YOLO models, augmented with multiple receptive fields, has been particularly effective in improving pulmonary TB detection. Guo et al. (2022) proposed a novel architecture named MIP-MY that integrates inverted residual channel attention modules and pyramid pooling to enhance feature extraction. The model demonstrated a significant reduction in missed detections for small pulmonary TB lesions, achieving a remarkable mean Average Precision (mAP) of 95.32% while reducing model parameters by 47%. This lightweight design is particularly suitable for deployment in environments with limited computational resources, such as rural healthcare facilities. Further advancements in TB diagnostics are attributed to YOLOv8, which incorporates good features like Selection Focal Fusion (SFF) blocks and gradient path optimization. (M et al. 2024) employed YOLOv8 on the TBX11K dataset, demonstrating its superiority in detecting small lesions and addressing dataset imbalances through data augmentation techniques. The model achieved improved precision and recall metrics, making it a robust tool for early TB detection. This research highlights YOLOv8's adaptability in managing imbalanced datasets and its potential for deployment in global health programs aimed at eradicating TB. Expanding YOLO's application to diverse diagnostic imaging tasks, Chen et al. (2024) introduced an improved YOLOv8s model

for detecting *Mycobacterium tuberculosis* in sputum smear images. The study addressed challenges related to the morphology and size variability of TB bacilli by incorporating multi-scale feature fusion (MSFF) and an Enhanced Coordinate Convolutional Self-Attention (ECCSA) mechanism. The improved model achieved an mAP of 85.7%, significantly outperforming earlier YOLO iterations and comparable models like SSD and YOLOv3. Despite a trade-off in processing speed, the model's precision underscores its suitability for high-stakes diagnostic tasks.

Yolo models have become increasingly helpful in object detection, evolution from YOLOv1 to YOLOv11 has brought improvements in handling dense and overlapping regions, making them suitable for medical datasets. The later YOLO models which is YOLOv5 to YOLOv11 have shown significant promise in this object detection where Their ability to process high resolution images, adapt to imbalanced datasets using augmentation techniques, and accurately localize small anomalies has made them invaluable tools in medical applications. By integrating advanced techniques like anchor-free detection (YOLOv7) and neural architecture search (YOLOv8).

2.1.3 Faster RCNN and Two-Stage Detectors

The development of two stage object detectors was the most important moment in computer vision, it combined high accuracy with robust localization capabilities. Among these, Faster RCNN was the most innovative model that has shaped the evolution of object detection. faster RCNN was introduced by Ren, Shaoqing, et al [Ren et al. \(2015\)](#), Faster RCNN improved the field by integrating a Region Proposal Network (RPN) to generate candidate regions of interest, which are subsequently classified and refined. This architecture addressed the computational inefficiencies , Faster RCNN eliminated the need for external region proposal methods such as Selective Search, significantly accelerating inference without sacrificing precision.

Two-stage detectors like Faster RCNN excel in tasks which require precise localization and class discrimination. By isolating object regions in the first stage and performs detailed analysis in the second, these models achieve high accuracy even for small and complex objects. As discussed by [\(Girshick 2015\)](#), this approach is particularly advantageous for applications in healthcare, where the detection of subtle abnormalities, such as nodules in chest X-rays, which requires exceptional precision. The adoption of the RPN within Faster RCNN enhanced its ability to focus on regions likely containing objects, streamlining the detection process and

improving its computational efficiency.

One of the defining features of two-stage detectors is their adaptability to complex tasks. Faster RCNN's modular architecture enables the integration of advanced backbones like ResNet and ResNeXt, further enhancing feature extraction and model performance. Despite its advantages, Faster RCNN's reliance on a two-stage process introduces a trade-off between accuracy and speed. The sequential nature of region proposal generation and classification inherently slows down inference, making Faster RCNN less suitable for real time applications compared to single stage detectors like YOLO. (Ganatra 2021) emphasized this limitation, noting that while Faster RCNN achieves superior accuracy in detecting small objects, its computational requirements may hinder its deployment in resource constrained environments or scenarios requiring immediate decision making, such as emergency diagnostics.

In conclusion, Faster RCNN and two-stage detectors contribute so much of modern object detection, achieving a balance of accuracy and flexibility . on the other hand their slower inference speeds may limit real time applications, their robustness and precision make them highly valuable in domains such as healthcare, where the stakes for accuracy are exceptionally high. With advancements in computational efficiency and feature extraction techniques, two-stage detectors like Faster RCNN continue to play a vital role in the evolving landscape of object detection.

Advances in deep learning and computer vision have significantly impacted healthcare, particularly in automating disease diagnosis through medical imaging. Chest X-rays, widely used for TB detection, have been enhanced by AI-driven models that address long-standing limitations of manual radiological assessments. Recent research highlights the use of YOLO models, such as YOLOv7 and YOLOv8, for detecting TB lesions in chest X-rays, leveraging their speed and accuracy for real-time analysis (Bista et al. 2023, M et al. 2024).

(Bista et al. 2023) demonstrated the efficacy of YOLOv7 in detecting active TB and absolute pulmonary TB using the TBX11K dataset. Their study incorporated data augmentation techniques and class weighting to mitigate data imbalance, achieving a mean Average Precision (mAP) of 0.587. Similarly, (M et al. 2024) employed YOLOv8, integrating Selection Focal Fusion blocks and attention mechanisms to enhance small-object detection, which is critical for identifying subtle TB lesions. Their results showcased significant improvements in precision and recall, addressing key challenges in TB diagnostics.

Comparative studies have also examined the performance of YOLO models relative to tradi-

tional detectors like Faster RCNN. (Diwan et al. 2023) noted that while Faster RCNN excels in accuracy, its inference speed lags behind YOLO's single-shot architecture. This trade-off highlights the importance of balancing speed and precision in clinical applications where timely diagnosis is paramount.

2.2 Relevance to the Project

Despite significant progress in applying deep learning-based object detection models to tuberculosis (TB) diagnostics, critical gaps remain in the current research landscape. While models such as YOLO and Faster RCNN have demonstrated strong performance in detecting abnormalities in chest X-rays, key challenges persist, particularly in addressing small-object detection, class imbalance, and model interpretability. and the most recent YOLO architectures, including YOLOv9 through YOLOv11, have not been evaluated more for their applicability to TB detection, presenting an opportunity to explore uncharted areas of research.

challenge lies in the detection of small TB lesions, which are often diffuse, subtle, and occupy minimal space in medical images. Earlier YOLO models, such as YOLOv4, struggled with small-object detection due to limitations in feature extraction capabilities (Guo et al. 2022). Enhancements introduced in YOLOv8 and its variants, including multi-scale feature fusion and advanced attention mechanisms like Enhanced Coordinate Convolutional Self-Attention (ECCSA), have improved the detection of small lesions. (Chen et al. 2024) demonstrated that an improved YOLOv8s model could effectively identify *Mycobacterium tuberculosis* in sputum smear images by integrating multi-scale feature refinement. However, while these advancements address specific limitations, their generalizability to diverse clinical datasets remains underexplored.

Handling class imbalance in datasets such as TBX11K is another critical issue. Medical datasets often have an unequal distribution of positive and negative cases or varying lesion types, skewing model predictions toward dominant classes. Techniques such as data augmentation and class weighting have been integrated into models like YOLOv8 to mitigate these biases, showing improved generalization across patient populations (M et al. 2024). However, the effectiveness of these methods in noisy or heterogeneous datasets, especially those involving pulmonary TB, requires further investigation.

Interpretability also remains a significant barrier to the adoption of AI-driven TB detec-

tion models. Deep learning models, particularly object detection architectures, often function as "black box" systems, making it difficult for clinicians to trust their predictions. Additionally, while lightweight YOLO architectures like YOLOv4 have proven suitable for resource-constrained settings, they often involve a trade-off between computational efficiency and detection accuracy. Conversely, more advanced models like YOLOv8 offer superior performance at the expense of increased computational demands, limiting their applicability in low-resource environments without hardware acceleration (Guo et al. 2022). This trade-off underscores the need for adaptable models that can balance efficiency and precision, especially for large-scale screening programs.

there is the lack of comparative analysis of the latest YOLO architectures, such as YOLOv9, YOLOv10, and YOLOv11, for TB detection. Existing research primarily focuses on earlier versions like YOLOv4 and YOLOv8, with no studies systematically evaluating how the most recent YOLO models can advance TB diagnostics. This dissertation fills this void by evaluating YOLO models from YOLOv5 to YOLOv11, alongside Faster RCNN, to assess their efficacy in detecting active and absolute pulmonary TB. By exploring these latest architectures, this research offers a novel contribution to the field, addressing unexplored opportunities in object detection for TB diagnosis.

Chapter 3

Methods

3.1 Dataset

Description of TBX11k dataset

The TBX11K dataset is taken from Kaggle ([Shams 2020](#)), serves as the foundation for this project and is a valuable resource for advancing tuberculosis (TB) detection using deep learning models. This dataset comprises 11,200 chest X-ray images, each annotated with bounding box information to highlight TB-affected regions. All the images are standardized to a resolution of 512x512 pixels, ensuring uniformity and compatibility with deep learning frameworks.

In this dataset, data was aggregated from four smaller datasets: DA, DB, Montgomery, and Shenzhen X-ray datasets, which contain 156, 150, 138, and 662 images, respectively. This aggregation makes the TBX11K dataset one of the largest publicly available collections for TB detection, having a diverse range of imaging conditions, patient demographics. Due to its origin as part of an online competition for computer-aided TB diagnosis, the ground truth for the testing set was not released. And this project excludes certain images, such as those in the test and extra categories.

After filtering the dataset, the dataset used for this study contains 8,452 images, distributed across several categories. Among these, 3,813 images belong to the "healthy" class, while 3,839 images are categorized as "sick." Of the sick images, 800 are annotated as TB-positive, which has TB-related abnormalities. The TB-positive images are further divided into active tuberculosis and obsolete pulmonary tuberculosis categories. Specifically, there are 972 images of active TB and 239 images of obsolete TB in the dataset. a small subset of images is annotated

with both labels, resulting in a total of 800 unique TB images. These overlaps highlight the complexity of TB diagnosis, where multiple conditions are coexisting in a single image.

The dataset has some challenges, including class imbalance. For example, the "healthy" and "sick" categories are balanced, but within the TB-positive subset, active TB images significantly outnumber obsolete pulmonary TB images. And, TB lesions often appear subtle or diffuse, occupying minimal portions of the X-ray. This complexity needs high-precision detection methods capable of accurately identifying small, overlapping regions of interest.

From this filtered TBX11K dataset, this project aims to address these challenges, advancing the development and evaluation of automated TB detection models that are robust, scalable.

3.2 preprocessing

3.2.1 scaling of bounding boxes

Scaling is one of the important steps because it ensures that the bounding box accurately align with the objects in the resized images. To ensure consistency across the dataset, all images were resized to a standard dimension of `new_width × new_height`. Since this resizing alters the scale of the images, the bounding box coordinates provided in the XML annotations are scaled proportionally using the following formula:

$$x_{\text{scaled}} = x_{\text{original}} \times \frac{\text{new_width}}{\text{original_width}}, \quad y_{\text{scaled}} = y_{\text{original}} \times \frac{\text{new_height}}{\text{original_height}}$$

This process ensures that bounding boxes accurately align with the objects in the resized images.

3.3 YOLO

YOLO, stands for "You Only Look Once," is a object detection framework introduced by (Redmon et al. 2015). Unlike other traditional object detection models that rely on a two stage process generating region proposals. YOLO operates as a single stage detector. It treats object detection as a single regression problem, simultaneously predicting the bounding boxes and class probabilities for multiple objects in an image in one forward pass through the network.

YOLO Detection Formula: YOLO divides the image into a grid of $S \times S$ cells and predicts

bounding boxes and class probabilities of each cell. The prediction formula can be described as:

$$\hat{y} = (p, b_x, b_y, b_w, b_h, c_1, c_2, \dots, c_C)$$

Where:

- p : The confidence score for the presence of an object in a grid cell.
- (b_x, b_y) : The normalized center coordinates of the bounding box relative to the grid cell.
- (b_w, b_h) : The width and height of the bounding box, normalized by the image dimensions.
- c_1, c_2, \dots, c_C : The class probabilities for C classes.

Each grid cell outputs:

$$p = P(\text{object}) \times \text{IOU}(\text{predicted box}, \text{ground truth})$$

Where:

- $P(\text{object})$: The probability that an object exists in the cell.
- $\text{IOU}(\text{predicted box}, \text{ground truth})$: The Intersection over Union, which measures the overlap between the predicted box and the ground truth box.

The bounding box parameters (b_x, b_y, b_w, b_h) are normalized to ensure they fit within the image dimensions. Class probabilities for each potential class in the grid cell are also outputted.

The model's ability to process an image just once to make all predictions. This approach makes YOLO incredibly fast compared to other two stage models like Faster RCNN, making it suitable for real-time applications. The key features of YOLO are Speed, Global Context and Unifies Architecture where YOLO processes images in real-time, making it ideal for applications requiring rapid detection. It analyzes the entire image in one go, helping it better understand contextual relationships between objects. Combines feature extraction, object detection, and classification in a single network.

YOLOv5 marked a significant development from earlier versions with improvements in usability, scalability, and performance. Comparing it to newer models like YOLOv6

through YOLOv11 provides insights into how the YOLO family has evolved to address challenges such as better small-object detection, handling of class imbalance, and improved computational efficiency.

YOLOv5 is widely used in industry due to its ease of deployment, while later versions like YOLOv8 and YOLOv11 introduce cutting edge features like advanced attention mechanisms, gradient path optimization, and neural architecture search (NAS). By comparing these models, it will be easier to determine which iteration best suits for specific application of TB detection. Earlier versions of YOLO like YOLOv3 and YOLOv4 have been extensively studied and applied in medical imaging. However, YOLOv8 and beyond are more recent and less explored in the context of TB detection, making this comparison a meaningful one. Each version from YOLOv5 to YOLOv11 incorporates incremental improvements, allowing to analyze how these enhancements impact performance in TB detection, particularly in handling challenges like small lesions and imbalanced datasets. While earlier YOLO versions and Faster RCNN are well-documented in medical imaging, there is limited comparative research on YOLOv5 through YOLOv11 for TB detection. Your project addresses this gap, contributing new knowledge to the field.

3.3.1 preprocessing

converting annotations into yolo format: YOLO models have their own standardized format for annotations, which differs from other object detection frameworks. The required YOLO annotation format is:

class_id x_{center} y_{center} width height

Where:

- x_{center} and y_{center} : The normalized coordinates of the bounding box center relative to the image dimensions.
- width and height: The normalized width and height of the bounding box relative to the image dimensions.
- class_id: A numeric identifier representing the object category (e.g., 0 for *Active Tuberculosis* and 1 for *Obsolete Pulmonary Tuberculosis*).

Converting to this format ensures that YOLO can correctly interpret the bounding boxes during training.

Normalization for Scale Invariance:

In the YOLO format:

All bounding box coordinates ($x_{\text{center}}, y_{\text{center}}, \text{width}, \text{height}$) are normalized to a range of 0 to 1 by dividing by the image dimensions (width, height).

Normalization ensures that the model is invariant to the absolute pixel dimensions of the images, enabling it to handle images of varying sizes effectively during training.

Example:

Original bounding box:

$$(x_{\min} = 100, y_{\min} = 50, x_{\max} = 200, y_{\max} = 150)$$

for a 400×300 image.

YOLO Format:

$$x_{\text{center}} = \frac{x_{\min} + x_{\max}}{2 \cdot \text{image_width}} = \frac{100 + 200}{2 \cdot 400} = 0.375$$

$$y_{\text{center}} = \frac{y_{\min} + y_{\max}}{2 \cdot \text{image_height}} = \frac{50 + 150}{2 \cdot 300} = 0.333$$

$$\text{width} = \frac{x_{\max} - x_{\min}}{\text{image_width}} = \frac{200 - 100}{400} = 0.25$$

$$\text{height} = \frac{y_{\max} - y_{\min}}{\text{image_height}} = \frac{150 - 50}{300} = 0.333$$

This process converts bounding boxes from XML to YOLO format and saves YOLO annotations to .txt files.and these YOLO annotations in .txt files will be ready for model training.

Since sick and healthy images in the TBX11k dataset does not have any bounding boxes, but yolo requires .txt file for every image, even if no objects are present.So creating an empty .txt files ensures YOLO interprets these images as background during training.so creating an empty .txt file for sick and healthy images are necessary. These empty .txt files are essential for

YOLO to correctly interpret such images as containing no objects during training. By providing empty annotations, YOLO learns to classify these images as background, improving its ability to distinguish between images with and without TB abnormalities.

Dataset Preparation for YOLO Training: The dataset is split into training (80%), validation (10%), and test (10%) sets using a stratified approach. Images and their corresponding YOLO .txt annotation files are copied to respective directories. Missing annotations for background images ('sick' and 'health') are handled by creating empty .txt files. A dataset.yaml file is generated to define the dataset structure, including the number of classes and their names, for YOLO training.

3.3.2 Training the model

The training process is conducted on YOLO models ranging from YOLOv5 to YOLOv11 to evaluate their performance on tuberculosis detection using the TBX11K dataset. Each model is fine-tuned to detect different classes.

For YOLOv5, YOLOv6, YOLOv8, YOLOv9, YOLOv10, and YOLOv11, the Ultralytics ([Ultralytics 2024](#)) library is used for training. These models are initialized with pre-trained weights from the COCO dataset and fine-tuned on the TBX11K dataset. A consistent training configuration was applied across these models for a fair comparison. unlike the other models, YOLOv7 is not part of the Ultralytics library. The implementation is sourced from an open-source GitHub repository ([WongKinYiu 2024](#)). this approach is called as transfer learning.

transfer learning: transfer learning is a machine learning technique in which a model trained on one task is reused or fine tuned for a different but related task. Instead of training a model from scratch transfer learning starts with a pre trained model, showing the knowledge it has already learned from a large dataset. and then fine tuning is done where the pre-trained weights are adopted to tbx11k dataset,enabling the yolo model to detect the abnormalities related to tuberculosis in x ray images.

A consistent training configuration is applied across these models for a fair comparison.Epochs=50,Image Size= 512 pixels, Batch Size= 16, Learning Rate: 0.001, Optimizer: SGD with momentum 0.9.

3.3.3 Data augmentation for YOLO

Initially, YOLO models (YOLOv5 to YOLOv11) are trained on the TBX11K dataset to detect tuberculosis-related abnormalities. After the training, the results revealed a significant issue with class imbalance. The dataset contained 972 images labeled as Active Tuberculosis and only 239 images labeled as Obsolete Pulmonary Tuberculosis. This class imbalance caused the models to perform poorly on the minority class that is Obsolete Pulmonary Tuberculosis, with low precision and recall scores, indicating that the models are biased toward the majority class. To overcome this challenge, data augmentation techniques are applied to the minority class. Augmentation is performed using Albumentations, a robust library for image augmentation with bounding box support. The transformations include HorizontalFlip (randomly flips the image horizontally with a 50% probability), RandomBrightnessContrast (randomly adjusts the brightness and contrast of the image), ShiftScaleRotate (applies random shifts, scaling, and rotations up to 15°) and Rotate (rotates the image within a range of ±25°).

The bounding boxes which are defined in YOLO format are adjusted during augmentation to maintain accuracy. Each image is augmented 5 times using the defined pipeline resulting in a more balanced dataset. This combined dataset (original + augmented images) increased class diversity, leading to a more balanced training process. This resulted in better detection metrics for the minority class and improved overall mAP. After augmenting the dataset, the YOLO models were re-trained with the combined original and augmented data. The re-trained models showed good improvements, particularly in the detection performance of the minority class. Metrics such as precision, recall, and mAP@50 showed notable gains, validating the effectiveness of data augmentation in addressing class imbalance.

3.4 Baseline model faster RCNN

Faster R-CNN (Region-based Convolutional Neural Network) is a two-stage object detection framework that achieves high accuracy for detecting and localizing objects in images. Faster R-CNN is an improvement over earlier versions which are R-CNN and Fast R-CNN, focusing on improving speed and efficiency without sacrificing accuracy. It introduces a Region Proposal Network (RPN) that generates candidate object proposals directly from the feature maps, removing the need for an external region proposal algorithm.

Faster R-CNN operates in a two-stage pipeline:

1. Region Proposal Generation (via the RPN) The Region Proposal Network (RPN) identifies potential regions of interest (RoIs) that are likely to contain objects:

- Each anchor (a predefined box of fixed size) is classified as foreground (contains an object) or background (does not contain an object).
- The RPN refines the coordinates of these anchors to improve localization.

The RPN loss is calculated as:

$$L_{\text{RPN}} = L_{\text{cls}} + \lambda \cdot L_{\text{reg}}$$

Where:

- L_{cls} : Binary classification loss (foreground vs. background).
- L_{reg} : Regression loss for anchor box refinement.
- λ : A balancing parameter.

Object Detection (via the Detection Head) The refined RoIs are passed through ROI Pooling to extract fixed-size feature maps. These features are then fed into the detection head for:

- **Object classification:** Identifying the object category.
- **Bounding box regression:** Refining the bounding box coordinates further.

The combined detection loss is:

$$L_{\text{det}} = L_{\text{cls}} + \alpha \cdot L_{\text{reg}}$$

Where:

- L_{cls} : Multi-class classification loss.
- L_{reg} : Bounding box regression loss.
- α : A weighting factor.

Final Model Loss

The final model loss combines both stages:

$$L = L_{\text{RPN}} + L_{\text{det}}$$

Region Proposal Network (RPN):

The RPN eliminates the need for external region proposal algorithms, allowing for efficient end to end training. This is crucial for medical datasets like TBX11K, where accurate region proposals improve detection quality.

In object detection, Faster R-CNN has established itself as a highly accurate model, especially in tasks requiring precise localization. This accuracy is primarily achieved through its two-stage detection pipeline, which generates region proposals in the first stage and classifies these proposals in the second. Having high-capacity backbones, such as ResNet-50, Faster R-CNN employs transfer learning to adapt these generalized features to specialized domains like medical imaging. This capability not only accelerates convergence but also enhances the model's performance on medical datasets, such as TBX11K, which are distinct from general-purpose datasets.

Given its robust two-stage architecture, Faster R-CNN excels in tasks requiring accurate bounding box localization. This makes it a strong baseline for high-accuracy object detection, providing a benchmark against which YOLO models can be evaluated.

The comparison between Faster R-CNN and YOLO highlights, particularly between accuracy and speed. Faster R-CNN is renowned for its high accuracy, making it suitable for scenarios where precision takes more importance over inference speed. However, its computational complexity and slower processing times can limit its use in real-time applications or large-scale screening programs. On the other hand, YOLO's single stage architecture is optimized for real time performance, allowing it to process images significantly faster than Faster R-CNN. While this speed advantage is valuable for large-scale or time-sensitive applications, it often comes with compromises in accuracy. This comparison thus serves to evaluate how well YOLO can balance these competing priorities in the context of medical imaging.

Another crucial aspect of this comparison is how well each model generalizes to medical datasets. Medical imaging datasets like TBX11K differ significantly from general-purpose datasets such as COCO, often presenting unique challenges, including imbalanced class distri-

butions and the presence of small or subtle abnormalities. Faster R-CNN, with its robust region proposal mechanism, is particularly used for detecting these subtle features. This makes it an excellent standard for assessing YOLO's ability to adapt to the nuances of medical datasets. By examining how both models handle the complexities of datasets like TBX11K, this comparison sheds light on their respective strengths and limitations in real-world medical applications.

The evaluation also explores the models efficiency in detecting and localizing small abnormalities, such as TB lesions, which are often diffuse and occupy minimal space within X-rays. Faster R-CNN's two-stage detection process ensures precise localization of such small features, providing a benchmark to gauge YOLO's performance. Additionally, the comparison investigates whether YOLO's speed advantage makes it a viable alternative for real-time applications, such as large-scale TB screening, without sacrificing the critical accuracy needed for medical diagnoses.

this comparison provides insights into their suitability for medical imaging tasks. Faster R-CNN's precision and accuracy make it a dependable baseline, while YOLO's real-time capabilities highlight its potential for rapid, large-scale deployment. The results of this comparison are essential for determining which model is better suited for the specific challenges of TB detection, balancing the need for accuracy with the demand for efficiency in medical field.

3.4.1 data augmentation for faster RCNN

The data augmentation process is implemented to address the challenges of class imbalance and to improve the ability of the model. To ensure consistency and comparability with the YOLO models, the same augmentation pipeline is applied for both Faster R-CNN and YOLO. The Albumentations library is used for its robust support of image transformations and precise adjustments for bounding boxes. The augmentation pipeline included a variety of transformations designed to simulate real world variations and increase dataset diversity. These transformations involved horizontal flipping, applied with a 50% probability to mimic changes in orientation, as well as brightness and contrast adjustments to emulate different imaging conditions. Additional transformations such as shifting, scaling, and rotating the images is also employed to introduce spatial variations, with shifts slightly displacing the bounding boxes, scaling to simulate zooming in and out, and rotations applied within a range of $\pm 15^\circ$. Furthermore, an additional rotation of up to $\pm 25^\circ$ was included to further diversify the augmented dataset.

Each image in the dataset is augmented five times just like in the yolo augmentation tech-

nique, with the augmented images and their corresponding annotations stored in dedicated directories. This consistent augmentation process across YOLO and Faster R-CNN ensures a fair comparison between the models. To verify the how right and diversity of the augmented data, a random subset of augmented images was visualized along with their bounding boxes. This step confirms that the transformations are applied correctly and that the annotations remains accurate.

3.4.2 Model training

To ensure compatibility with the Faster R-CNN framework, the TBX11K dataset is converted into the COCO format, which is widely supported by modern object detection models. The bounding boxes are converted to the COCO format (xmin, ymin, width, height) to align with Faster R-CNN's requirements. The COCO-style annotations are saved in a JSON file

Once the dataset is prepared, it is split into training (80%) and testing (20%) subsets using PyTorch's DataLoader. The augmented data is included in the training set to increase its diversity and address the class imbalance issue, while the validation and test sets remained unchanged to maintain unbiased evaluation metrics.

The Faster R-CNN model, pre-trained on the COCO dataset is selected for training. It utilizes a ResNet-50-FPN backbone, a robust architecture known for its ability to extract rich features from input images. The model is customized for this project by replacing its pre-trained classifier head with a new one tailored to detect the three classes in the TBX11K dataset

The training process was configured for optimal performance. The optimizer used is Stochastic Gradient Descent (SGD), with a learning rate of 0.001, momentum of 0.9, and a weight decay of 0.0005. A step decay scheduler was employed to reduce the learning rate by a factor of 0.1 every three epochs, ensuring steady convergence. The training was carried out for 50 epochs, allowing the model sufficient time to learn from the data.

During training, each epoch involved passing the images and their corresponding annotations to the model. Losses were computed for various tasks, including classification, bounding box regression, and region proposals. Gradients were calculated, and the optimizer updated the model weights to minimize the overall loss. training faster RCNN models took more time when comapred to training yolo models

After completing training, the model weights are saved preserving the best-performing version of the model for later evaluation and comparison.

3.5 Hyperparameter tuning

Hyperparameter tuning is performed on the YOLOv8 and YOLOv11s models, which were selected based on their superior performance during initial training and also they are the latest models. The tuning process was done to optimize the model's performance and gain deeper insights into their behavior. It was conducted using the Ultralytics Tuner framework ([Ultralytics n.d.](#)) , which systematically explores a range of hyperparameters across multiple iterations to maximize the model's fitness. The hyperparameters tuned included the learning rate (initial lr0 and final lrf), momentum, weight decay, and warmup parameters such as the number of epochs (warmup epochs) and momentum (warmup momentum) and , loss-specific parameters, including those for box loss (box), classification loss (cls), and distribution focal loss scaling (dfl), were optimized. The tuning process is carried out on an augmented dataset over 10 iterations, where each iteration represented a complete cycle of training and evaluation with a specific set of hyperparameters. This approach allowed for systematic exploration of the hyperparameter space, ensuring a thorough optimization of the models.

3.6 Testing and Evaluation

In testing and evaluation part, the trained models are assessed on the reserved test set, which comprised 10% of the TBX11K dataset. This test set is kept untouched during training and augmentation to ensure an unbiased evaluation of the model's performance. The goal is to evaluate the models ability to accurately detect tuberculosis related abnormalities, specifically distinguishing between "Active Tuberculosis," "Obsolete Pulmonary Tuberculosis," and background categories ("Sick" and "Healthy"). Key metrics, including precision, recall, mAP50, and mAP50-95, are used to measure the models' detection accuracy and effectiveness.

During testing, the models predicted bounding boxes, class labels, and confidence scores for each image, which were compared against the ground truth annotations using Intersection over Union (IoU) as the benchmark. This rigorous evaluation process not only highlighted the strengths and weaknesses of each model but also helps in comparing YOLO's real time performance with the precision-focused approach of Faster R-CNN, providing valuable insights into their applicability for tuberculosis detection in medical imaging.

Chapter 4

Results

This chapter shows the results obtained from applying the methods given in the Chapter 3. to evaluate the performance of yolo models from yolov5 to yolov10 and Faster RCNN for tuberculosis detection using the TBX11K dataset .The findings include model testing performance, the impact of data augmentation , evaluation metrics. Key metrics such as precision, recall, and mean Average Precision (mAP) are shown, along with visualizations of model predictions.

4.1 Dataset

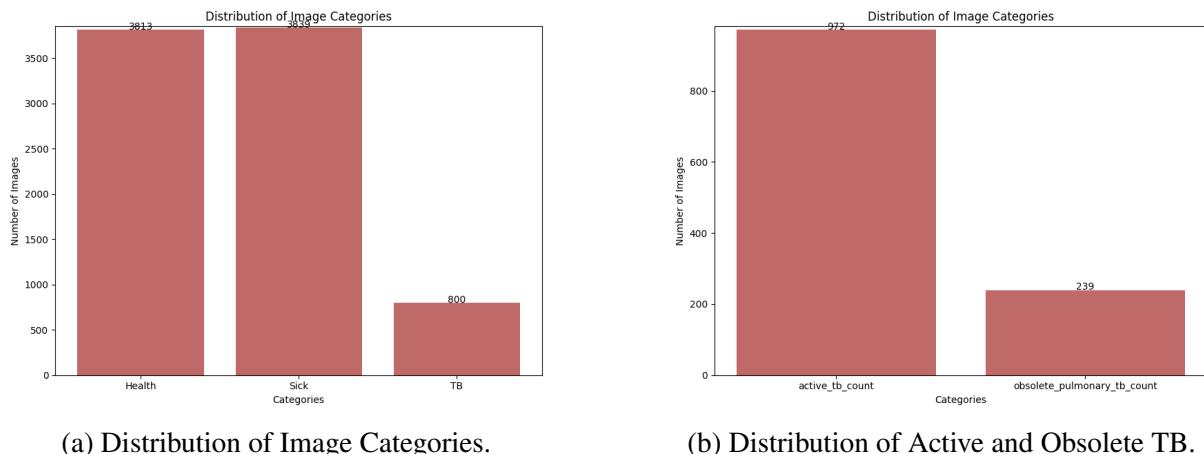


Figure 1: Comparison of Distributions: (a) Image Categories (b) Active vs. Obsolete TB.

As shown in [Figure 1](#), the graphs provide a detailed overview of the data distributions of TBX11K dataset ,the graphs showed the different class of the dataset and the distribution of active tuberculosis and obsolete pulmonary tuberculosis images in tuberculosis class.

4.2 Model Testing Results

Models	Precision	Recall	mAP@50	mAP@50-95
YOLOv5	0.7745	0.3316	0.3125	0.1350
YOLOv6	0.7327	0.2017	0.1974	0.0915
YOLOv7	0.6421	0.1834	0.1696	0.0518
YOLOv8	0.8619	0.3309	0.3715	0.1830
YOLOv9	0.7967	0.3105	0.3384	0.1690
YOLOv10	0.8025	0.2632	0.3213	0.1359
YOLOv11	0.8129	0.3105	0.3552	0.1661
Faster R-CNN	0.0471	0.0931	0.0972	0.0471

Table 1: Comparison of Object Detection Models Based on Precision, Recall, and mAP Metrics.

the performance of different yolo models from yolov5 to yolov11 and faster RCNN model is evaluated on the TBX11k dataset using the key metrics: Precision, Recall, mAP@50, and mAP@50-95 as shown in the [Table 1](#), the results shows that yolov8 and yolov11 performed better when compared to other models.

Faster R-CNN, serving as the baseline model is significantly underperforming when compared to YOLO models. Its precision, recall , and mAP@50 scores highlight its challenges in adapting to the TBX11K dataset. While Faster R-CNN is traditionally known for high accuracy in general object detection tasks, its two-stage detection pipeline and reliance on region proposals might have struggled .

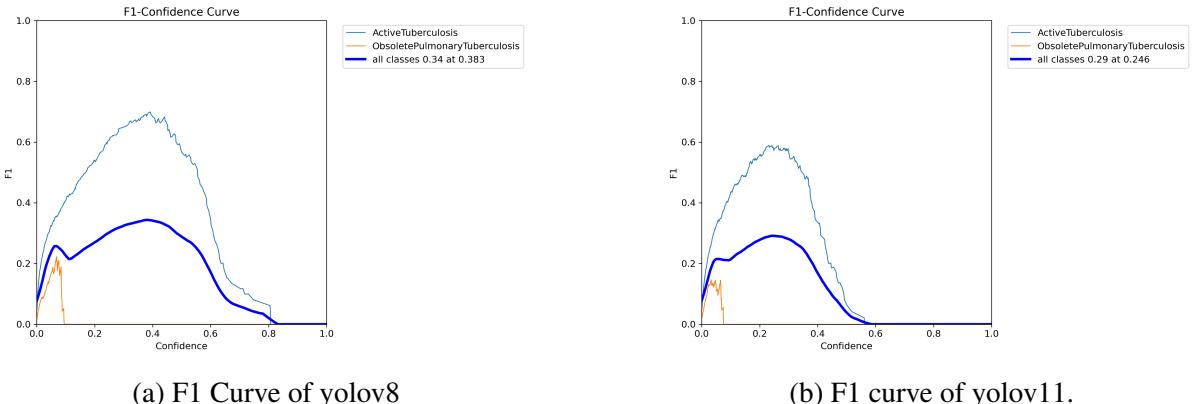


Figure 2: F1 curves of yolov11 and yolov8

The F1-Confidence Curve [Figure 2](#) shows the balance between precision and recall across various confidence thresholds. YOLOv8 and YOLOv11 models are performing better than the other models, where yolov8 is achieving a higher F1 score of 0.34 at 0.383 confidence and YOLOv11's F1 score 0.29 at 0.246 confidence.

even though both the models perform better than the other models, but [Figure 2](#) shows yolov8 and yolv11 struggles to perform consistently across both tuberculosis categories, with a noticeable disparity between the performance for Active Tuberculosis and Obsolete Pulmonary Tuberculosis. Both models perform relatively well in case of Active TB , In YOLOv8, the F1 score is around 0.8, indicating a good balance between precision and recall and yolov11 also achieves reasonable performance for this category, The dataset has 972 images of Active Tuberculosis, making it the majority class, this provides the models with more data to learn the patterns and features specific to this category. but in case of Obsolete pulmonary TB both the models struggle significantly. In YOLOv8, the F1 score for obsolete pulmonary TB peaks at 0.2 and rapidly declines, indicating poor precision and recall and in case of yolov11, the F1 curve for this class is nearly flat, with no significant peak, reflecting even worse performance than YOLOv8.

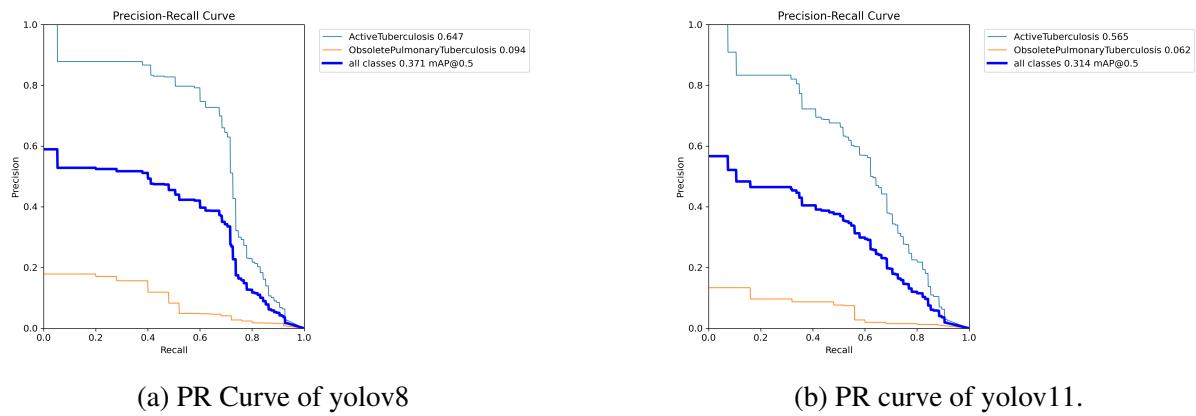


Figure 3: Precision and Recall curves of yolov11 and yolov8

The above graphs [Figure 3](#) shows clearly that the models struggles to balance performance between the two tuberculosis classes(Active Tuberculosis and Obsolete Pulmonary Tuberculosis). The Precision-Recall (PR) Curve reveals a significant disparity where the Active Tuberculosis class achieves a maximum precision of 0.647 with a relatively high recall, indicating that the model is effective at identifying and localizing true positives for this majority class. In contrast, the curve for Obsolete Pulmonary Tuberculosis is nearly flat, with a maximum precision of only 0.094, showing extremely poor detection capability.

4.3 Testing results after applying Data Augmentation

4.3.1 Data Augmentation

To address the major performance difference between the two classes(ActiveTB, ObsoletepulmonaryTB) and to remove the issue of class imbalance, data augmentation techniques are applied specifically to the minority class which is obsolete pulmonary TB.

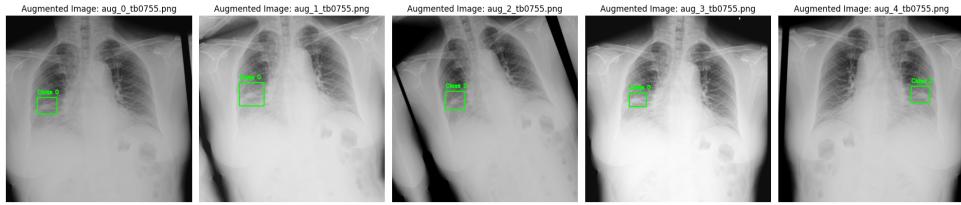


Figure 4: augmented images

Figure 4 shows an example image of an X-ray image ,where data augmentation is applied to minority class where where the images were augmented 4 times

Models after Augmentation	Precision	Recall	mAP@50	mAP@50-95
YOLOv5	0.8927	0.7491	0.8474	0.5653
YOLOv6	0.8549	0.3579	0.4677	0.2616
YOLOv7	0.7017	0.2965	0.4165	0.3567
YOLOv8	0.8533	0.7918	0.8917	0.6402
YOLOv9	0.7350	0.7638	0.8381	0.5479
YOLOv10	0.7388	0.7011	0.7790	0.5028
YOLOv11	0.8213	0.7196	0.8479	0.5432
Faster R-CNN	0.1987	0.2053	0.1587	0.1187

Table 2: Performance Metrics of Models after Augmentation.

The above Table 2 shows the different models performance after applying data augmentation to address the class imbalance. all the models are performing better than Table 1.

When compared to other models, YOLOv8 model is performing the best showing its superior capabilities in detecting tuberculosis abnormalities. YOLOv8 achieves the highest mAP@50 of 0.8917, indicating exceptional accuracy in detecting and localising objects. Its precision of 0.8533 reflects its ability to minimize false positives, while its recall of 0.7918 showcases its effectiveness in identifying true positives, even in challenging cases. Furthermore, YOLOv8's mAP@50-95 score of 0.6402, the best among all models, shows its robustness across varying object sizes and IoU thresholds, which is essential for refined detection in medical imaging.

Other models, such as YOLOv5 and YOLOv11, also show competitive performance, with YOLOv5 achieving high precision of 0.8927 and YOLOv11 maintaining a solid balance between precision (0.8213) and recall (0.7196). However, both fall short of YOLOv8's overall consistency and ability to maintain high recall while preserving precision. On the other hand, Faster R-CNN struggles significantly, with a precision of only 0.1987 and an mAP@50 of 0.1587, indicating that its two-stage architecture is less effective at handling the class imbalance and detecting subtle abnormalities in this dataset.

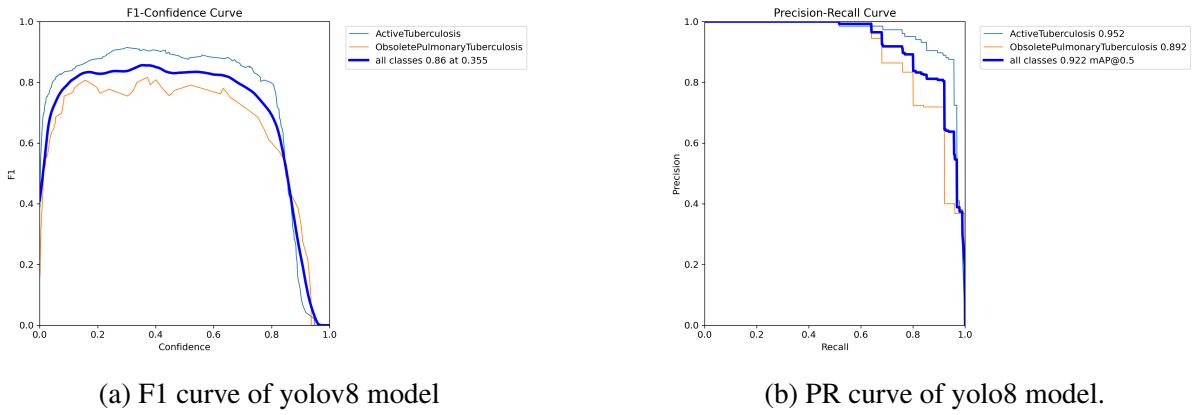


Figure 5: F1 and PR curve of yolo8 model

[Figure 5](#) shows the F1 and PR curve of yolo8 model after training with augmented images. After applying data augmentation, the YOLOv8 model exhibited significant improvements in detecting tuberculosis abnormalities, as shown by the F1-Confidence Curve and the Precision-Recall (PR) Curve in [Figure 5](#). The augmentation process focused on the minority class(Obsolete Pulmonary Tuberculosis).

The F1-Confidence Curve shows a clear picture of this improvement. For the majority class(Active TB), the F1 score peaked at an impressive 0.86, maintaining high values across a wide range of confidence thresholds. This indicates that the model preserved its ability to accurately detect this class while effectively balancing precision and recall. On the other hand, the minority class, Obsolete Pulmonary Tuberculosis, showed a dramatic improvement. Before augmentation, the F1 score for this class was significantly lower, barely reaching 0.2, reflecting the model's struggle to generalize for the minority class. After augmentation, the F1 score for this class increased close to 0.8, signaling that the model could now detect more true positives and reduce false positives.

The Precision-Recall Curve further tells that these findings For Active Tuberculosis, the PR curve remained nearly perfect, achieving a precision of 0.952 and good recall. This stability

demonstrates that augmentation did not compromise the performance of the majority class. the most important part of the lies in the performance of Obsolete Pulmonary Tuberculosis. The PR curve for this class, which was previously flat and indicative of poor detection, now shows a precision of 0.892 and higher recall.

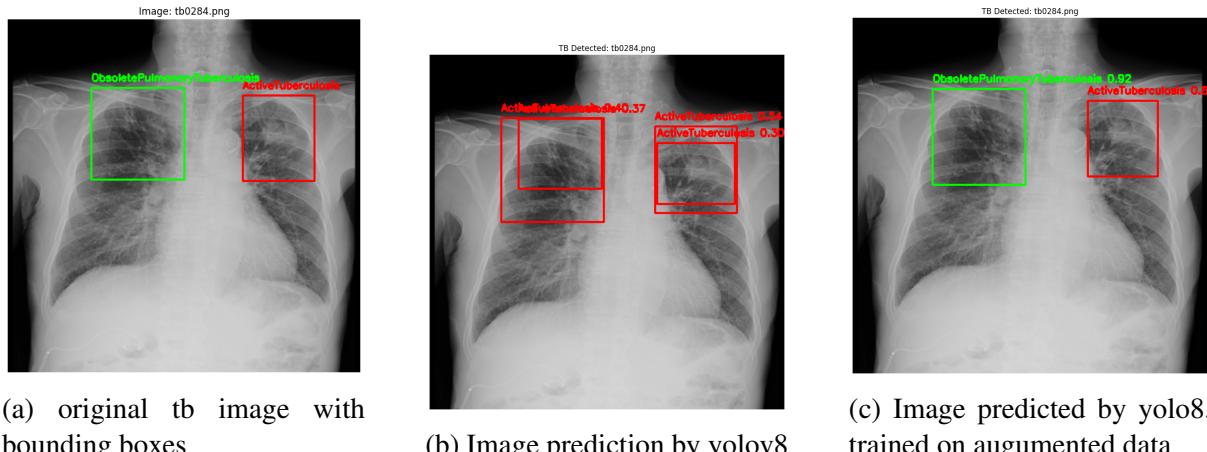


Figure 6: Comparison of images generated by yolov8.

The images in [Figure 6](#), show an example image generated by YOLOv8 model before and after applying data augmentation, [Figure 6b](#) shows the detection results produced by the YOLOv8 model after training on the original dataset without augmentation. Although the model successfully detects Active Tuberculosis with high precision and recall, it fails to detect Obsolete Pulmonary Tuberculosis, even though the bounding box for the minority class is clearly present in the ground truth.

[Figure 6c](#) demonstrates the detection results by the YOLOv8 model after training on the augmented dataset. Here, both Active Tuberculosis and Obsolete Pulmonary Tuberculosis are successfully detected, with precise enclosure boxes placed around abnormalities.

[Figure 7](#) is an example image of original chest x ray and predicted images detecting active tuberculosis where the image consists of four subfigures that compares the results of YOLO models (YOLOv5,YOLOv8,YOLOv11) prediction for detecting Active Tuberculosis in chest X-ray images

as shown in [Figure 7](#). The bounding box surrounding the abnormality is small and localized, which is a challenge for object detection models that is why out of all the YOLO models evaluated, YOLOv5, YOLOv8, and YOLOv11 were the only versions capable of successfully detecting the abnormality. This observation highlights the evolution in YOLO architectures, as newer models, such as YOLOv8 and YOLOv11, shows advanced features like multi scale de-

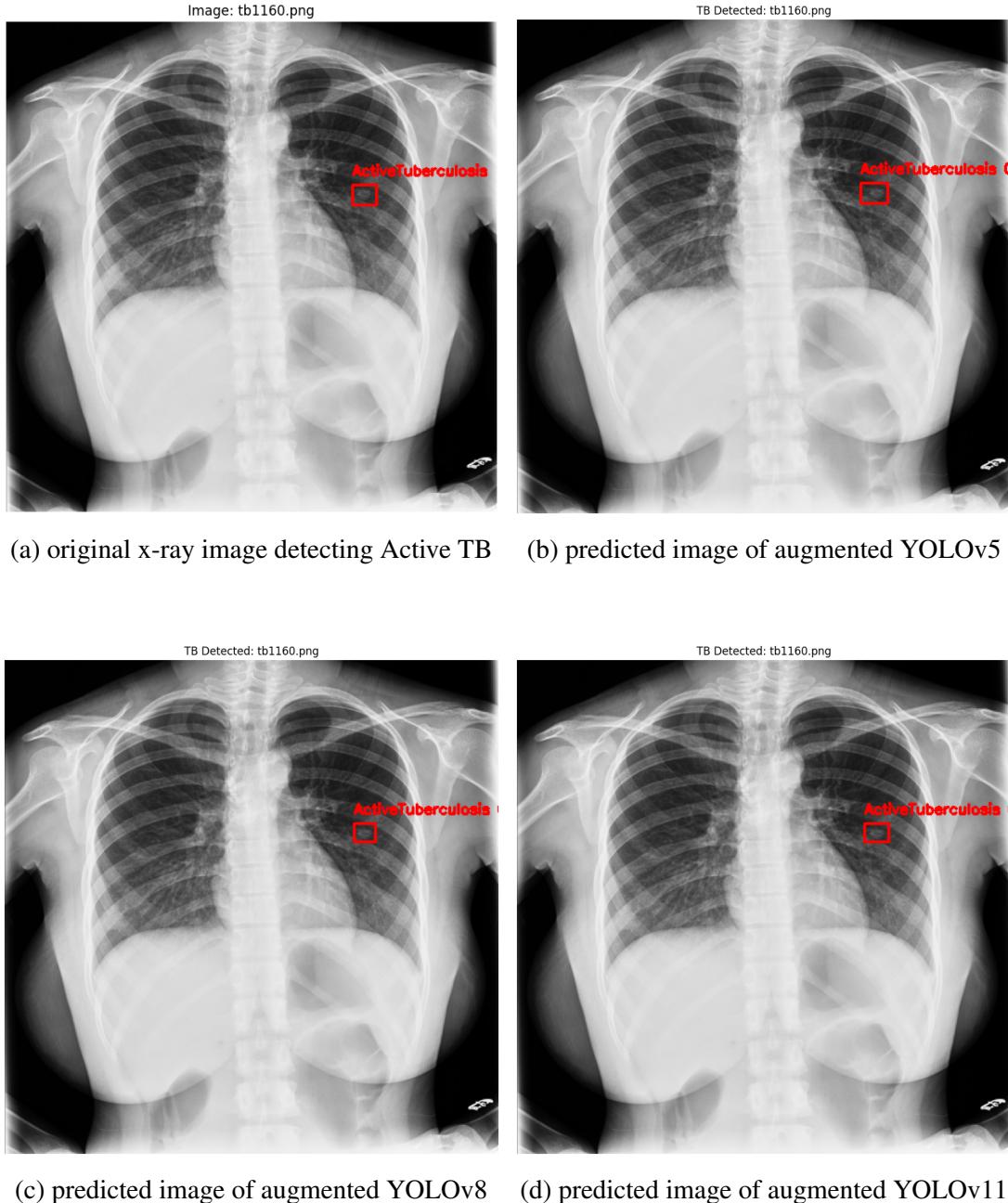


Figure 7: Comparison of original x-ray image of active TB with augmented yolo models(YOLOv5,YOLOv8,YOLOv11)

tection and cross-scale attention mechanisms, which are crucial for identifying small and subtle anomalies.

This [Figure 8](#) shows an example image of differnt models predicting obsolete pulmonary TB with its original x-ray image which acts as a ground truth

YOLOv5 [Figure 8b](#) does not align properly with the ground truth. The bounding boxes are overlapping, showing that YOLOv5 struggles to point the tuberculosis region accurately. This might be due to less robust training or architectural limitations in YOLOv5. where yolov6

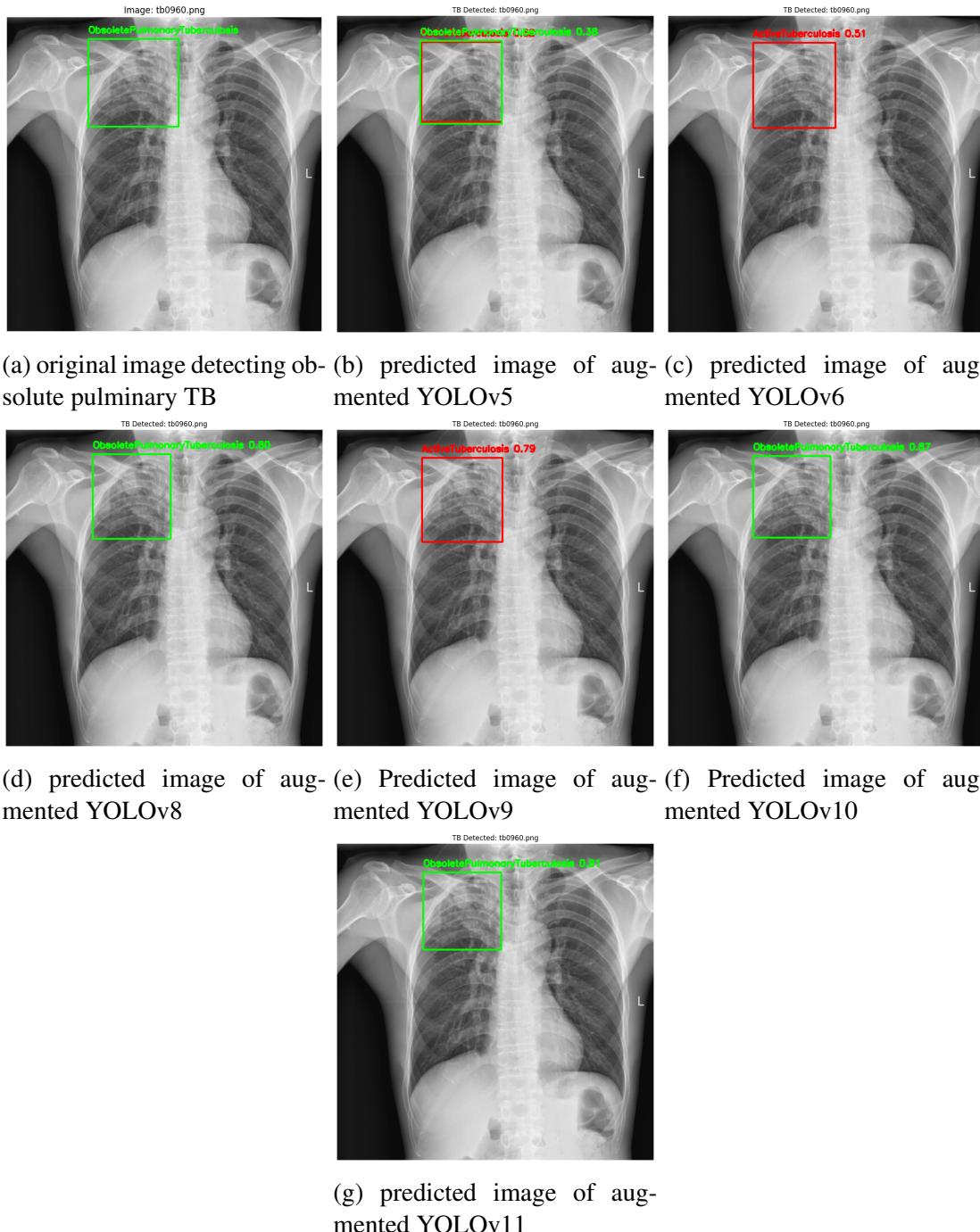


Figure 8: Comparison of original x-ray of absolute pulmonary TB with different augmented yolo models

[Figure 8c](#) and [yolov9](#) [Figure 8e](#), even though being the latest model ,they are not performing well in detecting the right TB

On the other hand YOLOv8,Yolov10,YOLOv11 models bounding boxes are aligning perfectly with the original ground truth box.

This [Figure 9](#) compares the performance of different YOLO model versions (YOLOv5 to

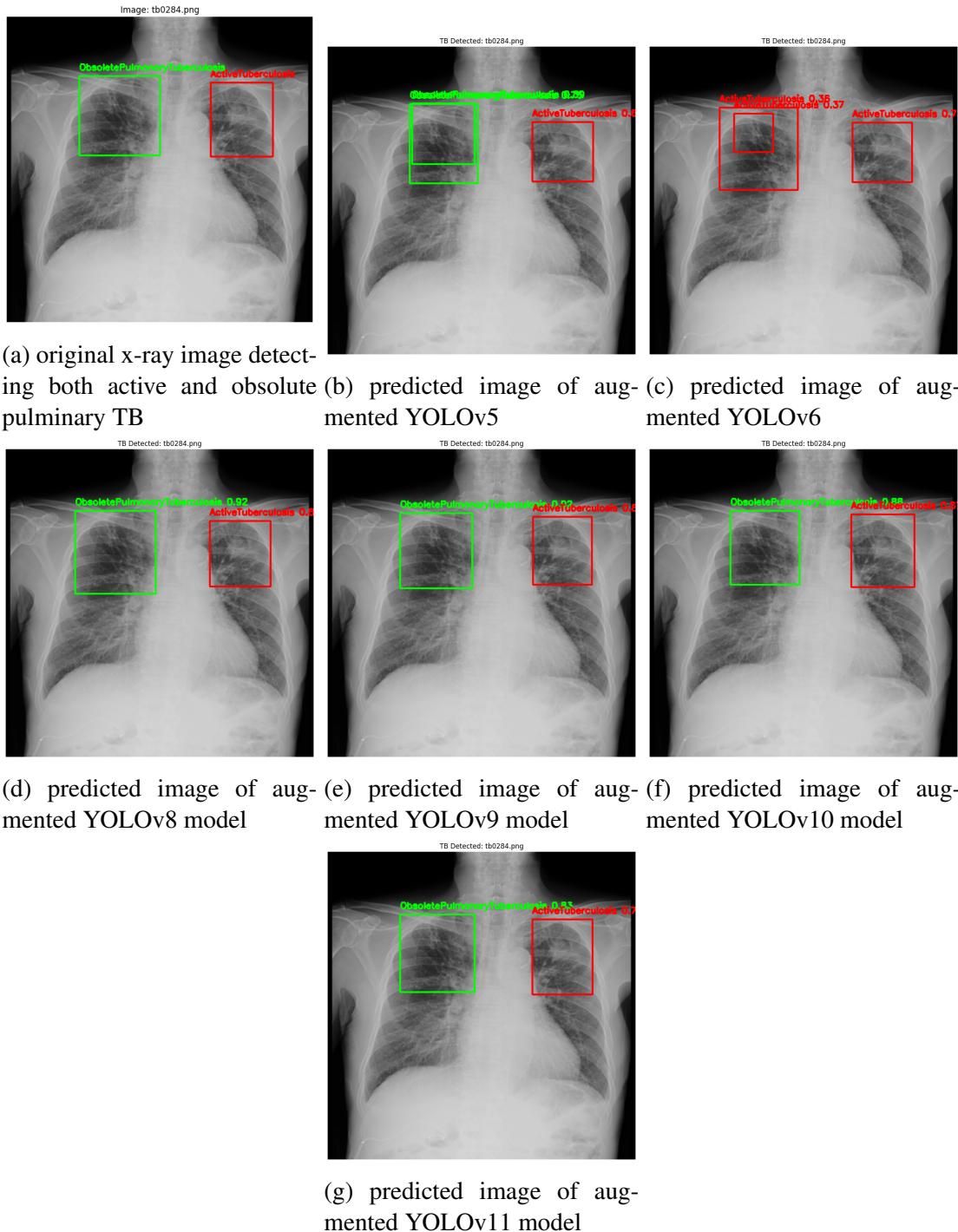


Figure 9: Comparison of 7 Images

YOLOv11) in detecting both active and obsolete pulmonary TB on chest X-rays.

YOLO models (YOLOv8,YOLOv9,YOLOv10,YOLOv11) are performing well in detecting both types of TB,but in YOLOv5 there are visible wrong alignments. The predicted boxes slightly deviate from the actual regions, and the model might be struggling with precise localization.and YOLOv6 performs well in detecting active tuberculosis but fails to detect obsolete pulmonary TB.

4.4 Hyperparameter tuning

In this project, I have focused on hyperparameter tuning only on YOLOv8 and YOLOv11 models as they consistently outperformed other versions in initial experiments. Their advanced models with high baseline metrics such as precision, recall, and mean Average Precision (mAP), made them ideal for further refinement. The goal of tuning is to know their full potential by optimizing key parameters, making sure that the models were not only accurate but also robust and generalizable across diverse scenarios.

YOLOv8 model with its anchor free detection mechanism and lightweight, modular design, showed exceptional precision during initial testing. These features allowed YOLOv8 to excel at detecting objects with high accuracy while maintaining efficiency. YOLOv11 introduced refinements in backbone networks, loss functions, and data augmentation strategies. These enhancements improved its ability to detect complex and challenging objects, particularly in scenarios requiring high recall. Given their strengths, both models were selected for hyperparameter tuning to further elevate their performance.

The hyperparameter tuning process included ten iterations for each model, systematically exploring combinations of parameters like learning rate, momentum, weight decay, and data augmentation techniques. For YOLOv8, the best results were achieved in iteration 8, with a fitness score of 0.46845. This iteration optimized key parameters, including a learning rate of 0.01139, momentum of 0.9171, and a mosaic augmentation factor of 0.85909. These adjustments resulted in precision (0.73988) and mAP50 (0.68162), showing YOLOv8's ability to balance accuracy and generalization effectively.

On the other hand, YOLOv11 reached its peak performance in iteration 2, achieving a fitness score of 0.33058. The optimal configuration for YOLOv11 included a learning rate of 0.01107 and a momentum value of 0.94819. While its precision and mAP scores were slightly lower than YOLOv8's, YOLOv11 excelled in recall, demonstrating its robustness in identifying difficult to detect objects. This strength makes it particularly valuable in applications where missing a detection carries significant consequences.

Comparing the two models, YOLOv8 emerged as the better performing option overall, with its superior precision and mAP metrics. However, YOLOv11's enhanced recall and refined backbone architecture offer complementary strengths, particularly for scenarios that prioritize comprehensive object detection.

Chapter 5

Discussion

5.1 Relation to objectives

The main goal of this project was to evaluate and compare the effectiveness of object detection models for detecting tuberculosis (TB) abnormalities using chest X-rays from the TBX11K dataset. Among these models, recent YOLO architectures performed well, with their modern design and real-time detection capabilities.

Trying to understand what makes YOLO more desirable than traditional object detection methods like Faster R-CNN. The results showed that, even though Faster R-CNN, with its two-stage detection approach, has been very popular, it struggled significantly with this dataset. Its precision was very less (0.1987), and the mAP@50 was 0.1587 which reflect its inability to handle the class imbalance and subtle abnormalities present in TB detection tasks. In contrast, YOLO models, particularly YOLOv8, achieved outstanding results. YOLOv8 maintained a high precision of 0.952 while also excelling in recall, making it most effective at identifying abnormalities. These results show that YOLO's single-stage architecture is better suited for tasks that require speed and adaptability, especially in medical imaging. Faster R-CNN, though reliable in some domains, was simply not equipped for the challenges of this project.

The evolution of YOLO models shows better results. By comparing YOLO versions from YOLOv5 to YOLOv11, it became clear how incremental changes in design can lead to good improvements. YOLOv5, didn't show promising accuracy and recall, even though yolov8 architectural evolution ([Alif & Hussain 2024](#)) is from yolov5, YOLOv5 (also authored by the same individuals) as YOLOv8, but struggled with precise localization, particularly for the minority class of obsolete pulmonary TB. Its bounding boxes often misaligned with the actual

abnormalities, highlighting architectural limitations. Then in case of YOLOv8 ,it diverges from traditional anchor-based methods. ([Alif & Hussain 2024](#)) Instead of relying on predetermined anchor boxes, YOLOv8 adopts an anchor-free approach by predicting the object's centre. This adjustment addresses challenges associated with anchor boxes that may not accurately represent custom dataset distributions, due to this anchor free approach the model performed very well . achieving precision of 0.952 and also excelled in balancing accuracy and efficiency.([Khanam & Hussain 2024](#)) YOLO11 extends and enhances the foundation laid by YOLOv8, YOLOv11 has introduced refined backbone networks and loss functions that enhanced recall, particularly in challenging cases.The model's improved feature extraction capabilities allow it to identify and process a broader range of patterns and intricate elements within ([Khanam & Hussain 2024](#)) While YOLOv11's precision and mAP were slightly lower than YOLOv8, its robustness in identifying objects marked a significant leap forward.

Class imbalance was one of the major challenges in this project, especially given the imbalance between active TB (the majority class) and obsolete pulmonary TB (the minority class). Initially, this imbalance effected more on the model's performance shown [Table 1](#) , with YOLOv8 and YOLOv11 struggling to detect the minority class [Figure 2](#) . The F1 score for obsolete pulmonary TB was barely reaching 0.2. but, data augmentation on the minority class turned out to be a very good solution. By augmenting the minority class, YOLOv8's F1 score as shown in [Figure 5](#) for obsolete pulmonary TB showed nearly 0.8, demonstrating a dramatic improvement in its ability to detect true positives while reducing false positives. This also didn't come at the expense of the majority class, YOLOv8 maintained its good performance in detecting active TB, with its precision and recall staying consistently high. The results emphasize the critical role of addressing class imbalance in medical datasets and the power of targeted augmentation to level the playing field.

5.2 Limitations in Hyperparameter Tuning Approach

Hyperparameter tuning is one of the important process in optimizing the performance of machine learning models. Initially, my goal was to perform detailed and extensive hyperparameter tuning for all YOLO models, from YOLOv5 to YOLOv11, to identify the optimal parameters for each version. This approach would have allowed for a more thorough understanding of each model's strengths and weaknesses under similar tuning conditions. It would also have

provided a better basis for comparison between the models in terms of precision, recall, and robustness. However, due to computational constraints, I was only able to conduct hyperparameter tuning for YOLOv8 and YOLOv11, the two most promising models identified from initial experiments.

For this project, I performed hyperparameter tuning on YOLOv8 and YOLOv11 using an L4 GPU, which is designed for efficient AI workloads. The tuning process proved to be extremely longer time and consumed lot of resource. Each iteration of tuning required significant time due to the large parameter space and the computational complexity of training these models on the TBX11K dataset. For YOLOv8, completing just 10 iterations of tuning took over 16 hours of continuous processing. Similarly, YOLOv11 required a comparable amount of time for 10 iterations. these limited iterations provided useful insights into the models performance, they did not explore the parameter space, and it became clear that a more comprehensive tuning process was beyond the capability of my available hardware.

This limitation in hyperparameter tuning introduces several challenges. First, by tuning only YOLOv8 and YOLOv11, I missed the opportunity to optimize YOLOv5, YOLOv6, YOLOv9, and YOLOv10. These models may have performed better with systematic tuning, potentially altering the final conclusions about their comparative performance. Second, the small number of iterations (10) restricted the exploration of the full hyperparameter space. As a result, there may be configurations that could further enhance the performance of YOLOv8 and YOLOv11, but these remain unexplored. Finally, the extensive time required for each iteration limited the feasibility of conducting a comprehensive tuning process. Running more iterations or tuning all models simultaneously was not possible due to hardware and time constraints.

5.2.1 Future Work

If sufficient computational resources were available, a more thorough approach could have been taken. First, I would have conducted hyperparameter tuning for all YOLO models to ensure a fair and comprehensive comparison across all versions. Second, I would have increased the number of iterations, exploring a larger parameter space to potentially reveal configurations that further enhance performance. Finally, leveraging distributed computing or cloud-based GPUs could have mitigated the computational limitations, enabling faster and more extensive tuning. Ideally, a comprehensive tuning process would have involved at least 50-300 iterations for each model to thoroughly explore their capabilities.

Chapter 6

Evaluation, Reflections, and Conclusions

6.1 Evaluation

This project aimed to assess and compare the performance of YOLO object detection models (YOLOv5 to YOLOv11) and Faster R-CNN for tuberculosis detection using chest X-rays from the TBX11K dataset. From experiments and analysis, the project addressed its objectives, uncovering key insights about the models and their suitability for medical imaging.

results of Objectives: The project effectively achieved its primary objectives. YOLO models, particularly YOLOv8 and YOLOv11, demonstrated good performance compared to Faster R-CNN in precision, recall, and mean Average Precision (mAP). This validated YOLO's single stage architecture is more efficient and better suited for real time applications, particularly in medical imaging where both accuracy and speed are critical.

Data augmentation showed an important component in addressing class imbalance, significantly enhancing the detection of Obsolete Pulmonary Tuberculosis, a minority class that initially struggled. For example, the F1-score for this class increased dramatically after augmentation, showing the importance of balancing datasets in improving model performance. And also, the analysis of YOLO's evolution revealed that newer versions, such as YOLOv8 and YOLOv11, introduced architectural improvements that directly enhanced detection capabilities, including better handling of small abnormalities and improved precision.

Effectiveness of Methods: The methods used in the project were highly effective in achieving its objectives, though some limitations were observed. Transfer learning was a key technique, allowing the YOLO models to adapt quickly to the TBX11K dataset by manipulating pre-trained weights. This approach proved essential for achieving strong baseline performance

and enabled a fair comparison between models. However, the use of YOLOv7 from an external repository introduced slight inconsistencies in implementation compared to the other YOLO versions, which were trained using the Ultralytics library.

Data augmentation played a very important role in improving the models performance, particularly for the minority class. Techniques such as flipping, rotation, and brightness adjustment resulted in a more balanced dataset, leading to better overall detection metrics. For example, YOLOv8 achieved the highest mAP@50 of 0.8917 and recall of 0.7918 after augmentation. This underscored the importance of targeted augmentation in addressing class imbalance and improving the robustness of object detection models in medical imaging.

Hyperparameter tuning, applied specifically to YOLOv8 and YOLOv11, allowed for further refinement of these models. but limited to 10 iterations due to computational constraints, the tuning process revealed valuable insights into their performance potential.

Results: The results clearly established YOLOv8 as the most effective model for tuberculosis detection in this study. Its better metrics across precision, recall, and mAP reflected its ability to accurately detect and localize abnormalities, even in challenging cases. Meanwhile, YOLOv11's strong recall demonstrated its capability in scenarios where missing detections must be minimized.

Data augmentation had a transformative impact, particularly on the minority class. Prior to augmentation, the models struggled while detecting Obsolete Pulmonary Tuberculosis, achieving low precision and recall. After augmentation, the F1 score for this class increased significantly, proving the effectiveness of balancing datasets in improving model performance.

The comparison between YOLO and Faster R-CNN provided valuable insights into their respective strengths and weaknesses. While Faster R-CNN's precision focused approach is well-suited for tasks requiring high accuracy, it fell short in scenarios involving class imbalance and small object detection. YOLO's single-stage architecture, on the other hand, excelled in handling such challenges, demonstrating its adaptability and efficiency in medical imaging.

6.2 Reflection

6.2.1 Acknowledgment of Existing Work

In this project, I have used existing tools, frameworks, and datasets to build upon and expand the scope of my work. Specifically, I utilized pre-existing YOLO architectures(YOLOv5 to

YOLOv11) from ultralytics ([Ultralytics 2024](#)) and github ([WongKinYiu 2024](#)) and the publicly available TBX11K dataset for tuberculosis detection.

My contributions lie in adapting these models for the specific task of TB detection, conducting experiments to compare their performance, applying advanced techniques like data augmentation and hyperparameter tuning, and analyzing the results to derive meaningful insights. The evaluation of the models strengths and limitations in medical imaging, as well as the exploration of how YOLO models evolve across versions.

6.2.2 Challenges faced

One of the most significant challenges was the computational time required for hyperparameter tuning. Despite using an L4 GPU, which is optimized for AI workloads, tuning even for 10 iterations took over 16 hours per model. This limited the ability to perform an exhaustive exploration of the hyperparameter space and prevented tuning for all YOLO models. As a result, the tuning process was restricted to YOLOv8 and YOLOv11, leaving potential performance improvements in other versions unexplored.

another challenge was the inclusion of YOLOv7, which was not part of the Ultralytics library used for training the other YOLO models. this model was taken from an external implementation from GitHub, introducing inconsistencies in the training process. While efforts were made to maintain consistency across models, these discrepancies may have impacted the overall comparison.

Faster R-CNN as a Baseline Model When starting this project, I had high expectations for Faster R-CNN to serve as a robust baseline for object detection. Its two-stage pipeline and reputation for precise localization in tasks like general object detection led me to believe it would perform well in this medical imaging application. Faster R-CNN has consistently demonstrated strong performance in scenarios requiring accurate bounding box predictions, and I anticipated better results with the TBX11K dataset.

but, the results were surprising. While Faster R-CNN did show precision in detecting certain features, its overall performance fell short when compared to YOLO models.

6.2.3 Future work

In this project, hyperparameter tuning was limited to YOLOv8 and YOLOv11 due to computational constraints. Future work could expand this process to include all YOLO models

(YOLOv5 to YOLOv11) and perform a more exploration of the hyperparameter space. This would provide a fairer basis for comparison and potentially reveal better-performing configurations for models like YOLOv5, YOLOv6, or YOLOv9.

further analysis and enhancement of the Faster R-CNN model can be done, particularly in addressing its underperformance observed in this study. eventhough the model is known to perform better ,it struggled to adapt to the challenges posed by the TBX11K dataset. A deeper investigation into its limitations is essential, starting with experimenting with more advanced backbone architectures, such as ResNet-101 or EfficientNet, which are known for their superior feature extraction capabilities.

in the future yolo models can be applied to more larger and diverse data ,and performance can be checked

6.2.4 Note for Healthcare Professionals

While the results of this project shows good performance of YOLO models, particularly YOLOv8 and YOLOv11 in detecting tuberculosis , it is crucial to emphasize that these models are not yet ready for direct deployment in medical practice. even though after achieving high precision and recall, there is presence of some false positives and false negatives ,which highlights the need for caution when considering their use in a highly sensitive field such as healthcare.

In medical diagnostics, accuracy is very important, and even a small rate of false positives or negatives can have serious consequences. For example, a false positive diagnosis of tuberculosis could lead to unnecessary stress, costly treatments, and further medical interventions for a patient who does not actually have the disease. and, a false negative could delay critical treatment for an actual TB patient.

To address these issues, future efforts should focus on refining these models to achieve perfect performance before considering clinical implementation.

6.3 Conclusion

In conclusion, this project explored the application of YOLO object detection models (YOLOv5 to YOLOv11) and Faster R-CNN as a baseline for tuberculosis detection using the TBX11K dataset. The study aimed to evaluate these model's performance in detecting Active and Obsolete Pulmonary Tuberculosis, with a focus on addressing class imbalance and optimizing

detection accuracy. The results demonstrated that YOLOv8 outperformed all other models, achieving the highest precision, recall, and mAP scores, highlighting its suitability for medical imaging tasks. Data augmentation proved to be a crucial step in removing the impact of class imbalance, significantly improving the detection performance of the minority class. Hyperparameter tuning further refined YOLOv8 and YOLOv11, showcasing their potential for accurate and robust detection. However, Faster R-CNN, despite its strong theoretical foundation, underperformed in this context, struggling to adapt to the challenges posed by the dataset. While the findings emphasize the promise of YOLO models in supporting medical diagnostics, it is important to acknowledge the limitations, including computational constraints and the models ability to false positives and false negatives. These factors underscore the need for further optimization and rigorous validation before these models can be deployed in clinical practice.

Chapter 7

Glossary

- **AI** - Artificial Intelligence
- **TB** - Tuberculosis
- **TBX11K** - Tuberculosis X-ray Dataset
- **YOLO** - You Only Look Once
- **R-CNN** - Region-based Convolutional Neural Network
- **mAP** - Mean Average Precision
- **IoU** - Intersection over Union
- **PR Curve** - Precision-Recall Curve
- **FPN** - Feature Pyramid Network
- **RPN** - Region Proposal Network
- **SGD** - Stochastic Gradient Descent
- **COCO** - Common Objects in Context (Dataset)
- **GPU** - Graphics Processing Unit
- **F1 Score** - Harmonic Mean of Precision and Recall
- **NMS** - Non-Maximum Suppression
- **LR** - Learning Rate

- **ROC Curve** - Receiver Operating Characteristic Curve

Bibliography

Alif, M. A. R. & Hussain, M. (2024), ‘Yolov1 to yolov10: A comprehensive review of yolo variants and their application in the agricultural domain’, *arXiv* .

URL: <https://arxiv.org/abs/2406.10139>

Bhalla, A. S., Goyal, A., Guleria, R. & Gupta, A. K. (2015), ‘Chest tuberculosis: Radiological review and imaging recommendations’, *Indian Journal of Radiology and Imaging* **25**(3), 213–225.

URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC4531444/>

Bista, R., Timilsina, A., Manandhar, A., Paudel, A., Bajracharya, A., Wagle, S. & Ferreira, J. C. (2023), ‘Advancing tuberculosis detection in chest x-rays: A yolov7-based approach’, *Information* **14**(12), 655. <https://doi.org/10.3390/info14120655>.

Bochkovskiy, A., Wang, C.-Y. & Liao, H.-Y. M. (2020), ‘Yolov4: Optimal speed and accuracy of object detection’, *arXiv preprint arXiv:2004.10934* .

URL: <https://doi.org/10.48550/arXiv.2004.10934>

Chen, H., Gu, W., Zhang, H., Yang, Y. & Qian, L. (2024), ‘Research on improved yolov8s model for detecting mycobacterium tuberculosis’, *Heliyon* **10**, e38088.

URL: <https://doi.org/10.1016/j.heliyon.2024.e38088>

Diwan, T., Anirudh, G. & Tembhurne, J. (2023), ‘Object detection using yolo: challenges, architectural successors, datasets and applications’, *Multimedia Tools and Applications* **82**, 9243–9275.

URL: <https://doi.org/10.1007/s11042-022-13644-y>

Ganatra, N. (2021), ‘A comprehensive study of applying object detection methods for medical image analysis’, *2021 8th International Conference on Computing for Sustainable Global*

Development (INDIACoM) pp. 821–826.

URL: [urlhttps://api.semanticscholar.org/CorpusID:235339509](https://api.semanticscholar.org/CorpusID:235339509)

Girshick, R. (2015), ‘Fast r-cnn’, *arXiv preprint arXiv:1504.08083* . To appear in ICCV 2015.

URL: <https://arxiv.org/abs/1504.08083>

Guo, Z., Wang, J., Wang, J. & Yuan, J. (2022), ‘Lightweight yolov4 with multiple receptive fields for detection of pulmonary tuberculosis’, *Computational Intelligence and Neuroscience* **2022**, 9465646.

Jocher, G. et al. (2020), ‘Yolov5: An improved real-time object detection model’, <https://github.com/ultralytics/yolov5>. Accessed: December 6, 2024.

Khanam, R. & Hussain, M. (2024), ‘Yolov11: An overview of the key architectural enhancements’, *arXiv preprint arXiv:2410.17725* .

URL: <https://doi.org/10.48550/arXiv.2410.17725>

M, P. R., WR, S. E., A, B. & M, M. A. (2024), ‘Yolov8’s advancements in tuberculosis identification from chest images’, *Frontiers in Big Data* **7**, 1401981.

URL: <https://doi.org/10.3389/fdata.2024.1401981>

Redmon, J., Divvala, S., Girshick, R. & Farhadi, A. (2015), ‘You only look once: Unified, real-time object detection’, *arXiv preprint arXiv:1506.02640* . To appear in CVPR 2016.

URL: <https://doi.org/10.48550/arXiv.1506.02640>

Redmon, J. & Farhadi, A. (2016), ‘Yolo9000: Better, faster, stronger’, *arXiv preprint arXiv:1612.08242* .

URL: <urlhttps://doi.org/10.48550/arXiv.1612.08242>

Ren, S., He, K., Girshick, R. & Sun, J. (2015), ‘Faster r-cnn: Towards real-time object detection with region proposal networks’, *arXiv preprint arXiv:1506.01497* .

URL: <https://doi.org/10.48550/arXiv.1506.01497>

Shams, U. (2020), ‘Tbx11k dataset’ . Accessed Dec 9, 2024.

URL: <https://www.kaggle.com/datasets/usmanshams/tbx-11>

Solawetz, J. & Francesco (2024), ‘What is yolov8? a complete guide’ . Published Oct 23, 2024. Accessed Dec 9, 2024.

URL: <https://blog.roboflow.com/what-is-yolov8/>

Ultralytics (2020), ‘Meituan yolov6 models’, <https://docs.ultralytics.com/models/yolov6/>. Accessed: December 6, 2024.

Ultralytics (2024), *Models Supported by Ultralytics*. Accessed: 2024-12-08.

URL: <https://docs.ultralytics.com/models/>

Ultralytics (n.d.), ‘Hyperparameter tuning guide’, <https://docs.ultralytics.com/guides/hyperparameter-tuning/>. Accessed: 2024-12-08.

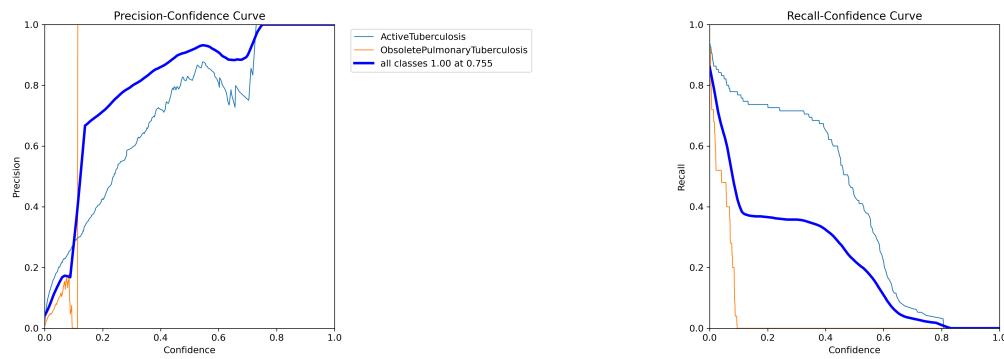
Wang, C.-Y., Bochkovskiy, A. & Liao, H.-Y. M. (2022), ‘Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors’, *arXiv preprint arXiv:2207.02696*.

URL: <https://doi.org/10.48550/arXiv.2207.02696>, doi = 10.48550/arXiv.2207.02696

WongKinYiu (2024), ‘Yolov7’, <https://github.com/WongKinYiu/yolov7>. Accessed: 2024-12-08.

Appendix A

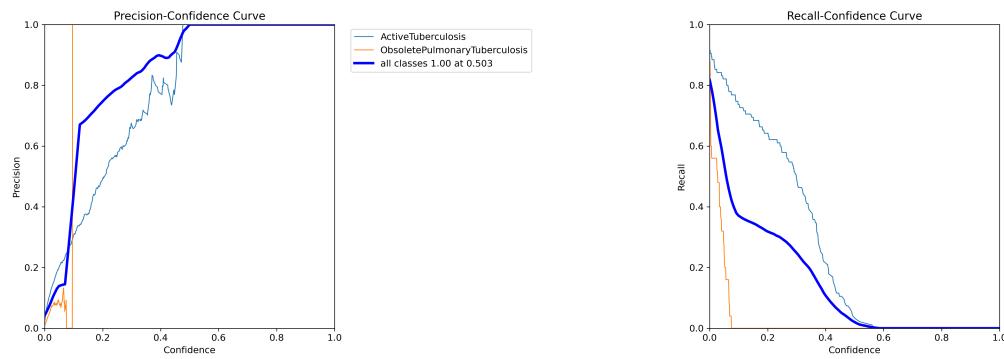
Graphs



(a) Precision curve of YOLOv8 model

(b) Recall curve of YOLOv8 model.

Figure 1: Precision curve and Recall cuve of YOLOv8



(a) Precision curve of YOLOv11 model

(b) Recall curve of YOLOv11 model.

Figure 2: Precision curve and Recall curve of YOLOv11

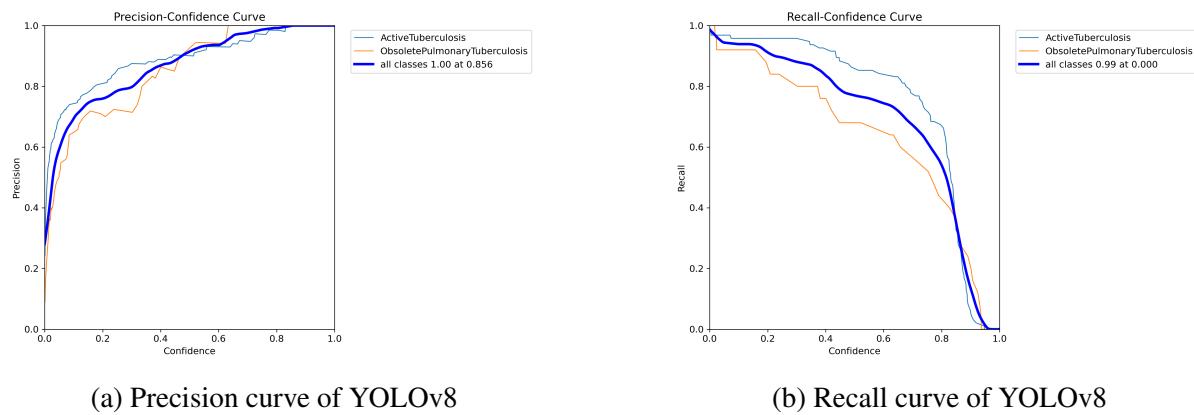


Figure 3: Precision curve and Recall curve of YOLOv8 model after applying data augmentation