Predictive analysis of Tuberculosis detection with X-Ray images using Yolov8

Manthena Karthik, Student, B. Sc. (AIML), Dept. of Computer Science, P.B. Siddhartha College of Arts & Science, AI Intern at Codegnan, Vijayawada, AP, India.
Syed Abdul Sattar, Student, B. Sc. (AIML), Dept. of Computer Science, P.B. Siddhartha College of Arts & Science, AI Intern at Codegnan, Vijayawada, AP, India.
Jitendra Chautharia, M.Tech (CPS, EE, IIT Jodhpur), B.Tech (EEE, RTU Kota), AI Engineer at Codegnan IT Solutions, Vijayawada, AP, India.

Abstract:
Tuberculosis (TB) is a highly infectious disease caused by the bacterium Mycobacterium tuberculosis, primarily affecting the lungs but capable of spreading to other organs. Despite significant advances in medical science, TB remains a global health challenge, with approximately 10 million new cases and 1.5 million deaths reported annually, according to the World Health Organization (WHO). The disease is transmitted through airborne particles, making it highly contagious, particularly in densely populated and low-resource settings.
By enabling early detection, these models play a critical role in preventative healthcare, helping to reduce the incidence and severity of heart disease. Our experimental results showcase the effectiveness of YOLOv8 in detecting fractures across various anatomical regions and fracture types, achieving competitive performance compared to existing methods. Furthermore, we explore the impact of dataset size, augmentation techniques, and model architecture variations on detection accuracy and robustness. The proposed system offers a promising solution for streamlining fracture diagnosis workflows, potentially reducing the burden on healthcare professionals and improving patient outcomes. Through comprehensive evaluation and comparison with traditional approaches, we validate the efficacy and practical utility of our proposed YOLOv8-based bone fracture detection system.

Introduction:
Accurate and timely diagnosis of Tuberculosis is crucial for effective medical intervention and patient care. In this paper, we propose a novel approach for automated Tuberculosis detection using YOLOv8, a state-of-the-art object detection framework. By training the YOLOv8 model on a dataset of radiographic images containing both fractured and intact lungs, we demonstrate its capability to accurately localize and classify fractures with high precision and recall. Our experimental results showcase the effectiveness of YOLOv8 in detecting lungs across various anatomical regions and types, achieving competitive performance compared to existing methods. Furthermore, we explore the impact of dataset size, augmentation techniques, and model architecture variations on detection accuracy and robustness. The proposed system offers a promising solution for streamlining fracture diagnosis workflows, potentially reducing the burden on healthcare professionals and improving patient outcomes. Through comprehensive evaluation and comparison with traditional approaches, we validate the efficacy and practical utility of our proposed YOLOv8-based Tuberculosis detection system.

OBJECT DETECTION:

Fig 1 chest Xray

Object detection is a computer vision task that involves identifying and locating objects within an image or a video sequence. It has numerous real-world applications like ranging from sevurity surveillance and autonomous driving to augmented reality and medical imaging. Object detection is the process of locating and classifying objects within in an images or videos. Unlike image classification, which only identifies the dominant object in an image, object detection algorithm provides information about the presence, location, and category of multiple objects in an image.

EXISTING SYSTEM:

Manual Interpretations By Radiologists: Traditional methods rely on radiologists manually examining X-ray images to identify fractures. Radiologists use their expertise to visually inspect the images, locate fractures, and provide diagnostic reports. However, this process is time-consuming, subjective, and prone to human error.

Rule-Based Approaches: Some existing systems use rule-based algorithms to detect fractures based on predefined criteria. These rules may involve analyzing pixel intensity, bone contours, and geometric patterns. However, rule-based methods may lack flexibility and struggle with complex fracture patterns.

Template Matching: Template matching techniques compare X-ray images with predefined templates of common fracture patterns. If a match is found, the system identifies the fracture. However, this approach may not handle variations well and requires an extensive template library.

Edge Detection and Thresholding: Edge detection algorithms identify bone edges in X-ray images. Thresholding techniques segment the image to highlight potential fracture regions. These methods are computationally efficient but may miss subtle fractures or produce false positives.

Region-Based Approaches: Some systems divide X-ray images into regions of interest (ROIs) and analyze these regions separately. Features like texture, shape, and intensity are extracted from ROIs to detect fractures. However, ROI selection and feature extraction can be challenging.

Machine Learning Based Approaches: Machine learning models, including YOLOv8, have been applied to fracture detection. YOLOv8, a state-of-the-art object detection model, learns to detect fractures directly from X-ray images. It considers contextual information, scales, and spatial relationships for accurate detection. YOLOv8's deep learning capabilities improve efficiency and reduce subjectivity.

Challenges in the Existing System: Variability in fracture patterns due to patient age, bone development, and injury type. Noise in X-ray images, artifacts, and variations in imaging quality. Balancing sensitivity (recall) and specificity (precision) in fracture detection.

In summary, while manual interpretation and rule-based methods have been used historically, machine learning approaches like YOLOv8 offer promising advancements in pediatric fracture detection. YOLOv8's ability to learn from data and adapt to complex fracture patterns makes it a valuable tool for improving clinical outcomes.

Processing Pipeline:

YOLOv8 processes images by

Resizing Input Image: It resizes the input image into a fixed size (In the dataset we have resized the image into 237x237 pixels).

Single Convolutional Neural Network: YOLOv8 runs a single convolutional neural network on the resized input image.

Thresholding Detections: The results which are being detected are thresholded based on the model's confidence scores.

Bounding Boxes: This will return the bounding boxes (coordinates) of detected objects along with their class labels.

Output Format:

For each detected object, YOLOv8 provided class label, confidence score, bounding box coordinates.

INPUT:

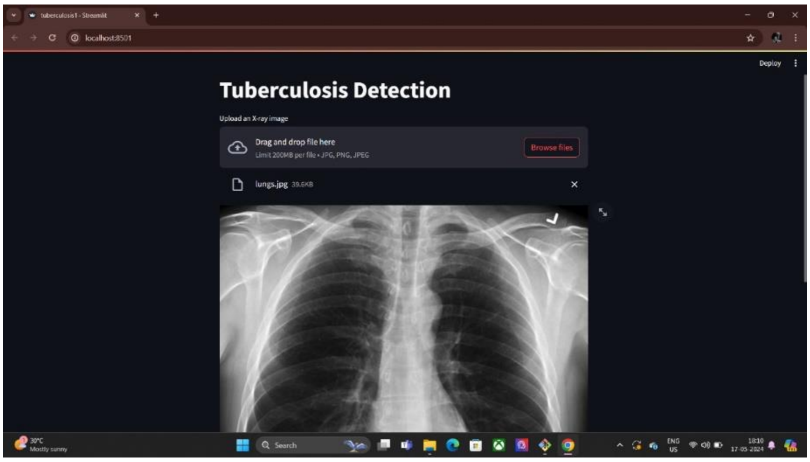

Fig-2 : It shows the X-ray of patient

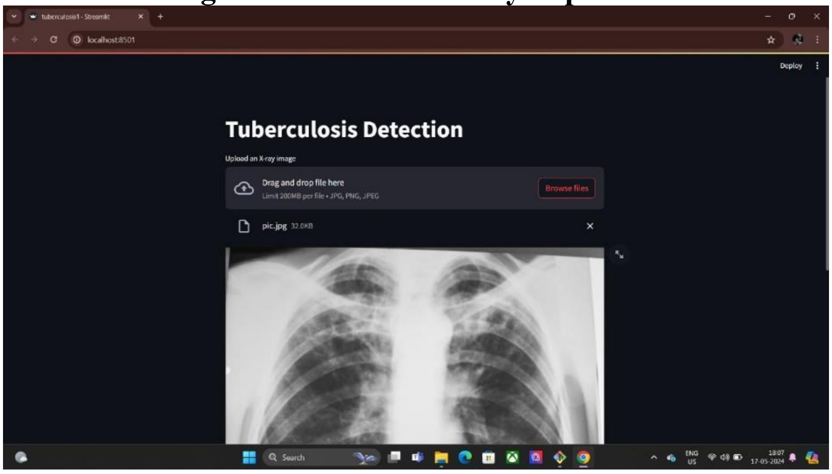

Fig-3 : The above fig indentifying the tuberculosis

INTERFACE:

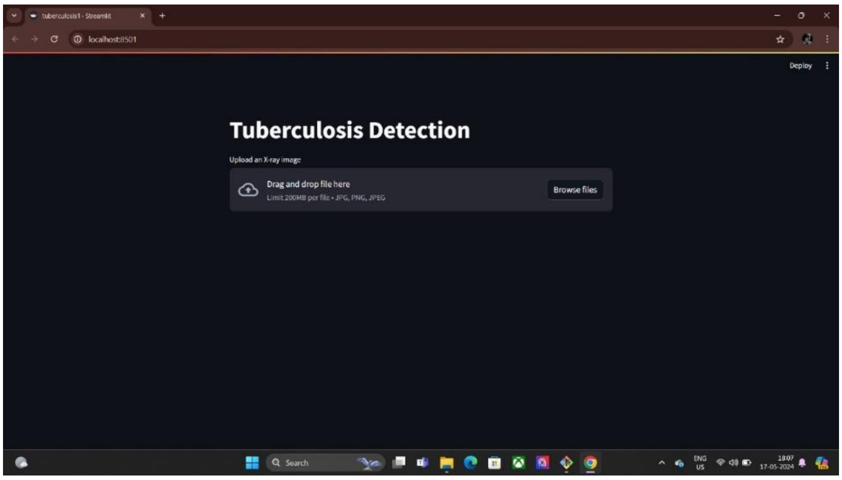

Fig-4: The above picture shows about the interface

OUTPUTS:

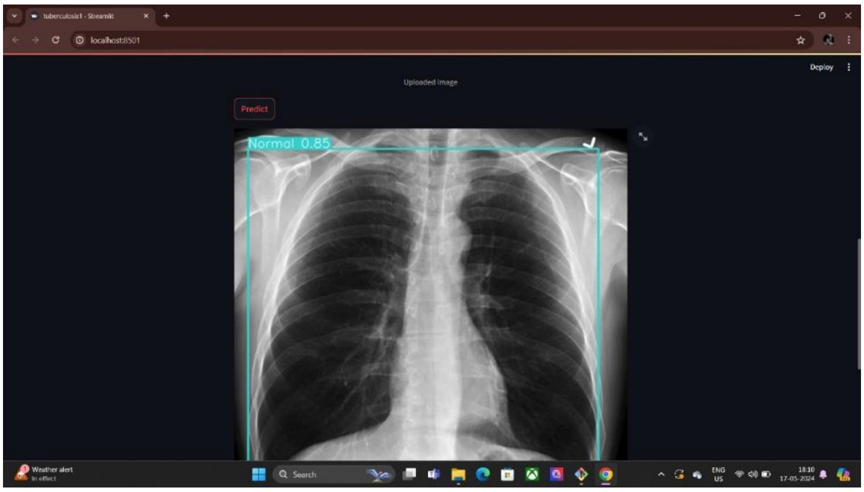

Fig-5

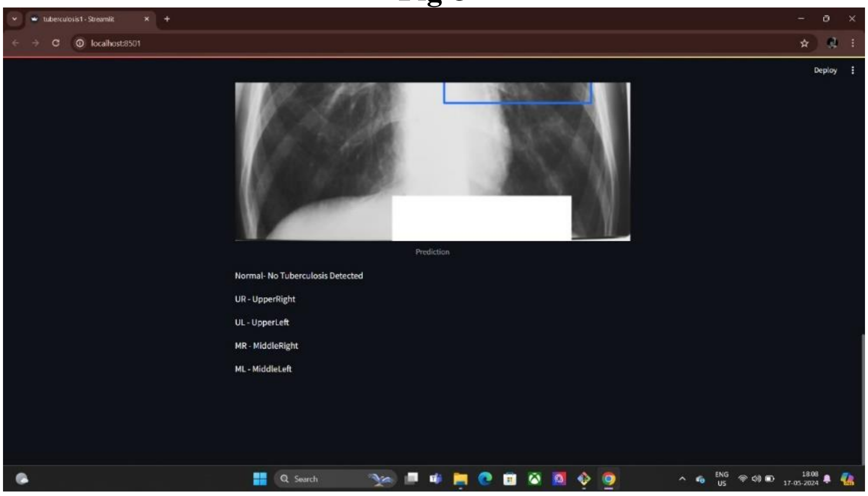

Fig-6

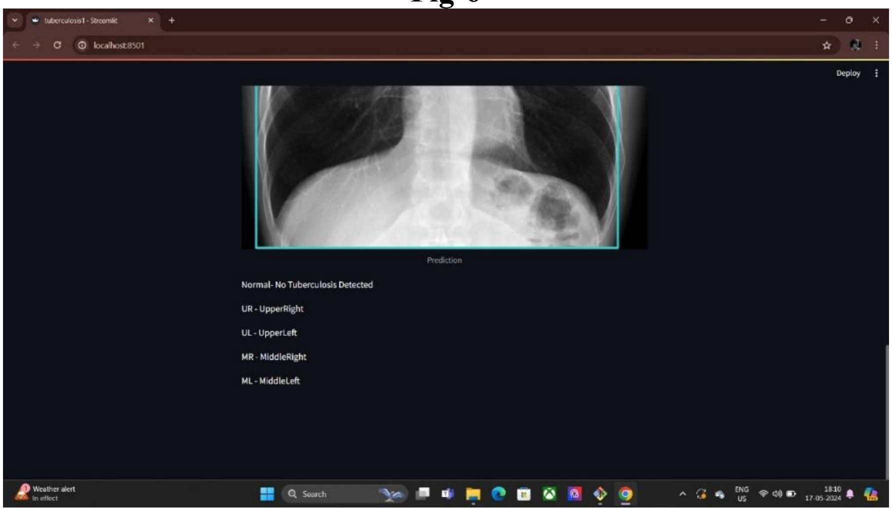

Fig-7

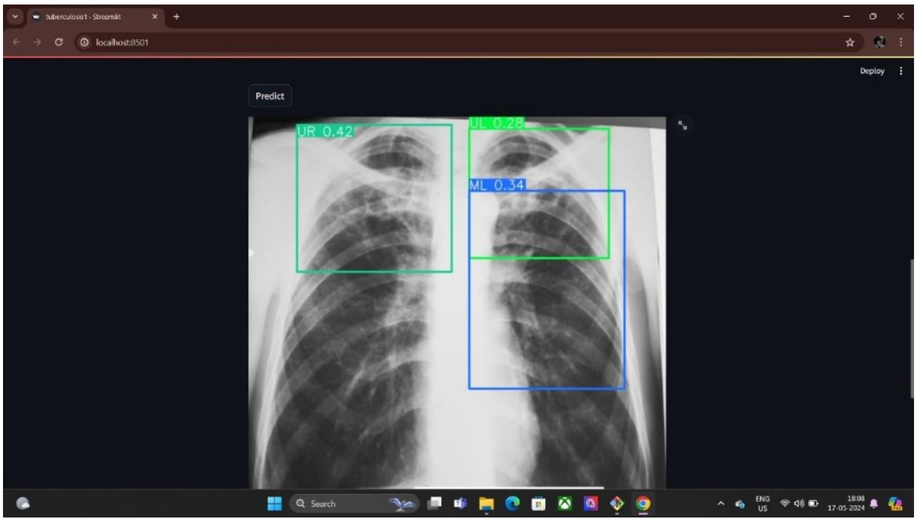

Fig-8

The above fig-5, fig-6, fig-7, fig-8 shows the patient is suffering with tuberculosis

MATERIALS AND METHODS:

The dataset we have used contains of total 3404 x-ray images. As YOLOv8 is a pre-trained model here 3161 X-ray images has been taken for training, 160 images for valid and 83 for testing. For preprocessing we have used Auto-oriented and done the resize stretch to 640X640 pixels. As for the augmentations we have used Flip-horizontal.

YOLOv8 Architecture:

YOLOv8 Architecture consists of lungs part which is specified for a specific task.

Lungs: This is the Convolutional Neural Network(CNN) responsible for extracting features from input image. YOLOv8 uses custom CSPDarknet53 backbone, which employs cross-stage partial connections to improve information flow between layers and boast accuracy.

Optimizer:

YOLOv8 uses AdamW optimizer during training. The AdamW optimizer is an extension of the popular Adam optimizer, which incorporates weight decay(L2 regularization) to prevent overfitting. It combines adaptive learning rates with weight decay, making it effective for training deep neural networks like YOLOv8. If we want to change the optimizer to something other than SGD, we can adjust the training configuration file(usually a .yaml file) to specify the desired optimizer(e.g., Adam or RMSprop).

Forward Pass: In the forward pass the YOLO model process an input image and predicts bounding boxes for objects within that image. Here the input image is divided into a grid of cells, each cell is responsible for predicting bounding boxes for ibjects that fall within it. For each cell, the model predicts object categories(using softmax activation). The bounding box coordinates(center coordinates, width, and height) relative to the cell. The final output is a tensor containing class probabilities and bounding box coordinates of all cells.

Loss Claculation: YOLOv8 uses a combination of loss functions to optimize its predictions

Classification Loss measures how well the model predicts object classes.

Localization Loss penalizes the discrepancy between predicted bounding box coordinates and ground truth coordinates.

Confidence Loss reflects the confidence score(objectness) for each bounding box.

The total YOLOv8 loss is weighted sum of these individual losses:

YOLOv8Loss $= \lambda_{Cls}.\,Classification\ Loss + \lambda_{loc}\ .Localization\ Loss + \lambda_{conf}\ .Confidence\ Loss$

Distribution Focal Loss (DFL):
An improved version of focal loss that focuses on hard-to-classify examples. It adjusts focal loss based on class distribution to make the model more sensitive to minority classes. DFL is used to improve bounding box regression, especially for objects with unclear boundaries.

Backpropagation and Weight Update: After calculating loss, gradients are computed with respect to the model's parameters(weights). Backpropagate these gradients through the network to update the model weights using optimization algorithms like stochastic gradient descent(SGD) or Adam.

The weight update rule for parameter(w) is

$$w_{\text{new}} = w_{\text{old}} - \eta \cdot \frac{\partial \text{Loss}}{\partial w}$$

Multi-GPU Training: Uf you have multiple GPU's, you can distribute the trianing workload across them for faster convergence. YOLOv8 supports multi-GPU training out of the box. This method is efficient and speeds up training compared to using 1 GPU.

Validation and Early Stopping: During trainig, the model is evaluated on a validation set. If the validation loss stops improving, early stopping is used to halt training. This helps prevent overfitting by stopping training when performance on the validation set deteriorates.

Fine-Tuning and Transfer Learning: Fine-Tuning involves adjusting hyperparameters or modifying the architecture of the pre-trained model. Transfer learning is a process where a pre-trained model(e.g., trained on other dataset) is used as a starting point for training on a new task. This can accelerate convergence and improve performance when the new task is similar to the task the model was originally trained on.

Post-Processing: After the model make predictions, post-processing steps are applied.Non-Maximum Suppresion(NMS) is used to remove redundant bounding boxes. Confidence score thresholds are set to filter out weak detections.

Evaluation Metrics: The trained model is evaluated on a test set using metrics like precision, recall, and F1-score. Average Precision(AP) at different Intersection over Union(IoU) thresholds is calculated. These metrics provide a comprehensive evaluation of the model's performance.

Deployment: Finally, the trained YOLOv8 model weights are saved for future inference. The model can be deployed for real-time fracture detection in medical applications.

Advantages:
High Accuracy and Speed: YOLOv8 can locate and classify objects in images with high accuracy and speed, making it suitable for detecting different types of fractures in various scenarios.

Fast and Reliable Diagnosis: YOLOv8 can provide a fast and reliable diagnosis of bone fractures, which can improve treatment outcomes.

Accurate Localization: YOLOv8 precisely locates the fracture site within the X-ray, aiding in accurate treatment planning2.

Disadvantages:
Information Bottleneck: As the network becomes deeper, an information bottleneck may occur, leading to loss functions that fail to produce useful gradients.

Slower Training Times: While YOLOv8 shows faster inference speed, some users report slower training times compared to previous versions.

Misinterpretation of X-ray Images: The percentage of X-ray images misinterpreted have reached 26%5. This could lead to incorrect fracture detection.

Conclusion:
The implementation of YOLOv8 for bone fracture detection shows promising results in automating the process of identifying fractures in medical imaging. By leveraging state-of-the-

art object detection techniques, the system achieves high accuracy and efficiency in detecting fractures, thus potentially aiding healthcare professionals in diagnosing injuries more effectively and swiftly. However, further validation on diverse datasets and rigorous evaluation against existing methods is necessary to establish its robustness and clinical applicability. Overall, the utilization of YOLOv8 presents a significant step towards enhancing fracture detection processes, offering potential benefits in patient care and medical workflow optimization.

References:

1. Machine Learning Prediction Model of Tuberculosis Incidence Based on Meteorological Factors and Air Pollutants by Na Tang, Maoxiang Yuan, Zhijun Chen, Jian Ma, Rui Sun, Yide Yang, Quanyuan He, Xiaowei Guo, ShixiongHu, andJunhuaZhou. https://doi.org/10.3390%2Fijerph20053910
2. Tuberculosis detection using deep learning and contrast enhanced canny edge detected X-Ray images by Hwa, Stefanus & Bade, Abdullah & Hijazi, MohdJeffree,Mohammad(2020). http://dx.doi.org/10.11591/ijai.v9.i4.pp713-720
3. A Comparative Study of Detection of Tuberculosis using Machine Learning & Deep Learning, by R. S. Prasad, R. C. Waghmare, T. B. Pajgade, R. R. Raut and M. L. Mahajan, https://ieeexplore.ieee.org/document/10112352
4. Machine and Deep Learning for Tuberculosis Detection on Chest X-Rays: Systematic Literature Review by Hansun S, Argha A, Liaw ST, Celler BG. https://www.jmir.org/2023/1/e43154/
5. Tuberculosis detection in chest radiograph using convolutional neural network architecture and explainable artificial intelligence. By Nafisah, S.I., Muhammad, G. https://doi.org/10.1007/s00521-022-07258-6.