University of Science and Technology of Hanoi

Midterm report

YOLO for object detection in medical images

Group 42

22BI13268	Vu Hai Thien Long
22BI13320	Cao Nhat Nam
22BI13120	Nguyen Duc Duy
22bi13470	Cao Hieu Vinh
22BI13306	Nguyen Van Minh

TABLE OF CONTENTS

1. Introduction

Accurate abnormality detection in medical imaging is vital for diagnostics. YOLO, a deep learning model, excels in real-time object detection and has been used for identifying kidney stones and bone fractures. YOLOv8 improves subtle pattern detection, aiding early diagnosis and prevention. This project explores YOLOv8's use in detecting kidney anomalies and bone fractures, demonstrating its adaptability and diagnostic accuracy.

2. Background

YOLO, a groundbreaking model in object detection, is known for its speed and accuracy. By treating object detection as a single regression problem, it predicts bounding boxes and class probabilities directly from the entire image, unlike traditional region-based methods. This makes YOLO ideal for real-time healthcare applications.

It works by dividing an image into a grid and predicting bounding boxes and class probabilities for objects:

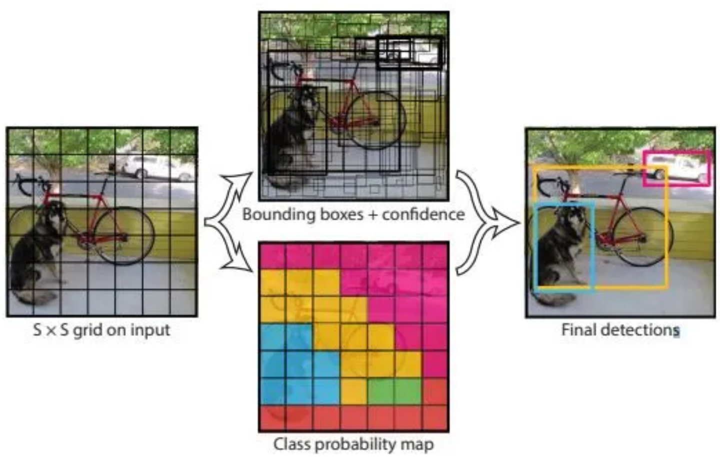

Figure 2.1. YOLO detection pipeline

The YOLO family has undergone several iterations, each improving on speed, accuracy, and feature detection. Below is a comparison of the key features across different YOLO versions:

YOLO Version	Key Features
YOLO v1	Grid-based detection, single-stage processing.
YOLO v2	Anchor boxes, batch normalization, multi-scale training.
YOLO v3	Darknet-53 backbone, feature pyramid networks (FPN).

YOLO v4	CSPNet architecture, k-means for anchors, GHM loss.
YOLO v5	EfficientDet, dynamic anchors, spatial pyramid pooling.
YOLO v6	EfficientNet-L2, dense anchor boxes for speed/accuracy.
YOLO v7	Nine anchor boxes, focal loss for small objects.
YOLO v8	Anchor-free detection, mosaic data augmentation.

In this project, we decided to choose **YOLOv8** as the main model because it is the most current stable version available, and it offers state-of-the-art performance with advanced features.

3. Exploratory Data Analysis

3.1. Data Understanding

Our project consists of 2 datasets: Bone Fracture dataset and Kidney Stone Image dataset:

3.1.1. Kidney Stone Image Dataset

The Kidney Stone dataset contained images focused on detecting a single class labeled as **'Tas_Var'**, indicating the presence of kidney stones. Given the simplicity of having only one class, this dataset allows for focused training and evaluation of the model's accuracy in detecting this specific abnormality.

Figure 3.1.1.1. An X-ray scan of kidney stones

3.1.2. Bone Fracture Image Dataset

To evaluate the performance of the YOLOv8 model on a more complex dataset with multiple classes, we selected a **Bone Fracture dataset** consisting of images categorized into seven distinct classes, as follows:

- **Elbow Positive**
- **Fingers Positive**
- **Forearm Fracture**
- **Humerus Fracture**
- **Humerus**
- **Shoulder Fracture**
- **Wrist Positive**

Figure 3.1.1.2. An X-ray scan of bone fracture

4. Data configuration and Training

4.1 Dataset configuration

The dataset consists of labeled medical images for object detection tasks. Each image contains specific medical abnormalities such as kidney anomalies or bone fractures.

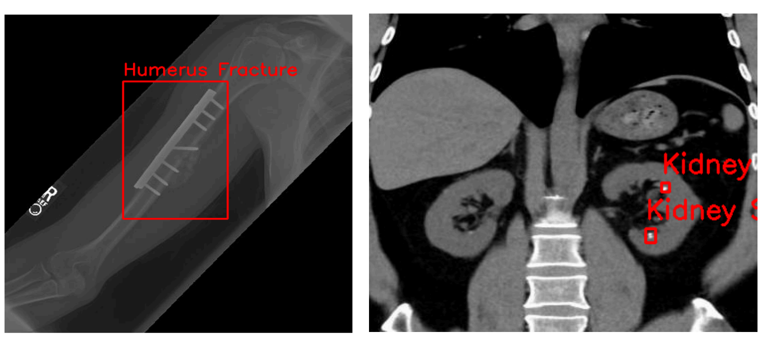

Figure 4.1 The Image with their labels box

The labels are stored in YOLO format, where each image has a corresponding .txt file containing the coordinates of bounding boxes and class labels. The dataset configuration is specified in a YAML file (**data.yaml**) which defines the paths for training and validation data, as well as the number and names of classes.

Before training, the model automatically applied the necessary preprocessing steps:

- **Normalization**: Image pixel values were scaled to the [0, 1] range.
- **Resizing**: All images were resized to 640x640.

4.2 Training Process

We initialized the YOLOv8n model using pre-trained weights (pretrained on COCO dataset). The model was trained with the following parameters:

- Epochs: 30
- Batch size: Auto-determined (-1)
- Optimizer: Auto-selected
- Data configuration: Specified by *'data.yaml'*

During training, various metrics were monitored and logged, including:

- Training losses: box loss, class loss, DFL loss
- Validation losses: box loss, class loss, DFL loss
- Performance metrics: precision, recall, mAP50, mAP50-95

5. Results and Finding

We conducted several performance evaluations of the YOLOv8n model using various plots to monitor and assess its effectiveness. Below are key findings:

5.1 Kidney Stone Image dataset

The overall performance is quite good, with the accuracy is **0.82**, which is good for a model being trained with only 30 epochs.

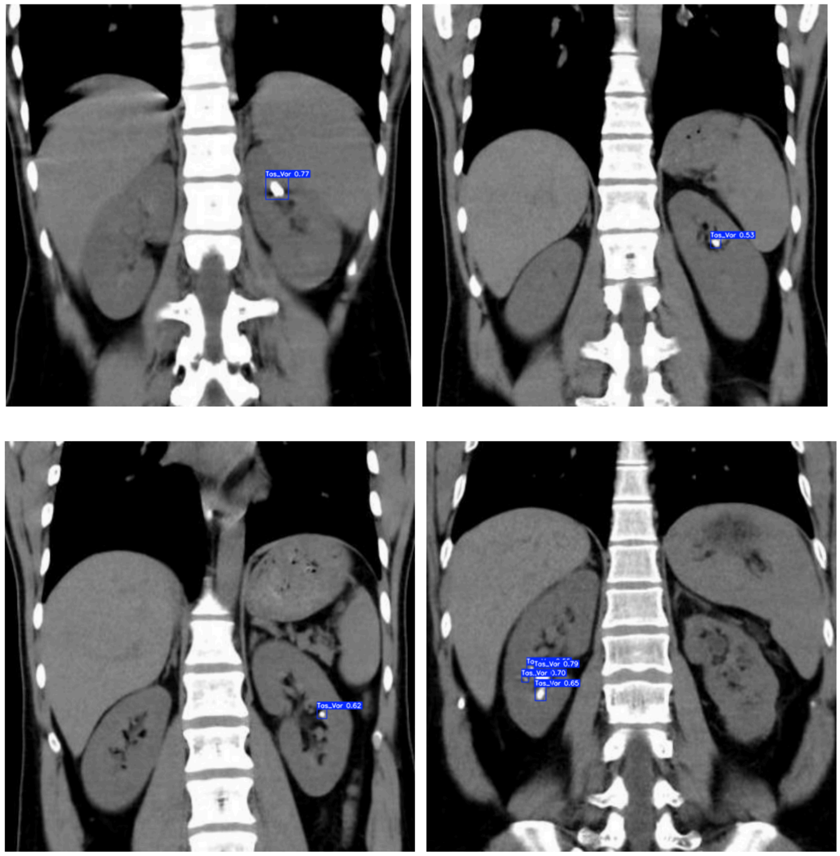

Figure 5.1.1. Kidney stone detections

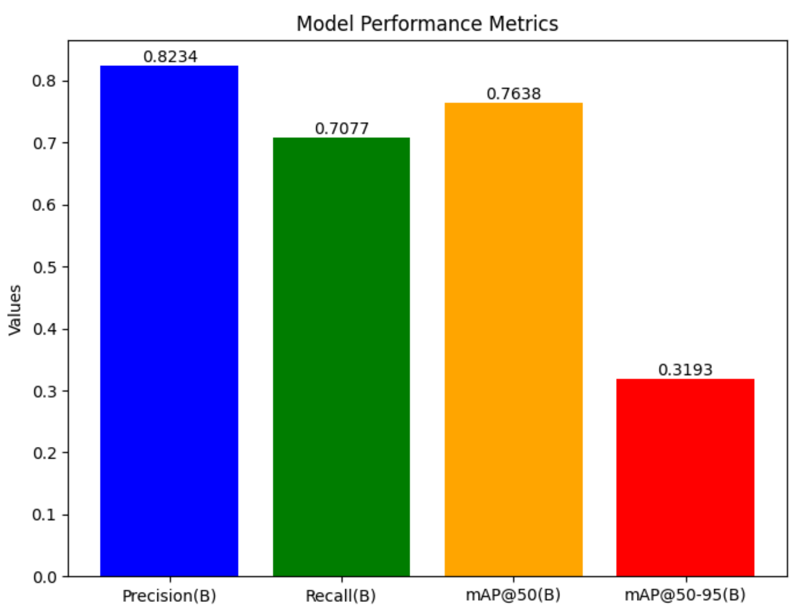

Figure 5.1.2. Model Performance Metrics

We also conduct some other plots to monitor and assess the performance:

Normalized Confusion Matrix:

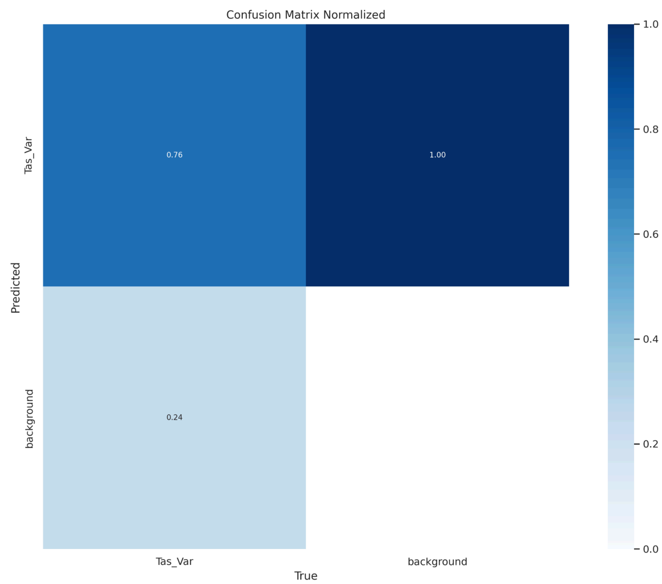

Figure 5.1.2 Normalized Confusion matrix of YOLOv8n for Kidney Dataset

We can see:

- *Tas_Var* class:
    - True Positive Rate (Recall) of 0.76
    - False Negative Rate of 0.24 (misclassified as background)
- *Background (no stone)* class:
    - True Negative Rate of 1.00
    - No false positives (0.00) for the background class

Observations:

- The model excels at correctly identifying the background class with perfect accuracy.
- There's room for improvement in detecting the Tas_Var class, with about 24% of actual Tas_Var instances being misclassified as background.
- No instances of background are misclassified as Tas_Var, indicating high precision for the Tas_Var class.

Precision-Recall Curve:

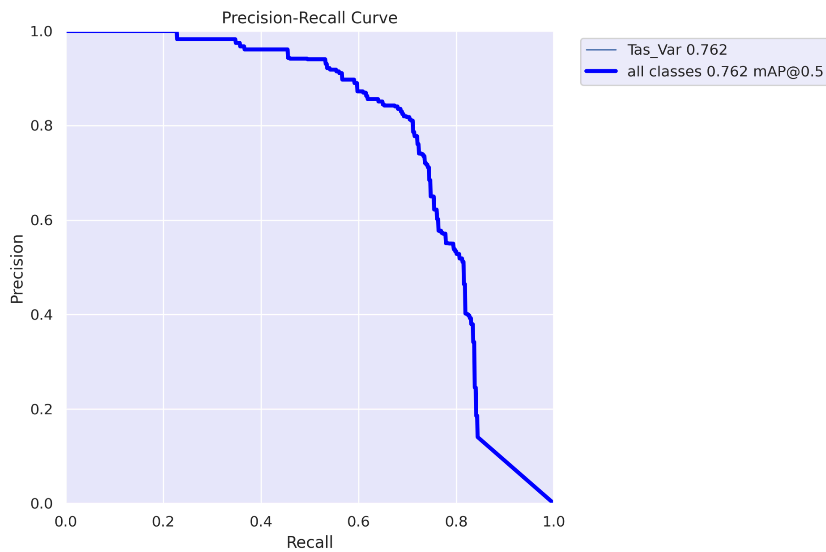

Figure 5.1.3. Precision-Recall plot of YOLOv8n for Kidney Dataset

- The model achieves a mean Average Precision (mAP@0.5) of 0.762 for all classes.
- The curve maintains high precision (near 1.0) for recall values up to about 0.6, indicating strong performance in this range.
- There's a noticeable drop in precision as recall increases beyond 0.6, suggesting a trade-off between identifying all positive instances (high recall) and maintaining accuracy (high precision).

This performance suggests the model is effective at identifying true positives with high confidence but may struggle with more challenging or ambiguous cases.

Model Performance Dashboard:

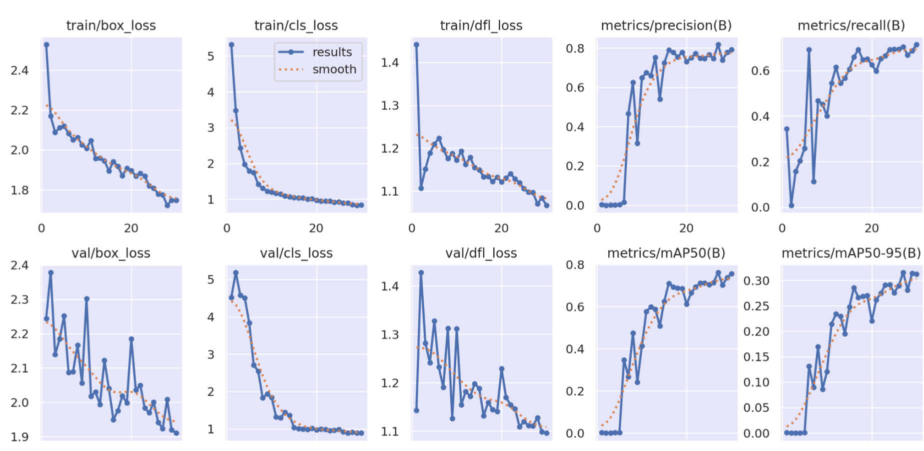

Figure 5.1.4. Performance dashboard of YOLOv8n for Kidney Dataset

- Throughout training, all performance metrics showed consistent improvement, indicating effective learning.
- No signs of overfitting were observed, as validation metrics followed the trend of training metrics, confirming the model's robustness during training.

5.2 Bone Fracture image dataset

On the other hand, the performance of the Bone Fracture image dataset is not very good, partly because the dataset has too many features and the dataset is not very big enough for the model to perform well.

Figure 5.2.1. Bone fracture detection

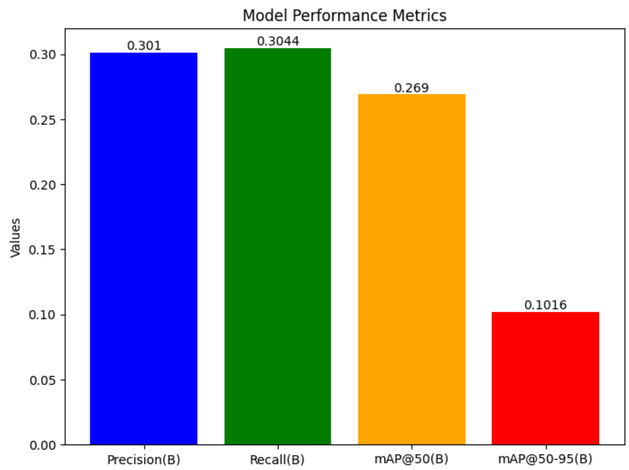

Figure 5.2.2. Model performance Metrics

Confusion Matrix Normalized:

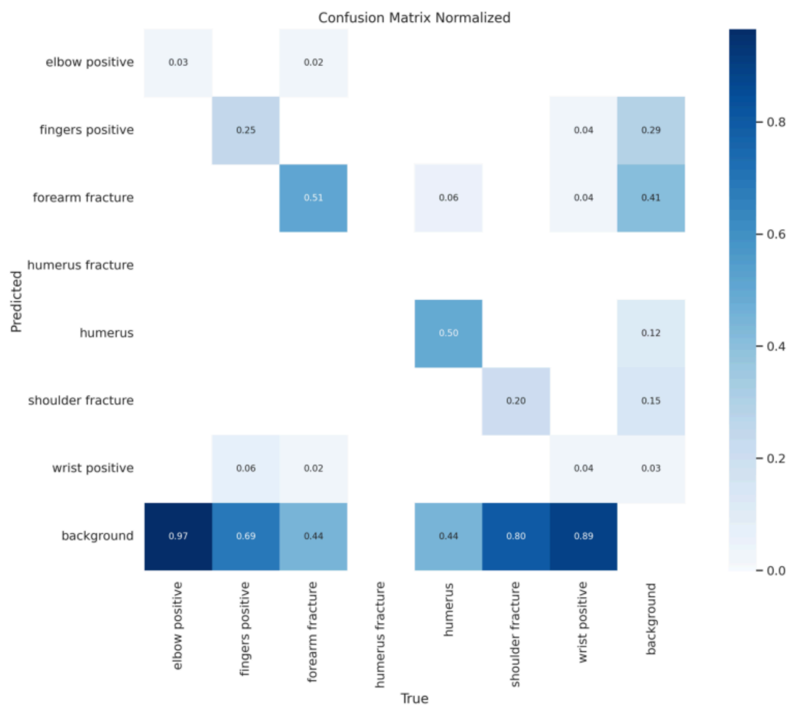

Figure 5.2.3. Confusion Matrix of YOLOv8n on Bone dataset

The analysis highlights strengths in detecting elbow fractures but reveals weaknesses in shoulder and wrist fracture detection, particularly due to high false negatives.

Model Performance Dashboard:

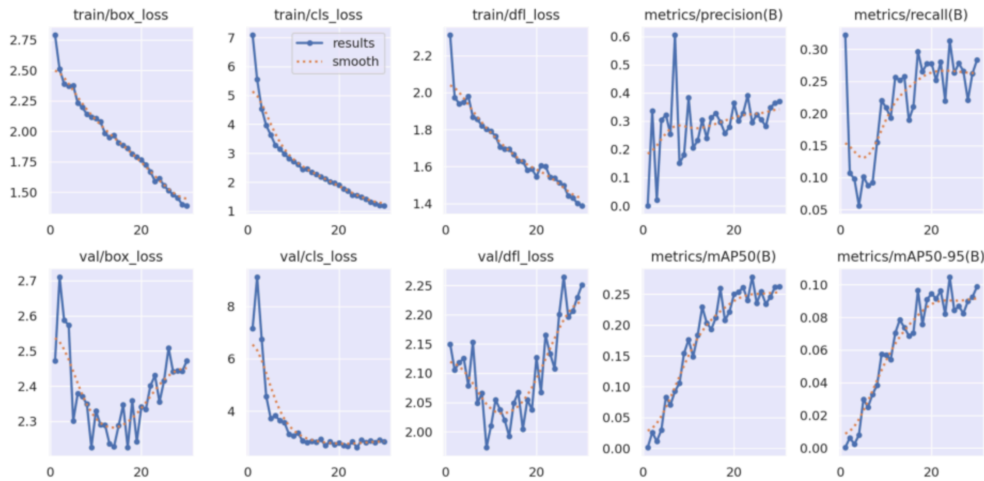

Figure 5.2.4. Model Performance Dashboard of YOLOv8n on Bone dataset

Precision, recall, mAP50, and mAP50-95 generally show improvement over time, peaking at 20 epochs.

6. Experimentation

In this part , we implemented the transfer learning on the Kidney Image dataset.

The training process was divided into two stages: **layer freezing** and **unfreezing**. Initially, a subset of the model's layers was frozen to retain the general features learned from the original dataset. After a few epochs, these layers were unfrozen to fine-tune the entire model to the new dataset.

Here are the configurations :

- **Freezing Layers**:
  For the first 5 epochs, the initial 30 layers were frozen to preserve low-level features like edge detection. The remaining layers adapted to the medical image classification task.
- **Unfreezing Layers**:
  After 5 epochs, all layers were unfrozen for the remaining 25 epochs, allowing the entire network to fine-tune for high-level features critical to medical diagnosis.

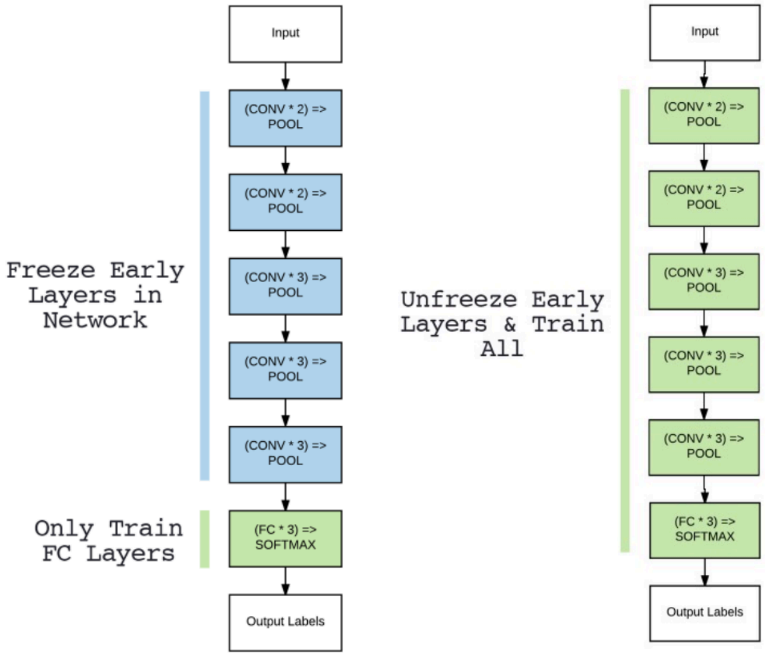

Figure 6.1. Freezing and Unfreezing method

Luckily, after many attempts, we finally improved the accuracies of our fine-tuned model with the metrics precision of **0.86**.

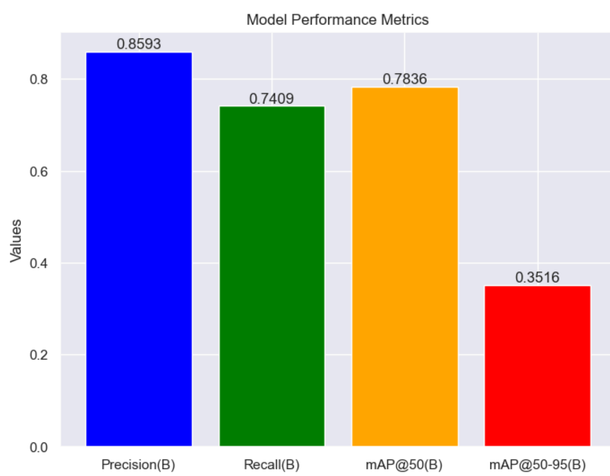

Figure 6.2. Model Performance Metrics of finetuned model

The metric precision shows great improvement from early steps, which is the freezing step, and gradually increase and top over the non-finetuned version at around 20 epochs.

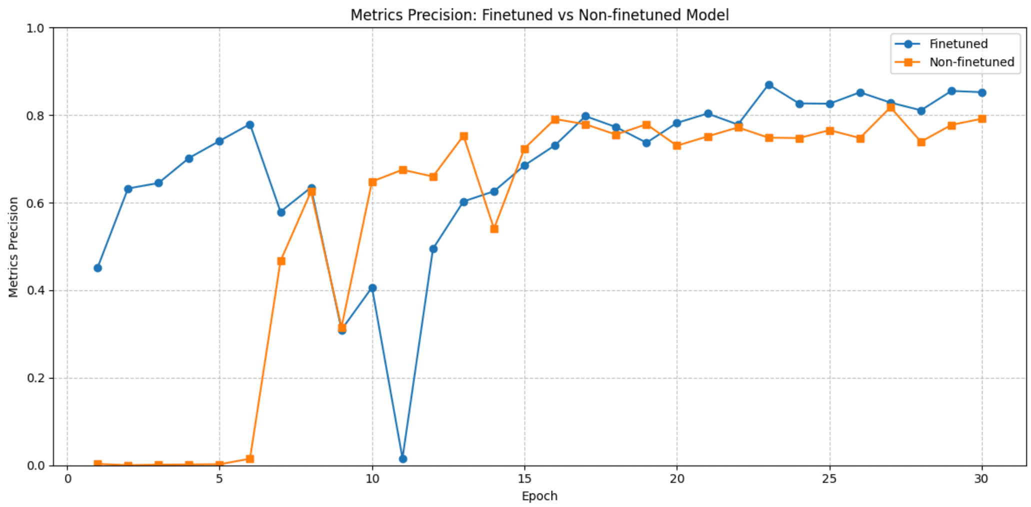

Figure 6.3. Comparison of Precision between 2 models

6. Discussion

There is still room for improvement for our model, other methods can be implemented like Hyperparameter Tuning, Learning Rate Scheduling,.... Despite the low accuracy for the Bone Fracture detection model, we have concluded several insights that the model may need further refinement or specialized techniques for handling subtle variations between classes, although YOLOv8 is powerful, additional work is required to optimize its performance in complex tasks like bone fracture detection, especially for harder-to-detect classes.

7. Conclusion

In this project, YOLOv8 proved effective for detecting kidney stones, achieving high accuracy with focused datasets. While the model struggled with more complex datasets, the use of transfer learning improved its precision. YOLOv8 can be a powerful tool in medical diagnostics, especially when fine-tuned for specific conditions. Further research could explore enhancing multi-class detection performance by increasing dataset size and adjusting model configurations.

References

[1]:https://jurnal.polgan.ac.id/index.php/sinkron/article/view/13744/2554

[2]:https://www.researchgate.net/figure/Process-flow-diagram-of-kidney-stone-detection-using-YOLO-models_fig4_376023128

[3]:https://www.researchgate.net/publication/379143641_Detection_of_Hand_Bone_Fractures_in_X-ray_Images_using_Hybrid_YOLO_NAS

[4]: S. C. Medaramatla, C. V. Samhitha, S. D. Pande and S. R. Vinta, "Detection of Hand Bone Fractures in X-Ray Images Using Hybrid YOLO NAS," in IEEE Access, vol. 12, pp. 57661-57673, 2024, doi: 10.1109/ACCESS.2024.3379760.