Bone Fracture Detection

Giuglan Cătălin-Ionuț
Faculty of Automatic Control and Computer Science
UNSTPB
Bucharest, Romania
catalingiuglan@yahoo.com

Brutaru-Mihăilișca Bogdan-Alexandru
Faculty of Automatic Control and Computer Science
UNSTPB
Bucharest, Romania
brutaru.m.bogdan.a@gmail.com

I. Introduction

In Milestone 2, we established a baseline YOLOv8 model for binary bone fracture detection. While the model performed adequately on the training set, evaluation on the test and validation sets revealed significant drops in performance, indicating overfitting and limited generalization. The dataset, although of good quality and moderately balanced after class merging, presented challenges such as high intra-class variation, subtle fracture patterns, and variability in image quality and patient positioning.

Several parameter combinations were explored, including different input resolutions, batch sizes, and YOLOv8 variants, yet most configurations resulted in either underfitting or overfitting.

II. Dataset Modification

In order to better reflect the real-world distribution of fractures in clinical settings, we modified the original dataset's class balance. While the initial dataset contained an approximately equal number of fractured and non-fractured X-ray images (roughly 50-50%), we adjusted the distribution to be closer to 10% non-fractured and 90% fractured images.

This adjustment aims to simulate more realistic clinical scenarios, where fractures are relatively rare compared to healthy cases, and allows the model to better learn to distinguish subtle fracture patterns without overfitting to an artificially balanced dataset. The modification was implemented by selectively reducing the number of fracture images and ensuring that both training and validation sets reflected the new distribution.

Despite this adjustment, one major limitation remained: the absolute number of images in the dataset was still relatively small. This restricted the model's ability to generalize to unseen X-rays, particularly for rare fracture types, and likely contributed to the lower mAP, precision, and recall observed on the test and validation sets.

III. Contribution Overview

In this milestone, our main contribution is the systematic optimization of the YOLOv8 model for bone fracture detection. We focused on evaluating the impact of key hyperparameters, particularly input image size ('imgsz') and batch size, on model performance. For each configuration, we recorded metrics including training and validation loss, precision, recall, and mAP, providing a comprehensive comparison to the baseline model established in Milestone 2.

The hyperparameter choices were guided by both practical constraints and empirical observations. For instance, smaller input image sizes allowed faster experimentation and reduced overfitting on limited data, while larger batch sizes stabilized gradient updates and improved feature learning. Each experiment generated valuable insights into the sensitivity of YOLOv8 to these parameters, revealing how subtle adjustments can impact the model's ability to detect small and faint fractures.

All experiments were conducted using the following common configuration:

- Model: YOLOv8n, Epochs: 150, Patience: 30
- Optimizer: auto, Task: detect, Pretrained: true

To clearly illustrate the effects of these hyperparameters, we generated two main visualizations. The first graph, shown in Figure 1, compares multiple YOLOv8 configurations with varying image sizes and batch sizes, highlighting trends in loss, precision, recall, and mAP across all experiments.

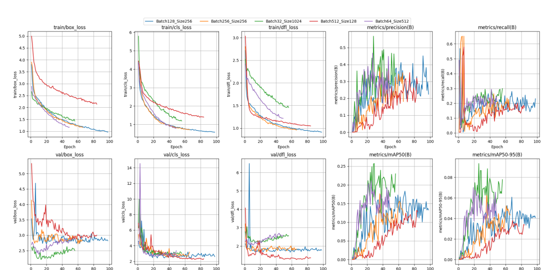


Fig. 1: YOLOv8 performance for various image sizes and batch sizes.

The second graph, shown in Figure 2, presents results for YOLOv8 combined with a SimCLR pretraining approach. This comparison provides visual evidence of how self-supervised pretraining can influence model performance on fracture detection.

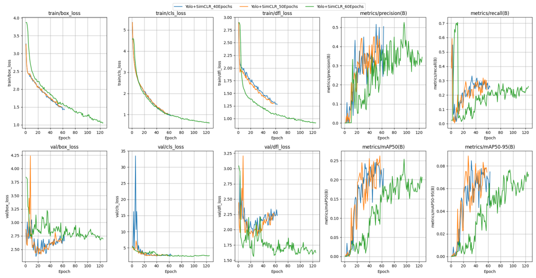


Fig. 2: Performance of YOLOv8 combined with SimCLR pretraining.

IV. Best Configuration and Results

From our experiments, the best performing configuration was obtained by first pretraining the model with SimCLR for 60 epochs to improve feature representation, followed by training YOLOv8n with a batch size of 512 and an input image size of 256 for 150 epochs. All other parameters remained consistent with the previous experiments (optimizer: auto, task: detect, pretrained: true).

Metric	Train Set	Test Set	Valid Set
mAP@0.5	0.9196	0.1549	0.2251
mAP@0.5:0.95	0.5891	0.0421	0.0730
Precision	0.9343	0.2578	0.4894
Recall	0.8304	0.2500	0.2206

TABLE I: Performance metrics for the best YOLOv8n + SimCLR configuration.

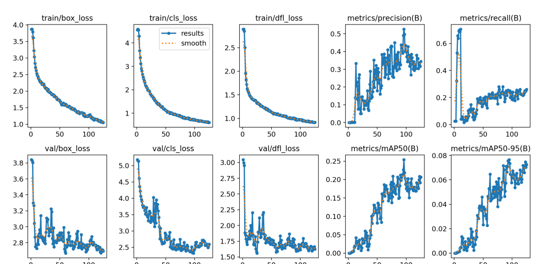
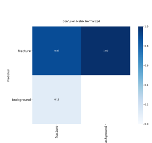
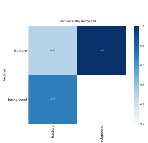


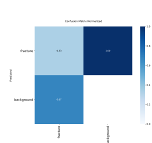
Fig. 3: Training results for our best YOLOv8n + SimCLR configuration.



(a) Train set



(b) Test set



(c) Validation set

Fig. 4: Confusion matrices for the best YOLOv8n + SimCLR configuration on train, test, and validation sets.

V. Comparison with Baseline

Metric	Train Set	Test Set	Valid Set
mAP@0.5	0.5013	0.1854	0.2501
mAP@0.5:0.95	0.2220	0.0625	0.0813
Precision	0.6349	0.2367	0.4555
Recall	0.4516	0.1979	0.2549

TABLE II: Baseline YOLOv8 performance from Milestone 2.

Compared to the baseline, our optimized configuration demonstrates a significant improvement on the training set:

- mAP@0.5 increased from 0.5013 to 0.9196
- mAP@0.5:0.95 increased from 0.2220 to 0.5891
- Precision improved from 0.6349 to 0.9343
- Recall improved from 0.4516 to 0.8304

VI. Limitations

Despite improvements from hyperparameter tuning and SimCLR pretraining, several limitations constrained the model's performance. First, the dataset remains relatively small, with limited examples for certain fracture types and orientations, which reduces generalization capability. Second, the variability in X-ray image quality, patient positioning, and subtle fracture patterns makes detection inherently challenging. Finally, while our experiments explored multiple configurations, the model sometimes struggled with borderline cases where fractures are faint or partially obscured.

VII. Our Contribution

Our contribution can be summarized as a combination of methodical experimentation, careful analysis, and practical insights into bone fracture detection using deep learning. Specifically, we:

- Conducted a comprehensive hyperparameter study for YOLOv8, exploring the effects of input image size and batch size on model performance.
- Applied SimCLR self-supervised pretraining to improve feature representation, enhancing detection of subtle fracture patterns.
- Produced detailed visual and quantitative analysis of model performance, including confusion matrices and performance graphs for multiple batch and image sizes.
- Offered an informed discussion of dataset limitations, including class imbalance and variability in image quality, highlighting their impact on generalization.

In conclusion, our work demonstrates how careful hyperparameter tuning and self-supervised pretraining can be combined to extract meaningful features from limited X-ray data, providing a solid foundation for future improvements in automated bone fracture detection.

References

[1] Justin Bolivar, "Bone Fracture Detection Using YOLOv8", Kaggle, 2024. https://www.kaggle.com/code/justinbolivar/bone-fracture-detection-using-yolov8/notebook