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

***Abstract*—Bone fracture detection in X-ray images is a clinically important yet challenging task due to large variations in anatomy, imaging conditions, and the often subtle visual appearance of fractures. Although deep learning techniques have shown promising results in medical image analysis, their performance is frequently limited by small datasets, class imbalance, and poor generalization to unseen data. In this paper, we investigate automated binary bone fracture detection using the YOLOv8 object detection framework, with a focus on improving feature representation and training stability under constrained data conditions.**

Starting from a baseline YOLOv8 model, we perform a systematic analysis of key hyperparameters, including input image resolution and batch size, to evaluate their impact on detection performance. Furthermore, we incorporate SimCLR-based self-supervised pretraining to enhance the quality of learned representations before supervised training. Experiments conducted on a publicly available X-ray dataset demonstrate that self-supervised pretraining significantly improves training convergence and feature learning, while also highlighting persistent challenges related to overfitting and limited generalization.

The results provide practical insights into the strengths and limitations of YOLO-based approaches for fracture detection and underline the importance of representation learning and careful model optimization when working with limited medical imaging data.

I. Introduction

Bone fractures represent one of the most common injuries encountered in clinical practice, affecting patients of all ages and anatomical regions. An accurate and timely fracture diagnosis is essential to ensure appropriate treatment, prevent complications such as malunion or nonunion, and reduce long-term disability. X-ray imaging remains the main diagnostic modality for fracture assessment due to its wide availability, low cost, and fast acquisition time. However, interpreting radiographs is a demanding task that requires significant expertise and attention, particularly in cases involving subtle fractures, overlapping anatomical structures, or suboptimal image quality.

Despite advances in medical imaging technology, manual analysis of X-ray images is still prone to diagnostic errors. Studies have shown that even experienced radiologists can miss fractures, especially when the fracture lines are faint or partially obscured by surrounding tissue or bone structures. These challenges motivate the development of automated fracture detection systems that can help clinicians by providing reliable and consistent decision support.

Early research in automated bone fracture detection relied primarily on classical image processing techniques, including edge detection, thresholding, segmentation, and handcrafted feature extraction. While these approaches demonstrated initial success, their performance was limited by sensitivity to noise, variations in exposure, and anatomical diversity across patients. As a result, their ability to generalize across datasets and clinical settings remained restricted.

The emergence of deep learning, particularly convolutional neural networks (CNNs), marked a significant shift in medical image analysis. CNN-based models are capable of learning hierarchical feature representations directly from data, reducing the need for manual feature engineering and improving robustness to image variability. Numerous studies have applied CNN architectures such as AlexNet, ResNet, and MobileNet to fracture detection tasks, often using transfer learning to mitigate the challenges posed by limited medical datasets. These approaches have demonstrated improved accuracy compared to classical methods, but often require substantial computational resources and large annotated datasets.

More recently, object detection frameworks have gained attention in fracture detection research. Among these, the You Only Look Once (YOLO) family of models has become particularly popular due to its ability to perform fast and accurate detection within a single forward pass. YOLO-based approaches have been successfully applied to various medical imaging tasks, including fracture detection, benefiting from real-time performance and strong localization capabilities. YOLOv8, one of the latest generation in this series, offers improved architectural design and training stability, making it a suitable candidate for medical applications under constrained hardware conditions.

Despite these advances, several challenges remain unresolved. Medical imaging datasets are often small, imbalanced, and heterogeneous, leading to overfitting and poor generalization of unseen data. In the context of fracture detection, this issue is worsened by high intra-class variability and the subtle visual appearance of certain fracture types. Furthermore, many existing studies focus on multi-class fracture classification, while binary fracture detection remains underexplored despite its relevance for initial screening and clinical triage.

In this work, we address these challenges by investigating automated binary bone fracture detection using the YOLOv8 framework. Rather than proposing a completely new ar-

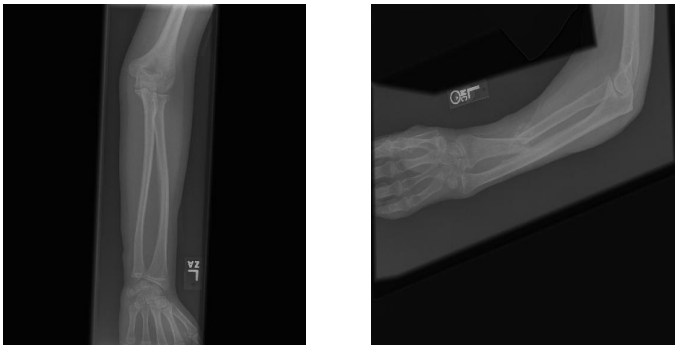
chitecture, our focus is on systematically improving model performance through careful optimization and representation learning. We explore the influence of key hyperparameters, such as input image size and batch size, on training stability and detection accuracy. Additionally, we incorporate self-supervised learning through SimCLR pretraining to enhance feature representation before supervised training.

The main contributions of this paper can be summarized as follows:

- We focus on binary fracture detection using a publicly available dataset and provide an additional discussion of its clinical limitations.
- We conduct a systematic hyperparameter study to evaluate the sensitivity of YOLOv8 to input resolution and batch size under limited data conditions.
- We apply SimCLR-based self-supervised pretraining to improve feature learning and assess its impact on fracture detection performance.
- We provide a detailed experimental analysis highlighting both performance gains and remaining challenges related to generalization and overfitting.

Through this study, our objective is to provide practical information on the application of modern object detection techniques for the detection of bone fractures and to establish a solid foundation for future improvements in automated medical image analysis.

To illustrate the visual variability encountered in practice, Figure 1 shows two representative examples from the dataset: one with a healthy bone and one containing a clear fracture line. Even in these relatively clean cases, the difference between fractured and non-fractured structures may be subtle, highlighting the difficulty of reliable fracture detection and the need for automated methods capable of learning fine-grained visual patterns.



(a) Normal bone X-ray (b) Fractured bone X-ray

Fig. 1: Visual comparison between a healthy bone and a fractured bone. The subtle appearance of fracture lines makes automated detection challenging, especially under variable image quality and anatomical complexity.

II. Related Work

Automated bone fracture detection has been an active research topic for several decades, evolving from traditional image processing techniques to modern deep learning and object detection frameworks. This section reviews the main categories of approaches proposed in the literature, highlighting their strengths and limitations, and positioning our work within the current research landscape.

A. Classical Image Processing Approaches

Early research on fracture detection relied primarily on classical image processing methods. These approaches typically employed edge detection, thresholding, morphological operations, and segmentation techniques to identify discontinuities in bone structures visible in X-ray images [12], [14]. Extracted handcrafted features were subsequently classified using traditional machine learning algorithms such as support vector machines, k-nearest neighbors, or ensemble-based classifiers [8], [13].

Although these methods provided valuable initial insights, their performance was highly sensitive to image noise, contrast variations, and differences in anatomical structures across patients. Furthermore, handcrafted features often failed to capture subtle fracture patterns, limiting generalization across datasets and clinical settings [12]. As a result, classical image processing approaches struggled to achieve reliable performance in real-world applications.

B. CNN-Based Fracture Detection

The introduction of convolutional neural networks (CNNs) marked a significant advancement in automated fracture detection. CNN-based models enabled automatic learning of hierarchical feature representations directly from X-ray images, reducing dependence on manual feature engineering. Early deep learning approaches demonstrated improved accuracy over classical techniques for fracture classification tasks [1], [9].

Subsequent studies explored deeper architectures and transfer learning strategies to further improve performance. Models based on architectures such as ResNet and MobileNet were trained using pretrained weights, allowing effective learning even with limited medical datasets [2], [10], [15]. Data augmentation techniques, including rotation, flipping, and scaling, were commonly applied to mitigate overfitting and increase robustness [3], [7]. These CNN-based approaches achieved notable improvements in classification accuracy, particularly for binary fracture detection.

Despite these successes, CNN-based methods often focus solely on image-level classification and do not provide explicit localization of fracture regions. This limitation can reduce interpretability and clinical usefulness, especially in scenarios where spatial context is important for diagnosis.

C. Object Detection and YOLO-Based Approaches

To address the lack of localization in classification models, recent research has increasingly adopted object detection frameworks for fracture detection. Object detectors provide both classification and spatial localization, offering more interpretable outputs. Among these frameworks, the You Only

Look Once (YOLO) family has gained widespread adoption due to its ability to perform real-time detection within a single forward pass [5].

Several studies have successfully applied YOLO-based models to bone fracture detection tasks. YOLOv8 and its variants have demonstrated competitive accuracy and robustness across different anatomical regions, while maintaining high inference speed suitable for clinical and mobile healthcare applications [4], [6], [11]. Comparative evaluations of YOLO variants further highlight the importance of architectural choices and parameter optimization for achieving stable performance [5].

However, many YOLO-based fracture detection studies focus on multi-class classification and assume relatively balanced datasets. In practice, medical imaging datasets are often small, imbalanced, and heterogeneous, which can lead to overfitting and reduced generalization when models are deployed in real-world clinical environments.

D. Self-Supervised Learning and Representation Learning

Self-supervised learning has emerged as a promising approach for improving representation learning in scenarios with limited labeled data. Contrastive learning methods aim to learn meaningful features by maximizing agreement between different augmented views of the same image, without relying on explicit annotations. Although self-supervised learning has been successfully applied in various computer vision and medical imaging domains, its adoption in bone fracture detection remains limited.

Most existing fracture detection studies rely exclusively on supervised learning paradigms, leaving an open research opportunity to explore whether self-supervised pretraining can improve feature extraction, training stability, and generalization in fracture detection tasks. This gap is particularly relevant for object detection frameworks such as YOLO, which require strong feature representations to identify subtle visual patterns.

E. Summary and Research Gap

In summary, prior work demonstrates a clear evolution from classical image processing techniques [12], [14] to CNN-based classification models [1], [9], [15] and, more recently, to YOLO-based object detection approaches [4], [6], [11]. While modern deep learning methods have significantly improved fracture detection performance, challenges related to limited data availability, class imbalance, and generalization persist.

Motivated by these limitations, our work focuses on binary bone fracture detection using YOLOv8, emphasizing systematic hyperparameter optimization and the integration of self-supervised SimCLR pretraining. By combining representation learning with careful model tuning, we aim to provide a practical contribution that addresses current gaps in fracture detection research.

III. Methodology

This section describes the methodology adopted for automated bone fracture detection, including the dataset used, preprocessing steps, baseline model configuration, and the proposed improvements applied to enhance performance. The overall approach follows a structured experimental pipeline designed to operate under limited data and hardware constraints.

A. Dataset Description

The experiments were conducted using a publicly available bone fracture X-ray dataset obtained from Kaggle [16]. The dataset contains approximately 3600 grayscale radiographic images acquired from multiple anatomical regions, such as wrist, ankle, and shoulder. Each image is provided together with annotation files formatted for compatibility with YOLO-based object detection frameworks.

The dataset exhibits significant variability in terms of image resolution, contrast, and patient positioning, reflecting real-world clinical conditions. Although the dataset quality is generally good, the limited number of samples and uneven representation of anatomical regions pose challenges for training deep learning models with strong generalization capabilities.

B. Label Simplification and Preprocessing

Originally, the dataset was annotated with multiple fracture categories corresponding to specific fracture types or anatomical locations. Since the objective of this work is binary fracture detection, all fracture-related classes were merged into a single *fracture* class, while images without fractures were assigned to a *non-fracture* class. This simplification focuses the task on identifying the presence of fractures rather than performing fine-grained classification.

To better reflect real-world clinical distributions, the dataset was rebalanced from an approximately equal class distribution to a more realistic configuration containing roughly 90% fractured images and 10% non-fractured images. This rebalancing was applied consistently across the training and validation sets.

Preprocessing steps included removal of corrupted or unreadable images, resizing all images to standardized input resolutions depending on the experimental configuration, and normalization of pixel intensities. After label merging and rebalancing, annotation files were manually verified to ensure correctness and consistency.

C. Baseline Model

As a baseline approach, we employed the YOLOv8 object detection framework due to its strong performance in detection tasks and favorable balance between accuracy and computational efficiency. Specifically, the YOLOv8n variant was selected because of its lightweight architecture, making it suitable for training under limited hardware resources.

The baseline model was initialized using pretrained weights and trained with the following configuration: task set to detection, optimizer selection set to automatic, input image size fixed at 256×256 pixels, batch size of 16, and a maximum of 150 training epochs. Early stopping with a patience of 30 epochs was used to prevent excessive overfitting. This configuration served as the reference point for all subsequent experiments.

D. Proposed Approach

To improve upon the baseline performance, we explored two complementary strategies: systematic hyperparameter optimization and self-supervised representation learning.

First, we conducted a hyperparameter study focusing on input image size and batch size. Smaller input resolutions were evaluated to reduce model complexity and overfitting, while larger batch sizes were tested to improve gradient stability and feature learning. All experiments were trained for the same number of epochs to ensure fair comparison across configurations.

Second, we incorporated self-supervised learning through SimCLR pretraining to enhance feature representation prior to supervised detection training. SimCLR employs contrastive learning to maximize agreement between different augmented views of the same image, enabling the model to learn meaningful representations without requiring labeled data. In our approach, the backbone network was pretrained using SimCLR for 60 epochs before being fine-tuned within the YOLOv8 detection framework.

By combining hyperparameter optimization with self-supervised pretraining, the proposed methodology aims to improve feature extraction, training stability, and overall detection performance, particularly in scenarios characterized by limited data availability and high intra-class variability.

IV. Experiments

This section outlines the experimental setup used to evaluate our fracture detection approach, including the training procedure, the tested configurations, and the metrics used to assess the performance of the model. The experiments are based on the baseline configuration developed in Milestone 2 and the optimized configurations explored in Milestone 3. All experiments were conducted using identical evaluation metrics and training procedures to ensure fair analysis.

A. Dataset Splits and Class Distribution

The dataset contains approximately 3600 X-ray images and was split into training, validation, and test subsets using a standard 70% / 15% / 15% ratio.

In Milestone 2, the dataset was configured with an approximately balanced class distribution, containing roughly 50% fractured images and 50% non-fractured images in the training directory. This setup was used to establish a baseline and to analyze initial model behavior.

In Milestone 3, the class distribution was modified to better reflect real-world clinical scenarios. The dataset was rebalanced to approximately 10% non-fractured images and 90% fractured images. This modification was applied to the training and validation sets, while the test set was kept unchanged to preserve unbiased evaluation.

B. Experimental Setup

All experiments were performed using the YOLOv8 object detection framework. The YOLOv8n variant was selected due to its lightweight architecture and suitability for limited hardware resources.

For Milestone 2, the model was initialized with pretrained weights and trained for up to 150 epochs with early stopping (patience of 30 epochs). The main training settings were:

- Model: yolov8n.pt
- Epochs: 150
- Batch size: 16
- Image size: 256x256
- Optimizer: auto
- Pretrained: true
- Other standard YOLOv8 defaults (cosine LR, augmentation off, etc)

In Milestone 3, we also introduced a SimCLR-based self-supervised pretraining stage to improve feature quality before running YOLOv8 detection. The idea was to train the early backbone layers to recognize general radiographic patterns without using labels. Each image was augmented twice, producing two different views of the same X-ray, and the model learned to bring their representations closer together. After this pretraining phase, the resulting backbone weights were transferred into YOLOv8 and used as initialization for the supervised fracture detection training.

All experiments were conducted using consumer-grade GPUs with 6 GB of memory and supplemented by Google Colab free-tier resources.

C. Evaluation Metrics

Performance was evaluated using standard object detection metrics. The primary metric was the Mean Average Precision at 0.5 (mAP@0.5), which measures how accurately the model detects and localizes fractures by combining both precision and recall into a single score. We also report mAP@0.5:0.95, which reflects performance across multiple confidence thresholds and provides a more comprehensive view of detection quality.

In addition, precision and recall were computed to assess false positive and false negative behavior, an essential aspect in medical imaging applications.

D. Baseline Results (Milestone 2)

The results obtained for the baseline YOLOv8n model trained on the balanced dataset are summarized in Table I. The corresponding training curves and prediction visualizations are shown in Figure 2.

Metric	Train Set	Test Set	Validation 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 I: Baseline YOLOv8

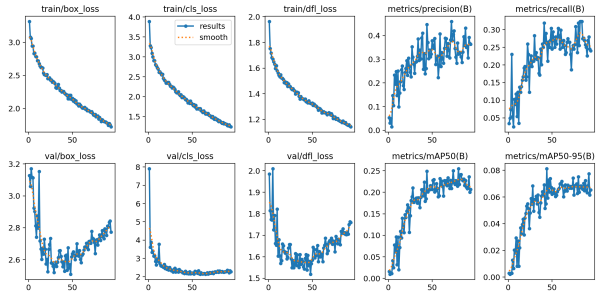


Fig. 2: Training metrics and prediction examples for the baseline YOLOv8 model (Milestone 2).

E. Optimized Results (Milestone 3)

The quantitative results for the best-performing configuration obtained in Milestone 3 are summarized in Table II. The corresponding training curves and prediction visualizations are shown in Figure 3. This configuration includes SimCLR pretraining and training on the rebalanced dataset.

Metric	Train Set	Test Set	Validation 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 II: YOLOv8n + SimCLR

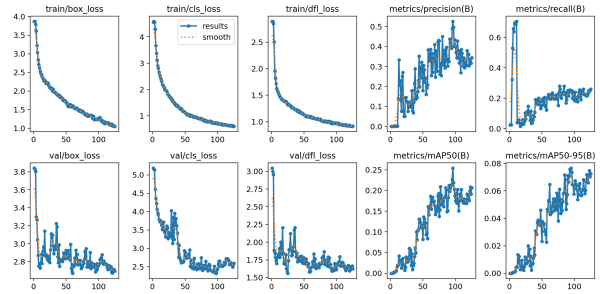


Fig. 3: Training metrics and prediction examples for the optimized YOLOv8n + SimCLR model (Milestone 3).

F. Experimental Observations

The experimental results highlight the impact of dataset balancing, hyperparameter tuning, and self-supervised pretraining on fracture detection performance. Training on a balanced dataset resulted in more stable validation metrics, while rebalancing improved fracture sensitivity at the cost of increased overfitting risk.

Larger batch sizes and SimCLR pretraining improved training convergence and feature representation; however, generalization remained limited by dataset size, variability in X-ray quality, and the subtle nature of fracture patterns.

V. Own Contribution

The primary contribution of this project consists of a structured and systematic optimization of a YOLOv8-based bone fracture detection pipeline. Our work focuses on experimentally analyzing the impact of training configurations, self-supervised pretraining, and dataset composition, rather than proposing a new detection architecture. All contributions are supported by quantitative metrics and visual analysis.

A. Hyperparameter Optimization of YOLOv8

A central contribution of this work is the comparative evaluation of multiple YOLOv8 configurations obtained by varying two key hyperparameters: input image resolution and batch size. These parameters were selected due to their strong influence on training stability, convergence speed, and generalization performance, especially under limited hardware resources.

Figure 4 presents a consolidated comparison of multiple YOLOv8 training configurations. The figure summarizes trends in loss, precision, recall, and mean Average Precision across different batch sizes and image resolutions.

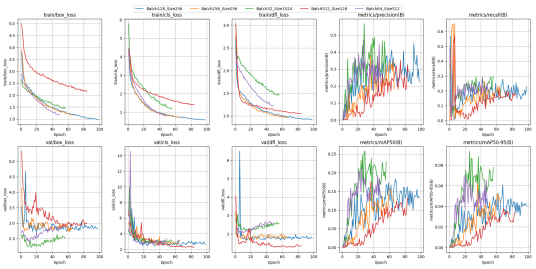


Fig. 4: Comparison of YOLOv8 performance across different batch sizes and input image resolutions.

This comparison allowed us to identify configurations that provide stable training behavior while minimizing overfitting. In particular, smaller image sizes combined with larger batch sizes showed improved convergence and reduced variance in training metrics.

B. Self-Supervised Pretraining with SimCLR

Another important contribution of this project is the integration and evaluation of SimCLR self-supervised pretraining prior to supervised YOLOv8 training. The backbone network was pretrained using contrastive learning for different numbers of epochs, allowing us to study how the duration of self-supervised training affects downstream fracture detection performance.

Figure 5 compares multiple YOLOv8 configurations with SimCLR pretraining, as well as different numbers of SimCLR pretraining epochs.

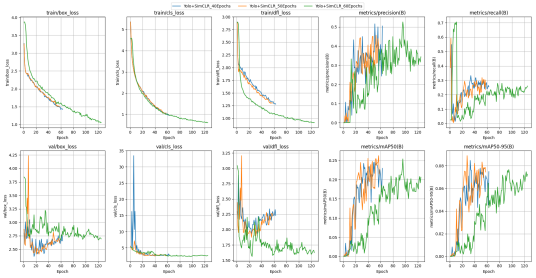


Fig. 5: Comparison of YOLOv8 performance with SimCLR pretraining for different numbers of pretraining epochs.

The results demonstrate that self-supervised pretraining improves feature representation and training convergence. Although generalization remains challenging, the inclusion of SimCLR consistently enhances training performance and accelerates learning.

C. Training Dynamics of the Optimized Configuration

After identifying the most promising configuration through hyperparameter tuning and self-supervised pretraining, we analyzed its training behavior in detail. Figure 6 illustrates the training loss and performance metrics for the best-performing YOLOv8n + SimCLR configuration.

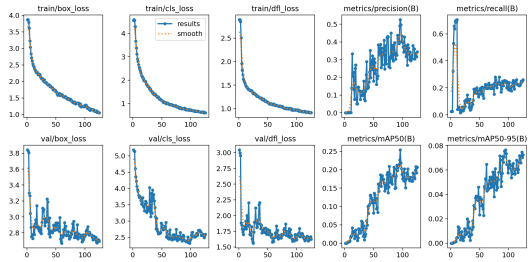


Fig. 6: Training results for the best YOLOv8n configuration combined with SimCLR pretraining.

The smooth convergence observed in this figure confirms the stability of the selected configuration and validates the optimization process employed in this project.

D. Confusion Matrix Analysis

To further assess model behavior beyond aggregate metrics, we performed a class-wise evaluation using confusion matrices for the training, test, and validation sets. These matrices provide insight into prediction bias, misclassification patterns, and generalization capability.

Figure 12 shows the confusion matrices obtained for each dataset split.

VI. Results and Discussion

This section analyzes the experimental results obtained during the development and optimization of the bone fracture detection system. The discussion focuses on performance trends, generalization behavior, and the impact of optimization strategies applied throughout the project.

A. Baseline versus Optimized Performance

Table III summarizes the performance of the baseline YOLOv8n model trained in Milestone 2 using a balanced dataset (50% fractured, 50% non-fractured). The results show moderate training performance but a substantial drop on the test and validation sets, indicating overfitting and limited generalization.

Metric	Train Set	Test Set	Validation 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 III: Baseline YOLOv8n performance using a balanced dataset (Milestone 2).

In contrast, Table IV presents the results of the optimized YOLOv8n configuration combined with SimCLR pretraining and trained on the rebalanced dataset (10% non-fractured, 90% fractured) in Milestone 3.

Metric	Train Set	Test Set	Validation 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 IV: Optimized YOLOv8n + SimCLR performance using a rebalanced dataset (Milestone 3).

B. Effect of Dataset Rebalancing

Comparing the results from Tables III and IV, it is evident that dataset rebalancing had a strong influence on model behavior. The rebalanced dataset significantly improved training performance, particularly in terms of precision and recall, indicating enhanced sensitivity to fracture patterns.

However, this improvement was not fully reflected in test and validation performance, where metrics remained relatively low. This behavior suggests that while the model learned fracture-specific features more effectively, the reduced diversity of non-fractured samples increased overfitting tendencies and reduced generalization.

C. Impact of SimCLR Pretraining

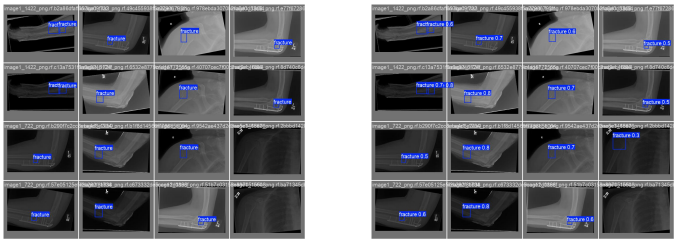
The integration of SimCLR self-supervised pretraining led to improved feature representation, as reflected by the higher training mAP and precision values in the optimized configuration. The contrastive learning phase enabled the backbone network to capture general radiographic patterns prior to supervised fine-tuning.

Despite these improvements, the gap between training and evaluation performance persisted. This observation indicates that self-supervised pretraining enhances learning efficiency but does not fully overcome the limitations imposed by small dataset size and high intra-class variability.

D. Generalization Analysis

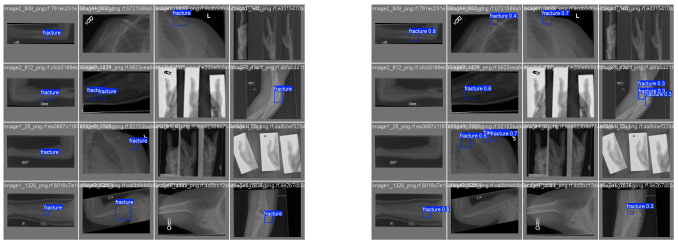
The persistent discrepancy between training and evaluation metrics across both baseline and optimized models highlights generalization as the primary challenge in this task. Even with optimized hyperparameters and pretraining, performance on unseen data remains limited.

This outcome is consistent with the inherent difficulty of fracture detection, where fractures may be subtle, partially occluded, or visually ambiguous. The results emphasize the need for larger and more diverse datasets, as well as advanced regularization and augmentation techniques.



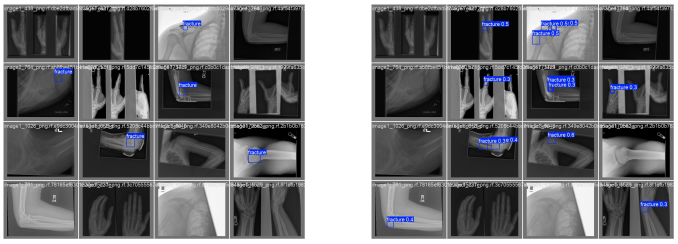
(a) Ground truth labels (b) Model predictions

Fig. 7: Qualitative comparison of ground truth labels and predictions on the train set.



(a) Ground truth labels (b) Model predictions

Fig. 8: Qualitative comparison of ground truth labels and predictions on the test set.



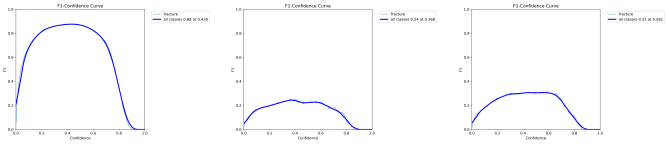
(a) Ground truth labels (b) Model predictions

Fig. 9: Qualitative comparison of ground truth labels and predictions on the validation set.

E. F1-Score and Precision-Recall Analysis

To complement the standard detection metrics, we further analyzed the model using F1-score curves and Precision–Recall (PR) curves for the training, test, and validation sets. These metrics provide a more balanced view of model performance, particularly under class imbalance conditions.

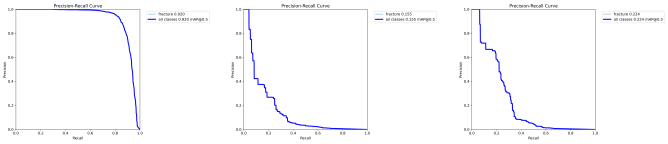
Figure 10 presents the F1-score evolution across dataset splits. The training set achieves consistently high F1 values, reflecting strong optimization and class separation. In contrast, both the test and validation sets exhibit lower peak F1 scores, confirming reduced generalization performance.



(a) Train (b) Test (c) Validation

Fig. 10: F1-score curves for the optimized YOLOv8n + SimCLR model.

Figure 11 shows the Precision-Recall curves for the same dataset splits. The training PR curve demonstrates high precision across a wide recall range, while the test and validation curves degrade more rapidly. This behavior indicates an increased number of false positives and false negatives on unseen data, which is consistent with the confusion matrix and mAP results.



(a) Train (b) Test (c) Validation

Fig. 11: Precision-Recall curves for the optimized YOLOv8n + SimCLR model.

F. Comparison Across Training, Test, and Validation Sets

A detailed comparison between the training, test, and validation results reveals a consistent performance gap across all evaluated metrics. As shown in Tables III and IV, both the baseline and optimized models achieve substantially higher performance on the training set compared to unseen data.

For the optimized YOLOv8n + SimCLR configuration, the training mAP@0.5 exceeds 0.91, while the corresponding test and validation scores remain below 0.23. This discrepancy indicates that although the model successfully learns fracture-specific visual patterns, these representations do not fully generalize to new samples.

The validation set generally achieves slightly better performance than the test set, suggesting that it remains closer to the training distribution. In contrast, the test set appears more challenging, likely due to increased variability in fracture appearance, imaging conditions, and anatomical regions.

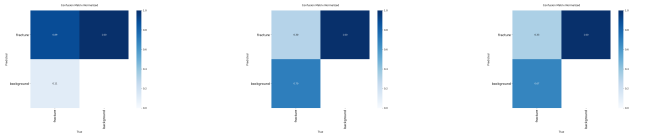
G. Confusion Matrix Interpretation

Figure 12 presents the confusion matrices obtained for the training, test, and validation sets using the optimized YOLOv8n + SimCLR model.

The training confusion matrix demonstrates strong class separation, with a high number of true positive detections and a low false negative rate. This confirms that the model effectively learns discriminative features for fracture detection during training.

In contrast, the test and validation confusion matrices show a noticeable increase in misclassifications. False negatives are more frequent, indicating missed fractures, while false positives also increase, suggesting reduced confidence in distinguishing fractured from non-fractured samples on unseen data. This behavior reflects the model's sensitivity to subtle fracture patterns and highlights limitations in generalization caused by dataset size and imbalance.

Overall, the confusion matrix analysis confirms that overfitting remains a primary challenge, despite the use of self-supervised pretraining and hyperparameter optimization.



(a) Train set (b) Test set (c) Validation set

Fig. 12: Confusion matrices for the optimized YOLOv8n + SimCLR configuration on training, test, and validation sets.

H. Key Observations

Based on the comparative analysis across dataset splits, several conclusions can be drawn:

- Training performance consistently overestimates real-world performance due to limited dataset diversity.
- Validation results provide a more reliable estimate than test results but still indicate limited generalization.
- Self-supervised pretraining improves learning stability but does not eliminate overfitting.
- Fracture detection remains highly sensitive to subtle visual differences and imaging variability.

I. Clinical Implications and Limitations

From a clinical perspective, the results demonstrate that while deep learning-based fracture detection systems can effectively learn fracture-related features, their reliability on unseen cases remains a concern. High false negative or false positive rates could have significant consequences in medical decision-making.

Therefore, the proposed system should be viewed as a supportive tool rather than a standalone diagnostic solution. Future improvements should focus on improving generalization, incorporating domain-specific augmentations, and validating performance on larger, multi-center datasets.

Overall, the results highlight both the potential and the current limitations of applying YOLO-based object detection models to medical imaging tasks.

J. Summary of Contributions

Overall, this project contributes:

- a systematic comparison of YOLOv8 configurations based on batch size and image resolution,
- an experimental evaluation of SimCLR self-supervised pretraining for medical image detection,
- a detailed visual analysis using training curves and confusion matrices,
- practical insights into optimizing deep learning models under limited computational resources.

These contributions provide a reproducible and well-structured framework for improving fracture detection systems and serve as a solid foundation for future research.

VII. Conclusion

In this paper, we investigated the problem of automatic bone fracture detection from X-ray images using deep learning-based object detection techniques. Starting from a baseline YOLOv8 model trained on a balanced dataset, we systematically explored optimization strategies aimed at improving fracture-specific feature learning under constrained data and hardware conditions.

Through a series of experiments, we demonstrated that dataset composition plays a crucial role in model behavior. While a balanced class distribution provided more stable evaluation metrics, rebalancing the dataset toward fractured samples improved the model's sensitivity to fracture patterns, at the cost of increased overfitting. Additionally, we showed that careful hyperparameter tuning, particularly batch size and input image resolution, significantly influenced training stability and convergence.

A key contribution of this work is the integration of SimCLR self-supervised pretraining with YOLOv8. This approach enhanced feature representation and substantially improved training performance, confirming the benefits of contrastive learning for medical imaging tasks where labeled data is limited. However, despite these improvements, a noticeable gap between training and evaluation performance persisted, highlighting generalization as the primary challenge.

The results emphasize the difficulty of fracture detection, where fractures can be subtle, visually ambiguous, and highly variable across patients and imaging conditions. While the proposed system demonstrates promising learning capability, it is not yet suitable for standalone clinical deployment and should be considered as a decision-support tool rather than a diagnostic replacement.

Future work could focus on improving generalization by expanding the dataset, incorporating more advanced data augmentation techniques, and exploring hybrid architectures. Additionally, validating the model on external datasets from different clinical sources would be an essential step toward assessing real-world applicability.

Overall, this work provides a solid experimental foundation for further research in automated bone fracture detection using deep learning.

REFERENCES

[1] D. P. Yadav and S. Rathor, "Bone Fracture Detection and Classification using Deep Learning Approach," in *2020 International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC)*, Mathura, India, 2020, pp. 282–285, doi: 10.1109/PARC49193.2020.236611.

[2] A. Kheaksong, P. Sanguansat, P. Samothai, T. Dindam, K. Srisomboon, and W. Lee, "Analysis of Modern Image Classification Platforms for Bone Fracture Detection," in *2022 6th International Conference on Information Technology (InCIT)*, Nonthaburi, Thailand, 2022, pp. 471–474, doi: 10.1109/InCIT56086.2022.10067836.

[3] L. Bisht, S. Katiyar, and Jyoti, "Bone Fracture Detection Using Python," in *2024 International Conference on Artificial Intelligence and Quantum Computation-Based Sensor Application (ICAIQSA)*, Nagpur, India, 2024, pp. 1–6, doi: 10.1109/ICAIQSA64000.2024.10882387.

[4] B. J, I. V. S. P. Varma, and A. Anand, "Bone Fracture Detection using YOLOv8 and OpenCV," in *2024 International Conference on Emerging Research in Computational Science (ICERCS)*, Coimbatore, India, 2024, pp. 1–5, doi: 10.1109/ICERCS63125.2024.10895558.

[5] P. Samothai, P. Sanguansat, A. Kheaksong, K. Srisomboon, and W. Lee, "The Evaluation of Bone Fracture Detection of YOLO Series," in *2022 37th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*, Phuket, Thailand, 2022, pp. 1054–1057, doi: 10.1109/ITC-CSCC55581.2022.9895016.

[6] S. A. Alqazzaz, A. A. A. A. Al-obaidi, Z. Al-Ibadi, and A. R. H. Khayeat, "Multi-Category Bone Fracture Detection Based on Deep Learning in X-ray Imaging Using YOLOv8s," in *2024 IEEE 9th International Conference on Engineering Technologies and Applied Sciences (ICETAS)*, Bahrain, 2024, pp. 1–4, doi: 10.1109/ICETAS62372.2024.11120177.

[7] K. Mittal, K. Singh Gill, P. Aggarwal, R. Singh Rawat, and G. Sunil, "Revolutionizing Fracture Diagnosis: A Deep Learning Approach for Bone Fracture Detection and Classification," in *2024 OPJU International Technology Conference (OTCON) on Smart Computing for Innovation and Advancement in Industry 4.0*, Raigarh, India, 2024, pp. 1–5, doi: 10.1109/OTCON60325.2024.10688357.

[8] R. Bagaria, S. Wadhwani, and A. K. Wadhwani, "Different Techniques for Identification of a Bone Fracture in Analysis of Medical Image," in *2020 IEEE 9th International Conference on Communication Systems and Network Technologies (CSNT)*, Gwalior, India, 2020, pp. 327–332, doi: 10.1109/CSNT48778.2020.9115760.

[9] S. Chauhan, "Bone Fracture Detection with CNN: A Deep Learning Approach," in *2024 5th International Conference on Smart Electronics and Communication (ICOSEC)*, Trichy, India, 2024, pp. 1253–1258, doi: 10.1109/ICOSEC61587.2024.10722699.

[10] I. M, V. I, A. J, P. P, and R. J, "Deep Learning Model to Detect and Classify Bone Fracture in X-Ray Images," in *2023 International Conference on System, Computation, Automation and Networking (ICSCAN)*, Puducherry, India, 2023, pp. 1–6, doi: 10.1109/ICSCAN58655.2023.10394986.

[11] P. Agarwal and P. Kumar, "Automated Bone Fracture Detection using YOLOv8," in *2025 3rd International Conference on Disruptive Technologies (ICDT)*, Greater Noida, India, 2025, pp. 1256–1260, doi: 10.1109/ICDT63985.2025.10986404.

[12] R. S. Upadhyay and P. Tanwar, "A Review on Bone Fracture Detection Techniques using Image Processing," in *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, Madurai, India, 2019, pp. 287–292, doi: 10.1109/ICCS45141.2019.9065874.

[13] V. L. F. Lum, W. K. Leow, Y. Chen, T. S. Howe, and M. A. Png, "Combining classifiers for bone fracture detection in X-ray images," in *IEEE International Conference on Image Processing*, Genova, 2005, pp. I-1149, doi: 10.1109/ICIP.2005.1529959.

[14] R. Kapse, S. Daware, R. Bawane, T. Cholkar, and D. Bambawale, "Image Processing for Detecting Bone Fractures," in *2023 4th International Conference on Electronics and Sustainable Communication Systems (ICESC)*, Coimbatore, India, 2023, pp. 1297–1302, doi: 10.1109/ICESC57686.2023.10193303.

[15] S. Thota, P. Kandukuru, M. Sundaram, A. Ali, S. M. Basha, and N. Hima Bindu, "Deep Learning based Bone Fracture Detection," in *2024 International Conference on Smart Systems for Applications in Electrical Sciences (ICSSES)*, Tumakuru, India, 2024, pp. 1–7, doi: 10.1109/ICSSES62373.2024.10561360.

[16] P. K. Darabi, "Bone Fracture Detection — Computer Vision Project," Kaggle Dataset, 2023. [Online]. Available: https://www.kaggle.com/datasets/pkdarabi/bone-fracture-detection-computer-vision-project