*AI Medicine*
https://www.sciltp.com/journals/aim

*Article*

# Comparative Analysis of Object Detection Frameworks for Fracture Detection in X-Ray Image

Zhihao Liu and Ruyi Zhang *

College of Medicine and Biological Information Engineering, Northeastern University, Shenyang 110016, China
* Correspondence: 2390160@stu.neu.edu.cn

**How To Cite**: Liu, Z.; Zhang, R. Comparative Analysis of Object Detection Frameworks for Fracture Detection in X-Ray Image. *AI Medicine* **2025**, *2*(2), 5. https://doi.org/10.53941/aim.2025.100005

Received: 6 July 2025
Revised: 12 September 2025
Accepted: 18 September 2025
Published: 25 September 2025

**Abstract:**Fracture detection plays a critical role in clinical examinations, especially in emergency surgery. Traditional fracture diagnosis relies on the experience of radiologists, which carries the risk of misdiagnosis. With the advancement of deep learning technologies, object detection methods have been widely applied to automated fracture detection, providing efficient and accurate solutions. This study aims to evaluate the performance of various object detection frameworks in the task of fracture detection by comparing one-stage and two-stage detectors, anchor-based and anchor-free methods. Thus, we selected nine representative object detection models for comparison, covering a variety of deep-learning architectures. Experimental results show that YOLOv10 not only achieves the highest accuracy but also demonstrates significant advantages in inference speed. Furthermore, Transformer-based models exhibit better precision in fracture detection, particularly showing potential in recognizing complex image features.

**Keywords:** deep learning; fracture detection; medical imaging; object detection

## 1. Introduction

Fractures refer to the partial or complete disruption of the continuity or integrity of bones caused by external forces or pathological factors. Approximately 150 to 200 million fractures occur worldwide each year, making fractures one of the significant global public health issues [1]. Traditionally, fracture detection relies on radiologists performing manual diagnoses using three types of medical imaging techniques: X-ray, computed tomography (CT), and magnetic resonance imaging (MRI) [2]. Among these, X-ray is often the primary method for fracture detection due to its advantages of lower radiation, faster imaging, and lower cost [3]. However, the quality of X-ray images can vary due to noise, blurriness, and other factors, and manual diagnosis carries the risk of misdiagnosis [4]. Research has shown that in emergency medical settings, the error rate for fracture diagnoses is about 26% [5,6]. Computer-aided diagnosis (CAD) of medical images provides decision-making support for radiologists and surgeons. With the continuous development of deep learning and advancements in medical image processing techniques [7–10], the automation, accuracy, and real-time capabilities of fracture detection have significantly improved, and an increasing number of researchers are applying neural models in CAD, including fracture detection.

To address the challenges in fracture image diagnosis, many researchers have conducted studies on automated fracture detection techniques. Early automated fracture detection mainly relied on traditional image processing techniques. Jones et al. [11] used the Canny edge detection algorithm to locate suspicious fracture regions, followed by Hough transform to identify linear fracture features, and finally classified them using a support vector machine (SVM). The innovative work of Zhang et al. [12] focused on fracture feature analysis. They found significant differences (p¡0.01) between fracture regions and normal bone tissue in features such as contrast and entropy in the gray-level co-occurrence matrix (GLCM). Based on this, the classification system designed by them showed good specificity in recognizing distal radius fractures. With the advancement of feature engineering techniques, researchers began combining multiple features for detection. Wang et al. [13] proposed a multi-feature fusion method that integrates Haralick fracture features with local binary patterns (LBP), significantly improving the detection sensitivity of femoral neck fractures.

The introduction of Convolutional Neural Network (CNN) marked a significant leap forward in computer-aided fracture diagnosis. The Kim team [14] was the first to apply transfer learning using ResNet-50 to fracture detection. Subsequent two-stage detectors, such as Faster R-CNN [15], demonstrated the ability to accurately localize fracture sites through a Region Proposal Network (RPN). More recently, the development of multimodal fusion and Transformer-based architectures has further advanced the field. For example, Liu et al. [16] proposed FracFormer, which integrates X-ray, CT, and clinical data through a cross-attention mechanism, while Chen et al. [17] introduced EdgeFracNet, employing Neural Architecture Search (NAS) to compress the model to just 8 MB with 92% accuracy, enabling deployment on mobile Digital Radiography(DR) devices.

In parallel, object detection frameworks have continued to evolve in fracture detection tasks. Ju et al. [18] applied YOLOv8 to pediatric wrist radiographs and demonstrated that one-stage detectors can achieve both high efficiency and competitive accuracy. Chen et al. [19] designed WCAY, a weighted channel attention YOLO variant, which enhanced feature representation and improved detection across multi-site fracture datasets. Furthermore, Tahir et al. [20] developed an ensemble deep learning framework, significantly boosting diagnostic accuracy. These technological advances highlight a clear trend towards lightweight architectures, attention mechanisms, and ensemble strategies, which not only improve detection accuracy but also enhance deployment feasibility in clinical practice.

However, few studies have compared the performance of different types of object detection models in fracture detection. In this paper, we selected nine popular object detection models and conducted a comprehensive evaluation of their performance in detecting pediatric wrist fractures.

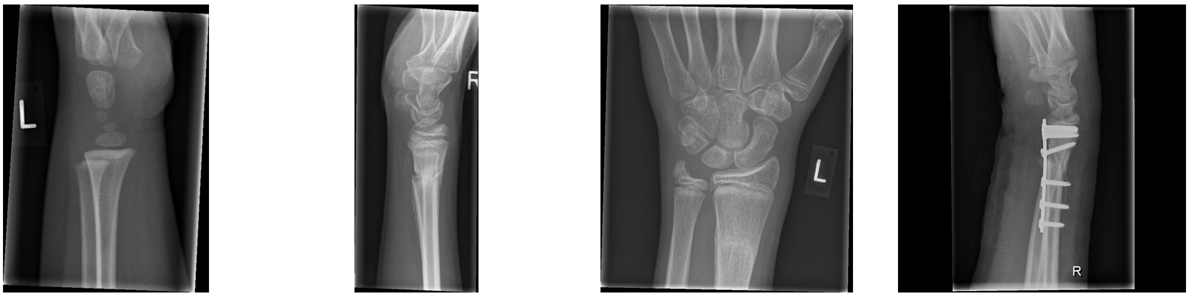
The main contributions of this paper are as follows: (1) We comprehensively evaluate nine object detection models on the GRAZPEDWRI-DX dataset. (2) We compare the performance of different models in detecting three key labels. (3) We make a comparison of the models' performance in terms of speed and computational complexity.

The rest of this paper is as follows: Section 2 introduces the datasets and models used in the study. Section 3 presents the performance metrics for evaluating the models and compares the results of different models. Section 4 discusses the results. Section 5 provides the conclusion.

## 2. Materials and Methods

### *2.1. Datasets*

The GRAZPEDWRI-DX dataset, released by the Medical University of Graz, is a publicly available collection of 20,327 pediatric wrist X-ray images. It was compiled by several pediatric radiologists at Graz University Hospital between 2008 and 2018, covering a total of 6091 patients and 10,643 studies. The dataset is annotated with 74,459 image labels and contains 67,771 labeled objects. Example images from this dataset are shown in Figure 1, with the detailed data distribution presented in Table 1.



(**a**) Healthy image with only textual labels.

(**b**) Right wrist image including fracture, periosteal reaction, and textual labels.

(**c**) Left wrist image including fracture, periosteal reaction, and textual labels.

(**d**) Image including fracture, metal, soft tissue, and textual labels.

**Figure 1.** Original images of the GRAZPEDWRI-DX dataset.

**Table 1.** Data distribution of GRAZPEDWRI-DX dataset.

| Label | Number | Ratio |
|---|---|---|
| Bone anomaly | 192 | 0.94% |
| Bone lesion | 42 | 0.24% |
| Foreign body | 8 | 0.04% |
| Fracture | 13,350 | 66.6% |
| Metal | 708 | 3.48% |
| Periosteal reaction | 2235 | 11.0% |
| Pronator sign | 566 | 2.78% |
| Soft tissue | 439 | 2.16% |
| Text | 20,274 | 99.74% |

### *2.2. Models*

Since the introduction of R-CNN, various high-precision object detection models based on deep learning have emerged. Typically, object detection models can be categorized into one-stage and two-stage models based on the number of detection stages, or into anchor-based and anchor-free methods depending on whether predefined anchor boxes are used. In this study, we selected nine popular object detection models and compared their performance in the task of pediatric wrist fracture detection. The nine models include: RetinaNet [21], YOLOF [22], YOLOX [23], YOLOv10 [24], CO-DETR [25], Sparse R-CNN [26], Faster R-CNN [27], Cascade R-CNN [28], and EfficientDet [29]. A detailed description of these models is provided in Table 2.

In the experiments, we replaced the backbones of some models to further compare their performance in pediatric wrist fracture detection. We selected ResNet [30], CSPNet [31], Swin Transformer [32], and DarkNet [33] as the backbones for most of the models.

**Table 2.** Description of object detection models.

| Model | Number of Stage | Anchor Setting | Model Description |
|---|---|---|---|
| RetinaNet | One-Stage | Anchor-based | a. RetinaNet introduces Focal Loss to address the class imbalance in object detection, focusing more on hard-to-classify examples.<br>b. It is a one-stage detector that combines the speed of single-stage models with the high accuracy of two-stage models [21]. |
| YOLOF | One-Stage | Anchor-based | a. YOLOF simplifies the model structure by reducing the number of feature pyramid layers, maintaining high detection performance. This design lowers computational costs while increasing speed.<br>b. It achieves state-of-the-art performance on several benchmark datasets, demonstrating its effectiveness in real-time object detection [22]. |
| YOLOX | One-Stage | Anchor-free | a. YOLOX introduces decoupled head and advanced augmentation techniques, which improve detection accuracy while maintaining high speed.<br>b. It achieves superior performance by using a simple yet effective architecture that scales well across different object detection tasks [23]. |
| YOLOv10 | One-Stage | Anchor-based | a. YOLOv10 introduces an end-to-end framework that significantly improves real-time object detection by optimizing speed and accuracy for various practical applications.<br>b. The model incorporates advanced techniques, such as feature fusion and multi-scale learning, to enhance detection performance, particularly in complex environments [24]. |
| CO-DETR | One-Stage | Anchor-free | a. CO-DETR integrates the contrastive learning into the DETR framework, significantly improving performance by learning better feature representations.<br>b. It eliminates the need for post-processing steps, such as non-maximum suppression, by directly producing high-quality object detections [25]. |
| Sparse R-CNN | One-Stage | Anchor-free | a. Sparse R-CNN uses sparse attention mechanisms to enhance the efficiency and effectiveness of object detection by focusing on only a small number of regions of interest.<br>b. By using dynamic proposals and sparse attention, Sparse R-CNN achieves competitive performance while significantly reducing the computation cost compared to traditional methods [26]. |
| Faster R-CNN | Two-Stage | Anchor-based | a. A two-stage object detector that first uses a Region Proposal model (RPN) to generate candidate regions.<br>b. RPN and the classifier share convolutional features, enabling efficient and end-to-end training.<br>c. It achieves high accuracy and is widely used in medical and general object detection tasks [27]. |
| Cascade R-CNN | Two-Stage | Anchor-based | a. Cascade R-CNN introduces a multi-stage detection framework that progressively improves detection performance by applying a series of detectors, each trained with increasing intersection over union (IoU) thresholds.<br>b. This approach allows for better handling of high-quality object detection tasks by refining results in each stage [28]. |
| EfficientDet | Two-Stage | Anchor-based | a. EfficientDet is a scalable and efficient object detection model that uses a compound scaling method to balance accuracy and efficiency.<br>b. The model achieves excellent performance with fewer parameters, making it suitable for real-time applications [29]. |

## 3. Results

In this section, we will introduce the performance metrics used in the experiments, the details of the experiments, and the comparison of performance results for each model.

### *3.1. Performance Metrics*

In this section, we use common performance metrics in object detection, including mean Average Precision (mAP), Precision, Recall, and F1 score, as the evaluation criteria for this study.

Average Precision(AP) represents the area under the Precision-Recall (PR) curve in object detection and is calculated using the following method. First, the intersection over union (IoU) between the predicted and ground truth bounding boxes is computed, along with a threshold (T) and the confidence score of the bounding box classification prediction. The formula is as follows:

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{1}$$

We can then calculate:

True Positives (TP): The prediction bounding box(BBox) with IoU $>$ T and meeting the category Confidence threshold.

False Positives (FP): The prediction BBox with IoU $<$ T and meeting the category Confidence threshold.

False Negatives (FN): The prediction BBox with IoU = 0.

Based on the TP, FP, and FN, we have,

$$\text{Precision} = \frac{\text{TP}}{\text{TP + FP}} \tag{2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP + FN}} \tag{3}$$

Based on different confidence thresholds for each category, we can plot the Precision-Recall (PR) curve to determine the AP value. The calculation process is as follows: For a category $c$, the Average Precision $AP_c$ is defined as the area under the Precision-Recall curve:

$$AP_c = \int_0^1 p_c(r)dr \tag{4}$$

Here, $p_c(r)$ is the precision value corresponding to the recall rate $r$. The final mAP is the average of the AP values across all $C$ categories:

$$\text{mAP} = \frac{1}{C}\sum_{c=1}^{C} AP_c \tag{5}$$

Finally, the mAP is obtained by averaging the values across all categories. By adjusting the IoU thresholds, we can compute mAP at various levels, such as mAP@0.25, mAP@0.50, and mAP@0.75.

### *3.2. Experiment Implementation*

This study is implemented based on MMDetection 3.3.0, with the runtime environment consisting of PyTorch 2.1.0 + CUDA 12.2. The hardware setup includes four RTX 3080 GPUs with 10 GB VRAM each. All models are trained for 100 epochs, and the model parameters that perform best on the validation set are retained as the final parameters.

The dataset is split into training, validation, and test sets with a ratio of 8:1:1. All pediatric wrist fracture X-ray images are resized to 640 × 640. The training is performed with an initial learning rate of 0.01 and a batch size of 2 per GPU (total batch size = 8 using 4 GPUs). A cosine annealing learning rate schedule is applied to progressively decay the learning rate to 1% of the initial value. Data augmentation methods, such as random flipping, are also employed. At the beginning of training, the pre-trained weights provided by MMDetection are used to facilitate convergence.

To ensure experimental fairness, the backbone configurations are clarified as follows: (1) Unified backbone models: For Faster R-CNN, RetinaNet, YOLOF, Cascade R-CNN, Sparse R-CNN, and CO-DETR, we uniformly use ResNet-50 as the backbone to guarantee comparability. (2) Default backbone models: For YOLOv10, YOLOX, and EfficientDet, we adopt their official lightweight configurations in MMDetection, namely YOLOv10-n, YOLOX-n, and EfficientDet-D0.

### *3.3. Results of Generalization Performance*

To comprehensively evaluate the performance of different object detection models in the fracture detection task, we compared various models based on key performance metrics, including mAP@0.25, mAP@0.50, mAP@0.75, Precision, Recall, and F1 Score. The results are shown in Table 3, and these metrics provide a comprehensive assessment of the models' performance in terms of accuracy, recall, and overall capability.

Based on the experimental results, YOLOv10 outperforms all other models across every metric. It achieves the highest detection accuracy at different IoU thresholds, with the mAP@0.25 of 0.650, mAP@0.50 of 0.632, and mAP@0.75 of 0.540. In addition, YOLOv10 records the best overall performance, reaching the Precision of 0.630, Recall of 0.550, and F1 Score of 0.580.

Following closely is Sparse R-CNN, which excels with the mAP@0.25 of 0.582 and mAP@0.50 of 0.527, showing a certain advantage in Precision at 0.590 and F1 Score at 0.530. However, its Recall of 0.480 is slightly lower than that of YOLOv10, suggesting it might have some issues with missed detections.

Although YOLOF's mAP@0.50 of 0.510 and mAP@0.75 of 0.420 are lower than those of YOLOv10 and Sparse R-CNN, its Precision of 0.580 still stands out.

In addition, while EfficientDet performs reasonably well inmAP@0.25, its performance inmAP@0.50 andmAP@0.75 is mediocre, with a relatively low Recall, indicating significant room for improvement in its detection capability. Although the model has a lighter computational overhead, it fails to outperform the YOLO series in terms of balancing performance and efficiency. In contrast, traditional anchor-based models such as Faster R-CNN, RetinaNet, and Cascade R-CNN generally underperform across most metrics, presenting significant performance bottlenecks in practical applications.

In summary, YOLOv10 demonstrates the best overall performance in the fracture detection task, particularly in terms of accuracy and recall, making it the most recommended model architecture at present.We selected 16 output images from the YOLOv10 model for demonstration, as shown in Figure 2. The model draws bounding boxes in different colors based on the predicted results for each data category: orange for predicted fracture regions, lemon yellow for regions predicted to contain metal,such as steel pins, light green for regions predicted to show periosteal reactions, and dark green for regions predicted to display text labels. The numbers above the bounding boxes represent the predicted confidence levels.

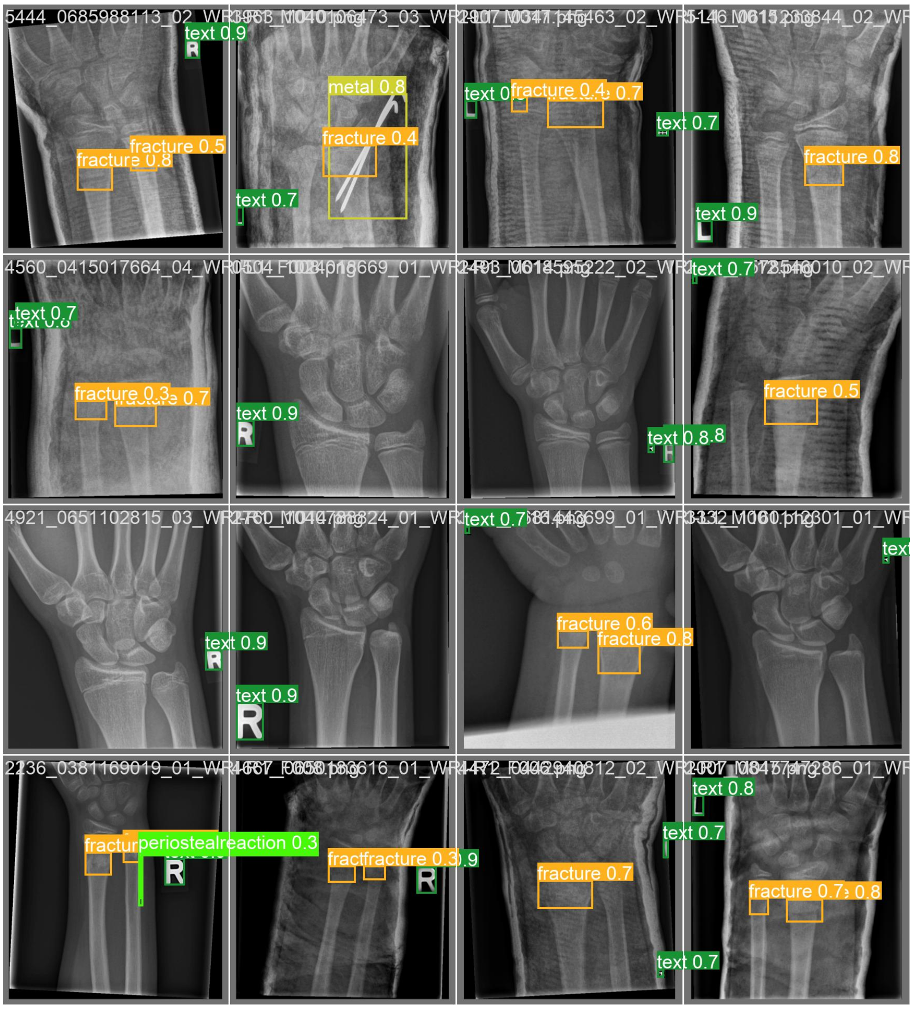


**Figure 2.** Visualization of YOLOv10 prediction results.

**Table 3.** Performance comparison of different object detection models on GRAZPEDWRI-DX.

| Model | mAP@0.25 | mAP@0.50 | mAP@0.75 | Precision | Recall | F1 score |
|---|---|---|---|---|---|---|
| Faster R-CNN-res50 | 0.474 | 0.330 | 0.143 | 0.430 | 0.350 | 0.372 |
| RetinaNet-res50 | 0.302 | 0.211 | 0.101 | 0.350 | 0.280 | 0.310 |
| **YOLOv10-n** | **0.650** | **0.632** | **0.540** | **0.630** | **0.550** | **0.580** |
| EfficienDet-D0 | 0.476 | 0.404 | 0.322 | 0.510 | 0.390 | 0.440 |
| YOLOF-res50 | 0.570 | 0.506 | 0.425 | 0.580 | 0.470 | 0.520 |
| Cascade R-CNN-res50 | 0.382 | 0.341 | 0.254 | 0.460 | 0.370 | 0.410 |
| CO-DETR-res50 | 0.441 | 0.421 | 0.344 | 0.470 | 0.380 | 0.420 |
| Sparse R-CNN-res50 | 0.582 | 0.527 | 0.420 | 0.590 | 0.480 | 0.530 |
| YOLOX-n | 0.402 | 0.367 | 0.270 | 0.460 | 0.370 | 0.410 |

Sparse R-CNN and YOLOF also show strong performance, making them suitable for applications where both precision and recall are critical. While EfficientDet's lightweight design offers advantages, its detection accuracy falls short compared to the YOLO series, making it less suitable for real-time fracture detection tasks that require high precision.

### *3.4. Performance Comparison of Three Key Labels*

As shown in Table 1, the GRAZPEDWRI-DX dataset includes nine different labels. For the fracture detection task, the three labels—Fracture, Metal, and Periosteal reaction—hold significant diagnostic value during the bone healing process. The Fracture label is used to mark fracture areas, the Metal label detects metal implants or fracture fixation devices, and the Periosteal reaction label pertains to the detection of periosteal reactions. To evaluate the performance of different object detection models on these clinically relevant categories, this section compares the performance metrics of nine detection models on these three labels, as shown in Figure 3.

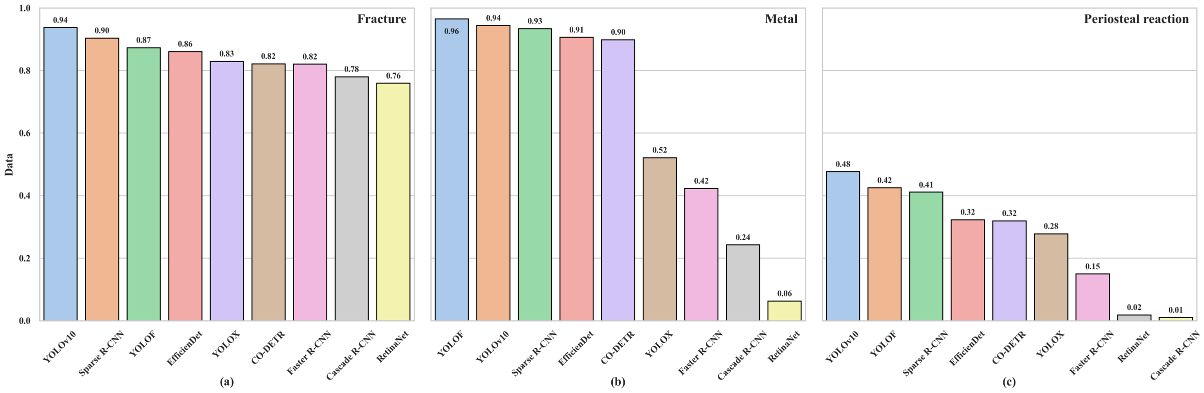


**Figure 3.** (**a**): Accuracy of Fracture labeling, (**b**): Accuracy of Metal labeling, (**c**): Accuracy of periosteal reaction labeling.

In the Metal label detection task, YOLOF demonstrates the best performance with the score of 0.965. YOLOv10 and Sparse R-CNN follow closely behind with scores of 0.944 and 0.934, respectively. EfficientDet and CO-DETR also show strong performance, with scores of 0.906 and 0.898, respectively, highlighting their effectiveness in detecting metal objects.

For the Fracture label detection task, YOLOv10 exhibits the best performance with the score of 0.937. It is followed by Sparse R-CNN with the score of 0.903, which also shows outstanding performance. YOLOF and EfficientDet perform well too, with scores of 0.873 and 0.86, respectively.

In the Periosteal Reaction label detection task, YOLOv10 achieved the score of 0.477. Although this is slightly lower compared to other categories, it still outperforms all other models. YOLOF and Sparse R-CNN also performed well, with scores of 0.425 and 0.411, respectively. EfficientDet and CO-DETR scored 0.323 and 0.319, respectively, showing relatively average performance.

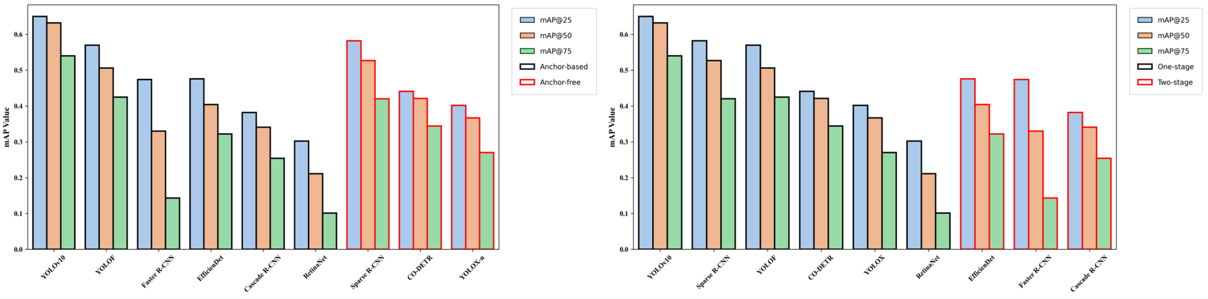
Overall, YOLOv10 and YOLOF stand out across multiple categories, demonstrating excellent performance. YOLOv10 is the best-performing model overall, particularly in the Fracture and Metal categories, where it achieves high precision and recall. Its simple design and fast inference speed make it ideal for real-time processing tasks. Sparse R-CNN shows strong detection capabilities in the Fracture and Metal categories but performs relatively weaker in the Periosteal reaction category, making it suitable for applications that require high-precision detection. EfficientDet performs well in the Metal category, but its performance in the Fracture and Periosteal reaction categories is relatively average.

### *3.5. Comparison of Detection Models: Anchor-Based vs. Anchor-Free and One-Stage vs. Two-Stage*

In this section, we will compare the performance of different model architectures from two perspectives: the use of anchor boxes, and whether the object detection process is based on a single-stage or two-stage approach.

As shown in Figure 4a, we first compare the performance of anchor-based and anchor-free models, with both categories ranked according to mAP from highest to lowest. As depicted in the figure, anchor-based models achieved the highest mAP on this dataset. Specifically, among anchor-based models, YOLOv10 demonstrated the best performance, while among anchor-free models, Sparse R-CNN outperformed the others.

Next, we compared the performance of one-stage and two-stage object detection models. As shown in Figure 4b, both types of models are ranked according to mAP from highest to lowest. From the figure, it can be seen that one-stage detectors generally achieved higher mAP on this dataset. Among one-stage detectors, YOLOv10 demonstrated the best performance, while among two-stage detectors, EfficientDet was the top performer. Overall, when comparing the average performance within the same category, one-stage detectors outperformed two-stage detectors across most metrics, ensuring higher detection efficiency and faster inference speed, while still handling the detection tasks with fine-grained precision.



(a) (b)

**Figure 4.** (**a**): Ranking of Anchor-based and Anchor-free models Based on mAP; (**b**): Ranking of One-stage and Two-stage models Based on mAP

### *3.6. Comparison of Model Calculation Efficiency and Reasoning Speed*

In fracture detection tasks, especially in resource-constrained environments, the design of lightweight models is crucial. The core objective of lightweight models is to reduce computational overhead, compress model size, and optimize memory usage, while maintaining diagnostic accuracy. This section compares the computational complexity and inference efficiency of various models, as these metrics are essential for evaluating the model's performance in real-world applications. We will focus on analyzing the model's floating point operations (FLOPs), number of parameters (Params), inference time per forward pass, and frames per second (FPS). Figure 5 and Table 4 provide a comprehensive comparison of these models.

**Table 4.** Efficiency and scale comparison of different object detection models.

| Model | FLOPs(G) | Params(M) | Inference(ms) | FPS |
|---|---|---|---|---|
| Faster R-CNN | 58.888 | 41.753 | 72.1 | 13.88 |
| RetinaNet | 48.082 | 36.496 | 65.2 | 15.34 |
| YOLOF | 39.43 | 42.52 | 27.98 | 36.56 |
| YOLOv10 | 8.354 | **2.802** | **15.78** | **63.37** |
| Cascade R-CNN | 51.691 | 51.4 | 165.8 | 6.03 |
| EfficientDet | **3.927** | 4.308 | 90.7 | 11.03 |
| Sparse R-CNN | 82.814 | 37.7 | 103.1 | 9.70 |
| CO-DETR | 25.357 | 43.45 | 79.4 | 12.59 |
| YOLOX | 13.39 | 8.94 | 17.82 | 56.12 |

The lightweight nature of object detection models is typically measured by Params and FLOPs, both of which directly relate to the model's storage usage and computational resource consumption. Comparing the models, YOLOv10 stands out in terms of lightweight design, using only 2.802M parameters and 8.354G FLOPs, significantly lower than traditional two-stage methods like Faster R-CNN and Cascade R-CNN. Additionally, EfficientDet has the 3.927G FLOPs and only 4.308M parameters, but its inference time does not show a significant reduction, indicating that despite its lower structural complexity, its inference efficiency is not ideal. Overall, YOLOv10 and other YOLO models like YOLOX achieve significant model size compression while maintaining detection capabilities, making them more suitable for embedded or resource-constrained real-world applications.

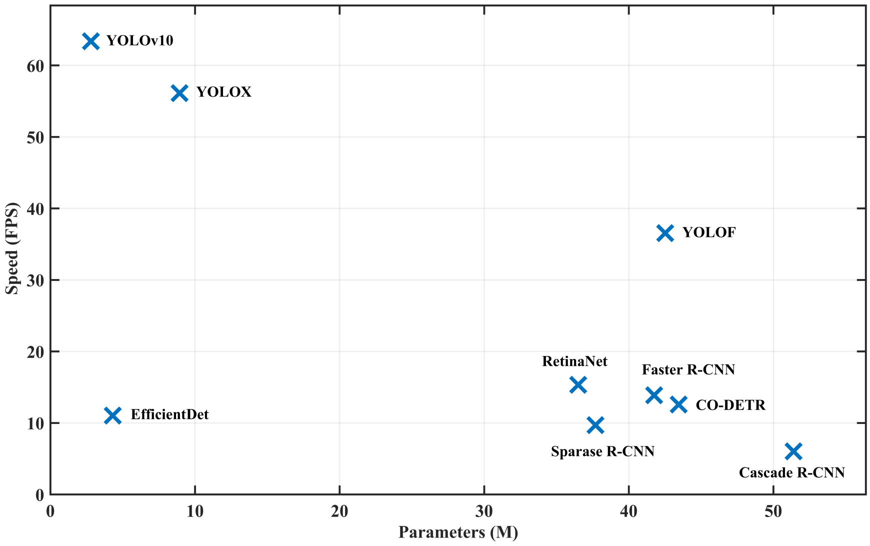


**Figure 5.** Comparison of model scale and inference speed.

## 4. Discussion

In this study, we compared the performance of nine object detection models on a public fracture dataset, covering aspects such as overall performance, detection of three key labels, anchor-based vs. anchor-free methods, one-stage vs. two-stage models, as well as real-time capabilities and model lightweight design. According to the experimental results, YOLOv10 outperforms other models in multiple areas. Additionally, YOLOF also shows impressive performance in detecting key labels. The following sections will discuss each of these results in detail.

In the comparison,YOLOv10 excels across all metrics, particularly in mAP@0.50 and mAP@0.75, where it significantly outperforms other models, indicating its exceptional balance between detection accuracy and recall. Sparse R-CNN follows closely behind, with a minimal gap in mAP@0.50 compared to YOLOv10, demonstrating strong overall performance.Notably, YOLOv10 demonstrated superior performance in fracture detection compared to other detectors, particularly in the identification of small-scale features such as periosteal reactions. This advantage may be attributed to several factors. First, the enhanced backbone of YOLOv10 (e.g., Pyramid Split Attention(PSA)/Cross Stage Partial(CSP) modules) enables more efficient feature reuse and gradient flow, allowing the model to better capture subtle textural differences in fracture regions. Second, the improvements in multi-scale feature fusion and anchor-free label assignment strategies make YOLOv10 more robust in learning sparse and fine-grained features, thereby significantly enhancing its small-object detection capability. In addition, compared with traditional two-stage or anchor-based detectors, YOLOv10 achieves a more favorable balance between accuracy and inference speed, which is particularly beneficial for real-time clinical applications. Finally, the lightweight design of YOLOv10 reduces the risk of overfitting and improves generalization across diverse imaging conditions, ensuring its applicability in various clinical settings.

However, even for YOLOv10, which achieves the best overall performance, certain cases of missed or incorrect detections are still observed. These errors may primarily stem from the following factors: first, small fractures often lose critical details during image resizing and feature downsampling; second, normal bone textures, anatomical structures, or imaging artifacts can resemble fracture lines, leading to model confusion; and finally, if the feature fusion mechanism fails to adequately preserve small-scale information, the model struggles to capture both local details and global context, which further complicates accurate detection.

In the detection tasks for the three key labels (Fracture, Metal, Periosteal Reaction), YOLOF and YOLOv10 stand out in the Fracture and Metal categories, with YOLOF achieving the highest mAP50 in the Metal label. This indicates that both YOLOF and YOLOv10 are highly effective at precise localization and classification when dealing with objects with larger sizes and clear features. For the Periosteal Reaction category, YOLOv10 performs the best.

This task is more challenging, especially since periosteal reactions are typically small and irregular in shape, which makes the models' performance on small targets relatively weaker. Additionally, Sparse R-CNN and EfficientDet also perform excellently in the Metal category detection tasks.

In the comparison of single-stage and two-stage models, our results show that single-stage detectors have comprehensively outperformed two-stage detectors. YOLOv10, as a representative single-stage model, demonstrates outstanding performance by achieving both high accuracy and fast inference speed. In contrast, Sparse R-CNN, a typical two-stage model, shows some accuracy advantages but suffers from slower inference, especially when processing large volumes of data, which limits its practical applicability. This performance gap is largely due to the structural differences: two-stage models rely on region proposal and subsequent classification/regression steps, which increase computational overhead and inference latency. On the other hand, single-stage models directly perform detection and classification in an end-to-end manner, eliminating redundant proposal generation, thereby achieving higher efficiency. These results suggest that the lightweight design and efficient feature utilization of one-stage detectors make them more suitable for real-time clinical fracture detection, where both accuracy and speed are critical.

In the comparison of anchor-based and anchor-free models, anchor-free approaches avoid the challenges associated with anchor matching, making them more flexible and efficient in complex object detection tasks, with overall performance surpassing that of anchor-based models. In the context of fracture detection, this advantage is particularly evident: anchor-free methods (e.g., YOLOX, YOLOv10) directly predict object centers or bounding boxes, thereby circumventing the mismatch issues inherent to anchor design. This enables them to achieve superior performance in detecting fine-grained features such as subtle fracture lines and periosteal reactions. By contrast, anchor-based methods (e.g., Faster R-CNN, RetinaNet) rely on predefined anchor priors, which provide greater stability in detecting large fracture regions or fragments with well-defined boundaries, but they are less adaptable to small objects and long-tailed distributions. Notably, among the anchor-based methods, only YOLOv10 demonstrates outstanding performance, which may be attributed to its enhanced feature fusion mechanisms and efficient label assignment strategies.

## 5. Conclusions

This study provides a comprehensive comparison of the performance of nine mainstream object detection models on the publicly available GRAZPEDWRI-DX pediatric wrist trauma dataset. The comparison covers overall performance, detection of key categories, anchor-based vs. anchor-free designs, one-stage vs. two-stage models, as well as real-time efficiency and lightweight characteristics. Among the models, YOLOv10 demonstrates impressive performance, highlighting its potential in real-time object detection tasks. Additionally, Sparse R-CNN and EfficientDet perform well in certain image categories, particularly in scenarios requiring high precision, showcasing their advantages in small object detection and accurate localization.

Despite the promising results achieved in this study, there are still some limitations. Future work could focus on validating the models on larger and more diverse datasets to further enhance their generalization ability. Additionally, improving the model architecture or incorporating more auxiliary information could help increase the accuracy for different label categories. Furthermore, future research will also aim to optimize the inference speed and real-time performance, especially for applications on mobile or embedded devices.

## Author Contributions

Z.L.: conceptualization, methodology, data curation, investigation, writing—original draft preparation; R.Z.: methodology, formal analysis, writing—review & editing, supervision. All authors have read and agreed to the published version of the manuscript.

## Funding

This research received no external funding. The APC was funded by the authors.

## Institutional Review Board Statement

Not applicable. The study did not involve human participants or animals.

## Informed Consent Statement

Not applicable. The study did not involve human participants.

## Data Availability Statement

The dataset used in this study, GRAZPEDWRI-DX, is a publicly available pediatric wrist X-ray dataset. It can be utilized for related studies such as bone age assessment. The GRAZPEDWRI-DX dataset is available for access at https://figshare.com/articles/dataset/GRAZPEDWRI-DX/14825193. If direct access is unavailable, it is recommended to contact the dataset provider or relevant research institutions to request access. The use of this dataset must comply with its original licensing agreement and ensure adherence to ethical standards for medical data.

## Conflicts of Interest

The authors declare that there are no commercial or financial relationships that could be construed as a potential conflict of interest, and all authors have no relevant conflicts of interest.

## References

1. Büsching, P.; Karhade, A.V. Fracture incidence and risk factors: A global review. *J. Trauma Acute Care Surg.* **2021**, *70*, 512–520.
2. Smith, W.E.; Brown, G.W. Imaging modalities in fracture diagnosis: A review of X-ray, CT, and MRI. *Eur. Radiol.* **2019**, *29*, 4732–4745.
3. Liu, H.; Zhang, X. Advantages and challenges of X-ray imaging in bone fracture diagnosis. *Med. Imaging J.* **2018**, *45*, 80–85.
4. Li, J.; Wang, Z. Diagnostic errors in emergency medicine: The impact of imaging modalities in fracture diagnosis. *J. Emerg. Med.* **2020**, *58*, 584–591.
5. Erhan, E.; Kara, P.; Oyar, O.; et al. Overlooked extremity fractures in the emergency department. *Turk. J. Trauma Emerg. Surg.* **2013**, *19*, 25–28.
6. Mounts, J.; Clingenpeel, J.; McGuire, E.; et al. Most frequently missed fractures in the emergency department. *Clin. Pediatr.* **2011**, *50*, 183–186.
7. Adams, S.; Henderson, R.; Yi, X.; et al. Artificial intelligence solutions for analysis of X-ray images. *Can. Assoc. Radiol. J.* **2021**, *72*, 60–72.
8. Choi, J.; Cho, Y.; Lee, S.; et al. Using a dual-input convolutional neural network for automated detection of pediatric supracondylar fracture on conventional radiography. *Investig. Radiol.* **2020**, *55*, 101–110.
9. Chung, S.; Han, S.; Lee, J.; et al. Automated detection and classification of the proximal humerus fracture by using deep learning algorithm. *Acta Orthop.* **2018**, *89*, 468–473.
10. Tanzi, L.; Vezzetti, E.; Moreno, R.; et al. Hierarchical fracture classification of proximal femur X-ray images using a multistage deep learning approach. *Eur. J. Radiol.* **2020**, *133*, 109373.
11. Jones, D.; Smith, R.; Brown, C. Automated detection of fractures in radiographs using edge detection and machine learning. *J. Med. Imaging* **2005**, *14*, 101–110.
12. Zhang, L.; Wang, F.; Li, Y. Fracture detection based on texture features in medical images. *IEEE Trans. Med. Imaging* **2008**, *27*, 163–170.
13. Wang, X.; Zhang, Y.; Lee, J. Feature fusion for automated detection of hip fractures from radiographs. *Comput. Med. Imaging Graph.* **2012**, *36*, 253–260.
14. Kim, Y.; Lee, C.; Park, J. Deep learning for detecting bone fractures in X-ray images using ResNet-50. *J. Med. Imaging* **2017**, *44*, 33–39.
15. Wang, L.; Zhang, H.; Li, X. Deep learning-based two-stage detector for fracture detection using Faster R-CNN. *IEEE Trans. Med. Imaging* **2017**, *36*, 853–860.
16. Liu, M.; Zhang, T.; Li, X. FracFormer: A Transformer-based model for complex pelvic fracture detection with multimodal data fusion. *IEEE Trans. Med. Imaging* **2022**, *41*, 4381–4390.
17. Chen, S.; Wang, J.; Liu, J. EdgeFracNet: A lightweight deep learning model for fracture detection with neural architecture search. *J. Med. Imaging* **2023**, *56*, 11–18.
18. Ju, R.Y.; Li, X.; Wang, Y. Fracture detection in pediatric wrist trauma X-ray images using YOLOv8. *Sci. Rep.* **2023**, *13*, 20077.
19. Chen, P.; Liu, S.; Lu, W.; et al. WCAY object detection of fractures for X-ray images of multiple sites. *Sci. Rep.* **2024**, *14*, 26702.
20. Tahir, A.; Saadia, A.; Khan, K.; et al Enhancing diagnosis: ensemble deep-learning model for fracture detection using X-ray images. *Clin. Radiol.* **2024**, *79*, e1394–e1402.
21. Lin, T.Y.; Dollar, P.; Girshick, R.B.; et al. Focal loss for dense object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 2980–2988.
22. Chen, Q.; Wang, Y.; Yang, T.; et al. You only look one-level feature. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13039–13048.
23. Ge, C.; Zhang, S.; Li, Q.; et al. YOLOX: Exceeding YOLO series in 2021. *arXiv* **2021**, arXiv:2107.08430.

24. Wang, A.; Chen, H.; Chen, K.; et al. YOLOv10: Real-time end-to-end object detection. *arXiv* **2024**, arXiv:2405.14458.
25. Dai, J.; Xie, Y.; Wang, X.; et al. CO-DETR: Contrastive learning for object detection in transformers. *arXiv* **2021**, arXiv:2106.04751.
26. Zhu, X.; Liang, Z.; Liu, Y.; et al. Sparse R-CNN: End-to-end object detection with sparse features. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 3355–3365.
27. Ren, S.; He, K.; Girshick, R.; et al. Faster R-CNN: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99.
28. Cai, Z.; Vasconcelos, N. Cascade R-CNN: Delving into high quality object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 1244–1256.
29. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and efficient object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 3958–3968.
30. He, K.; Zhang, X.; Ren, S.; et al. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
31. Huang, Z.; Wang, X.; Li, W. CSPNet: A new backbone that can enhance learning capability of CNN. *arXiv* **2020**, arXiv:2004.08955.
32. Liu, Z.; Lin, Y.; Cao, Y.; et al. Swin Transformer: Hierarchical vision transformer using shifted windows. *arXiv* **2021**, arXiv:2103.14030.
33. Redmon, J.; Divvala, S.; Girshick, R.; et al. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.