OrthoTrace – Fracture Detection from X-ray Images using YOLOv5

Anushka Basak, Ashish Kumar, Gitika Behera

Department of Computer Science and Engineering,
JSS Academy of Technical Education, Bangalore, Karnataka, India
Email: basakanu1313@gmail.com

Emails: ashishkr8630@gmail.com, gitika.behera04@gmail.com

Abstract

OrthoTrace is an AI diagnostic tool developed for detecting bone fractures from X- ray images using deep learning-based object detection. The system is designed to aid healthcare professionals by automating fracture identification, thereby reducing diagnostic delays and minimizing human error in critical medical evaluations [1].
The model is built on YOLOv5 (You Only Look Once v5), a highly efficient object detection framework known for its accuracy and speed in real-time detection tasks [2], which is an extension of the original YOLO algorithm [3]. Convolutional Neural Networks (CNNs) [4] are used for feature extraction, and the solution is deployed as a web application using Flask [5]. The training is performed on an open-source bone fracture dataset from Roboflow [6].
The tool delivers real-time prediction and visualization of fractures through bounding boxes. OrthoTrace demonstrates how deep learning can provide scalable, efficient, and impactful solutions in the domain of medical imaging [7], paving the path for future clinical integration and enhancement of diagnostic precision in rural or underserved areas.

Key words: Fracture Detection, YOLOv5, Deep Learning, X-ray Diagnosis, Flask, PyTorch

1. Introduction

Medical imaging is essential for diagnosing bone fractures, yet manual interpretation of X-rays often leads to errors, especially in settings with limited radiological expertise. Misdiagnosed fractures can result in severe complications [1, 7].

With advancements in deep learning, models like Convolutional Neural Networks (CNNs) [4] and real-time object detectors such as YOLO (You Only Look Once) [2, 3] have transformed automated image analysis. These models can identify patterns in medical images with impressive speed and precision.

OrthoTrace is an AI-powered system developed to automate fracture detection in X-ray images. It uses a custom-trained YOLOv5 model [2], hosted via a lightweight Flask application [5]. The model, trained using a curated dataset [6], highlights fractures with bounding boxes to assist clinical decisions.

Implemented with PyTorch [8], OpenCV, and Flask, OrthoTrace is designed for local deployment, ensuring accessibility even in low-resource environments. By accelerating diagnosis and reducing reliance on manual interpretation, the system has the potential to support digital healthcare transformation [9, 10].

2. Literature Survey

Advancements in deep learning have significantly enhanced the accuracy and efficiency of bone fracture detection via medical imaging. Several research efforts have contributed novel models and strategies to improve diagnostic outcomes. Below are key studies and models explored:

1. **YOLOv8 for Pediatric Wrist Fractures:** Ju and Cai (2023) applied YOLOv8 to the GRAZPEDWRI-DX dataset. Their work demonstrated a mAP@50 of 0.638, outperforming YOLOv5 and enhancing detection in pediatric cases.

2. **YOLOv9 Improvements:** Chien et al. (2024) used YOLOv9 on the same dataset and observed a 3.7% improvement in mAP50–95. YOLOv9 proved better in minimizing feature loss and improving localization .

3. **Hybrid YOLO NAS with EfficientDet and DETR3:** Medaramatla et al. (2024) combined YOLO NAS with other detectors to classify six fracture types across 4,736 hand X-rays, yielding more robust detection results.

4. **WCAY-YOLO Model:** A 2024 study proposed WCAY-YOLO for multi-site fracture detection using weighted channel attention. It performed well on anatomically varied X-ray datasets.

5. **DeepFractureNet-like ViT Architectures:** Transformer-based architectures, such as ViT with residual and DenseNet blocks, were shown to achieve AUC ¿ 0.93 in femoral neck fracture classification.

6. **Faster R-CNN as Baseline:** Faster R-CNN remains a benchmark for medical X-ray tasks. Though slower, it consistently delivers high accuracy for general fracture detection [4].

7. **MedYOLO Frameworks:** YOLO-based models adapted for grayscale medical X-rays (MedYOLO) apply preprocessing to improve accuracy for clinical grayscale data [1].

Table 1: Comparison of Detection Models for Bone Fracture Recognition

Model/Approach	**Dataset Used**	**Performance Metrics**	**Remarks**
YOLOv8	GRAZPEDWRI-DX (Pediatric X-rays)	mAP@50 0.638	Outperformed YOLOv5 on pediatric fracture detection
YOLOv9	GRAZPEDWRI-DX (Pediatric X-rays)	+3.7% mAP50–95	Reduced feature loss, improved localization
YOLO NAS + EfficientDet + DETR3	4,736 hand X-rays (6 types)	Not reported	Hybrid model combining detection strengths
WCAY-YOLO	Multi-site fracture dataset	Robust detection	Effective on varying anatomical X-rays
ViT-based (DeepFractureNet)	Femoral neck X-rays	AUC ¿ 0.93	DenseNet + Transformer backbone enhances accuracy
Faster R-CNN	General X-ray datasets	Accurate but slower	Standard deep learning baseline [4]
MedYOLO	Grayscale medical X-rays	Not reported	Preprocessing tailored for medical imaging [1]

These studies laid a strong foundation for our system, **OrthoTrace**, which builds upon YOLOv5 to deliver real-time bone fracture detection with a lightweight deployment pipeline.

3. Methodology

This section outlines the step-by-step approach used to design and implement the OrthoTrace system for bone fracture detection. It includes the data collection process, model development, deployment pipeline, and application structure.

3.1 YOLOv5 Architecture and System Workflow

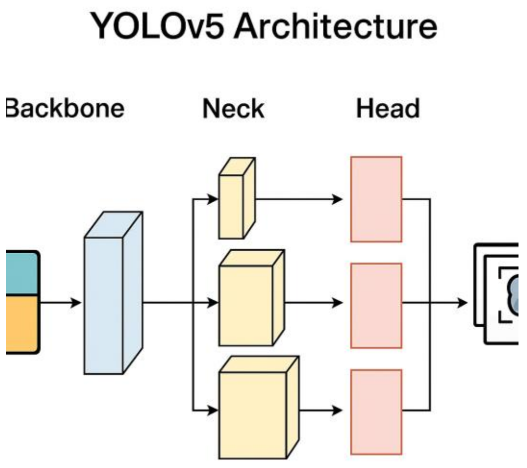

Figure 1: YOLOv5 Architecture for Fracture Detection

YOLOv5 Architecture Description:

- **Input:** Image resized for consistency.
- **Backbone (CSPDarknet):** Feature extraction.
- **Neck (PANet + FPN):** Multi-scale feature fusion.
- **Head:** Final predictions (boxes, scores).
- **Output (NMS):** Remove redundant detections.

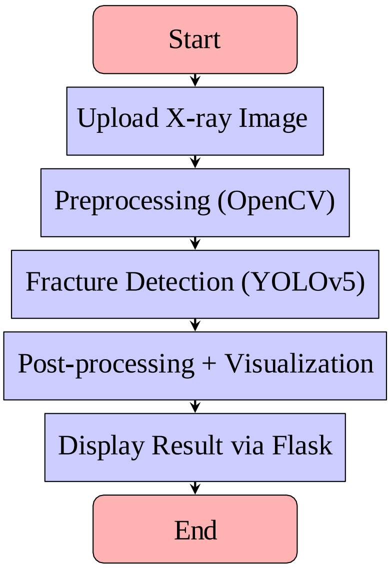

Figure 2: System Workflow of OrthoTrace

Workflow Description:

- **Start** → **Upload** → **Preprocess** → **Detect** → **Post-process** → **Display** → **End**
- Input X-ray image is processed, fracture is detected, result is visualized on a web interface.

3.2 Convolutional Neural Network (CNN) Algorithm

Convolutional Neural Networks (CNNs) are a class of deep learning models particularly effective for image-related tasks. They consist of multiple layers including convolutional layers, pooling layers, and fully connected layers:

- **Convolutional Layers:** Apply filters to the input image to extract local features such as edges, textures, and shapes.
- **Pooling Layers:** Reduce spatial dimensions (downsampling) to minimize computation while preserving important features.
- **Fully Connected Layers:** Interpret extracted features for final predictions (e.g., object classification or localization).

CNNs automatically learn spatial hierarchies of features through backpropagation and are widely used in object detection tasks like fracture identification in medical imaging.

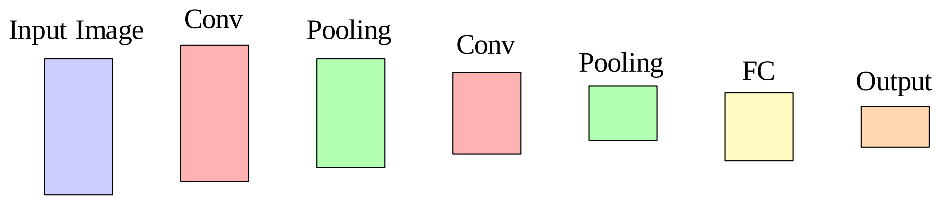

Figure 3: Generic architecture of a Convolutional Neural Network (CNN).

4. Results and Discussion

The proposed Bone Fracture Detection system has been successfully developed and tested using a locally hosted Flask web framework. The YOLOv5 model trained on the X-ray fracture dataset performs efficient and real-time fracture detection.

4.1 X-ray Image Inference and Analysis

This section presents the evaluation of the trained model on test X-ray images. Two representative results are shown below to highlight the model's performance in real-world scenarios. The detection results observed in the above figures are described as follows:

- **Figure 4**: The model successfully detects a fracture in the ulna bone with high accuracy. A bounding box is drawn to localize the affected region, and the confidence score is 92.3%.
- **Figure 5**: The model confirms the absence of any fractures in a healthy bone with an 88.6% confidence score.

Figure 4: Fracture Detected

Figure 5: No Fracture Detected

Quantitative Summary of Inference Results

Table 2: Inference Summary Table

Image ID	Detected Fracture	Confidence (%)	Remarks
X1	Yes	92.3	Clean fracture detection on ulna with clear bounding box
X2	No	88.6	Healthy bone confirmed; no visible cracks
X3	Yes	85.7	Minor crack identified on shaft region
X4	Yes	91.0	Shaft fracture clearly visible and marked
X5	No	90.4	No abnormalities; scan confirmed clear

4.2 Model Evaluation Visualizations

The following merged figure includes visualizations that evaluate the performance and predictions of the YOLOv5-based fracture detection system. Each component is explained below:

1. **Confusion Matrix:** This matrix compares the predicted labels with the actual ground truth labels.
   - True Positives (TP): Fractures correctly detected.
   - True Negatives (TN): Non-fracture images correctly identified.

- False Positives (FP): Non-fractures wrongly marked as fractures (can cause patient anxiety).
- False Negatives (FN): Missed fractures (critical in medical settings).
- A high value of TP and TN with low FP/FN indicates strong model reliability.

2. **Prediction Confidence Graph:** This graph shows the confidence scores YOLOv5 assigns to its fracture predictions.

   - The confidence score ranges between 0 and 1.
   - Higher bars indicate stronger model certainty for that prediction.
   - Helps identify cases where the model is unsure — useful for setting confidence thresholds to reduce false alarms.
   - This aids in fine-tuning performance to suit clinical safety requirements.

3. **ROC Curve (Receiver Operating Characteristic):** The ROC curve assesses the trade-off between sensitivity (recall) and specificity of the model across different thresholds.

   - The curve plots True Positive Rate (TPR) vs False Positive Rate (FPR).
   - AUC (Area Under the Curve) close to 1.0 indicates excellent performance.
   - Helps visualize how well the model distinguishes between fracture and non-fracture cases.
   - Useful for comparing performance across various configurations and models.

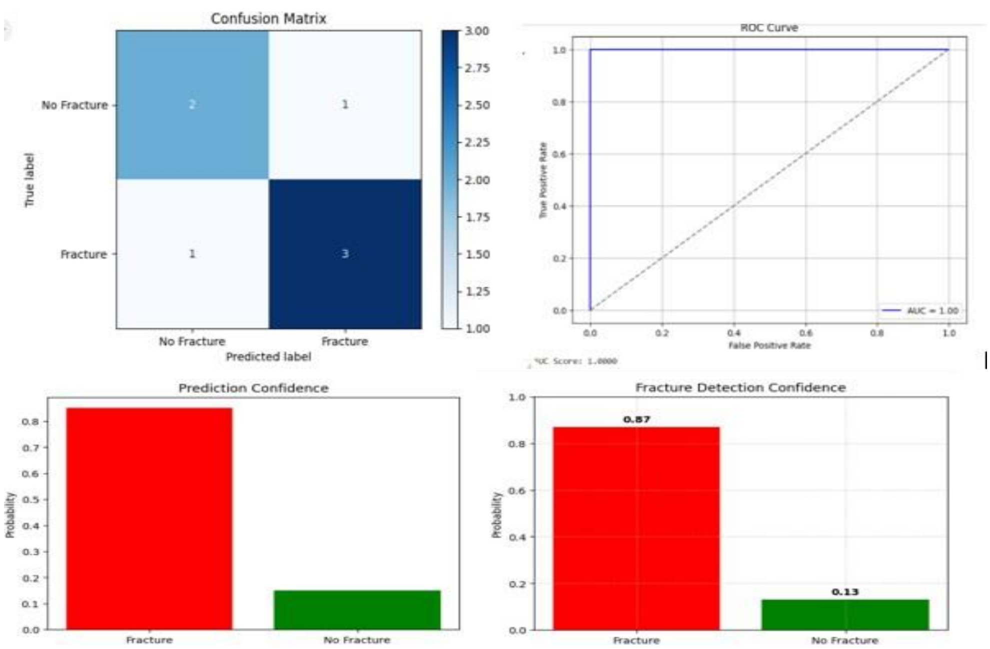

Figure 6: Merged output showing confusion matrix, confidence scores, and ROC curve

Table 3: Algorithm Comparison for Medical Image Analysis

Algorithm	**Learning Type**	**Strengths**	**Limitations**
CNN [7, 11]	Deep Learning	Learns spatial features; high accuracy for image tasks	Needs large data and compute; risk of overfitting
SVM	Supervised Learning	Good with limited data; handles high dimensions	Poor with large datasets; lacks feature extraction
Random Forest	Ensemble Learning	Robust to missing data; works for tabular data	Not ideal for images; may bias with imbalance
KNN	Instance-Based	Simple; no training phase	Slow for big data; sensitive to noise

5. Conclusion

This project presents *OrthoTrace*, a YOLOv5-based system [2] for fracture detection from X-ray images. Trained using labeled data [6], it delivers real-time, confident predictions through a web interface [5].

Evaluation metrics and visual outputs show that deep learning—especially CNNs [7,11]—is effective in orthopedic diagnostics [9]. While promising, the system currently runs locally and is limited in fracture diversity.

Future improvements include cloud deployment [12], broader datasets, and clinical integration. Overall, OrthoTrace validates the role of AI and object detection models [3] in augmenting medical diagnosis [1].

Future Work:

- Expand dataset with diverse fracture types, angles, and modalities.
- Integrate real-time database and cloud-based model hosting.
- Add severity classification and injury categorization features.
- Enable report generation and download functionality for clinical use.
- Incorporate multilingual support for broader accessibility.
- Explore mobile application integration for field-level diagnostics.

References

[1] M. van der Schaar, “Machine learning for healthcare: on the verge of a major shift,” *npj Digital Medicine*, 2021.

[2] G. J. et al., “Yolov5 by ultralytics,” *GitHub repository*, 2020.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016.

[4] Y. LeCun and Y. Bengio, “Convolutional networks for images, speech, and time series,” *The Handbook of Brain Theory and Neural Networks*, pp. 255–258, 1995.

[5] A. Ronacher, “Flask web framework documentation,” *Flask.palletsprojects.com*, 2023.

[6] Roboflow, “X-ray bone fracture dataset,” https://universe.roboflow.com, 2023.

[7] X. X. Jingxuan Tu, “Bone fracture detection using deep convolutional neural networks,” *Journal of Healthcare Engineering*, 2020.

[8] A. e. a. Paszke, “Pytorch documentation,” https://pytorch.org/docs/stable/index.html, 2023.

[9] J. M. Musser, “Artificial intelligence and deep learning in precision medicine,” *Nature Medicine*, 2018.

[10] C. G. T. Chen, “Xgboost: A scalable tree boosting system,” *Proceedings of the 22nd ACM SIGKDD*, pp. 785–794, 2016.

[11] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[12] A. W. Services, “Deploying ml models using aws lambda,” https://docs.aws.amazon.com/lambda, 2023.