BT5151 Advanced Analytics and Machine Learning

Group Project Report - Intelligent Bone Fracture Detection System

Team Members:
Raveena Sivagurunathan (A0176727R)
Sneha More (A0297761M)
Swedha Rajarajan (A0297326U)
Rickson Hu Hong Rui (A0296664L)

1 Executive Summary **3**
2 Introduction **3**
2.1 Problem Statement 3
2.2 Solution 3
2.3 Business Goal 4
3 Methodology **4**
3.1 Project Objective 4
3.2 Data Source 4
3.3 Model Architecture- YOLOv8 (For speed) 4
3.4 Model Architecture - Faster-CNN (For accuracy) 5
3.5 Model Evaluation 5
4 Implementation **6**
4.1 Preprocessing Pipeline 6
4.2 Model Training and Inference 6
4.2.1 YOLOv8 6
4.2.2 Faster R-CNN 7
4.3 Evaluation Summary: YOLOv8 vs. Faster R-CNN 9
5 Conclusion **9**
6 References **10**
7 Appendix **11**

1 Executive Summary

With Singapore's rapidly ageing population and the increasing burden on healthcare systems, the timely and accurate diagnosis of bone fractures has become a critical challenge. An intelligent bone fracture detection and classification system was developed to address this issue using state-of-the-art deep learning models — YOLOv8 and Faster R-CNN.

After performing preprocessing steps such as image resizing, annotation scaling, and data cleaning, the dataset was enhanced through data augmentation.YOLOv8 was chosen for its real-time detection capabilities, while Faster R-CNN was selected for its superior classification accuracy. Both models were trained and evaluated using standard object detection metrics, including mean Average Precision (mAP), Precision-Recall curves, and confusion matrices.

The YOLOv8 model achieved a mAP@0.5 of 0.287, mAP@0.5–0.95 of 0.10, with an overall precision of 0.30 and recall of 0.31, which indicates moderate localisation performance but limited sensitivity to fractures. On the other hand, the Faster R-CNN model achieved a mAP@0.5 of 0.34, mAP@0.5–0.95 of 0.10, with an overall precision of 0.42 and recall of 0.51, which showcases higher detection accuracy and better localisation, but at the cost of more computation and time. The choice of model ultimately boils down to the clinical context of whether speed or precision is a higher priority.

2 Introduction

2.1 Problem Statement

Singapore, like many developed countries, is facing a demographic shift toward an ageing population. By 2030, individuals aged 65 and above are expected to make up 25% of the population, according to the Department of Statistics Singapore. Ageing increases the risk of bone fragility from conditions like osteoporosis, leading to a higher likelihood of fractures.

Ultimately, this has led to more hospital admissions due to fall-related bone injuries. Prompt and accurate fracture diagnosis is vital for elderly patients to prevent mobility loss, extended recovery, or further health decline. Fracture detection traditionally depends on radiologists manually inspecting X-ray images—a process that is time-consuming, prone to human error, and challenged by subtle or complex fractures. The rising volume of imaging data also strains radiology departments, contributing to diagnostic delays.

2.2 Solution

Given this context, there is a pressing need for an automated AI-driven solution that can assist medical professionals.. A reliable AI-based diagnostic tool can significantly improve efficiency, reduce radiologist workload, minimise diagnostic errors, and ultimately enhance patient outcomes.

The traditional process of diagnosing bone fractures through manual examination of X-ray images presents several challenges like **time consumption, human error and subjectivity, scalability issues, and inconsistent results.** Thus, by leveraging deep learning and computer vision techniques, we aim to automate the detection and classification of bone fractures in X-ray images, enabling faster, more accurate decisions.

2.3 Business Goal

Proposing an AI-driven solution has a lot of profitability potential in the following areas:

- **Operational Efficiency:** Hospitals and clinics face increasing pressure to deliver faster and more accurate diagnoses. The solution leads to savings in costs as fewer hours are spent per image, labour costs are reduced, resulting in quick patient turnover.
- **Scalable Solution for Private Healthcare Providers:** Private healthcare providers, imaging centres, and telemedicine companies can integrate the AI system to differentiate themselves in a competitive market and serve a broader patient base. This can result in new revenue streams and expand market reach.
- **Commercial Licensing of AI Software:** The AI model can be monetised in multiple commercial formats, such as direct license to hospitals, private clinics, and diagnostic laboratories or packaged as a **SaMD** (Software as a Medical Device) product.
- **Sustainable Competitive Advantage Through Continuous AI Learning**
  A key advantage of AI models is their ability to improve over time by learning from new data. Once deployed, they continually refine their performance, making them a high-return, one-time investment that grows more accurate and valuable. This self-improving nature offers a strong competitive edge, positioning the product as a standout in a fast-evolving market.

3 Methodology

3.1 Project Objective

This project aims to design and implement a deep learning solution for the automatic detection and classification of bone fractures in X-ray images. Using object detection frameworks, it localises fracture regions and categorises them into clinically relevant anatomical groups: **bones, elbow positive, fingers positive, forearm fracture, humerus, humerus fracture, shoulder fracture, and wrist positive.**

To achieve this, we adopted a two-model strategy (YOLOv8 and Faster R-CNN), balancing speed and accuracy. The system is modular, reproducible, and designed for real-world deployment in radiology workflows. The project also evaluates trade-offs between model architectures and examines how preprocessing, data augmentation, and annotation formats affect detection accuracy and generalisation across fracture types.

3.2 Data Source

The model will be trained and tested using publicly available unstructured X-ray datasets, more specifically, the bone fracture detection dataset from Kaggle that contains labelled fracture images, as well as the COCO-Annotated dataset from Kaggle that provides object detection annotations to facilitate model training. The Kaggle bone fracture dataset contains over 3,000 labelled X-ray images of fracture types, providing a balanced representation for model training. Since X-ray scans are unstructured data, preprocessing and annotating them appropriately before model training is necessary.

3.3 Model Architecture- YOLOv8 *(For speed)*

YOLO (You Only Look Once) operates as a single-stage detector, enabling it to process images quickly by providing bounding boxes with class confidence scores. It follows a modular architecture (see Appendix - Figure 1) comprising three core components:

- **Backbone**: Responsible for feature extraction from the input image. YOLOv8 uses an advanced CSPDarknet-based backbone that enhances gradient flow and reduces computational load.
- **Neck**: Aggregates features from different scales using a PANet-like structure, which strengthens the model's ability to detect objects of various sizes.
- **Head**: Performs final object detection by predicting bounding boxes, class probabilities, and objectness scores.

3.4 Model Architecture - Faster-CNN *(For accuracy)*

Unlike YOLOv8's real-time, single-stage approach, Faster R-CNN is a two-stage model that prioritises accuracy over speed (see Appendix – Figure 2). It is well-suited for medical imaging, where precise localisation and minimising false negatives are critical. The key difference lies in their detection approach: YOLOv8 detects objects in a single pass, while Faster R-CNN has a two-stage process where it first generates region proposals and then classifies them, enabling more focused, fine-grained detection.

Faster R-CNN consists of the following components:

- **Backbone (Feature Extractor)**: Typically a deep convolutional neural network like ResNet-50 or ResNet-101, used to extract feature maps from the input image.
- **Region Proposal Network (RPN)**: A lightweight neural network that scans the feature maps and proposes candidate object regions (Regions of Interest or RoIs) based on anchor boxes.
- **RoI Pooling Layer**: Converts each region proposal into a fixed-size feature map by aggregating features, ensuring consistency before classification.
- **Head Network (Classification and Regression)**: For each proposed region, the head network classifies the object and refines its bounding box using regression.

3.5 Model Evaluation

YOLOv8 and Faster R-CNN are evaluated using the following metrics.

1. **Mean Average Precision (mAP)**
   Summarises the model's ability to accurately predict fractures in the right locations and classify them correctly.
   - **mAP at 0.5:** This metric evaluates how well the predicted bounding boxes overlap with the ground-truth (actual labelled fractures) boxes at an Intersection-over-Union (IoU) threshold of 0.5.
   - **mAP at 0.5-0.95: This** metric is stricter. It calculates the average precision over multiple IoU thresholds (from 0.5 up to 0.95 in increments of 0.05), providing a broader picture of model robustness.

Note: **IoU** is calculated by dividing the area of overlap between the predicted and actual boxes by the area of their union. A higher IoU (> 0.5) indicates good localisation. For mAP@0.5, any prediction with an IoU less than 0.5 is considered incorrect.

2. **Precision and Recall**
   High precision means the model rarely makes mistakes by incorrectly labelling healthy bone regions as fractured. High recall indicates the model is good at finding nearly all actual fractures and not missing many.

4 Implementation

4.1 Preprocessing Pipeline

A structured preprocessing pipeline was developed to convert the raw, unstructured X-ray images and annotations into a clean, consistent, and model-compatible format.

1. **Image resizing:** All raw X-ray images were resized to 512×512 pixels to standardise input dimensions across the dataset. Unreadable or corrupted image files were identified and skipped.
2. **Annotation Update and Cleaning:** Original annotations were in COCO format (bounding boxes and class labels). Bounding box coordinates were accurately scaled to match the resized image dimensions. Cleaning was performed to remove any missing or invalid annotations.
3. **Data Train-Test split:** The dataset was split into 80% training and 20% validation to ensure the models had sufficient data for learning while preserving a holdout set for evaluation.
4. **Data Augmentation (For training set only):** Augmentations were applied dynamically using the Albumentations library during model training, saving storage space and improving efficiency. Examples of augmentation strategies applied:
    - **Random Horizontal Flip** (probability 0.5)
    - **Gentle Brightness/Contrast Adjustment** (preserving medical clarity)
    - **Random Rotation** (up to ±10°)
    - **Random Scaling** (up to 10%)
    - **Normalisation** (using ImageNet statistics)
5. **Oversampling underrepresented classes:** Additional augmented images were generated for classes with low frequencies within the training set in order to improve the model's exposure to diverse fracture patterns and reduce bias toward majority classes.

4.2 Model Training and Inference

4.2.1 YOLOv8

The training and validation loss curves indicate a healthy and stable learning process (see Appendix - Figure 3). Box_loss, cls_loss, and dfl_loss steadily decreased across epochs for both training and validation sets, with minimal divergence. This indicates the model is effectively learning to localise and classify fractures without overfitting. Precision (metrics/precision(B)) started high (~0.8) but dropped significantly after 10 epochs, becoming more erratic later in training. This may indicate instability from the confidence threshold or a dataset imbalance. In contrast, recall (metrics/recall(B)) steadily improved, reaching ~0.3 by the end, suggesting the model is learning to detect more true positives, though some fractures are still missed.

The mAP50 score, evaluating detection at an IoU threshold of 0.5, steadily increased to about 0.28 by the end of training. Meanwhile, the mAP50-95, representing the average mAP across stricter thresholds, peaked at 0.10. These modest values suggest the model is learning to detect fractures but still has limitations in localisation and classification, making it unsuitable for clinical use at this stage. The current results are promising, showing strong initial progress in detecting bone fractures. With further enhancements, the model provides a solid foundation for developing a practical, robust fracture detection system in medical imaging.

The results in Appendix - Figure 4 showcase the model’s ability to successfully identify and localise various fracture types, including forearm fracture, shoulder fracture, elbow and humerus, with labelled

bounding boxes in distinct colours. From the visualisation, we can tell that the model demonstrates reasonable localisation accuracy, with most bounding boxes correctly surrounding the fractured areas. It is able to recognise different anatomical regions (e.g., forearm, elbow, shoulder) and label them appropriately, which is promising for multi-site fracture detection. A few images do not have any bounding boxes, which may indicate false negatives (missed detections) or true negatives (no fracture present), depending on the ground truth. While the detections are largely correct, the relatively modest performance seen in the earlier mAP metrics suggests that not all fractures are detected, or some bounding boxes may not be precise enough. Improvement could be achieved with more annotated training data, better image preprocessing, or tuning the anchor box settings.

The diagram in Appendix - Figure 5 displays the normalised confusion matrix. It reveals that our YOLOv8 model performs strongly in identifying background regions, with very high accuracy across nearly all classes (e.g., 0.97 for elbow positive, 0.95 for shoulder fracture, and 0.93 for wrist positive). This indicates a conservative prediction strategy, where the model favours background classifications to minimise false positives. However, this cautious behaviour results in a notable under-detection of actual fractures, as many true fracture instances are misclassified as background. For instance, 57% of true forearm fracture cases and 44% of humerus instances are predicted as background, highlighting a tendency to overlook subtle fracture signals. The model performs better in detecting humerus (0.56) and forearm fractures (0.47), indicating more distinguishable visual patterns for these classes. However, true positive rates are low for critical classes like elbow positive (0.03), fingers positive (0.06), and shoulder fracture (0.05). The limited misclassification between fracture types suggests the model can differentiate anatomical regions, with sensitivity being the main challenge rather than specificity.

The Precision-Recall (PR) curve (see Appendix - Figure 6) provides a class-wise and overall evaluation of our YOLOv8 model's ability to detect different types of bone fractures. The overall mean Average Precision at IoU 0.5 (mAP@0.5) across all classes is 0.287, indicating moderate performance with encouraging results in some categories and room for improvement in others. Among the individual classes, humerus achieves the highest AP of 0.585, showing that the model is most effective at detecting fractures in this region. Forearm fracture also shows strong performance with an AP of 0.465, indicating the model has learned meaningful features for this class. Fingers positive (0.249) and elbow positive (0.190) have modest detection, while shoulder fracture (0.149) and wrist positive (0.083) have the lowest AP scores, highlighting challenges for the model in these classes. The PR curves show a sharp decline in precision as recall increases, indicating the model confidently predicts at low recall but struggles with precision as it captures more true positives. This trade-off reflects a conservative approach, prioritising precision and minimising false positives, but at the cost of missing more fractures.

4.2.2 Faster R-CNN

The Faster-RCNN model, developed using Detectron2 with a ResNeXt-101-FPN backbone, was trained over 3000 iterations on a custom bone fracture dataset annotated in COCO format. Dataset registration was handled via the register_coco_instances() function for both training and validation sets. The training pipeline was configured using Detection2's DefaultTrainer, and evaluation was conducted using the COCOEvaluator.

The dataset used for training and validation was preprocessed to a fixed resolution of 512x512 pixels. All images were resized uniformly, and their corresponding COCO-format annotations were updated accordingly to scale the bounding box coordinates. Additionally, a custom FractureDataset class was implemented to load the data, applying Albumentations-based augmentations during training. These included horizontal flips, brightness/contrast changes, rotations, and random scaling, designed to improve

model robustness to positional and lighting variations. For validation, only normalisation was applied to maintain consistency without introducing noise.

During training, the total loss, which includes classification loss, bounding box regression loss, and region proposal network (RPN) loss, was logged and visualised using TensorBoard. The loss curve (Figure 6) showed a steep drop during the first 500 iterations, followed by stabilisation with minor fluctuations, indicating effective convergence. Toward the later stages, the curve hinted at potential overfitting beyond iteration 2500.

Evaluation on the validation set using inference_on_dataset() and COCOEvaluator revealed that the model achieved a mean Average Precision (mAP) of approximately 0.40 at IoU=0.50 (mAP@50) and around 0.10 at IoU=0.50–0.95 (mAP@50–95). These metrics suggest that while the model could correctly detect fracture regions with reasonable bounding box overlap, it struggled with stricter localisation thresholds.

To better understand the model's predictions, bounding box visualisations were generated on validation X-ray images. The model successfully identified and localised various fracture types, such as a shoulder fracture with 88% confidence and a forearm fracture with 90% confidence, demonstrating its strength in detecting larger or clearer fractures. However, in several instances, no bounding boxes were produced, indicating either missed detections or conservative confidence thresholds.

In addition to COCO metrics, we generated a confusion matrix and precision-recall (PR) curves to evaluate class-level performance. The confusion matrix highlighted that some classes, such as "humerus fracture" and "forearm fracture", were frequently confused with the background class, reflecting a lack of sensitivity in certain regions. The PR curves further confirmed this trend: although some classes achieved high AP, others had flatter curves and lower precision at higher recall values.

A class-wise analysis of Average Precision (AP) revealed significant variation:

- Humerus fracture: AP = 0.525
- Forearm fracture: AP = 0.332
- Fingers positive: AP = 0.247
- Elbow positive: AP = 0.033

These results show that while the model performed well on more prominent or better-represented classes, it struggled with subtle or underrepresented types like elbow fractures. The overall mAP@0.5 across all classes was approximately 0.34, reinforcing that the model tends to prioritise precision over recall, often missing less obvious fractures.

The model successfully identifies and localises various fracture types and bone regions (see Appendix - Figure 7), including fingers (confidence: 81%), forearm fractures (88%), and shoulder fractures (86% and 88%). These examples illustrate the model's ability to detect anomalies across different anatomical areas.

The total loss (see Appendix - Figure 8) dropped sharply within the first 100 iterations, reflecting rapid initial learning. Subsequently, the loss curve fluctuated around 0.5–1.0, indicating model convergence with minimal overfitting.

The model performs well on classes such as elbow positive, wrist positive, and humerus, with accuracy above 95%. However, it struggles with forearm and shoulder fractures, indicating potential confusion between similar anatomical regions (see Appendix - Figure 9).

The highest average precision (AP) was achieved for shoulder fracture (AP=0.437), followed by fingers positive (AP=0.332). The overall mean Average Precision at IoU threshold 0.5 (mAP@0.5) across all classes is 0.213, suggesting moderate detection performance and room for improvement (see Appendix - Figure 10).

In summary, the Faster-RCNN model demonstrates promising potential in detecting bone fractures from X-ray images, particularly for prominent and well-represented classes such as shoulder and forearm fractures. The training process showed effective convergence with minimal overfitting, and the model achieved a moderate mAP@0.5 of 0.34, reflecting its ability to detect fractures with reasonable accuracy. However, evaluation metrics such as the confusion matrix and PR curves revealed challenges in classifying more subtle or underrepresented fracture types, pointing to the need for improved sensitivity and better handling of class imbalance.

4.3 Evaluation Summary: YOLOv8 vs. Faster R-CNN

A clear trade-off exists between speed and accuracy in the performance of YOLOv8 and Faster R-CNN. YOLOv8 offers faster inference and is better suited for real-time applications, but may miss subtle fractures due to its lower sensitivity. In contrast, Faster R-CNN achieves higher detection accuracy and better localisation, especially for complex or small fractures, though it requires more computation and time. The model choice ultimately depends on the clinical context — YOLOv8 for speed and Faster R-CNN for precision.

Metric	YOLOv8	Faster R-CNN
mAP at 0.5	0.287	0.34
mAP at 0.5-0.95	0.10	0.10
Precision	0.30	0.42
Recall	0.31	0.51

Table 1: Summary of Metrics for YOLOv8 and Faster R-CNN models

5 Conclusion

We developed and evaluated an intelligent bone fracture detection system using state-of-the-art deep learning models, YOLOv8 and Faster R-CNN. A robust preprocessing pipeline ensured data consistency and improved model generalisation across fracture types. YOLOv8 excelled in real-time detection with moderate accuracy, while Faster R-CNN achieved higher precision and recall for complex cases, though at a greater computational cost.

Our results highlight the trade-off between speed and accuracy in medical imaging AI. YOLOv8 suits fast-paced clinical settings, whereas Faster R-CNN is better for scenarios prioritising diagnostic precision. Both models still have room for improvement, particularly in detecting subtle fractures and addressing class imbalances.

Future work could focus on advanced augmentation, improved loss functions, and exploring ensemble or transformer-based models. With ongoing refinement and clinical deployment, this AI system could significantly enhance fracture diagnosis, reduce radiologist workload, and improve patient care.

6 References

[1] “Deep Learning Archives,” NVIDIA Blog, https://blogs.nvidia.com/blog/category/enterprise/deep-learning/ (accessed Apr. 14, 2025).

[2] S. Karim, Y. Zhang, S. Yin, I. Bibi, and A. A. Brohi, “A brief review and challenges of object detection in optical remote sensing imagery,” Multiagent and Grid Systems, vol. 16, no. 3, pp. 227–243, Oct. 2020. doi:10.3233/mgs-200330

[3] Darabi, P. K. (2024, July 15). *Bone Fracture Detection: Computer Vision Project*. Kaggle. https://www.kaggle.com/datasets/pkdarabi/bone-fracture-detection-computer-vision-project

[4] Arturo-Bandini-Jr. (2024, January 14). *Bone Fracture Detection Detection (coco annots)*. Kaggle. https://www.kaggle.com/datasets/banddaniel/bone-fracture-detection-detection-coco-annots

7 Appendix

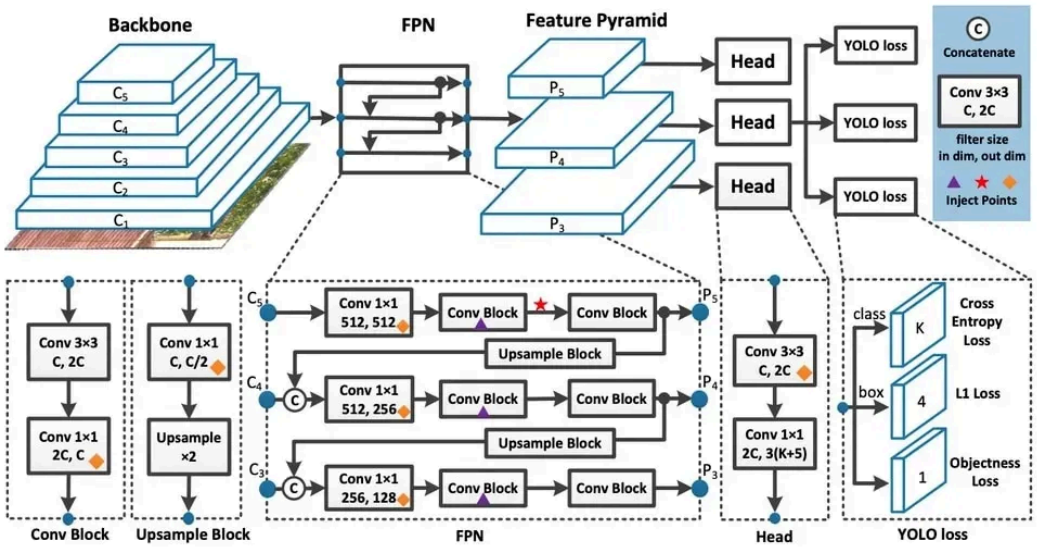

Figure 1: YOLOv8 Architecture [1]

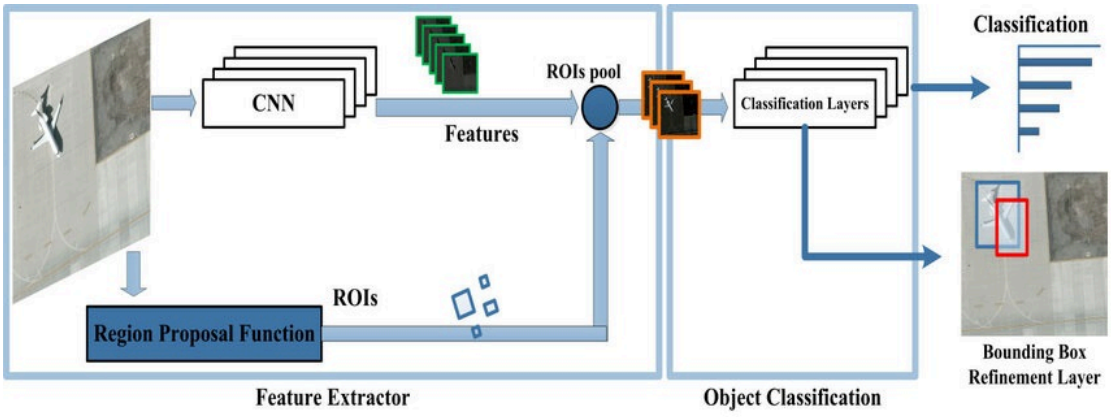

Figure 2: R-CNN Architecture [2]

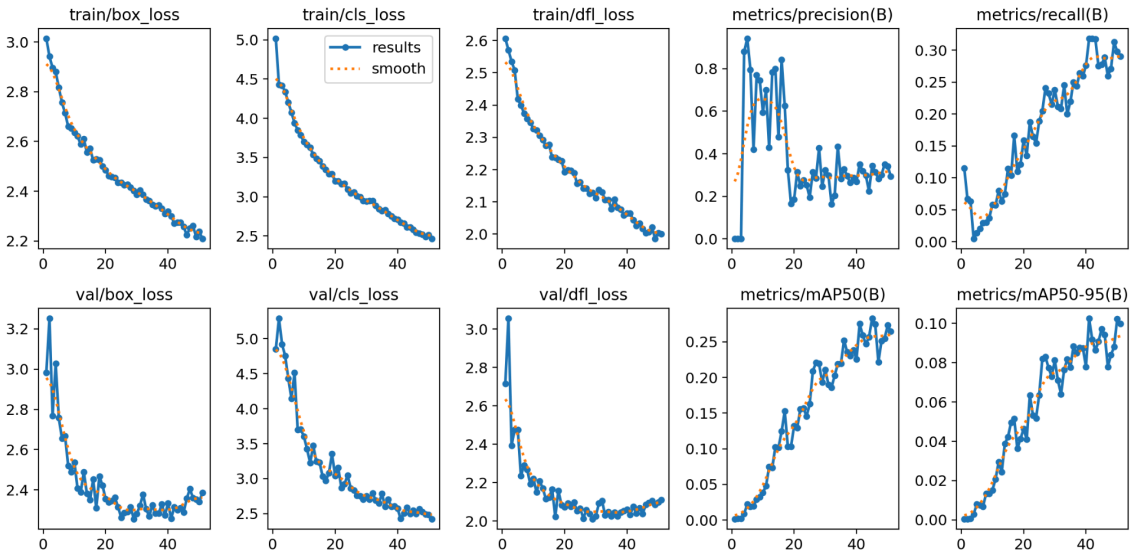

Figure 3: Performance Metrics and Loss Trends for YOLOv8 Training and Validation

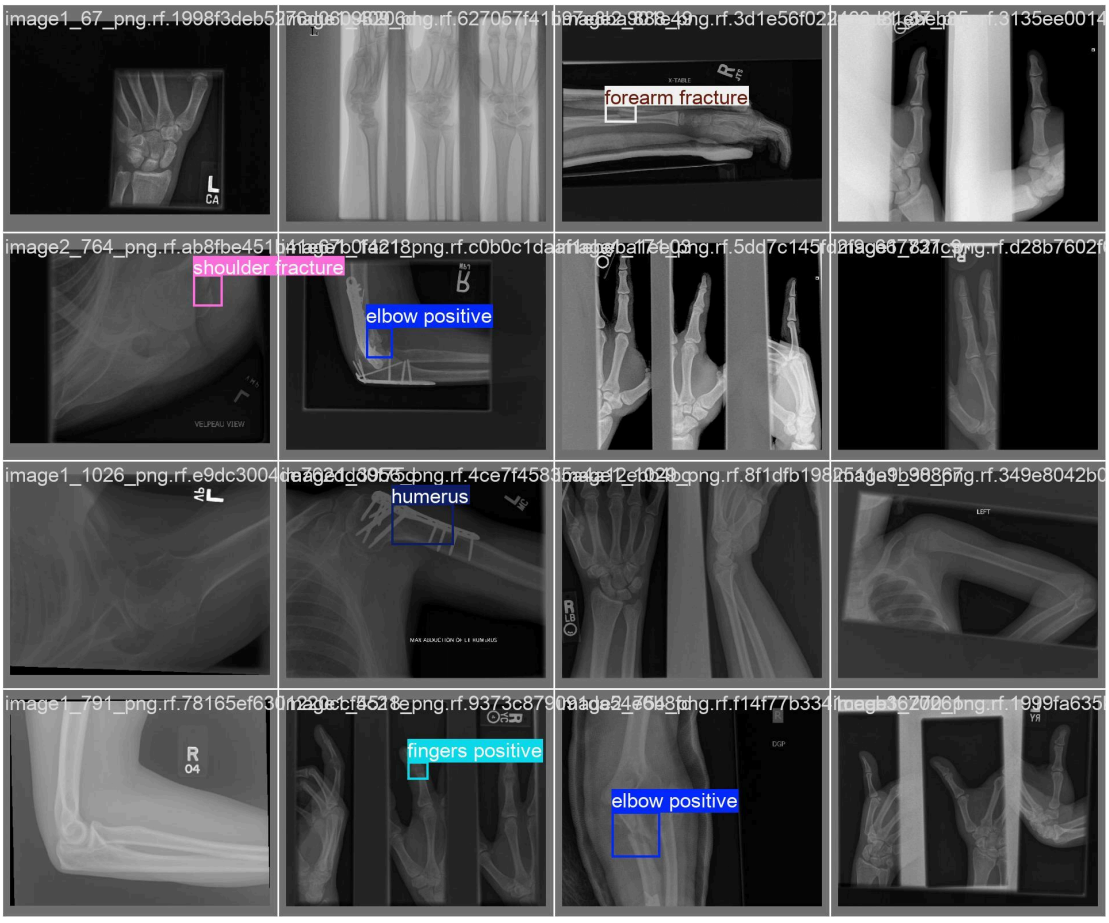

Figure 4: Annotated X-ray Images Showing Fracture Types Across Anatomical Regions

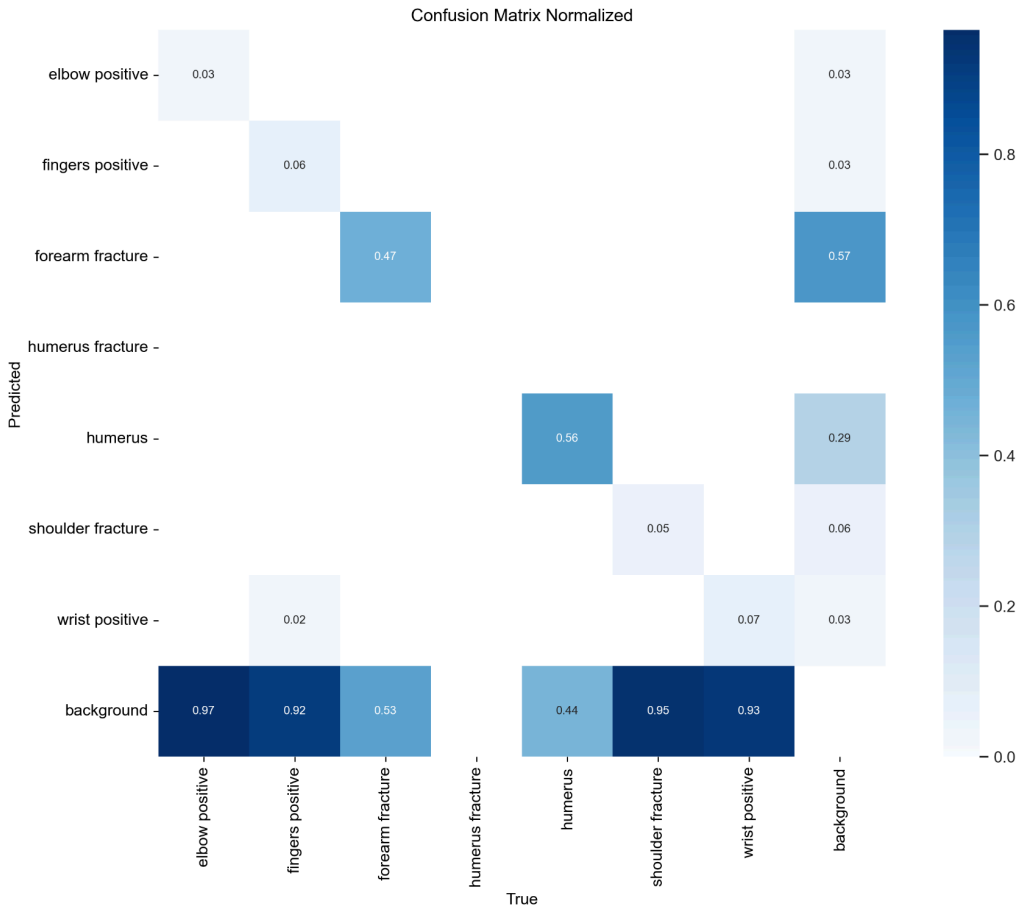

Figure 5: Normalised Confusion Matrix for YOLOv8 Predictions on Validation Set

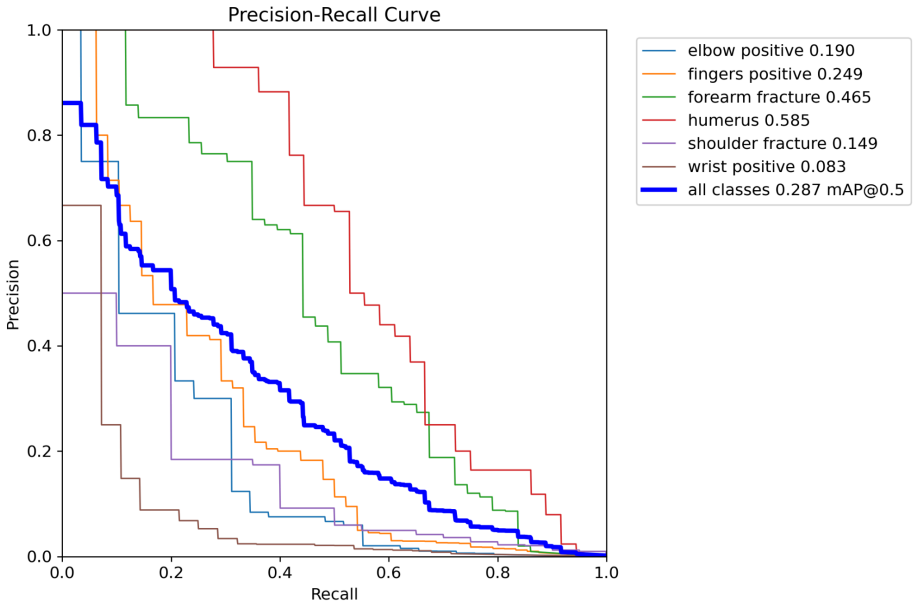

Figure 6: Precision-Recall Curve for YOLOv8 Across Fracture Classes on Validation Set

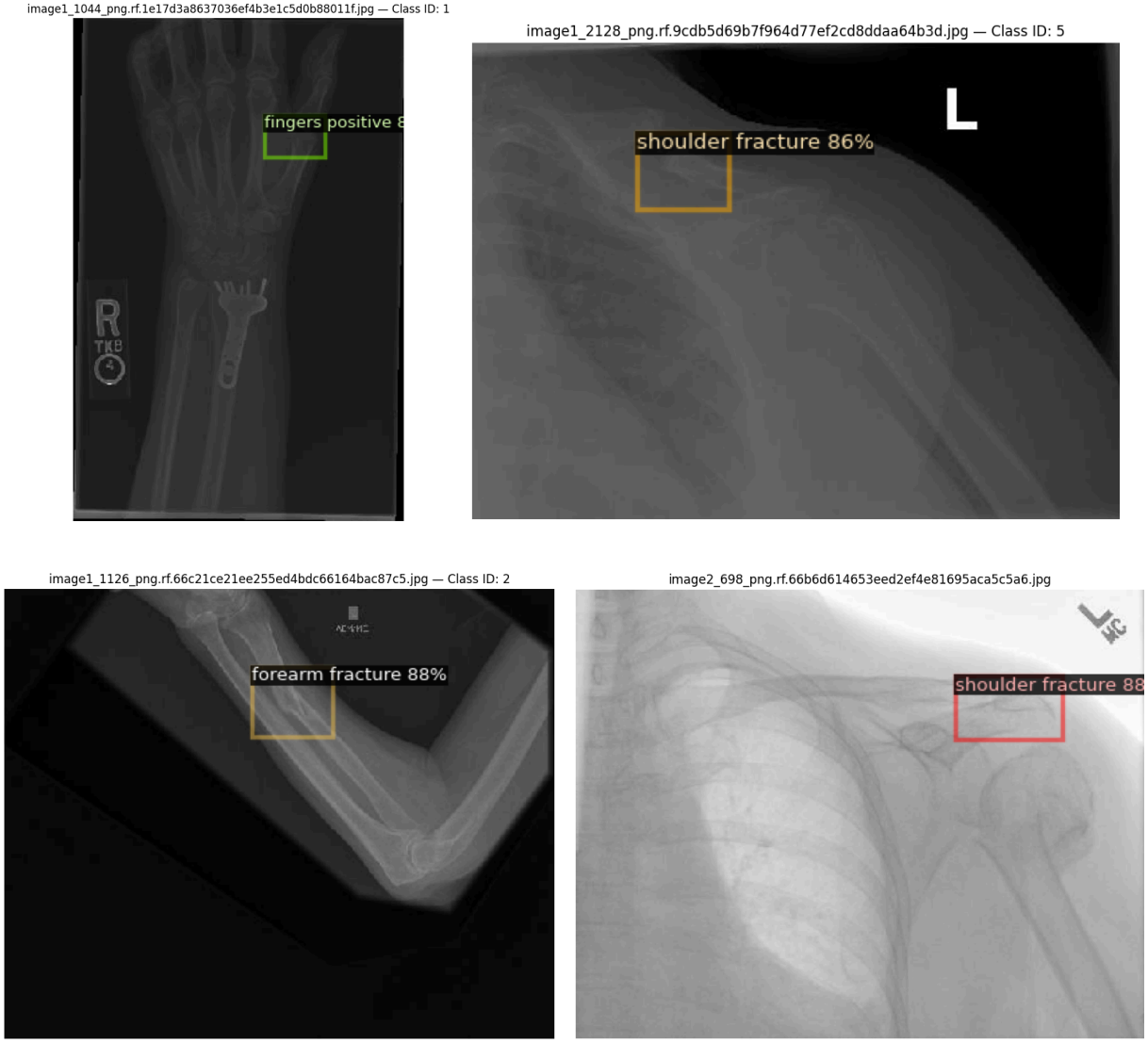

Figure 7: Sample predictions from the Faster-RCNN model on validation X-ray images

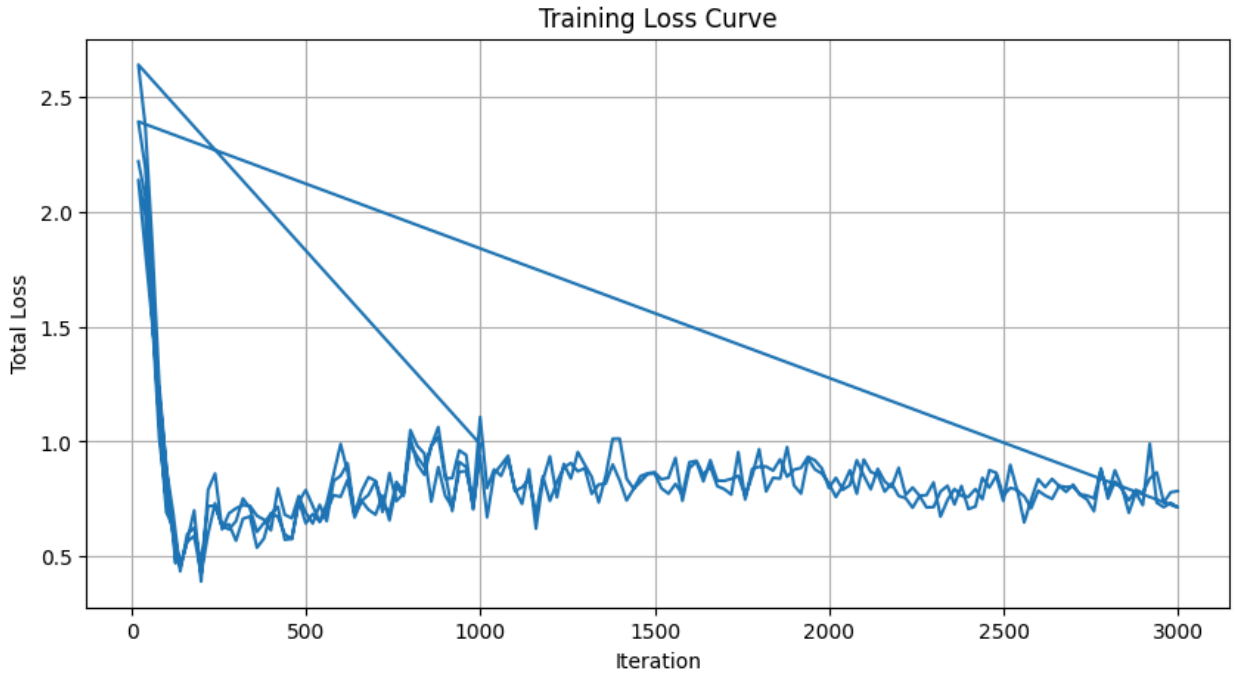

Figure 8: Training loss curve of the Faster-RCNN model over 3000 iterations

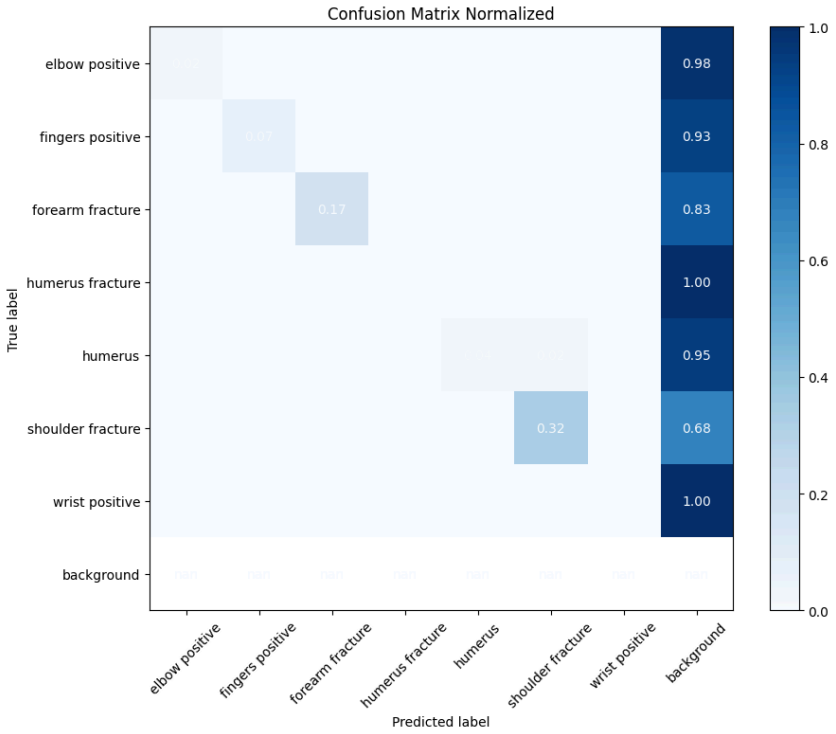

Figure 9: Normalised confusion matrix showing the classification performance of the Faster-RCNN model across different classes

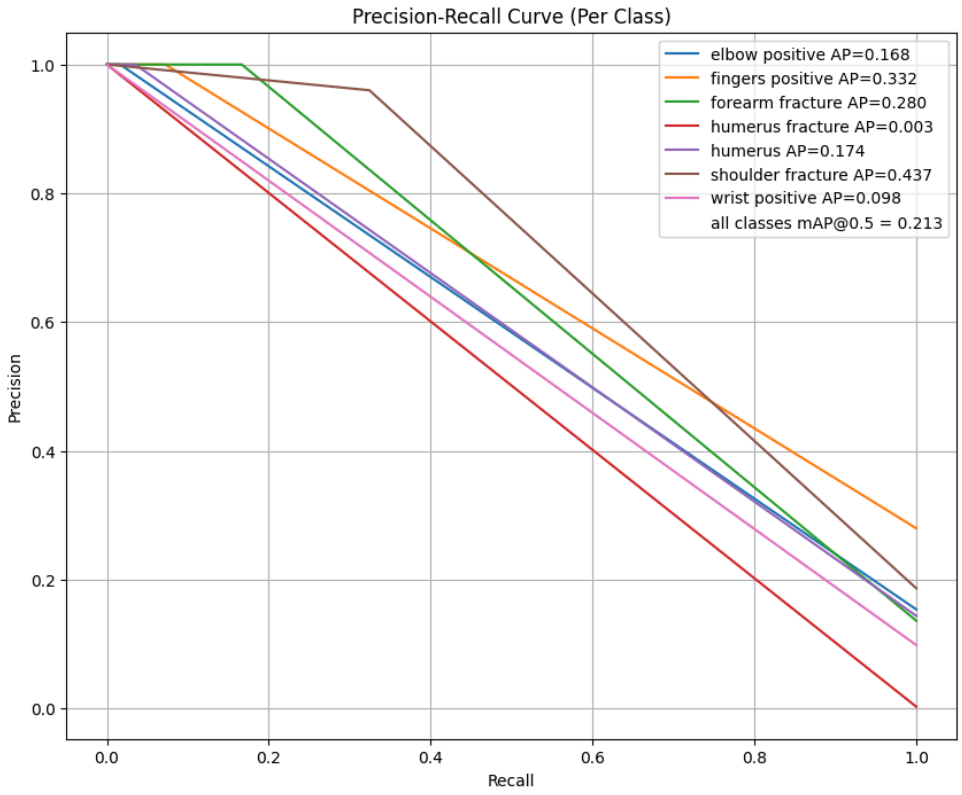

Figure 10: Precision-Recall (PR) curves per class for the Faster-RCNN model