

Methodologically Sound Class-Imbalance Handling in YOLOv8 Hand Fracture Detection via Training-Only Oversampling

Alfaturachman Maulana Pahlevi ^{1*}, Rohimatun Nisfah ^{2*}, Meysilla Cindy Silsiana ^{3*}, Cinantya Paramita ^{4**}

^{*} Department of Informatics Engineering, Universitas Dian Nuswantoro, Semarang

^{**} Dinus Research Group for AI in Medical Science (DREAMS), Department of Informatics Engineering, Universitas Dian Nuswantoro, Semarang

111202516609@mhs.dinus.ac.id ¹, 111202415915@mhs.dinus.ac.id ², 111202415824@mhs.dinus.ac.id ³,
cinantya.paramita@dsn.dinus.ac.id ⁴

Article Info

Article history:

Received 2026-07-10

Revised 2026-09-11

Accepted 2026-10-05

Keyword:

Class imbalance,
Hand fracture detection,
Object Detection,
Oversampling,
YOLOv8.

ABSTRACT

Hand fractures are frequently encountered in emergency settings and require prompt radiological interpretation. YOLOv8-based detection frameworks offer strong potential for automated fracture detection, but class imbalance, in which fracture images are substantially outnumbered by normal cases, can reduce model sensitivity. This study proposes a leakage-aware evaluation protocol in which oversampling is strictly confined to the training partition, thereby preventing contamination of the validation and test sets. This design addresses a methodological gap that has not been consistently applied in prior fracture detection studies, including those using the FracAtlas dataset. The study evaluates and compares training-set oversampling on YOLOv8s and YOLOv8m using a hand-only FracAtlas subset (1,538 images; fracture-to-non-fracture ratio 1:2.5). Under imbalanced conditions, Recall was low for both models (YOLOv8s: 0.482; YOLOv8m: 0.469). Following oversampling, YOLOv8m achieved Recall of 0.812 (+73.2%) and mAP@50 of 0.832 (+66.6%), while YOLOv8s reached Recall of 0.723 (+50.2%) and mAP@50 of 0.815 (+59.3%). Image-level evaluation yielded Specificity above 0.96, ROC-AUC above 0.967, and an FNR of 0.061 for both models, indicating suitability for clinical screening. Efficiency benchmarking showed that YOLOv8s runs at 93.61 FPS with a peak VRAM of 73.97 MB, compared to 39.25 FPS and 194.97 MB for YOLOv8m. Grad-CAM analysis confirmed that both models focus their activation on anatomically relevant bone regions. Despite a modest accuracy difference ($\Delta\text{mAP@50} = 0.017$), YOLOv8s requires 2.3 times fewer parameters, making it a practical option for resource-constrained settings. The results suggest that leakage-aware oversampling is a promising strategy for improving fracture detection sensitivity in imbalanced medical imaging datasets, though conclusions are based on single-seed training runs and require multi-seed validation and statistical hypothesis testing to confirm generalizability.



This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

I. INTRODUCTION

Hand fractures rank among the most frequently encountered musculoskeletal conditions managed in emergency settings, with upper-limb injuries contributing substantially to both emergency department admissions and functional disability among patients of working age [1], [2]. Because the hand plays an essential role in daily activities,

delays or errors in diagnosis may lead to complications such as malunion, joint stiffness, and permanent functional loss [3].

Conventional diagnosis relies on radiographic interpretation by trained clinicians. Although plain X-ray radiography remains the primary imaging modality because it is inexpensive, rapid, and widely available, its interpretation requires considerable expertise [4]. Non-displaced and

hairline fractures in particular present with subtle radiological signs that can be easily overlooked, especially under the high-workload conditions commonly found in emergency settings. These challenges have increased interest in deep learning-based CAD systems that can support faster and more consistent radiographic interpretation [6], [7].

The YOLO series has attained considerable popularity within deep learning-based object detection frameworks, owing to its favorable trade-off between detection precision and computational economy, which renders it particularly well-suited for time-critical clinical deployment [8], [9]. Several studies have applied YOLO variants to fracture detection across diverse anatomical sites, including pediatric wrist [10], forearm [11], and mandible [12], with competitive results. For instance, Meza et al. [11] evaluated a modified YOLOv8 model on the FracAtlas hand dataset; however, their approach did not explicitly address class imbalance, resulting in suboptimal recall and missed fracture detections. Furthermore, many existing fracture detection studies suffer from methodological flaws such as applying data augmentation prior to dataset partitioning, which causes severe data leakage and artificially inflates performance metrics [14], [15].

Class imbalance remains a critical bottleneck in medical image analysis because pathological images are naturally far less common than normal radiographs. In the hand-only subset of FracAtlas used in this study, fracture images constitute only 28.5% of the dataset (ratio 1:2.5). When trained on imbalanced data, models develop a bias toward the majority normal class, resulting in low Recall (sensitivity) and increased risks of missed clinical diagnoses [13].

To address class imbalance without introducing optimistic evaluation bias, this study proposes a leakage-aware oversampling framework. The novelty of this work is twofold: (1) establishing a rigorous, leakage-aware evaluation protocol for hand fracture detection in which oversampling and augmentation are strictly confined to the training partition, thereby preserving the authentic imbalanced distribution in the validation and test sets. This addresses a methodological gap that, to the best of our knowledge, has not been explicitly considered in prior FracAtlas hand-subset benchmarks, including Meza et al. [11], whose YOLOv8-based approach did not incorporate class-imbalance handling and consequently reported suboptimal recall; and (2) providing the first empirical model-scale selection guideline for this dataset by comparing YOLOv8s and YOLOv8m across clinical sensitivity (Recall and FNR), image-level screening metrics (Specificity and ROC-AUC), and measured inference efficiency (FPS and peak VRAM), all of which are

directly relevant to resource-constrained emergency deployment.

Specifically, the objectives of this study are: (1) evaluate the baseline detection performance of YOLOv8s and YOLOv8m under class imbalance; (2) implement and validate the controlled training-set oversampling strategy; and (3) perform a comprehensive quantitative and qualitative evaluation encompassing object detection metrics (mAP, Recall, Precision, F1-Score, FNR), image-level clinical metrics (Specificity, ROC-AUC), inference efficiency (latencies, FPS, peak VRAM), and Explainable AI feature activation maps (Grad-CAM).

By directly contrasting with unhandled imbalanced baselines (e.g., Meza et al. [11]) under a strict leakage-aware protocol, this study provides a reliable methodological reference for deploying lightweight object detectors in resource-constrained emergency healthcare settings.

II. METHODS

A. Dataset and Hand-Subset Filtering

This study uses FracAtlas, an open-source and anonymized orthopedic X-ray dataset developed for the tasks of fracture categorization, anatomical localization, and segmentation research [16]. The full repository comprises 4,083 radiographs spanning multiple skeletal regions. To maintain focus on hand fracture detection, only images showing the hand region were selected. After the filtering process, 1,538 images remained, consisting of 438 fracture images (28.5%) and 1,100 non-fracture images (71.5%). The dataset contains 549 fracture instances (bounding box annotations), corresponding to an average of 0.357 bounding boxes per image, reflecting that not every image contains a visible fracture lesion.

It should be noted that FracAtlas does not provide patient identifiers; image filenames are sequential numeric codes (e.g., IMG0000040.jpg). Consequently, data partitioning was conducted at the image level in the absence of patient identifiers. The possibility of radiographs originating from the same patient appearing in different subsets cannot be completely ruled out and is considered a limitation of this experimental setup.

B. Stratified Dataset Split

The dataset was partitioned using stratified splitting with a 70:15:15 proportion allocated to the training, validation, and test partitions, in that order. Stratification was applied at the image level to preserve a consistent fracture-to-non-fracture proportion across all three subsets, as detailed in Table I.

TABLE I
DATASET DISTRIBUTION PRIOR TO OVERSAMPLING (IMBALANCED CONDITION)

Split	Total Images	Fracture Images	Non-Fracture Images	Fracture Instances
Train	1,076	306	770	383
Validation	231	66	165	81
Test	231	66	165	85
Total	1,538	438	1,100	549

C. Oversampling Strategy for the Training Subset

The 1:2.5 class imbalance was addressed through oversampling restricted exclusively to the training data. Random oversampling, which replicates minority-class images with light augmentation, was applied to handle class imbalance within deep learning frameworks, especially in scenarios characterized by limited data availability [2].

Fracture images in the training subset were replicated with minimal, anatomically safe augmentations (rotation $\pm 10^\circ$ and

horizontal flip) until the fracture-to-non-fracture ratio approached 1:1. Replicating images accompanied by light transformations expands visual pattern diversity, helping to mitigate exact-duplicate overfitting. This procedure yielded a final training subset of 1,540 images: 770 original non-fracture images, 306 original fracture images, and 464 augmented fracture replications. This resulted in an average replication factor of approximately 1.5 per fracture image. The complete post-oversampling distribution is presented in Table II.

TABLE II
DATASET DISTRIBUTION AFTER OVERSAMPLING

Split	Total Images	Fracture Images	Non-Fracture Images	Fracture Instances
Train	1,540	770	770	945
Validation	231	66	165	81
Test	231	66	165	85
Total	2,002	902	1,100	1,111

All replication and augmentation steps were performed after dataset splitting was finalized and applied solely to the training subset. No replicated or augmented images entered the validation or test subsets. This design helps prevent data leakage that may occur if oversampling is applied before splitting, where original and replicated images could appear in different subsets [14], [15]. By maintaining validation and test subsets in their original imbalanced distribution (28.5% fracture), the improvement observed on the test subset can be associated with the oversampling strategy rather than changes in the evaluation data distribution.

TABLE III
AUGMENTATION POLICY APPLIED DURING OVERSAMPLING

Augmentation	Value	Justification
Horizontal Flip	$p = 0.5$	Anatomically safe; left-right symmetry of the hand
Rotation	$\pm 10^\circ$	Accommodates patient positioning variation during imaging
Brightness/Contrast	Random	Simulates X-ray exposure and bone density variability
Translation	5%	Accounts for positional variation of the hand within the frame
Scale	$\pm 15\%$	Accounts for variable patient-detector distance
Vertical Flip	0.0 (disabled)	Reverses proximal-distal bone orientation, critical for fracture interpretation
Shear	0.0 (disabled)	Produces anatomically unrealistic distortions
Perspective	0.0 (disabled)	Invalid for X-ray, which is an orthogonal projection
Hue/Saturation	0.0 (disabled)	Irrelevant; X-ray images are grayscale

Oversampling & Augmentation Visual Results (FracAtlas Hand Subset)

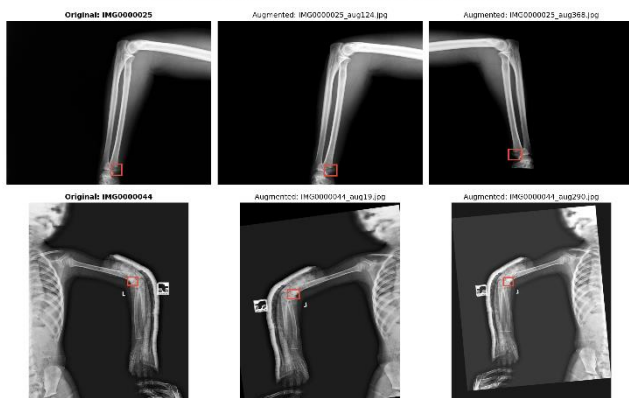


Figure 1. Examples of oversampled fracture images after augmentation.

D. Augmentation Policy During Oversampling

Augmentations applied during the oversampling process were restricted to transformations that are anatomically safe for X-ray images. Table III summarizes the augmentation policy along with the rationale for each parameter.

E. Model Architecture: YOLOv8s and YOLOv8m

Two variants of the YOLOv8 architecture developed by Ultralytics [17], namely YOLOv8s (small) and YOLOv8m (medium), were selected for this study to compare model performance based on computational requirements and feature extraction capability.

YOLOv8n (Nano, 3.2M parameters) was not included because its substantially smaller capacity may limit its ability to represent subtle bone fractures. Conversely, larger variants (YOLOv8l and YOLOv8x) carry significantly higher computational and memory demands with an elevated risk of overfitting on a modest dataset of 1,538 images. Therefore, YOLOv8s and YOLOv8m were selected as balanced representatives for lightweight and medium model scales. The architectural specifications of both models are summarized in Table IV.

TABLE IV
ARCHITECTURAL AND INFERENCE EFFICIENCY
SPECIFICATIONS OF YOLOV8S AND YOLOV8M

Specification/Metric	YOLOv8s	YOLOv8m
Parameters	11.16 million	25.89 million
GFLOPs (imgsz=640)	28.7	79.1
Inference Latency	10.68 ms / image	25.48 ms / image
Throughput	93.61 FPS	39.25 FPS
Peak Inference VRAM	73.97 MB	194.97 MB
Backbone	CSPDarknet (C2f)	CSPDarknet (C2f)
Neck	PAN-FPN	PAN-FPN
Detection Head	Decoupled Detect	Decoupled Detect
Pre-trained on	COCO (80 classes)	COCO (80 classes)

YOLOv8s contains approximately 11.16 million parameters and requires 28.7 GFLOPs. In empirical benchmark testing (at input resolution 640×640), YOLOv8s achieved an average inference latency of 10.68 ms per image (93.61 FPS) with a peak VRAM footprint of only 73.97 MB. These low resource requirements confirm its suitability for deployment in resource-constrained computing environments, including edge devices and local clinic workstations. In comparison, YOLOv8m possesses 25.89 million parameters (2.3× larger) and requires 79.1 GFLOPs. During inference, YOLOv8m registered a latency of 25.48 ms per image (39.25 FPS) and consumed 194.97 MB of peak VRAM representing a 2.4× increase in latency and a 2.6× increase in memory consumption relative to YOLOv8s [18]. Both models maintain real-time throughput (>30 FPS), establishing their practical viability for integration into clinical diagnostic pipelines.

Both models were fine-tuned from COCO-pretrained weights with a single-class detection head configured for the fracture class. Transfer learning from COCO was applied to utilize previously learned feature representations and support model training on the relatively small medical imaging dataset.

F. Training Configuration

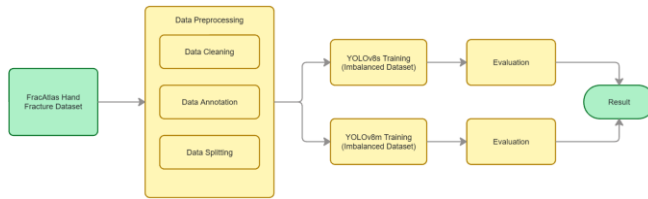


Figure 2. Training pipeline under the imbalanced (baseline) condition.

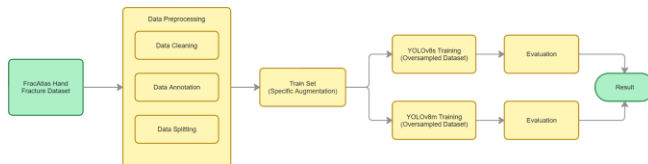


Figure 3. Training pipeline with the oversampling strategy applied.

Training hyperparameters for both models are detailed in Table V. To guarantee experimental reproducibility, each configuration underwent a single training session with a deterministic random seed (seed = 42).

TABLE V
TRAINING HYPERPARAMETER CONFIGURATION

Hyperparameter	Value	Justification
optimizer	AdamW	Stable optimizer for fine-tuning on small medical datasets
lr0	0.001	Reduced from default 0.01 to prevent gradient overshoot in early epochs
lrf	0.01	Final learning rate ratio with cosine decay scheduler
epochs	100	With early stopping (patience = 20 epochs)
imgsz	640	Balances fracture detail preservation and VRAM capacity
batch	4	Adjusted for available VRAM capacity (NVIDIA GeForce RTX 3050 Laptop GPU)
cos_lr	True	Cosine learning rate scheduling for smoother convergence
warmup_epochs	5	Extended warmup for training stability in early epochs
weight_decay	0.0001	Light regularization to retain fine-grained fracture features
amp	True	Automatic Mixed Precision for VRAM efficiency
seed	42	Fixed seed for experimental reproducibility

AdamW was selected as the optimization strategy due to its separation of weight decay regularization from the adaptive learning rate adjustment mechanism, thereby applying regularization more consistently regardless of gradient magnitude scaling [19]. On small medical datasets, training stability is particularly important given the elevated risk of overfitting [20]. The starting learning rate was lowered to 0.001, compared to the default value of 0.01 to prevent gradient overshoot in early epochs, while cosine decay scheduling and extended warmup facilitated a smoother domain transition from natural images to X-ray radiographs [21].

Training was conducted using Ultralytics framework v8.4.64 using Python 3.12.13, PyTorch 2.11.0+cu128, and CUDA 12.8, accelerated by a NVIDIA GeForce RTX 3050 Laptop GPU. Total training time was 1 hour 18 minutes for YOLOv8s and 2 hours 4 minutes for YOLOv8m, each reaching the early stopping criterion (patience = 20 epochs) or the maximum of 100 epochs.

G. Conservative Data Augmentation During Training

In addition to the oversampling augmentation policy, the default YOLOv8 on-the-fly training augmentations were also adjusted to suit X-ray image characteristics, as summarized in Table VI. The mosaic augmentation probability was reduced to 0.5 (from the default 1.0) to limit image-merging artifacts that could disrupt the spatial integrity of bone structures. The hsv_v value was retained at 0.2 to simulate inter-patient variation in X-ray exposure and bone density [22].

TABLE VI
ON-THE-FLY TRAINING AUGMENTATION CONFIGURATION

Parameter	Value	Rationale
degrees	10.0°	Simulates patient hand positioning variation
fliplr	0.5	Safe; hand anatomy is relatively bilaterally symmetric
flipud	0.0 (disabled)	Proximal–distal orientation is diagnostically critical
shear	0.0 (disabled)	Produces anatomically unrealistic deformations
perspective	0.0 (disabled)	X-ray images are orthogonal projections
mosaic	0.5	Reduced from default 1.0 to prevent image-merging artifacts
mixup	0.1	Low intensity; used for regularization only
copy_paste	0.0 (disabled)	Pasting fracture regions across images is medically invalid
hsv_h / hsv_s	0.0 (disabled)	Irrelevant for grayscale X-ray images
hsv_v	0.2	Simulates X-ray exposure and bone density variation across patients

H. Evaluation Metrics

To comprehensively assess model performance across both bounding box localization and clinical screening utility, evaluation encompassed object-level detection metrics (Precision, Recall/Sensitivity, F1-Score, mAP@50, mAP@50–95, and False Negative Rate) as well as image-level classification metrics (Specificity and ROC-AUC), as detailed in Table VII. In clinical triage, evaluating whether an image contains any fracture (binary image-level classification) is paramount alongside localized detection. Object-level metrics (Precision, Recall, F1-Score, and FNR) were evaluated at a fixed confidence threshold of 0.25 and an IoU threshold of 0.50, whereas mAP@50 and mAP@50–95 were derived across varying confidence levels using standard Ultralytics protocols. For image-level evaluation, an image was categorized as positive if at least one bounding box met or exceeded the 0.25 confidence threshold (taking the maximum confidence score as the image-level score), enabling the calculation of image-level Specificity, False Negative Rate (FNR = 1 - Sensitivity), and ROC-AUC across the 1,100 non-fracture and 438 fracture images.

TABLE VII
EVALUATION METRICS USED IN THIS STUDY

Metric	Description	Priority
mAP@50	Mean Average Precision at IoU ≥ 0.50	Primary
Recall (Sensitivity)	Proportion of true fracture instances correctly detected	Primary
Specificity	Proportion of non-fracture images correctly identified (Image-Level)	Primary
FNR	False Negative Rate; proportion of true fracture cases missed (1 - Recall)	Primary
ROC-AUC	Area under the ROC curve based on maximum image confidence score	Secondary
mAP@50–95	Mean Average Precision averaged over IoU thresholds 0.50–0.95	Secondary

Each metric is defined and described individually below.

Precision represents the fraction of positive detections that genuinely correspond to actual fracture cases. This metric describes the ability of the model to reduce false positive predictions, where non-fracture images are incorrectly detected as fractures. A high Precision value indicates that most fracture detections produced by the model are correct.

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

where TP (True Positive) denotes the count of accurately identified fractures, whereas FP (False Positive) refers to the number of detections that do not align with genuine fracture instances.

Recall reflects the model's ability to correctly identify genuine fracture cases. In this study, Recall is considered an important evaluation metric because missed fracture cases (false negatives) may occur when the model fails to identify existing fractures. Low Recall values indicate that more fracture cases are undetected, which may affect clinical decision-making and potentially lead to delayed treatment or complications such as malunion and functional impairment.

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

where FN (False Negative) denotes the count of genuine fracture cases that went undetected by the model.

False Negative Rate represents the proportion of actual fracture cases that the model fails to detect. In clinical triage, minimizing FNR is paramount because missed fractures (false negatives) carry higher risks of delayed treatment than false positives.

$$FNR = \frac{FN}{TP + FN} = 1 - Recall \quad (3)$$

Specificity measures the proportion of non-fracture (normal) images correctly identified as negative. It evaluates the model's ability to avoid false alarms on healthy radiographs.

$$Specificity = \frac{TN}{TN + FP} \quad (4)$$

where TN (True Negative) denotes non-fracture images correctly predicted without any bounding box above the confidence threshold, and FP (False Positive) represents non-fracture images incorrectly flagged with fracture bounding boxes.

ROC-AUC measures the discriminatory capacity of the model across all possible confidence thresholds by plotting Sensitivity against 1 - Specificity. For each image, the prediction score is defined as the maximum bounding box confidence score.

The F1-Score, defined as the harmonic mean of Precision and Recall, offers a unified metric that balances both aspects of model performance. In contrast to the arithmetic mean, the harmonic mean imposes a greater penalty when there is a substantial discrepancy between Precision and Recall, making it a more informative metric in situations where both false positives and false negatives entail significant consequences.

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (5)$$

An F1-Score close to 1.0 indicates that the model achieves both high Precision and high Recall simultaneously.

IoU measures the overlap between the predicted bounding box and the ground-truth bounding box. This metric is used to determine the localization accuracy of detected objects. A prediction is considered a True Positive when the IoU value between the predicted box and the ground-truth annotation reaches or exceeds the defined threshold (0.50 in this study).

$$\text{IoU} = \frac{\text{Area}(B_{\text{pred}} \cap B_{\text{gt}})}{\text{Area}(B_{\text{pred}} \cup B_{\text{gt}})} \quad (6)$$

where B_{pred} denotes the predicted bounding box and B_{gt} represents the ground-truth bounding box. An IoU value of 0.50 or higher serves as the conventional threshold in standard object detection evaluation frameworks.

Mean Average Precision at IoU = 0.50 (mAP@50) is the primary localization-aware detection metric used in this study. For every class, Average Precision (AP) is derived by calculating the area beneath the Precision-Recall curve using a fixed IoU threshold of 0.50. Because this study involves a single detection class (fracture), mAP@50 equals the AP for that class. Unlike Precision or Recall evaluated at a single confidence threshold, mAP@50 integrates performance across all confidence levels, offering a more thorough assessment of the model's ability to distinguish between classes.

$$\text{mAP}_{50} = \text{AP}_{@ \text{thinsplIoU}=0.50} \quad (7)$$

A higher mAP@50 indicates stronger overall detection performance, reflecting how reliably the model identifies fractures with both accuracy and precision across different confidence settings.

Mean Average Precision from IoU 0.50 to 0.95 (mAP@50–95) serves as a more stringent performance indicator, computed by averaging detection results over ten IoU thresholds spanning from 0.50 to 0.95 at 0.05 intervals.

This metric imposes increasingly stringent localization requirements and penalizes bounding box predictions that are categorically correct but spatially imprecise. As a result, mAP@50–95 more accurately reflects the model's bounding box localization quality.

$$\text{mAP}_{50:95} = \frac{1}{10} \sum_{i=0}^9 \text{AP}_{@ \text{thinsplIoU}=0.50+0.05i} \quad (8)$$

In this study, mAP@50–95 is treated as a secondary metric, since the clinical priority is fracture detection (recall) rather than pixel-level localization precision.

This study used a single dataset split and a single training run per configuration (seed = 42). Because only one training run was conducted per condition, the variability of performance metrics attributable to random initialization has not been quantified, and no formal statistical test (such as McNemar's test or bootstrap analysis) was performed to verify that the observed improvements are statistically significant rather than an artifact of a favorable random seed. Readers are advised to interpret the reported performance differences as descriptive comparisons rather than statistically validated conclusions. Multi-seed training and hypothesis testing remain necessary to establish stronger statistical reliability and are identified as a priority for future work.

III. RESULT AND DISCUSSION

The performance of YOLOv8s and YOLOv8m under the original imbalanced dataset and after oversampling is summarized in Table VIII. Given that all configurations underwent training under uniform experimental conditions, any subsequent performance discrepancies can be attributed directly to the influence of dataset composition on model behavior. The quantitative results presented in Table VIII form the basis for the subsequent analyses using Precision-Recall curves, training curves, and confusion matrices.

TABLE VIII
MODEL PERFORMANCE COMPARISON: IMBALANCED CONDITION VS. OVERSAMPLING

Model	Condition	Precision	Recall	F1-Score	mAP@50	mAP@50–95	FNR
YOLOv8s	Imbalanced	0.7606	0.4815	0.5896	0.5117	0.1974	0.5185
YOLOv8s	Oversampling	0.8971	0.7232	0.8008	0.8150	0.4210	0.2768
YOLOv8m	Imbalanced	0.8029	0.4691	0.5923	0.4995	0.2062	0.5309
YOLOv8m	Oversampling	0.8234	0.8124	0.8178	0.8320	0.4387	0.1876

Under the imbalanced training condition, both models produced relatively high Precision but considerably lower Recall. YOLOv8s obtained a Precision of 0.7606 and a Recall of 0.4815, while YOLOv8m reached 0.8029 and 0.4691, respectively. Such behavior indicates that fracture predictions were generally reliable when made, but a substantial proportion of true fracture cases went undetected. The behavior is consistent with the effect of class imbalance, where the predominance of non-fracture samples biases the learned decision boundary toward the majority class, reducing sensitivity to fracture instances [13].

Following oversampling, both models improved across all evaluation metrics, with Recall gains considerably exceeding

those in Precision. For YOLOv8s, Recall increased from 0.4815 to 0.7232 while Precision rose from 0.7606 to 0.8971. YOLOv8m showed a more pronounced asymmetry: Recall climbed from 0.4691 to 0.8124, whereas Precision changed modestly from 0.8029 to 0.8234. At the mAP@50 level, improvements were equally substantial, rising from 0.5117 to 0.8150 for YOLOv8s and from 0.4995 to 0.8320 for YOLOv8m.

The larger improvement in Recall compared with Precision is consistent with the objective of oversampling. Increasing the representation of fracture samples in the training subset allowed the models to learn fracture patterns more effectively and improved their sensitivity to fracture cases. Precision, by

contrast, depends on the model's capacity to distinguish fractures from visually similar normal anatomical structures, a capability that distributional rebalancing alone does not directly address. The concurrent improvements in F1-Score and mAP indicate that the benefits of oversampling outweighed the relatively small change in Precision.

This study did not include an empirical comparison between oversampling and alternative class-imbalance strategies such as Focal Loss or Class-Weighted Loss. While these alternatives differ mechanistically from oversampling (loss-reweighting adjusts gradient magnitudes during backpropagation but does not augment the input feature distribution, whereas sampling-level oversampling directly increases model exposure to minority-class examples), their relative effectiveness on this dataset and architecture was not empirically tested. Consequently, the claim that oversampling is superior to these alternatives cannot be made on the basis of the present results alone. This constitutes a boundary of interpretation for the current study, and a controlled empirical comparison using the same dataset split is recommended in future work.

Comparing the two model variants under the oversampled condition, YOLOv8m surpassed YOLOv8s across the majority of primary performance indicators, attaining superior results in mAP@50 (0.8320 compared to 0.8150), Recall (0.8124 versus 0.7232), F1-Score (0.8178 against 0.8008), and mAP@50–95 (0.4387 relative to 0.4210). Conversely, YOLOv8s demonstrated a higher Precision score (0.8971 versus 0.8234). Under the imbalanced condition, however, the two variants performed at a comparable level, suggesting that the larger capacity of YOLOv8m provides no meaningful advantage when fracture samples are scarce and insufficiently varied. Once oversampling introduced greater diversity into the training subset, the additional parameters of YOLOv8m appear to have been utilized more effectively, contributing to its higher Recall and mAP.

Despite this advantage, the performance gap between the two models remained narrow, with a mAP@50 difference of only 0.017. In terms of computational efficiency, YOLOv8s requires fewer than half the parameters (11.16M vs. 25.89M) and operates at substantially lower computational cost (28.7 vs. 79.1 GFLOPs), which represents a considerable efficiency gain at a relatively minor accuracy cost.

From a deployment perspective, this trade-off has practical implications. For settings where computational resources are limited, YOLOv8s offers a competitive alternative. Where higher detection sensitivity is the priority, such as emergency department triage, the Recall advantage of YOLOv8m may justify the additional computational requirements. In either case, model selection for clinical deployment should ultimately be evaluated against institution-specific data and operational constraints.

To further assess suitability for clinical deployment, inference efficiency was evaluated on an NVIDIA GeForce RTX 3050 Laptop GPU at an input resolution of 640×640 pixels. Inference speed was recorded in milliseconds (ms) per

image and converted to Frames Per Second (FPS), while peak GPU memory (VRAM) allocation was monitored throughout the inference phase. The benchmarking results demonstrated a clear trade-off between architectural complexity and computational efficiency. YOLOv8s exhibited a highly efficient average inference time of 10.68 ms per image (93.61 FPS), with a peak VRAM consumption of only 73.97 MB. Conversely, YOLOv8m required an average inference time of 25.48 ms per image (39.25 FPS) and consumed 194.97 MB of peak VRAM approximately 2.6 times higher than YOLOv8s. Crucially, both models maintain inference speeds well above real-time requirements (>30 FPS), indicating that either variant can be integrated into clinical Picture Archiving and Communication Systems (PACS) without introducing perceptible delays in radiographic interpretation.

The performance gains observed in this study also compare favorably with prior work on the same dataset. Meza et al. [11] reported an mAP@50 of 0.713 using a modified YOLOv8 model on the FracAtlas hand dataset without explicit class imbalance handling, whereas the proposed models achieved mAP@50 values of 0.8150 and 0.8320 for YOLOv8s and YOLOv8m, respectively. Although the experimental protocols were not identical, the shared dataset makes this the most relevant available comparison. These gains are likely related to the anatomy-specific filtering, leakage-aware oversampling, and conservative augmentation applied during dataset preparation.

The oversampling strategy applied in this study combined two distinct mechanisms simultaneously: class ratio rebalancing (shifting the fracture-to-non-fracture ratio from 1:2.5 to approximately 1:1) and visual variability expansion via light augmentation applied to duplicated fracture images. Because dedicated ablation experiments (such as comparing oversampling without augmentation versus oversampling with augmentation, or testing intermediate sampling ratios such as 1:2 versus 1:1) were not conducted, the individual contribution of each component could not be empirically isolated. Consequently, it remains unquantified whether performance gains stem primarily from class ratio adjustment or from augmentation-induced feature diversity. This constitutes a boundary of interpretation for the current study, and systematic ablation experiments separating these components are identified as an essential direction for future research.

A consistent disparity between mAP@50 and mAP@50–95 was evident throughout all experimental settings. For YOLOv8m with oversampling, mAP@50 reached 0.8320 while mAP@50–95 stood at only 0.4387. Such a gap reflects the stricter localization requirements embedded in mAP@50–95, which aggregates detection outcomes over multiple IoU thresholds well above 0.50 and penalizes bounding boxes that are categorically correct but spatially imprecise. Hand fractures frequently present as small, irregular, or poorly demarcated regions, making precise localization inherently challenging even when the fracture itself is correctly identified. The effect of oversampling on the overall detection

trade-off can be observed more directly from the Precision-Recall curves presented in Figures 4–7.

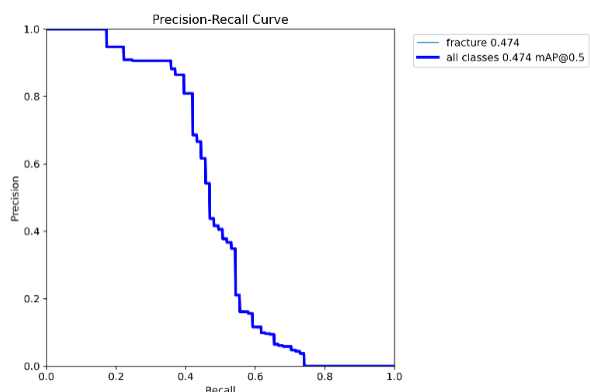


Figure 4. Precision-Recall curve for YOLOv8s under the imbalanced condition.

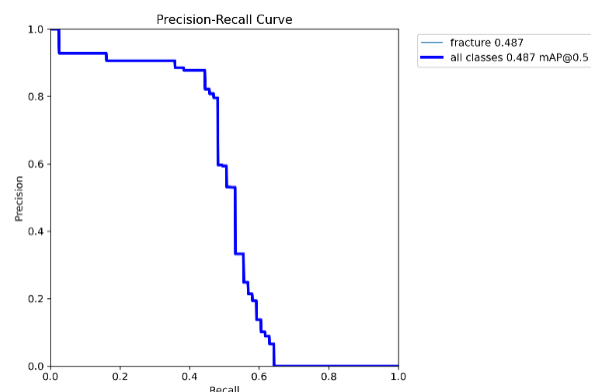


Figure 5. Precision-Recall curve for YOLOv8m under the imbalanced condition.

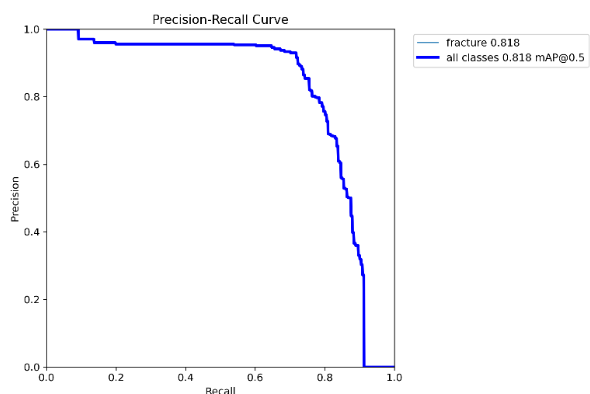


Figure 6. Precision-Recall curve for YOLOv8s under the oversampling condition.

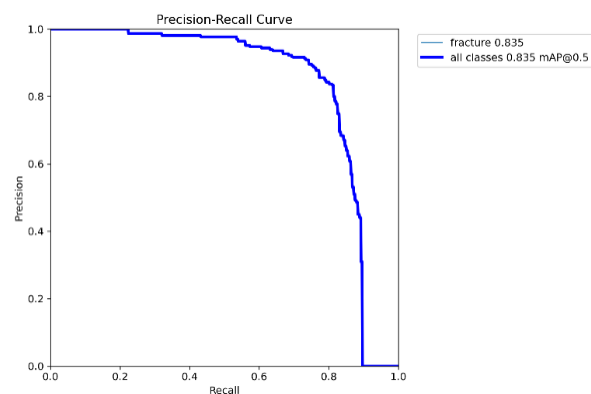


Figure 7. Precision-Recall curve for YOLOv8m under the oversampling condition.

Models trained on the imbalanced dataset exhibited curves that contracted sharply as Recall increased, reflecting the difficulty of sustaining Precision when the model was pressed to detect a larger proportion of fracture cases. After oversampling, both architectures sustained a considerably expanded area beneath the Precision-Recall curve throughout

the entire Recall spectrum, confirming that the rebalanced training distribution improved not only peak performance but also the robustness of the overall detection trade-off.

The training curves shown in Figures 8–11 illustrate the optimization process for each experimental configuration.

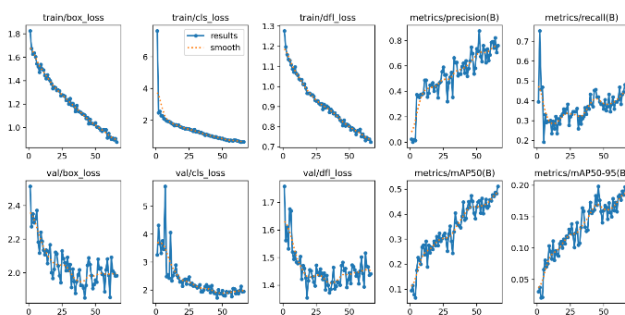


Figure 8. Training curves for YOLOv8s under the imbalanced condition.

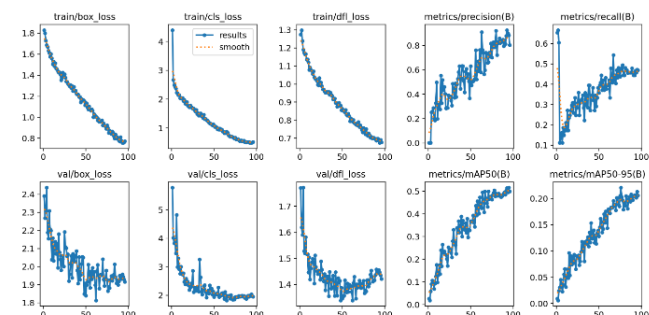


Figure 9. Training curves for YOLOv8m under the imbalanced condition.

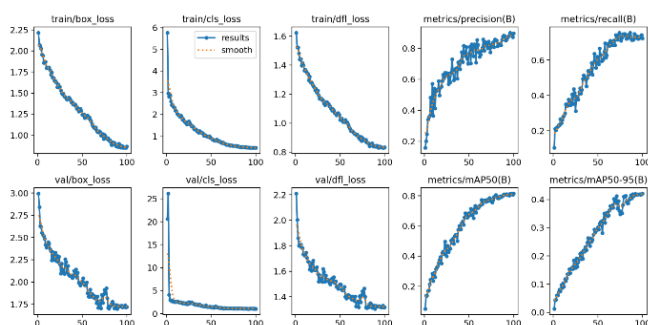


Figure 10. Training curves for YOLOv8s under the oversampling condition.

Across all four configurations, training and validation losses decreased steadily without noticeable divergence, while Precision, Recall, mAP@50, and mAP@50–95 showed gradual improvement before leveling off in the later epochs. The convergence pattern was consistent across both model variants and both training conditions, indicating stable

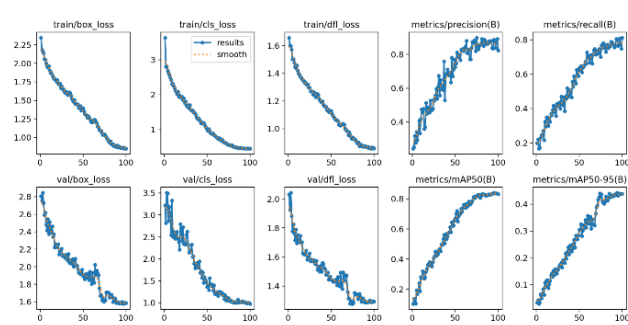


Figure 11. Training curves for YOLOv8m under the oversampling condition.

optimization throughout. Evidence of substantial overfitting was not observed, as the margin between training and validation metrics remained within an acceptable range.

Beyond the quantitative metrics, the confusion matrices afford a more detailed view of the classification mistakes made by both architectures under each experimental setting.

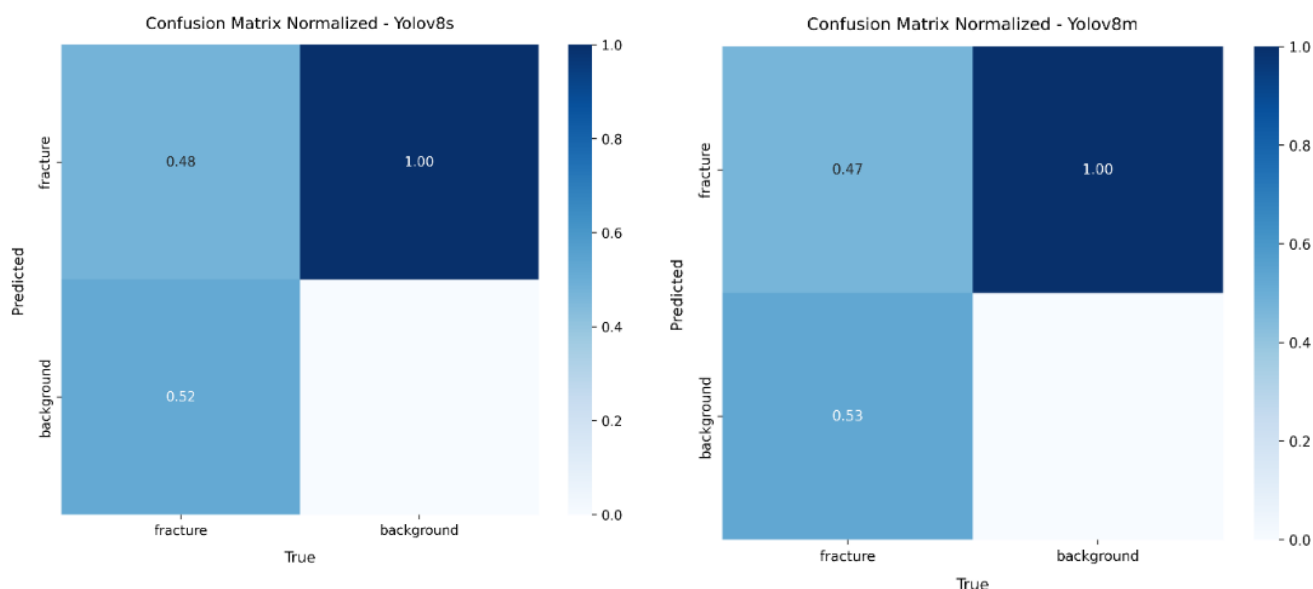


Figure 12. Normalized confusion matrices for YOLOv8s and YOLOv8m under the imbalanced condition.

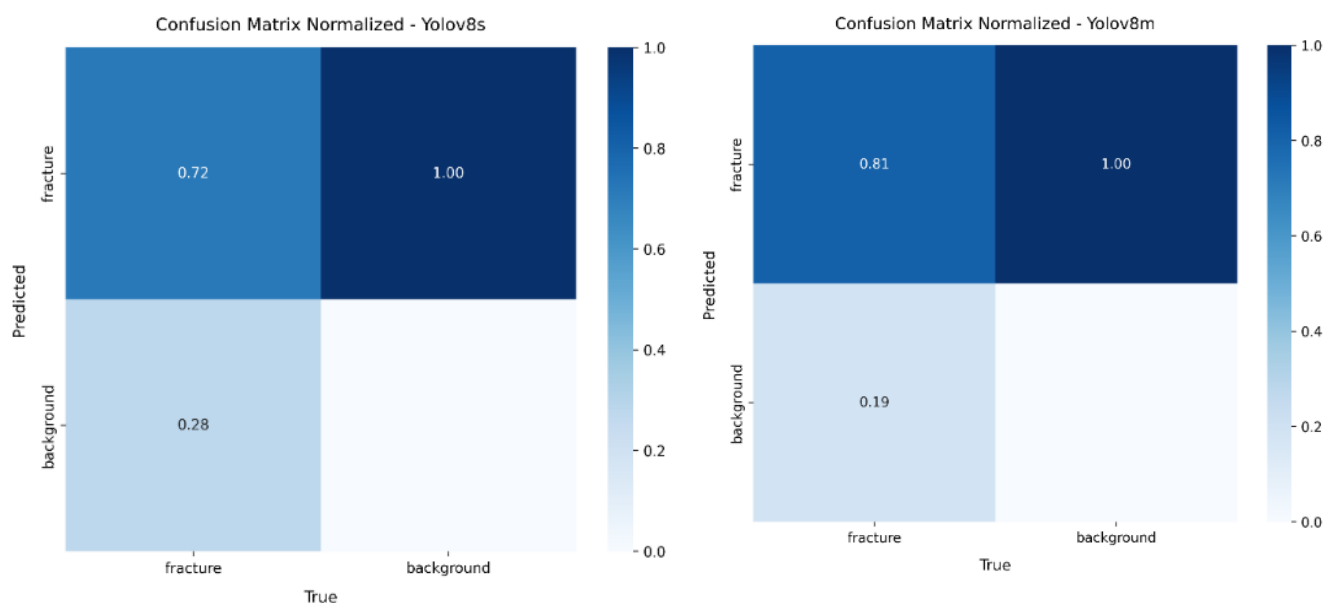


Figure 13. Normalized confusion matrices for YOLOv8s and YOLOv8m under the oversampling condition.

Under the imbalanced condition, both models generated a high proportion of false negative predictions, consistent with the Recall values of approximately 0.47 to 0.48. After oversampling, the false negative rate decreased from approximately 0.52 to 0.28 for YOLOv8s and from 0.53 to 0.19 for YOLOv8m, closely mirroring the Recall improvements in Table VIII. A qualitative examination of the remaining errors revealed that fractures occupying small annotated regions, including hairline fractures and thin cortical fissures, were most frequently missed, likely because fine spatial details become less distinguishable during feature extraction within the detection network. Missed detections were also observed when overlapping anatomical structures in lateral or oblique projections obscured the fracture boundary. False positive predictions, by contrast, were most commonly associated with joint spaces and normal bone line patterns that carry visual characteristics resembling fracture lines. These observations are qualitative and were not accompanied by a systematic quantification of error frequency by type.

The reduction in false negatives carries direct clinical relevance. In medical imaging, false negatives are generally considered more consequential than false positives, as an undetected fracture may lead to delayed treatment and poorer clinical outcomes [23]. A false positive typically prompts additional clinical evaluation, adding workload without directly compromising patient safety [24]. Against this background, the Recall improvements achieved in this study, from 46.91% to 81.24% for YOLOv8m and from 48.15% to 72.32% for YOLOv8s, represent a meaningful gain in clinical utility. A computer-aided detection system built on these models, particularly the oversampled configurations, could serve effectively as a first-line screening tool to help clinicians prioritize cases warranting further assessment, without replacing the final diagnostic judgment of a qualified

radiologist or physician. Such a system is particularly relevant in settings where specialist access is limited, where automated pre-screening may help reduce diagnostic delays.

Since the primary objective in clinical screening is to minimize false negative diagnoses without overwhelming radiologists with false alarms, we further evaluated model performance using image-level binary classification metrics. A radiographic image was classified as positive (fracture) if the model predicted at least one bounding box with a confidence score equal to or greater than the 0.25 threshold; otherwise, it was classified as negative (normal). Under this image-level evaluation, both YOLOv8 variants demonstrated exceptional discriminatory capacity. YOLOv8s achieved an image-level Specificity of 0.9686 and a Receiver Operating Characteristic Area Under the Curve (ROC-AUC) of 0.9686. Similarly, YOLOv8m achieved a Specificity of 0.9748 and an ROC-AUC of 0.9674. Both models recorded an identically low False Negative Rate (FNR) of 0.0606 at the image level, meaning that only approximately 6% of patients with actual fractures would be missed during initial automated screening. The high Specificity (>0.96) indicates that the models are highly reliable at confirming normal cases, successfully identifying non-fracture radiographs with very few false alarms.

To further validate the clinical credibility of the detection models and ensure that predictions are driven by genuine pathological features rather than spurious background artifacts, Explainable AI (XAI) analysis was conducted using Gradient-weighted Class Activation Mapping (Grad-CAM). In this study, Grad-CAM hooks were applied to the C2f modules within the Path Aggregation Network (PAN) neck (specifically layers 15, 18, and 21 of the YOLOv8 architecture). These layers were selected because they aggregate multi-scale contextual features immediately before

the final detection head, providing a highly representative spatial activation map.

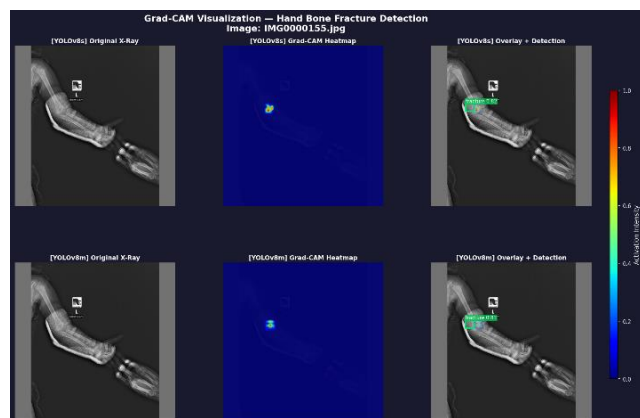


Figure 14. Grad-CAM comparison between YOLOv8s and YOLOv8m on representative hand fracture test images.

In successful detection cases (True Positives), as illustrated in Figure 14, the Grad-CAM heatmaps demonstrate that the models accurately focus attention on specific cortical discontinuities and irregular bone alignments indicative of a fracture. The activation is strictly confined to the anatomical region of interest, effectively ignoring the dark background and irrelevant soft tissue, confirming that the models have learned clinically valid representations. Conversely, in certain subtle or hairline fracture cases where neither model predicted a bounding box above the confidence threshold, the corresponding Grad-CAM activation map remains blank. This absence of strong focal activation indicates that the model's feature extraction at the PAN level did not accumulate sufficient confidence for the subtle fracture pattern. Visualizing both successful and failed detections not only enhances the transparency of the deep learning "black box," but also highlights the clinical value of XAI in building trust in computer-aided diagnostic systems before deployment in emergency triage settings.

Although the results are encouraging, certain limitations remain. Both models continue to miss a portion of subtle fracture cases, particularly non-displaced and hairline fractures characterized by limited visual contrast and small spatial extent. Addressing these patterns may require more targeted approaches, such as enhanced small-object feature representation or multi-scale detection strategies, and is left for future work. Despite these limitations, the findings demonstrate that addressing class imbalance through anatomy-specific dataset preparation and oversampling can substantially improve fracture detection performance without modifying the underlying YOLOv8 architecture.

IV. CONCLUSION

This study evaluated the effect of leakage-aware training-only oversampling on hand fracture detection using YOLOv8s and YOLOv8m, applied to a hand-only subset of the FracAtlas dataset under a 1:2.5 class imbalance ratio. By

strictly confining oversampling and augmentation to the training partition while preserving the original imbalanced distribution in the validation and test sets, this work provides a methodologically controlled performance reference for fracture detection under realistic evaluation conditions.

Oversampling substantially improved detection performance for both architectures. YOLOv8m achieved the highest Recall (0.812) and mAP@50 (0.832), making it more suitable for sensitivity-critical applications such as emergency triage. YOLOv8s achieved comparable mAP@50 (0.815) at considerably lower computational cost, making it better suited to resource-constrained deployment. At the image level, both models demonstrated high Specificity (above 0.96) and a low False Negative Rate (0.061), confirming their potential as first-line screening tools. These results indicate that leakage-aware oversampling is a practical and architecture-agnostic strategy for addressing class imbalance in medical object detection, without requiring modifications to the loss function or model architecture.

The principal contribution of this work is a leakage-aware evaluation framework and an empirical model-scale selection guideline for hand fracture detection under class imbalance. The findings are subject to several interpretive boundaries: all metrics derive from a single training run, no formal statistical significance testing was conducted, alternative imbalance-handling strategies such as Focal Loss were not empirically benchmarked, and no ablation experiments were performed to isolate individual oversampling components. Generalization to broader clinical populations also requires external multi-center validation. Future work should prioritize multi-seed experiments with variance reporting, ablation studies separating ratio rebalancing from augmentation diversity, and empirical comparison against loss-reweighting baselines under identical experimental conditions.

REFERENCES

- [1] V. Hansen, J. Jensen, M. W. Kusk, O. Gerke, H. B. Tromborg, and S. Lysdahlgaard, "Deep learning performance compared to healthcare experts in detecting wrist fractures from radiographs: A systematic review and meta-analysis," *Eur. J. Radiol.*, vol. 174, p. 111399, May 2024, doi: 10.1016/j.ejrad.2024.111399.
- [2] M. Dashti *et al.*, "Evaluation of deep learning and convolutional neural network algorithms for mandibular fracture detection using radiographic images: A systematic review and meta-analysis," *Imaging Sci. Dent.*, vol. 54, no. 3, pp. 232–239, Sep. 2024, doi: 10.5624/isd.20240038.
- [3] H. Bouziri, Y. Roquelaure, A. Descatha, W. Dab, and K. Jean, "Temporal and spatial distribution of musculoskeletal disorders from 1990 to 2019: a systematic analysis of the global burden of disease," *BMJ Public Health*, vol. 1, no. 1, Dec. 2023, doi: 10.1136/bmjph-2023-000353.
- [4] Z.-L. Han *et al.*, "A systematic review and meta-analysis of artificial intelligence software for tuberculosis diagnosis using chest X-ray imaging," *J. Thorac. Dis.*, vol. 17, no. 5, pp. 3223–3237, May 2025, doi: 10.21037/jtd-2025-604.
- [5] L. Mattijssen-Horstink, J. J. Langeraar, G. J. Mauritz, W. van der Stappen, M. Bagelaar, and E. C. T. H. Tan, "Radiologic discrepancies in diagnosis of fractures in a Dutch teaching emergency department: a retrospective analysis," *Scand. J. Trauma Resusc. Emerg. Med.*, vol. 28, no. 1, p. 38, May 2020, doi: 10.1186/s13049-020-00727-8.

- [6] V. Ponkilainen, I. Kuitunen, R. Liukkonen, M. Vaajala, A. Reito, and M. Uimonen, "The incidence of musculoskeletal injuries: a systematic review and meta-analysis," *Bone Jt. Res.*, vol. 11, no. 11, pp. 814–825, Nov. 2022, doi: 10.1302/2046-3758.1111.BJR-2022-0181.R1.
- [7] A. Azizi, M. Azizi, and M. Nasri, "Artificial Intelligence Techniques in Medical Imaging: A Systematic Review," *Int. J. Online Biomed. Eng. IJOE*, vol. 19, no. 17, pp. 66–97, Dec. 2023, doi: 10.3991/ijoe.v19i17.42431.
- [8] T. Diwan, G. Anirudh, and J. V. Tembhurne, "Object detection using YOLO: challenges, architectural successors, datasets and applications," *Multimed. Tools Appl.*, vol. 82, no. 6, pp. 9243–9275, 2023, doi: 10.1007/s11042-022-13644-y.
- [9] J. Terven and D. Cordova-Esparza, "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Mach. Learn. Knowl. Extr.*, vol. 5, no. 4, pp. 1680–1716, Nov. 2023, doi: 10.3390/make5040083.
- [10] R.-Y. Ju and W. Cai, "Fracture detection in pediatric wrist trauma X-ray images using YOLOv8 algorithm," *Sci. Rep.*, vol. 13, no. 1, p. 20077, Nov. 2023, doi: 10.1038/s41598-023-47460-7.
- [11] G. Meza, D. Ganta, and S. G. Torres, "Deep Learning Approach for Arm Fracture Detection Based on an Improved YOLOv8 Algorithm," *Algorithms*, vol. 17, no. 11, Oct. 2024, doi: 10.3390/a17110471.
- [12] T. Zhou, K. Chen, H. Lu, S. Yu, W. Chai, and Q. Liu, "PSCA-YOLO: An enhanced YOLO with position-semantics coupled attention for mandibular fracture detection," *iScience*, vol. 29, no. 4, p. 115404, Apr. 2026, doi: 10.1016/j.isci.2026.115404.
- [13] I. Araf, A. Idri, and I. Chairi, "Cost-sensitive learning for imbalanced medical data: a review," *Artif. Intell. Rev.*, vol. 57, no. 4, p. 80, Mar. 2024, doi: 10.1007/s10462-023-10652-8.
- [14] J. F. Cohen and M. D. F. McInnes, "Deep Learning Algorithms to Detect Fractures: Systematic Review Shows Promising Results but Many Limitations," *Radiology*, vol. 304, no. 1, pp. 63–64, Jul. 2022, doi: 10.1148/radiol.212966.
- [15] R. Aggarwal *et al.*, "Diagnostic accuracy of deep learning in medical imaging: a systematic review and meta-analysis," *Npj Digit. Med.*, vol. 4, no. 1, p. 65, Apr. 2021, doi: 10.1038/s41746-021-00438-z.
- [16] I. Abedeen, M. A. Rahman, F. Z. Protyasha, T. Ahmed, T. M. Chowdhury, and S. Shatabda, "FracAtlas: A Dataset for Fracture Classification, Localization and Segmentation of Musculoskeletal Radiographs," *Sci. Data*, vol. 10, no. 1, p. 521, Aug. 2023, doi: 10.1038/s41597-023-02432-4.
- [17] R. Sapkota and M. Karkee, "Ultralytics YOLO Evolution: An Overview of YOLO26, YOLO11, YOLOv8 and YOLOv5 Object Detectors for Computer Vision and Pattern Recognition," Mar. 16, 2026, *arXiv: arXiv:2510.09653*. doi: 10.48550/arXiv.2510.09653.
- [18] S. M. Alkentar, B. Alsahwa, A. Assalem, and D. Karakolla, "Practical comparison of the accuracy and speed of YOLO, SSD and Faster RCNN for drone detection," *J. Eng.*, vol. 27, no. 8, p. 19, Aug. 2021, doi: 10.31026/j.eng.2021.08.02.
- [19] Y. Shao *et al.*, "BDS-Adam optimizer integrating adaptive variance rectification with semi-adaptive gradient smoothing," *Sci. Rep.*, vol. 15, no. 1, p. 36906, Oct. 2025, doi: 10.1038/s41598-025-20788-y.
- [20] A. Espis, C. Marzi, and S. Diciotti, "Comparative analysis of supervised and self-supervised learning with small and imbalanced medical imaging datasets," *Sci. Rep.*, vol. 15, no. 1, p. 32345, Sep. 2025, doi: 10.1038/s41598-025-99000-0.
- [21] C. Zhang *et al.*, "The WuC-Adam algorithm based on joint improvement of Warmup and cosine annealing algorithms," *Math. Biosci. Eng.*, vol. 21, no. 1, pp. 1270–1285, 2024, doi: 10.3934/mbe.2024054.
- [22] T. Kumar, R. Brennan, A. Mileo, and M. Bendeche, "Image Data Augmentation Approaches: A Comprehensive Survey and Future Directions," *IEEE Access*, vol. 12, pp. 187536–187571, 2024, doi: 10.1109/ACCESS.2024.3470122.
- [23] B. Kocak *et al.*, "Evaluation metrics in medical imaging AI: fundamentals, pitfalls, misapplications, and recommendations," *Eur. J. Radiol. Artif. Intell.*, vol. 3, p. 100030, Sep. 2025, doi: 10.1016/j.ejrai.2025.100030.
- [24] M. H. Bernstein *et al.*, "Can incorrect artificial intelligence (AI) results impact radiologists, and if so, what can we do about it? A multi-reader pilot study of lung cancer detection with chest radiography," *Eur. Radiol.*, vol. 33, no. 11, pp. 8263–8269, 2023, doi: 10.1007/s00330-023-09747-1.