# Albument-NAS: An Enhanced Bone Fracture Detection Model

Evandiaz Fedora[1], Alexander Agung Santoso Gunawan[2]
Computer Science Department-Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia[1]
Computer Science Department-School of Computer Science, Bina Nusantara University, Jakarta, Indonesia[2]

***Abstract*—Diagnosing fracture locations accurately is challenging, as it heavily depends on the radiologist's expertise; however, image quality, especially with minor fractures, can limit precision, highlighting the need for automated methods. The accuracy of diagnosing fracture locations often relies on radiologists' expertise; however, image quality, particularly with smaller fractures, can limit precision, underscoring the need for automated methods. Although a large volume of data is available for observation, many datasets lack annotated labels, and manually labeling this data would be highly time-consuming. This research introduces Albument-NAS, a technique that combines the One Shot Detector (OSD) model with the Albumentation image augmentation approach to enhance both speed and accuracy in detecting fracture locations. Albument-NAS achieved a mAP@50 of 83.5%, precision of 87%, and recall of 65.7%, significantly outperforming the previous state-of-the-art model, which had a mAP@50 of 63.8%, when tested on the GRAZPEDWRI dataset—a collection of pediatric wrist injury X-rays. These results establish a new benchmark in fracture detection, illustrating the advantages of combining augmentation techniques with advanced detection models to overcome challenges in medical image analysis.**

***Keywords*—Albumentation; augmentation; bone fracture; deep learning; object detection; YOLO-NAS**

## I. Introduction

Medical images are essential for modern healthcare and diagnostics. However, their limited resolution can make it challenging for healthcare providers to fully evaluate a patient's condition [1]. Medical image analysis also faces obstacles like insufficient data and the complexity of interpreting outcomes [2]. Recently, deep learning has shown promising potential to automate and enhance medical image analysis, boosting accuracy and efficiency in diagnosing fractures. Yet, limited data and challenges in result interpretation continue to hinder its application [3]. The lack of large, high-quality medical datasets is a major barrier in training effective deep learning models. Furthermore, although deep learning can deliver accurate predictions, healthcare professionals often find it challenging to interpret the outcomes produced by these models [4].

Detecting bone fractures is essential for timely medical intervention and effective rehabilitation [5]. Traditional methods depend on manual examination of images, a process that is both time-consuming and susceptible to human error [6]. Surgeons typically require comprehensive patient histories and detailed X-ray analysis, which call for specialized expertise and training [7].

In terms of data availability, hospital datasets, such as X-rays, vary in quality and completeness of information [8]. Sometimes, image quality can impact the speed of disease analysis for patients. Additionally, not all datasets have labels or annotations, making automated detection challenging [9]. Experts need to manually label these datasets, which is very time-consuming. Furthermore, when using Deep Learning model, a large amount of data is required, which is difficult to achieve in a short time [10] [11].

A useful strategy for overcoming dataset limitations is data augmentation. This technique creates synthetic data that introduces greater variation than the original dataset, expanding the data pool with a wider range of examples [12]. This increased diversity helps the model learn more effectively and generalize across various dataset conditions, including both high and low-quality data [13]. By simulating multiple potential scenarios, augmentation strengthens the model's ability to handle real-world data variability, while also reducing the reliance on large, high-quality labeled datasets. As a result, this process accelerates training and improves overall model performance [14] [15]. One of many augmentation method worth considering is Albumentation. It strives to achieve a balance between several key factors, delivering excellent performance across a wide range of transformations while offering a concise API and a flexible, extensible design [22].

This research aims to improve the accuracy of bone fracture detection using 'Albument-NAS'. The approach seeks to automatically identify and locate fractures in medical images, addressing challenges posed by resolution limitations that hinder accurate assessment of patient conditions. The research highlights the use of data augmentation to expand the variety of training data, which is anticipated to enhance the model's ability to recognize different patterns in medical images, ultimately leading to more accurate detection of bone fracture locations. Albument-NAS not only improves fracture detection accuracy but also has the potential to assist radiologists in real-time diagnostics, reducing diagnostic time and improving accessibility in under-resourced medical settings.

## II. Literature Review

### A. Research about Bone Fracture

Through data augmentation techniques, the recent study by Ju and Cai [7] improves YOLOv8's performance on the GRAZPEDWRI-DX dataset, which is a collection of pediatric wrist injury X-rays. With a state-of-the-art mean average accuracy (mAP@50) of 0.638, their suggested model

outperformed the original YOLOv8 model and an enhanced YOLOv7 model, which had respective scores of 0.634 and 0.636. This work contributes to the development of object detection models by demonstrating the importance of data augmentation in boosting YOLOv8's efficacy, especially for pediatric X-ray analysis.

Ahmed and Hawezi's research [16] addresses inaccuracies in bone fracture diagnoses due to blurry images from conventional X-ray scanners, which increase misdiagnosis risks. Their study aims to develop a machine learning-based system to assist surgeons in detecting fractures more accurately. Among several algorithms tested, the Support Vector Machine (SVM) model achieved the highest accuracy at 92.8%, followed by the Random Forest model with 85.7%, highlighting the potential of machine learning to improve diagnostic precision.

To meet the demand for quick, precise fracture diagnosis utilizing X-ray and CT images, Hareendranathan *et al*. [17] employed classification approaches to differentiate bone fractures from normal bone. Large data quantities and visual blurriness in MRI and CT scans make manual diagnosis difficult. The goal of this project was to develop an image-processing system that can classify fractures with speed and accuracy. On a dataset of 100 training and testing photos, the system demonstrated a high accuracy rate of 99.5%, indicating potential for effective fracture diagnosis.

### *B. Research about Data Augmentation*

Su et al. [18] use Generative Adversarial Networks (GANs) for data augmentation in order to handle class imbalance and data scarcity. Particularly in minority classes, traditional GANs suffer from mode collapse or unequal distributions. They suggest Self-Transfer GAN (STGAN), a two-stage technique for producing varied 256×256 skin lesion pictures, as a solution to this problem. For a high-quality synthesis, STGAN first learns generic information from all classes and then blends it with information unique to each subject. When tested on the HAM10000 dataset, STGAN outperformed StyleGAN2 by up to 33% in terms of FID, Inception Score, Precision, and Recall. The STGAN framework shown efficacy for balanced classification with 98.23% accuracy, 88.85% sensitivity, 90.23% precision, 89.48% F1-score, and 98.34% specificity.

Cubuk et al. [19] introduce AutoAugment, an automated method for finding optimal data augmentation policies, enhancing image classifier accuracy. Unlike manual approaches, AutoAugment uses an automated search, achieving state-of-the-art accuracy on datasets like CIFAR-10, CIFAR-100, SVHN, and ImageNet. Notably, it achieved 83.5% Top 1 accuracy on ImageNet, surpassing the previous 83.1%, and reduced CIFAR-10's error rate to 1.5%, a 0.6% improvement.

The research by Elbattah *et al*. [8] presents an interesting study in the field of data augmentation. The data they aim to augment is a representation of eye-tracking known as scanpath. This research is intriguing because various aspects can be extracted from human eye movements, such as emotion recognition. The data augmentation method used is Variational Autoencoder (VAE) [20]. The results show that the accuracy of the model without augmentation increased from 67% to 70% with the use of augmentation.

## III. Problem Statement

The challenges in bone fracture detection include issues with image quality and the expertise of doctors in analyzing X-ray scan results. Sometimes, the fractures are very small and almost imperceptible. This difficulty can certainly consume time when analyzing to determine the exact location of the fracture. There are many publicly available datasets that can be used for machine learning training, but not all of them are good quality. Some have poor image quality, some lack annotations for fracture locations, and others have excessive labels that provide unnecessary information for identifying fracture locations. Therefore, proper processing of the dataset is needed so that it can be easily learned by the model, allowing the model to capture patterns to predict the location of bone fractures.

## IV. Methodology

Bone fracture detection in medical imaging plays a crucial role in various applications, particularly in supporting faster and more accurate diagnoses and treatments. This research aims to enhance bone fracture detection performance by applying data augmentation techniques within a one-shot detector model, namely as Albument-NAS method. In this chapter, the methodology employed in this research is described in detail. The process begins with the collection of medical image samples containing fractures, followed by data augmentation to increase data variability, and concludes with the implementation of the detection model architecture. Sample images and the overall procedure will be presented and thoroughly discussed.

### *A. Dataset*

For this study, two datasets will be used: the COCO Bone Fracture Dataset and the GRAZPEDWRI dataset. The COCO dataset will be used for training and validation, while the GRAZPEDWRI dataset, similar to the dataset used by previous researchers Ju and Cai [3], will be used as test data to compare with the model applied in this research. Both datasets contain the same types of data—images and annotation files. However, the COCO dataset is in COCO format, while the GRAZPEDWRI dataset is in Pascal VOC format. For the COCO dataset, all images have a uniform size of 416x416, while for the GRAZPEDWRI dataset, the images vary in size. 908 images from COCO dataset will be augmented to varying dataset, and for testing will be using 300 images from GRAZPEDWRI dataset. For the image quality of each dataset, both datasets exhibit similar levels of variation in terms of brightness. Some images have high brightness, while others are relatively dim. The COCO Bone dataset (Fig. 1) has additional variation in terms of image color, with some images featuring a blue background, while the majority are grayscale. In contrast, the GRAZPEDWRI dataset consists entirely of grayscale images. Regarding data distribution, all the provided images depict fractures; there are no images of normal bones, and the specific types of fractures or patient details are not explained. This indicates that both datasets are specifically designed for fracture detection research.

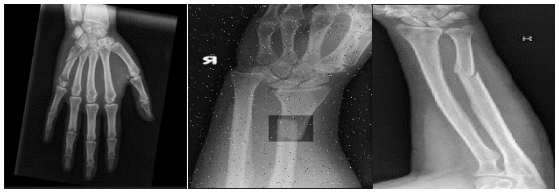

Fig. 1. COCO bone fracture dataset sample.

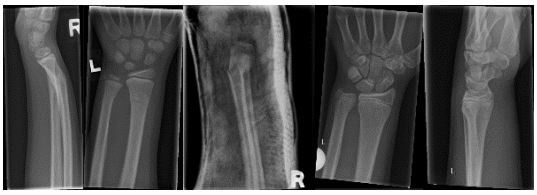

Fig. 2. GRAZPEDWRI dataset sample.

### B. Preprocessing

Preprocessing is necessary to ensure that the images are prepared for training with the chosen deep learning model. Since the model to be used is YOLO, it is beneficial to resize all images to a uniform size. For this purpose, the target size is set to 640x640 to provide better clarity of the fracture areas. The image size of 640x640 was selected as it provides a good balance between computational efficiency and model accuracy. Larger image sizes can increase accuracy but also require more computational resources, while smaller sizes can reduce processing time but may miss finer details. The chosen size ensures that the model can detect fractures effectively while maintaining reasonable training and inference times.

The next step is to apply bounding box labels using the annotation files provided for the training and validation datasets. YOLO requires images to be pre-labeled for object detection tasks. The test data will also be labeled, but only for comparison and validation purposes to check whether the model's predicted bounding boxes are accurate. However, it appears that in the GRAZPEDWRI dataset (Fig. 2), there are some labels that are not necessary for locating fractures. Therefore, label extraction will be performed to retain only the fracture location bounding boxes, while other labels will be disregarded.

### C. Albumentation

The study by Ju & Cai [7] provided brightness and exposure enhancement augmentation for each image. This research will do the same, but with the addition of several other augmentations. Augmentation will be carried out using the Albumentations method provided by Roboflow, allowing the author to directly utilize the augmented dataset. As a result, Roboflow produced 2,360 augmented images to be used as the training and validation dataset. What sets Albumentations apart from typical augmentation techniques is its ability to apply augmentations probabilistically. This means that augmentations are applied randomly to images based on predefined constraints and probability values.

Brightness in an image refers to the overall light intensity or luminance, which affects how light or dark an image appears. Increasing brightness makes an image appear lighter, while decreasing it makes it darker. The brightness of each pixel can be adjusted by adding a constant value to the pixel's intensity, represented by Formula 1:

$$I' = I + \beta \tag{1}$$

where $I$ is the original intensity of the pixel, $I'$ is the adjusted intensity, and $\beta$ is a constant that controls the level of brightness.

Exposure in an image relates to the amount of light captured by the camera sensor, influencing the brightness and detail visible in the image, especially in the highlights and shadows. Proper exposure ensures that an image retains detail without being overly bright (overexposed) or too dark (underexposed). Exposure adjustments can be achieved by scaling pixel intensity, often represented by Formula 2:

$$I' = I \times \alpha \tag{2}$$

where $I$ is the original pixel intensity, $I'$ is the adjusted intensity, and $\alpha$ is the multiplier that controls the exposure level. Values of $\alpha > 1$ will increase exposure, brightening the image, while $0 < \alpha < 1$ reduces exposure, darkening it.

The final augmentation performed by the author is the addition of noise. In this case, the applied noise is salt and pepper noise. Salt and pepper noise is a type of image noise characterized by random occurrences of white (salt) and black (pepper) pixels throughout the image, creating isolated bright and dark spots that disrupt the image's smooth appearance. The effect of salt and pepper noise is typically applied by randomly setting a percentage of pixels to the minimum intensity (0, representing black) or maximum intensity (255, representing white) in an 8-bit grayscale image. Mathematically, it can be represented as Formula 3:

$$I(x,y) = \begin{cases} 0 \; with \; probability \; p_s \\ 255 \; with \; probability \; p_p \\ I(x,y) \; otherwise \end{cases} \tag{3}$$

pixel intensity, $p_s$ is the probability of salt noise, and $p_p$ is the probability of pepper noise. The addition of this noise enriches the dataset quality and trains the model to learn from various unpredictable dataset conditions. Fig. 3 shows the sample of augmented images.

Fig. 3. Augmented images sample.

Fig. 4 shows the overall architecture and functionality of the proposed work, which includes augmentation with Roboflow Albumentation and object detection using YOLO-NAS creating author proposed method, Albument-NAS, while Algorithm I shows the pseudocode for the proposed method.

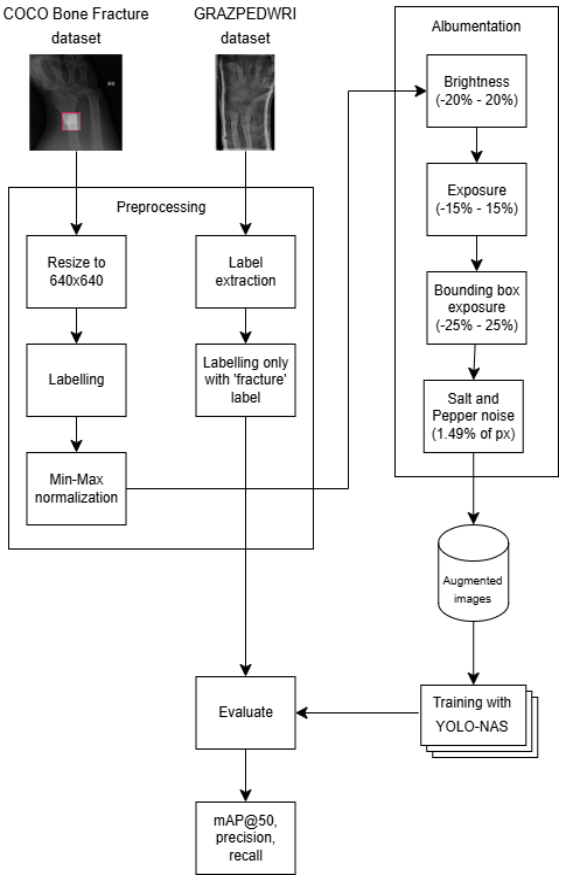


Fig. 4. Overall procedure of albument-NAS.

**Algorithm 1:** Albument-NAS method: Albumentation augmentation and YOLO-NAS detection

***Inputs***: COCO Bone Fracture dataset
***Outputs***: X-Ray images with bounding box labelled
1. Initialize image
2. Initialize Albumentation technique
3. Compute augmentation into images
4. Initialize YOLO-NAS model and hyperparameter
5. While (training not converged or early stopping not reached) do
6. Load augmented images and their bounding box label
7. For (every image in batch) do
8. For (every region in the image with a bounding box) do
9. Apply YOLO-NAS detection
10. Update
11. Adjust model parameter based on accuracy and loss
12. End
13. End
14. End
15. Evaluate mAP@50, precision, recall
16. End

### D. YOLO-NAS Model

For the object detection model, the author used YOLO-NAS. YOLO-NAS is an object detection model developed by Deci AI using neural architecture search (NAS) techniques to automatically design the network architecture. This allows the model to find the optimal settings for recognizing objects in images or videos without requiring extensive human input. The advantage of YOLO-NAS over other YOLO models is its higher accuracy, faster speed, and better efficiency, making it ideal for use on low-power hardware such as mobile devices. YOLO-NAS is based on the modified CSPNet backbone architecture and uses the YOLOv5 detection head. The combination of the efficient CSPNet backbone, the powerful YOLOv5 detection head, and advanced NAS enables YOLO-NAS (Fig. 5) to achieve exceptional accuracy and speed.

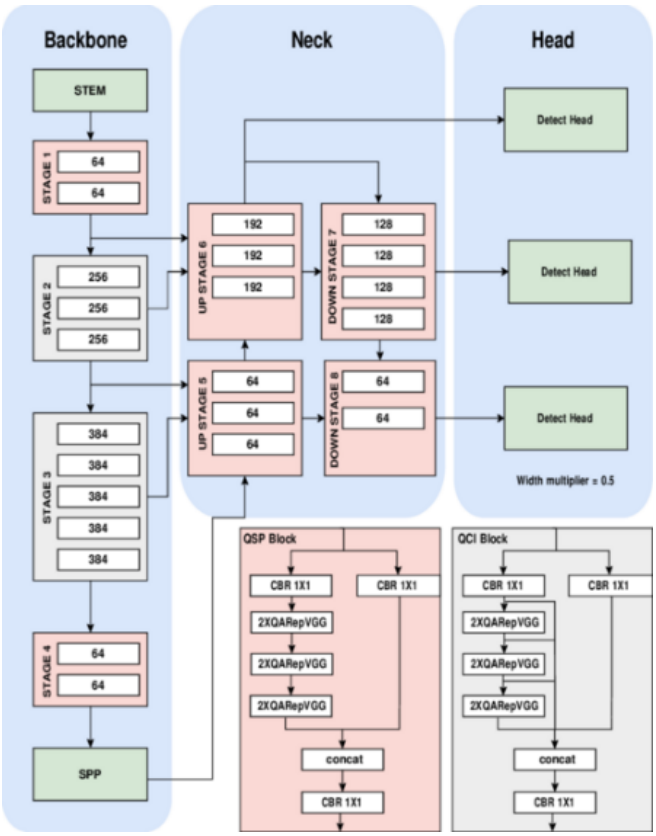


Fig. 5. YOLO-NAS architecture.

As for the backbone using CSPNet, Formula 4 shows how the backbone works:

$$F_{backbone}(I) = FeatureExtraction(I) \quad (4)$$

where $I$ is the input image, and $F_{backbone}(I)$ produces the feature used by detection head. YOLOv5 detection head is used to predict $B_{pred}$, which is the bounding box and object class for image $I$. The function can be seen in Formula 5 below.

$$B_{pred} = YOLOv5(F_{backbone}(I)) \quad (5)$$

For the optimal network architecture searching, YOLO-NAS uses NAS (Neural Architecture Search) method to optimize the model's architecture by selecting the best components such as the number of layers, the number of neurons, and connections between layers.

$$A^* = \arg max\ L(A, D) \quad (5)$$

where $A$ is the network architecture being searched, $L(A, D)$ is the loss function for the architecture $A$ on dataset $D$ and $A^*$ is the optimal architecture that maximizes performance.

YOLO-NAS also compute loss function, typically combines classification loss and bounding box regression loss. Formula 6 shows how YOLO-NAS handle loss function:

$$L_{total} = L_{cclassification} + L_{bounding\_box} + L_{objectness} \quad (6)$$

$L_{classification}$ measures the error in classifying object class, $L_{bounding\_box}$ measures the error in predicting the bounding box location, and $L_{objectness}$ measures the error in predicting the probability of the object presence in a grid.

## V. Result and Discussion

This research is implemented in Python 3.10. The environment used for this study, including the hardware and software specifications, is outlined in Table I. These specifications define the necessary resources for implementing the proposed object detection system effectively. The hardware setup provides sufficient processing power, while the software components offer the required tools and libraries to support model training and evaluation.

TABLE I. Research Environment Details

| Type | Device Type | Name |
|---|---|---|
| Hardware | Processor | 13th Gen Intel(R) Core™ i7-13620H, 2400 Mhz |
| | Graphics processor | NVIDIA GeForce RTX 4060 Laptop GPU |
| | Storage | 1 TB |
| | Memory | 16 GB |
| Software | IDE | Visual Studio Code |
| | Albumentation | Roboflow |
| | Important libraries | cpython, cython, ipython, ipykernel, jedi, numpy, onnx, pillow, pytorch (make sure it is CUDA supported), scipy, super-gradients (YOLO model), supervision, torchvision, torchaudio |

This research will observe three key metrics employed to evaluate the some of the object detection model's performance. By focusing on these metrics, a comprehensive understanding of the model's detection precision and overall effectiveness can be observed.

### A. Metrics

To assess how well the model's bounding box predictions identify fracture locations, the mAP@50 metric will be analyzed. A higher mAP value indicates that the model is more successful in accurately placing bounding boxes that match the ground truth. Additionally, precision will be calculated to evaluate how accurately the proposed model predicts the correct class, and recall will be used to determine how many fracture images are correctly identified as fractures.

TABLE II. Model Comparison on GRAZPEDWRI Dataset

| Model | Metric | | |
|---|---|---|---|
| | ***mAP*** | ***Precision*** | ***Recall*** |
| YOLOv8 (SOTA) [3] | 63.8% | - | - |
| Baseline (ResNet50) | 71.4% | 80.5% | 61.3% |
| Baseline + Albumentation | 76.2% | **88.2%** | 63.6% |
| Faster R-CNN | 73.3% | 81.4% | 62.3% |
| Faster R-CNN + Albumentation | 78.3% | 83.4% | 64.3% |
| YOLO-NAS | 74.3% | 79.2% | 59.8% |
| **Albument-NAS (proposed method)** | **83.5%** | 87% | **65.7%** |

Table II compares performance metrics across different models, focusing on mAP, precision, and recall. YOLOv8 (SOTA) achieves a mAP of 63.8%, though Precision and Recall metrics are not available. The baseline model (ResNet50) attains a mAP of 71.4%, with a precision of 80.5% and a Recall of 61.3%, demonstrating strong true positive accuracy but moderate recall. With Albumentation augmentation, the Baseline model improves to a mAP of 76.2%, while Precision rises to 88.2% and recall to 63.6%, indicating a notable increase in true positive identification. Faster R-CNN achieves a mAP of 73.3%, with a Precision of 81.4% and a Recall of 62.3%. This result highlights its strong precision and balanced performance in recall, surpassing the Baseline model in mAP. When augmented with Albumentation, Faster R-CNN improves further to a mAP of 78.3%, with a Precision of 83.4% and a Recall of 64.3%. This demonstrates that augmentation enhances both precision and recall, solidifying Faster R-CNN as a robust model for object detection tasks. YOLO-NAS without augmentation achieves a mAP of 74.3%, with a Precision of 79.2% and a Recall of 59.8%, slightly below the Baseline with augmentation. However, the proposed method, Albument-NAS, achieves the highest mAP at 83.5%, with a precision of 87% and recall of 65.7%, demonstrating that augmentation significantly enhances both accuracy and completeness of detections across all metrics. Precision and recall values for YOLOv8 were not available due to limitations in the original study's reporting. This metric should be considered when comparing the performance of YOLO-based models. The inclusion of Faster R-CNN results underscores its competitive performance, especially when paired with augmentation, although it does not surpass the Baseline with augmentation or Albument-NAS in overall accuracy.

The results demonstrate that an appropriate augmentation process can enhance object detection accuracy, specifically in identifying fracture locations. The YOLO-NAS model also proves superior due to its ability to optimize parameters during training. Consequently, the architecture of YOLO-NAS achieves greater convergence compared to YOLOv8, whose structure is already pre-defined and fixed. The ResNet50 model achieves higher precision because its architecture is optimized to focus on fine-grained feature extraction [21], which reduces

false positives. ResNet50's deep residual connections help capture subtle details and ensure that detected regions are more likely to correspond accurately to true fractures, thus improving precision. This focus makes ResNet50 particularly effective in tasks where high specificity (true positive accuracy) is crucial, even though it might not reach the same level of overall recall as YOLO-based models optimized for real-time, comprehensive detection. Additionally, Faster R-CNN achieves higher precision compared to ResNet50 due to its two-stage detection mechanism, which separates region proposal from classification. This architecture allows Faster R-CNN to focus on high-quality region proposals, reducing the likelihood of false positives during object classification.

The YOLO-NAS model without augmentation performed below the baseline augmented model due to the limited diversity of the training data. Data augmentation significantly improves model generalization by simulating various real-world conditions, thus enhancing its ability to detect fractures across different image qualities.

### B. Bone Fracture Detection

To ensure a fair comparison in bone fracture detection, the detection results will be evaluated using the GRAZPEDWRI dataset, which serves as the testing dataset. The object detection results can be seen in Fig. 6 below.

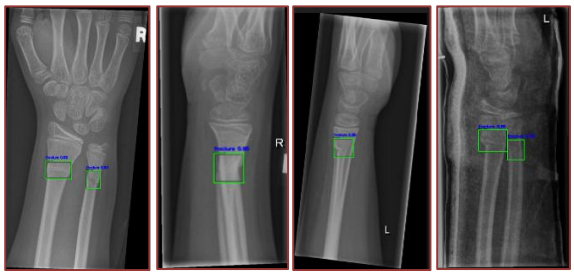

Fig. 6. Fracture detection result by proposed method, albument-NAS.

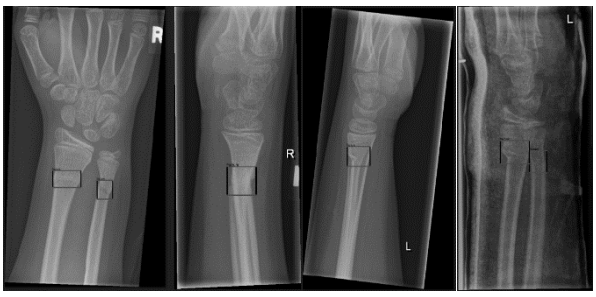

Fig. 7. Fracture detection ground truth.

Based on Fig. 6, Albument-NAS successfully detects bone fractures accurately. Fig. 7 displays the actual locations of the fractures. Despite the inconsistent quality of the X-ray images (with the fourth image in Fig. 7 appearing darker than the others), the model still performs well in making inferences. These variations are largely attributed to challenges in detecting small or subtle fractures in images with low resolution or significant noise. Nonetheless, the model demonstrates strong performance in accurately identifying larger fractures.

The Albument-NAS model has the potential to be incorporated into clinical workflows via cloud-based diagnostic systems, providing remote access for healthcare professionals in underprivileged regions. Additionally, its capability to swiftly and accurately detect fracture locations can help alleviate the workload of radiologists, especially in emergency scenarios where prompt action is essential.

### C. Grad-CAM Visualization

Grad-CAM (Gradient-weighted Class Activation Mapping) can help radiologists and healthcare providers understand how the model identifies fracture locations by generating visual heatmaps that highlight the regions of an image most influential to the model's decision. When applied to X-ray images, Grad-CAM produces an overlay that indicates which areas the model "focused on" to predict the presence of a fracture. Radiologists are more inclined to embrace AI systems when they can confirm that the model's decision-making process aligns with their clinical knowledge and judgment. By providing visual insights into how predictions are generated, Grad-CAM facilitates a collaborative dynamic between AI and healthcare providers, positioning the AI as a supportive tool rather than a replacement. This clarity not only strengthens trust but also encourages broader adoption of AI solutions in clinical settings by ensuring patient safety and adhering to medical decision-making standards. As for this research, Fig. 8 shows how Grad-CAM can clarify the model's prediction results by focusing on the fracture location.

Fig. 8. Grad-CAM visualization.

## VI. Conclusion and Future Work

The results of this study demonstrate that appropriate data augmentation, specifically with Albumentation, significantly improves object detection performance in identifying fracture locations. The Albument-NAS model achieved the highest performance in terms of mAP@50, Precision, and Recall when augmented, suggesting its advantage over other models due to its ability to dynamically optimize parameters during training. This adaptability allows YOLO-NAS to achieve greater convergence, whereas YOLOv8, with a more rigid architecture, is less flexible in parameter optimization. Meanwhile, the ResNet50 model, although not reaching the highest recall, excels in precision due to its residual architecture, which effectively captures detailed features, thereby reducing false positives. This characteristic makes ResNet50 particularly useful in applications requiring high specificity and accuracy. The reason behind the significant drop in mAP@50 for the SOTA model is that it was designed for multi-object detection, requiring accurate predictions across a large number of classes. However, this approach is less relevant to the original goal, which is to precisely determine the location of bone fractures. Although YOLO-based models like YOLOv5 and YOLOv8

perform well in general object detection tasks, they frequently encounter challenges in detecting small objects like bone fractures due to resolution constraints and the high precision requirements of medical imaging. The Albument-NAS model overcomes these challenges by utilizing advanced augmentation methods and refining its architecture to enhance the detection of fine-grained fracture details.

Future research could further investigate the impact of different types of augmentation techniques on fracture detection, especially in enhancing model robustness across diverse and inconsistent image qualities. Additionally, integrating hybrid architectures that combine the fine-grained feature extraction of ResNet50 with the dynamic optimization abilities of YOLO-NAS could yield a model with both high precision and recall. Finally, evaluating these models on a broader dataset of fracture types or on 3D imaging data might enhance their applicability and reliability in clinical diagnostics. Another hybrid method is combining YOLO-NAS with transformer-based models like DETR (DEtection TRansformers) could enhance the model's capability to identify intricate fracture patterns. This approach would capitalize on YOLO-NAS's spatial efficiency and the transformers' ability to capture contextual information.

## REFERENCES

[1] Y. Bao, "Medical image super-resolution reconstruction: A comprehensive investigation of Generative Adversarial Networks," Appl. Comput. Eng., vol. 51, pp. 14–19, 2024. doi: 10.54254/2755-2721/51/20241148.

[2] F. D. Pérez Cano, G. Parra-Cabrera, I. Vilchis-Torres, J. Reyes-Lagos, and J. Jimenez-Delgado, "Exploring Fracture Patterns: Assessing Representation Methods for Bone Fracture Simulation," J. Pers. Med., vol. 14, p. 376, 2024. doi: 10.3390/jpm14040376

[3] S. Tippannavar, Y. S. D., G. Shivakumar, and E. Madappa, "Unveiling the Spectrum: Versatile Image Processing Techniques in Bone Fracture Detection - A Comprehensive Review," J. Artif. Intell. Capsule Networks, vol. 5, pp. 499–520, 2023. doi: 10.36548/jaicn.2023.4.004

[4] M. Li, Y. Jiang, Y. Zhang, and H. Zhu, "Medical image analysis using deep learning algorithms," Front. Public Health, vol. 11, 2023. doi: 10.3389/fpubh.2023.1273253

[5] D. Yadav et al., "Hybrid SFNet Model for Bone Fracture Detection and Classification Using ML/DL," Sensors, vol. 22, p. 5823, 2022. doi: 10.3390/s22155823

[6] A. Makhlouf, M. Maayah, N. Abughanam, and C. Catal, "The use of generative adversarial networks in medical image augmentation," Neural Comput. Appl., vol. 35, 2023. doi: 10.1007/s00521-023-09100-z.

[7] R.-Y. Ju and W. Cai, "Fracture detection in pediatric wrist trauma X-ray images using YOLOv8 algorithm," Sci. Rep., vol. 13, 2023. doi: 10.1038/s41598-023-47460-7.

[8] K. T. Krishnan, L. Kavyaa, and J. Sugumar, "Automated Bone Fracture Detection Using Convolutional Neural Network," J. Phys. Conf. Ser., vol. 2471, 2023. doi: 10.1088/1742-6596/2471/1/012003

[9] K. Dimililer, "IBFDS: Intelligent bone fracture detection system," Procedia Comput. Sci., vol. 120, pp. 260–267, 2017. doi: 10.1016/j.procs.2017.11.237.

[10] I. Sary, S. Andromeda, and E. Armin, "Performance Comparison of YOLOv5 and YOLOv8 Architectures in Human Detection using Aerial Images," Ultima Comput.: J. Syst. Komput., pp. 8–13, 2023. doi: 10.31937/sk.v15i1.3204.

[11] A. Mumuni, F. Mumuni, and N. Gerrar, "A Survey of Synthetic Data Augmentation Methods in Machine Vision," Mach. Intell. Res., 2024. doi: 10.1007/s11633-022-1411-7.

[12] A. H. Basori, S. Malebary, and S. Alesawi, "Hybrid Deep Convolutional Generative Adversarial Network (DCGAN) and Xtreme Gradient Boost for X-ray Image Augmentation and Detection," Appl. Sci., vol. 13, p. 12725, 2023. doi: 10.3390/app132312725.

[13] E. Saraswathi and J. Banu, "Hybrid CGAN-based plant leaf disease classification using OTSU and surf feature extraction," Neural Comput. Appl., pp. 1–13, 2024. doi: 10.1007/s00521-024-09812-w.

[14] M. Zhu, C. Liu, and T. Szirányi, "A Global Multi-Temporal Dataset with STGAN Baseline for Cloud and Cloud Shadow Removal," in Proc. Int. Conf. Image Process. Theory, Tools Appl. (IPTA), pp. 206–212, 2023. doi: 10.5220/0012039600003497.

[15] O. Ezeme, Q. Mahmoud, and A. Azim, "Design and Development of AD-CGAN: Conditional Generative Adversarial Networks for Anomaly Detection," IEEE Access, vol. 8, pp. 1–16, 2020. doi: 10.1109/ACCESS.2020.3025530.

[16] K. D. Ahmed and R. Hawezi, "Detection of bone fracture based on machine learning techniques," Measurement: Sensors, vol. 27, p. 100723, 2023. doi: 10.1016/j.measen.2023.100723.

[17] Hareendranathan et al., "Deep Learning Approach for Automatic Wrist Fracture Detection Using Ultrasound Bone Probability Maps," SN Compr. Clin. Med., vol. 5, 2023. doi: 10.1007/s42399-023-01608-8.

[18] Q. Su, H. Hamed, M. Isa, X. Hao, and X. Dai, "A GAN-Based Data Augmentation Method for Imbalanced Multi-Class Skin Lesion Classification," IEEE Access, 2024. doi: 10.1109/ACCESS.2024.3360215.

[19] E. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. Le, "AutoAugment: Learning Augmentation Strategies From Data," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 113–123, 2019. doi: 10.1109/CVPR.2019.00020.

[20] M. Elbattah, C. Loughnane, J.-L. Guerin, R. Carette, F. Cilia, and G. Dequen, "Variational Autoencoder for Image-Based Augmentation of Eye-Tracking Data," J. Imaging, vol. 7, p. 83, 2021. doi: 10.3390/jimaging7050083.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.

[22] Buslaev, Alexander & Iglovikov, Vladimir & Khvedchenya, Eugene & Parinov, Alex & Druzhinin, Mikhail & Kalinin, Alexandr. (2020). Albumentations: Fast and Flexible Image Augmentations. Information. doi: 11. 125. 10.3390/info11020125.