Evaluating Fine Tuned Deep Learning Models for Real-Time Earthquake Damage Assessment with Drone-Based Images

Furkan Kızılay[1], Mina R. Narman[2], Hwapyeong Song[3], Husnu S. Narman[3*], Cumhur Cosgun[4], Ammar Alzarrad[5]

[1]Department of Computer Engineering, Manisa Celal Bayar University, Şehit Prof. Dr. İlhan Varank Kampüsü, Manisa, Turkey.
[2]Department of Computer Engineering, Acıbadem Mehmet Ali Aydınlar University, , Istanbul, Turkey.
[3]Department of Computer Sciences and Electrical Engineering, Marshall University, 1 John Marshall Dr, Huntington, 25755, WV, USA.
[4]Pennsylvania Department of Transportation, 1101 S. Front St., Harrisburg, 17104, PA, USA.
[5]Department of Civil Engineering, Marshall University, 1 John Marshall Dr, Huntington, 25755, WV, USA.

*Corresponding author(s). E-mail(s): narman@marshall.edu;
Contributing authors: 190315027@ogr.cbu.edu.tr; Mina.Narman@live.acibadem.edu.tr; song24@marshall.edu; ccosgun@pa.gov; alzarrad@marshall.edu;

Abstract

Earthquakes pose a significant threat to life and property worldwide. Rapid and accurate assessment of earthquake damage is crucial for effective disaster response efforts. This study investigates the feasibility of employing deep learning models for damage detection using drone imagery. We explore the adaptation of models like VGG16 for object detection through transfer learning and compare their performance to established object detection architectures like YOLOv8 (You Only Look Once) and Detectron2. Our evaluation, based on various metrics including mAP, mAP50, and recall, demonstrates the superior performance of YOLOv8 in detecting damaged buildings within drone imagery, particularly for cases with moderate bounding box overlap. This finding suggests its potential suitability for real-world applications due to the balance between accuracy and

efficiency. Furthermore, to enhance real-world feasibility, we explore two strategies for enabling the simultaneous operation of multiple deep learning models for video processing: frame splitting and threading. In addition, we optimize model size and computational complexity to facilitate real-time processing on resource-constrained platforms, such as drones.
This work contributes to the field of earthquake damage detection by (1) demonstrating the effectiveness of deep learning models, including adapted architectures, for damage detection from drone imagery, (2) highlighting the importance of evaluation metrics like mAP50 for tasks with moderate bounding box overlap requirements, and (3) proposing methods for ensemble model processing and model optimization to enhance real-world feasibility. The potential for real-time damage assessment using drone-based deep learning models offers significant advantages for disaster response by enabling rapid information gathering to support resource allocation, rescue efforts, and recovery operations in the aftermath of earthquakes.

Keywords: Earthquake Damage Detection, Deep Learning for Object Detection, Drone Imagery Analysis, Model Processing for Video Analysis, Real-Time Disaster Response Systems

1 Introduction

Earthquake is one of the most catastrophic natural disasters, inducing widespread devastation and significant mortality. Prompt and precise assessment of earthquake damage is vital for an effective disaster response. This facilitates the targeted allocation of resources for search and rescue initiatives, infrastructure repair, and reconstruction endeavors. Traditional methods for damage assessment often rely on manual field surveys, which can be time-consuming, labor-intensive, and potentially dangerous in post-earthquake environments (Ünlü & Kiriş, 2022). Building upon the limitations of traditional methods, recent advancements in unmanned aerial vehicle (UAV) technology, commonly referred to as drones, offer a promising solution for expedited and safe post-disaster data collection. These advancements enable drones to capture high-resolution aerial imagery of disaster zones, providing crucial insights into the magnitude and severity of the damage (Abdi & Jabari, 2021). This data can be used to create detailed damage maps, aiding decision-making for rescue teams and disaster management authorities.

Several studies have explored the application of remote sensing techniques for earthquake damage detection using satellite and aerial imagery (Hasanlou et al., 2021; Rao et al., 2023; J. Zhang & Gong, 2013). These methods often rely on change detection between pre- and post-earthquake images to identify areas with structural alterations (Joshi et al., 2017; Khodaverdi zahraee & Rastiveis, 2017). However, limitations exist, such as the requirement for pre-disaster imagery and the potential for inaccuracies due to factors like cloud cover or low spatial resolution (Hasanlou et al., 2021).

Recent studies have further advanced UAV-based seismic damage assessment methodologies. For instance, Xiong et al. (2020) introduce an automated approach combining UAVs and Convolutional Neural Networks (CNNs) to assess building seismic damage. Their method involves three key components: data preparation, building image segmentation, and CNN-based damage assessment, achieving a damage distribution accuracy of 89.39%. This approach highlights the potential of integrating UAV technology with advanced neural network models for effective damage evaluation. Additionally, Fernandez Galarreta et al. (2015) address the challenges of structural damage assessment using UAVs equipped with oblique imagery. Their research integrates 3-D point-cloud data with object-based image analysis (OBIA) to evaluate building façades and roofs, focusing on creating comprehensive damage scores and identifying major damage features. This study underscores the utility of multi-perspective UAV imagery for detailed damage classification and highlights the importance of high-resolution data in post-disaster assessments.

While the potential of drone imagery for earthquake damage detection is recognized, there is a gap in the literature regarding the development of datasets and models specifically tailored for this task. Existing deep learning models for damage classification frequently rely on publicly available datasets comprised of satellite or aerial imagery (Fujita et al., 2017). However, these datasets may not provide the requisite detail and resolution captured by drone images (Hong et al., 2022; C. Liu et al., 2022). Additionally, these models might not be optimized for the specific challenges associated with drone-captured data, such as variations in lighting conditions, camera angles, and image quality (Kalantar et al., 2020).

This paper aims to bridge the aforementioned gap in the literature by not only acknowledging the potential of UAV technology for earthquake damage assessment but also by actively contributing to its development. To achieve this, we propose a three-pronged approach. First, we will develop a comprehensive drone image dataset specifically designed for earthquake damage detection. Our proposed approach comprises three distinct stages: identifying building demolition, classifying building damage types, and detecting specific wall damage. To address the regional structural damage identification within the context of this work, the proposed approach also aims to extend its applicability to broader regional assessments. By leveraging the high-resolution imagery and detailed damage annotations provided by drone technology, this research facilitates the identification of structural damage not only at the building level but also across larger geographic areas. This enables a more comprehensive analysis of regional damage patterns, which can be instrumental in evaluating the overall impact of the earthquake on infrastructure and guiding resource allocation for recovery efforts. The identification of wall damage types is particularly crucial, as vertical cracks, often resulting from structural movements, can compromise the stability of load-bearing walls and pose significant safety concerns (S. J. Kim et al., 2011). This dataset will be meticulously annotated with detailed labels for various damage features, enabling the effective training and evaluation of deep learning models. Secondly, we will conduct a comparative analysis of the performance of different deep learning architectures for object detection in this domain. Specifically, we will evaluate the effectiveness of YOLOv8, Detectron2, and an adaptation of VGG16 for this task. The selection of these models was guided by the balance between computational efficiency and accuracy. YOLOv8 and Detectron2 were chosen due to their proven effectiveness in real-world scenarios, providing a robust balance between speed and accuracy. YOLOv8 is recognized for its rapid processing capabilities, making it suitable for real-time applications, while Detectron2 offers high accuracy with its advanced object detection features. Although transformer-based models exhibit high accuracy, they were not selected due to their larger model sizes and greater computational requirements. A comprehensive evaluation of a wide range of models led to the focus on those that offer both practical performance and applicability for earthquake damage detection using drone imagery. Finally, we will assess the potential of the proposed approach for real-world applications through qualitative and quantitative analyses. By addressing these objectives, this work contributes to the development of more accurate and effective methods for earthquake damage detection, thereby enhancing disaster response efforts with timely and detailed information to guide critical search and rescue operations, as well as reconstruction endeavors.

The rest of the paper is organized as follows: In Section 2, we describe the most relevant and recent work on the problem of earthquake damage assessment. In Section 3, the dataset preparation steps and dataset information of three different models used for earthquake damage detection are mentioned. In Section 4, the working principles of object detection and the architecture of the 3 different models used are mentioned. In Section 5, the training and test phases of the models are mentioned. Section 6 includes a comparison of the error metrics of all trained models, a comparison of the models against each other, and an evaluation of each class in the datasets. In Section 7, threading and model size reduction approaches are mentioned to increase the usability of models to be integrated into the limited hardware capabilities of drones. Section 8 includes which model outperforms other models in earthquake damage detection using drone imagery, highlights its superior accuracy and efficiency, and explores strategies for real-time deployment on resource-constrained platforms. Section 9 discusses the expansion of datasets for future research, the exploration of alternative deep learning architectures, and the integration of additional image analysis techniques. Section 10 demonstrates that YOLOv8 is effective for earthquake damage detection using drones and includes future research opportunities to improve real-world applications.

2 Related Work

Detection and assessment of earthquake damage through remote sensing techniques have been a significant focus of research, with conventional methodologies often relying on pre- and post-earthquake satellite or aerial imagery for change detection (Hasanlou et al., 2021; Rao et al., 2023; J. Zhang & Gong, 2013). While these methodologies provide valuable insights, they are constrained by certain limitations. For instance, the necessity for pre-disaster imagery can impede swift assessment in the aftermath of an earthquake (Khodaverdi zahraee & Rastiveis, 2017), and factors such as cloud cover or low spatial resolution of satellite imagery can compromise the accuracy of damage detection (Hasanlou et al., 2021).

The advent of deep learning has introduced more sophisticated approaches to damage detection, with studies exploring the application of Convolutional Neural Networks (CNNs) to classify damage severity in post-disaster imagery (Hong et al., 2022; Joshi et al., 2017). However, these approaches often rely on publicly available datasets comprised of satellite or aerial imagery, which may not capture the level of detail and resolution offered by drone imagery (Hong et al., 2022; C. Liu et al., 2022). Consequently, these models may not be readily applicable to the unique challenges posed by drone-captured data, such as variations in lighting conditions, camera angles, and image quality (Kalantar et al., 2020). A recent investigation delves into the potential of deep learning for post-earthquake building damage detection using very high-resolution (VHR) TerraSAR-X imagery (Turker & Sumer, 2008). Their findings underscore the efficacy of deep learning in this domain when leveraging high-resolution satellite data. Nevertheless, satellite imagery remains subject to limitations such as resolution constraints and potential occlusions due to cloud cover.

Recent advances in deep learning methods have further enhanced damage assessment capabilities. For instance, Wang et al. (2023) introduced the QuakeCityNet (QCNet) model, which integrates a novel geometric consistency enhanced (GCE) loss function to address the challenge of precise pixel-level recognition of earthquake-damaged buildings using UAV images. This model demonstrates robust performance across various weather conditions and has shown a significant reduction in false-positive noise, emphasizing the model's effectiveness in handling complex scenarios often encountered in urban seismic damage assessment. Additionally, Hoskere et al. (2022) leverage computer generated imagery (CGI) to construct a 3D synthetic environment for the simulation of post-earthquake inspections. By incorporating physics-based graphics models (PBGMs), their approach allows for comprehensive evaluation and validation of autonomous inspection strategies. This methodology supports various aspects of autonomous post-earthquake assessment, including simulation of UAV path planning, automatic image labeling, and direct comparison of different autonomous assessment approaches. Furthermore, Y. Liu et al. (2019) propose a deep learning model that integrates spatial pyramid pooling with an encoder-decoder architecture for high-resolution building segmentation in remote sensing imagery. Their approach addresses the challenges of feature resolution loss in CNNs, providing qualitative and quantitative improvements over existing models such as U-Net and SegNet. This model's performance highlights its potential utility in practical scenarios involving high-resolution remote sensing data.

The exploration of object detection models for earthquake damage assessment using drone imagery is an emerging field of study. However, there is a paucity of research directly comparing the performance of different deep learning architectures for this specific task. Existing comparative works on object detection models often center on general-purpose datasets or applications beyond the scope of earthquake damage detection (Bhowmick et al., 2020; Mangalathu et al., 2020). A study juxtaposes the performance of YOLOv8 with other detection models using the Microsoft COCO dataset, a widely recognized benchmark for general object detection tasks (Mangalathu et al., 2020). While informative regarding the relative merits of these models, such findings may not directly translate to the challenges inherent in earthquake damage detection using drone imagery.

Several studies address the challenge of concurrent object detection from multiple video streams using deep learning models while considering resource constraints on processing hardware. Abri et al. (2020) propose a multi-thread frame tiling model specifically for YOLOv3 object detection, achieving significant improvements in frames per second (FPS) for processing multiple video streams concurrently on CPUs and GPUs without sacrificing detection accuracy. R. Kim et al. (2020) similarly explore resource optimization for deep learning object detection at the edge using TVM, an automated compiler that tailors the models to specific hardware characteristics. Their approach leverages task-level pipeline parallelism to improve overall object detection performance on multi-stream video workloads running on edge devices with limited computational capabilities. Lee and Hwang (2022) finally acknowledge the computational demands of YOLO for real-time object detection on embedded systems. They propose a novel YOLO architecture with adaptive frame control (AFC) that addresses these limitations by dynamically adjusting frame processing rates. Their findings demonstrate that AFC maintains high detection precision while enabling real-time processing on resource-constrained hardware. These works all highlight the importance of optimizing resource utilization for efficient concurrent object detection from multiple video streams using deep learning models.

This research endeavors to address this gap by developing a comprehensive drone image dataset tailored specifically for object detection in earthquake damage assessment. We aim to assess the efficacy of prominent deep learning architectures, including YOLOv8, Detectron2, and a VGG16 adaptation, to ascertain the most suitable model for this task. Through rigorous evaluation of our dataset, we seek to glean valuable insights into the strengths and limitations of these architectures within the context of drone-based earthquake damage detection.

3 Creating a Dataset

This research employs drone imagery captured in the aftermath of the devastating 2023 Hatay-Maraş earthquake to curate comprehensive datasets for earthquake damage detection. The datasets are created by collecting video footage taken by drones and extracting high-resolution photos from these videos. Drone imagery presents a significant advantage over traditional satellite or aerial data by offering high-resolution views of affected areas, facilitating the identification of finer-grained damage features (Joshi et al., 2017). To streamline the data collection and labeling process, we leverage the platform Roboflow in tandem with Label Studio, a powerful open-source annotation tool. Roboflow provides a convenient interface for uploading and managing drone images, while Label Studio facilitates the creation of high-quality annotations for various damage features.

However, the creation of robust datasets for earthquake damage detection using drone imagery poses unique challenges. Data collection efforts may encounter hindrances such as airspace restrictions or safety concerns in post-earthquake environments (C. Liu et al., 2022). Moreover, the labeling process demands meticulous attention to detail, as annotators should accurately identify and categorize diverse damage features within the high-resolution drone images (Hong et al., 2022). One specific challenge in working with drone imagery is the occurrence of potential image orientation inconsistencies. These inconsistencies may arise due to variations in drone flight paths, camera angles, or post-processing steps, which could result in images being rotated or flipped in unintended ways. To address this, an Auto-Orient step was implemented across all three datasets. This preprocessing step automatically corrects the orientation of images by referencing the embedded metadata, such as the EXIF orientation tag, ensuring that each image is displayed in its intended orientation. This step is crucial as it guarantees uniformity in image presentation, thereby preventing any misinterpretation of damage features during the subsequent labeling and analysis processes. By recognizing and addressing these challenges through efficient data collection and labeling strategies, we aim to develop a high-quality dataset that effectively captures the complexities of earthquake damage as observed through drone imagery (Pi et al., 2021). This dataset will serve as a valuable resource for training and evaluating deep learning models for object detection in earthquake damage assessment.

To ensure the quality and utility of the datasets, two primary categories of techniques were employed: preprocessing and augmentation. Preprocessing techniques are foundational in image processing projects, as they prepare the data for subsequent deep learning model training (Vocaturo et al., 2018). This initial step ensures consistency within the dataset and adapts the images into a format suitable for the models. For instance, all three datasets underwent an Auto-Orient step to rectify any potential image orientation inconsistencies. Additionally, resizing the images to specific dimensions was imperative. The building classification dataset was resized to 640x640 pixels, while the damage level classification and wall damage detection datasets were resized to 416x416 pixels and 640x640 pixels, respectively. These resize dimensions were selected based on the specific requirements of the deep learning architectures employed for each task. The damage type dataset is composed of a single class (collapsed), whereas the damage level dataset comprises four distinct classes (Heavy Damage, Minor Damage, Moderate Damage, Undamaged). Lastly, the wall damage dataset includes two classes (vertical and horizontal), focusing on the detection of different orientations of wall cracks.

Following preprocessing, data augmentation techniques were applied to artificially expand the dataset size and introduce variations in image appearance. This is a crucial step in image processing projects as it enhances the fine-tuned model's ability to learn robust features that generalize well to unseen data during testing (Ayan & Ünver, 2018). Real-world data often exhibits variations in factors such as lighting, orientation, and sensor noise. By introducing these variations through augmentation, the model's capacity to handle such variations and perform accurately on new, unseen images is enhanced. Through the application of these preprocessing and augmentation techniques, our aim was to develop datasets that more accurately represent real-world scenarios and enhance the generalization capabilities of the deep learning models utilized for earthquake damage detection. To address the complexities of the earthquake damage detection task, where capturing the diverse and intricate features of damage types, levels, and wall damages is crucial, the proportion of the training set has been intentionally kept high in the proposed system to ensure a more comprehensive training set. Consequently, the training sets for each dataset comprise 86% of the images, while the validation and test sets consist of 9% and 5%, respectively. To ensure robust model performance evaluation and fine-tuning, a higher proportion of images 9% was allocated to the validation set compared to the test set 5%, allowing for more comprehensive model adjustment and hyperparameter optimization while reserving a smaller portion for final performance assessment. Table 1 summarizes the three datasets and their status following the preprocessing and augmentation steps applied to each of them:

Table 1 Dataset Information

Dataset	Train Set	Validation Set	Test Set
Damage Type	86% (3182 images)	9% (332 images)	5% (185 images)
Damage Level	86% (9322 images)	9% (976 images)	5% (541 images)
Wall Damage Detection	86% (4480 images)	9% (469 images)	5% (260 images)

4 Model Architectures

This research investigates the performance of three prominent deep learning architectures for object detection in earthquake damage assessment using drone imagery: YOLOv8, Detectron2 with a Faster R-CNN backbone, and a VGG16 adaptation. Fig. 1 summarizes the mechanics of two common object detection methods: two-stage and one-stage detectors. Two-stage detectors perform region proposal in one stage and classification and bounding box regression in another stage. One-stage detectors perform these tasks in a single stage.

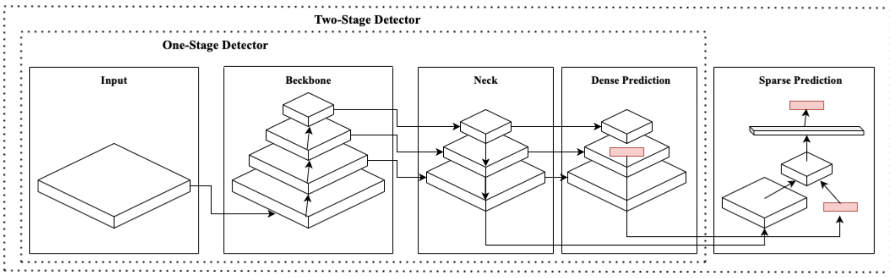

Fig. 1 The process of an object detection model

YOLOv8 is a single-stage object detection model known for its real-time processing capabilities and high accuracy (Mangalathu et al., 2020). It employs a single neural network to simultaneously predict bounding boxes and class probabilities for objects within an image. YOLOv8 offers pre-trained models with varying complexities, allowing us to explore the trade-off between accuracy and computational efficiency. In this study, we utilize the YOLOv8 Nano and YOLOv8x models, representing lightweight and high-precision versions, respectively. YOLOv8 Nano is the smallest and the fastest while the YOLOv8x is the most accurate and slowest among the other YOLOv8 models. Table 2 summarizes the performance metrics for two YOLOv8 models, YOLOv8n and YOLOv8x. It compares factors like model size, mean Average Precision (mAP) on validation data, inference speed on CPU and Nvidia A100 GPU, number of trainable parameters, and floating-point operations (FLOPs).

Table 2 YOLOv8 Performance Metrics

Model	Size (pixels)	mAP val	Speed (CPU ONNX ms)	Speed (A100 TensorRT ms)	Parameters (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

Detectron2 is a versatile open-source platform offering a suite of pre-trained object detection models based on various architectures (Gerke & Kerle, 2011). It facilitates experimentation with different backbones and configurations. For this research, we leverage the faster-rcnn-R-50-FPN-3x model, which utilizes a Faster R-CNN architecture with a ResNet-50 backbone trained on the COCO dataset. Faster R-CNN is a two-stage object detection model that employs a region proposal network (RPN) to generate candidate object bounding boxes, followed by a classification and bounding box regression stage for refinement (Adams, 2004). The VGG16 architecture, originally designed for image classification, is adapted for object detection in this work. VGG16 utilizes a series of convolutional layers with increasing filter sizes and decreasing depths, effectively extracting features from images at different scales (Dell'Acqua & Gamba, 2012). While powerful for classification, modifications are necessary for object detection tasks. These modifications involve replacing the final fully connected layers in VGG16, designed for classification using softmax activation, with layers suitable for object detection (Adriano et al., 2021; Gong et al., 2013). We employ linear activation in these layers and adjust the output dimensionality to predict bounding box coordinates (startX, startY, endX, and endY). This allows the model to directly regress bounding boxes rather than solely predicting class probabilities. Furthermore, to facilitate precise bounding box regression, we append additional dense layers to the VGG16 architecture (Naito et al., 2020). These layers are strategically positioned after the convolutional blocks to leverage the hierarchical feature representations learned by the network. By applying non-linear transformations through these new dense layers, the model learns to predict precise bounding box coordinates. This augmentation enables the VGG16 architecture to perform both feature extraction and bounding box regression within a unified framework. By comparing the performance of these three architectures, YOLOv8, Detectron2 with Faster R-CNN, and the VGG16 adaptation, we aim to identify the most suitable model for object detection in the context of earthquake damage assessment using drone imagery.

5 Experimental Setup

This section details the hardware and software configurations used for model training and evaluation, along with the rationale behind the chosen deep learning architectures and the strategies employed to ensure a fair comparison between them.

The experimental setup leveraged Google Colab Pro+ for training the deep learning models. This cloud-based platform provided access to a powerful A100 Tensor Core GPU, accelerating the training process significantly. The software environment primarily relied on Python programming language with deep learning libraries such as Ultralytics (for YOLOv8) and Detectron2 (for Faster R-CNN).

1. Rationale for Deep Learning Architecture Selection Two prominent object detection architectures were chosen for this study: YOLOv8 and Detectron2's implementation of Faster R-CNN with R50-FPN backbone. Both architectures were selected due to their suitability for multi-object detection tasks. The VGG16 model's results were not included in the evaluation due to its inability to make multiple predictions on a single image, rendering it unsuitable for real-world scenarios.
2. YOLOv8: This single-stage detector offers a compelling balance between real-time processing capabilities and high accuracy (Mangalathu et al., 2020). Its ability to simultaneously predict bounding boxes and class probabilities for objects within an image makes it a strong contender for real-world applications like drone-based damage detection.
3. Detectron2's Faster R-CNN with R50-FPN Backbone: This two-stage detector offers high accuracy and has been successfully applied to various object detection tasks (Gerke & Kerle, 2011). The R50-FPN backbone utilizes a residual network with feature pyramid networks, potentially improving the model's ability to detect objects at various scales, a crucial aspect for earthquake damage assessment in drone imagery.

The rationale behind selecting YOLOv8 and Detectron2 lies in their complementary strengths; YOLOv8's speed and efficiency make it ideal for real-time applications, while Detectron2's robust architecture excels in accuracy and detailed object recognition. YOLOv8 is particularly well-suited for scenarios where quick decision-making is crucial, such as in real-time monitoring and immediate damage assessment during disaster response (Reis et al., 2023). On the other hand, Detectron2's advanced two-stage detection process and the use of a residual network with feature pyramid networks (R50-FPN) enhance its capability to detect objects at various scales and resolutions, making it particularly adept at identifying detailed damage patterns in complex scenes (Park et al., 2023).

These models are particularly suitable for earthquake damage detection using drone imagery due to their ability to handle complex scenes and detect multiple objects with varying sizes and orientations. YOLOv8's architecture is designed to perform well even with lower resolution images, which is common in drone footage, ensuring that smaller and potentially critical details are not missed (Li et al., 2023). Detectron2, with its robust feature extraction capabilities, can efficiently process high-resolution images to capture fine-grained details of the damage, which is essential for comprehensive damage assessment and planning of repair strategies.

Their performance is critical in addressing challenges such as varying image quality due to different lighting conditions, diverse types of structural damage, and the need for rapid processing to facilitate timely disaster response. YOLOv8's real-time processing capabilities allow for the immediate identification and localization of damaged areas, enabling quick decision-making and prioritization of response efforts (Lou et al., 2023). Detectron2, with its superior accuracy, helps in generating detailed reports of the damage, which are crucial for documentation and further analysis. Both models' adaptability through transfer learning ensures that they can be fine-tuned to the specific characteristics of the earthquake damage detection task, thus improving their overall effectiveness and reliability in real-world applications. By leveraging these advanced architectures, the study aims to enhance the effectiveness of drone-based earthquake damage assessment, ultimately contributing to more efficient disaster management and recovery efforts.

Furthermore, both YOLOv8 and Detectron2 offer pre-trained models on extensive datasets, such as the COCO (Common Objects in Context) and ImageNet datasets. COCO contains over 200,000 labeled images with more than 80 object categories, providing a diverse range of scenarios that enhance the models' ability to generalize across various tasks. ImageNet, on the other hand, consists of over 14 million images spanning thousands of categories, making it one of the most comprehensive datasets available for image classification and object detection tasks. Fine-tuning these pre-trained models on our custom drone image dataset allows for adaptation to the unique challenges of earthquake damage detection, potentially leading to improved accuracy compared to training models from scratch. To ensure a fair comparison between YOLOv8 and Detectron2, we adopted a meticulous approach to training hyperparameter configuration.

In this study, hyperparameter optimization was carried out using a combination of grid search and Bayesian optimization techniques to maximize model performance. Grid search systematically tested predefined combinations of hyperparameters, such as learning rates and batch sizes, providing a foundational understanding of their effects. However, due to its computational intensity, it was complemented by Bayesian optimization, which employs a probabilistic model to iteratively select promising hyperparameter configurations. This approach efficiently navigates the hyperparameter space, balancing exploration and exploitation to converge on optimal settings more effectively.

The optimization focused on key parameters such as learning rate, weight decay, and training epochs. YOLOv8's learning rate and batch size were adjusted for better convergence and efficiency, while Detectron2's learning rate schedule and proposal numbers were fine-tuned to enhance detection accuracy. The use of these advanced techniques ensured that both models were meticulously tuned, resulting in improved performance for earthquake damage detection and more accurate, reliable results in practical applications.

Max Iteration Calculation: When training models with Detectron2, we determined the optimal number of iterations based on the training process. Subsequently, we employed the formula

$$\text{Epoch} = \frac{\text{Max Iteration} \times \text{Batch Size}}{\text{Train Size}} \tag{1}$$

to calculate the corresponding number of epochs required for training the YOLOv8 model. This approach ensures both models are trained for a comparable amount of time, allowing for a more balanced evaluation of their performance. An epoch represents one full pass through the entire training dataset, while an iteration corresponds to a single update of the model's parameters based on a subset of the data, known as a batch. The relationship between epochs, iterations, batch size, and the size of the training dataset allowed us to determine how many epochs would result in the same total number of updates as the iterations used in Detectron2.

By calculating the number of epochs in this way, we ensured that both YOLOv8 and Detectron2 were trained for a comparable amount of time. This approach was crucial for making a balanced evaluation of their performance, as it allowed us to compare the models under similar training conditions.

The training process involved meticulous data preparation, model configuration, parameter tuning, and evaluation. Here's a detailed breakdown of the steps undertaken for both YOLOv8 and Detectron2 fine-tuned models:

5.1 Data Preparation:

The custom drone image dataset was split into training, validation, and test sets. This ensures the model is trained on a representative sample of the data, the validation set helps fine-tune hyperparameters to avoid overfitting, and the test set provides an unbiased evaluation of the model's generalization ability. Data cleaning and preprocessing techniques were applied, such as handling missing values and normalizing pixel intensities, to ensure the data is suitable for training the deep learning models (Salvi et al., 2021). Furthermore, during the preprocessing phase, handling missing values was considered critical, particularly in scenarios where incomplete metadata or corrupted images could compromise the dataset's integrity. Missing values were specifically addressed by implementing data imputation techniques or by excluding any images or annotations that could not be reliably corrected. By ensuring that all images and their associated metadata were complete and accurate, the risk of introducing noise into the training process was minimized, thereby enhancing the reliability and performance of the trained models.

5.2 Model Configuration:

For Detectron2, the configuration file was meticulously tailored to the specific task. This involved specifying the Faster R-CNN with R50-FPN backbone architecture, and defining essential training parameters like batch size, learning rate, and the number of iterations (epochs) as calculated previously.

For YOLOv8, the models YOLOv8 Nano and YOLOv8x were based on the desired trade-off between accuracy and computational efficiency. Detecting damage types and wall damage types requires fast processing and accurate feature extraction, thus, YOLOv8 Nano among the detailed models of YOLOv8 was configured. For damage severity classification, YOLOv8x, which is suitable for processing complex nuances rather than speed, was used.

5.3 Parameter Fine Tuning and Optimization:

The training process of YOLOv8 models encompasses hyperparameter tuning, a crucial aspect of real-world training (Isa et al., 2022). This iterative process involves adjusting parameters like learning rate, batch size, and optimizer choice to optimize model performance. Efficient exploration of the hyperparameter space was achieved through the employment of techniques such as grid search and automated tools. Detectron2 offers more flexibility for hyperparameter tuning within its configuration file (Ali et al., 2022). To potentially enhance model performance, parameters such as the learning rate scheduler and weight decay were adjusted.

5.4 Model Training and Monitoring:

The training process involved running the models for the designated number of iterations (epochs) while monitoring training metrics like loss and validation accuracy. This monitoring allows for early detection of potential issues like overfitting and enables adjustments to the training strategy if necessary.

5.5 Model Evaluation:

This evaluation process assesses the model's ability to generalize to unseen data and accurately identify objects of interest (e.g. damaged buildings). By comparing the AP and AR metrics between YOLOv8 and Detectron2, we determined which model performed better in our specific earthquake damage detection task. Fig. 2 shows a precision-recall curve for an object detection model. The precision-recall curve plots the trade-off between precision and recall for different classification thresholds (X. Zhang et al., 2015). Precision is the proportion of true positives among the predicted positives, while recall is the proportion of true positives among all the actual positives (Saito & Rehmsmeier, 2015).

The line definitions in the figure are as follows:

- Blue line: HorizontalCrack 0.822
- Orange line: VerticalCrack 0.965
- Dark blue line: all classes 0.894 mAP@0.5

5.6 Inference:

Following successful training and evaluation, the final models were used to perform inference on unseen drone images. This allows for visualization of the model's predictions and practical assessment of its effectiveness in real-world damage detection scenarios. While the overall training process follows the outlined structure for both YOLOv8 and Detectron2, the specific software libraries necessitate some distinct configurations.

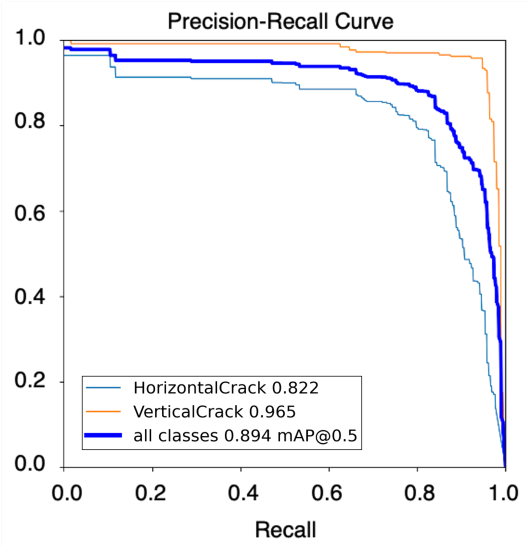

Fig. 2 Precision recall curve comparison for different crack types

By employing these software-specific approaches and adhering to the overarching training principles, we ensured a controlled and optimized training environment for both YOLOv8 and Detectron2, enabling a fair comparison of their performance in earthquake damage detection using drone imagery.

6 Results and Analysis

This section presents the results obtained from training and evaluating YOLOv8 and Detectron2 models on the custom earthquake damage detection datasets. The selection of YOLOv8 for damage type and wall damage type detection, and YOLOv8x for damage level classification, was driven by the specific strengths and requirements of each task. YOLOv8's architecture, with its balance between speed and accuracy, was well-suited for distinguishing damage types and wall cracks, where real-time processing and precise feature extraction are crucial. YOLOv8x, with its enhanced capacity and improved performance on complex tasks, was chosen for damage level classification to handle the nuanced differences in damage severity more effectively. The models were meticulously trained using tuned hyperparameters to ensure optimal performance. Here, we present the evaluation metrics alongside a discussion of the insights gleaned from the comparison and their implications for future research.

6.1 Evaluation Metrics

To evaluate the performance of the YOLOv8 and Detectron2 models in detecting earthquake damage within drone imagery, we employed standard object detection metrics. Here, we provide a brief explanation of each metric used:

Average Precision (AP): This metric calculates the mean value of precision across all recall levels for a specific object class. It provides a comprehensive measure of a model's ability to both correctly identify objects and retrieve a high proportion of all relevant objects within a class (Ramzi et al., 2021).

Mean Average Precision (mAP): This metric averages the AP scores across all object classes present in the dataset. It offers a single value to represent the overall detection performance of the model for all classes (Henderson & Ferrari, 2017).

$$\mathrm{mAP} = \frac{1}{k}\sum_{i=1}^{k} \mathrm{AP}_i \tag{2}$$

mAP50: This metric is a variant of mAP that calculates the mean AP considering only objects with Intersection over Union (IoU) thresholds of 0.5. IoU measures the area of overlap between the predicted bounding box and the ground truth bounding box for an object. This metric provides insights into the model's ability to detect objects with varying degrees of accuracy (Znamenskaya et al., 2023).

Recall: This metric represents the proportion of all actual positive cases that were correctly identified by the model. It focuses on the model's ability to retrieve a high percentage of relevant objects within a class (Wu et al., 2019).

$$\mathrm{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{3}$$

While these evaluation metrics are well-established in the field, our research further investigates their applicability in complex real-world scenarios, particularly in earthquake damage detection using drone imagery. By integrating these metrics with innovative model adaptations and data augmentation techniques, we aim to push the boundaries of current object detection performance, offering new insights into the deployment of deep learning models in disaster response contexts. These enhancements contribute to the development of more robust and adaptable models capable of handling the unique challenges presented by post-disaster environments.

6.2 Results

Table 3 compares the performance of two damage-type detection models: Detectron2 and YOLOv8 Nano. It evaluates them based on three metrics: mean Average Precision at an IoU threshold of 0.5 (mAP50), recall, and Average Precision (AP) for a collapsed category.

Table 3 Damage Type Models Results

Metric	Detectron2 (15960 iteration)	YOLOv8 Nano (80 epoch)
mAP50	48.83	92.1
Recall	-	82.1
AP (Collapsed)	48.3	92.1

To provide a clearer understanding of the performance differences between the two models in detecting damage types, we present visual results from both Detectron2 and YOLOv8 Nano in Fig. 3 and Fig. 4. These visuals illustrate the models' outputs on sample images, highlighting how each model performs in identifying and classifying the types of damage.

Fig. 3 Damage type detection results using YOLOv8 Nano

Fig. 4 Damage type detection results using Detectron2

Table 4 compares the performance of two object detection models, Detectron2 and YOLOv8x, on the task of classifying the damage level of objects. The metrics include mAP50, Recall, and Average Precision (AP) for different damage levels (Heavy, Minor, Moderate, and Undamaged). Overall, YOLOv8x performs better than Detectron2 on all metrics except mAP50.

The prediction of damage levels presents a more complex challenge compared to detecting damage types or wall cracks, primarily due to the nuanced distinctions required in assessing the severity of damage. This task involves interpreting subtle variations in visual features, which can be particularly challenging for image processing models. The model's lower performance in damage level prediction reflects these inherent difficulties and the limitations of current object detection frameworks in capturing the intricate details needed for accurate classification. Moreover, the accurate determination of damage levels often necessitates the expertise of earthquake engineers, who can assess structural damage based on detailed criteria beyond visual cues alone. The system developed in this study provides a preliminary detection capability with sufficient accuracy, yet it is not a substitute for expert evaluation. Future research and improvements are essential to enhance the model's performance and reliability for more precise damage level assessment, supporting the system as a valuable tool for initial evaluation and aiding expert decision-making in earthquake damage assessment.

To provide a clearer understanding of the performance differences between the two models in detecting damage levels, we present visual results obtained from both Detectron2 and YOLOv8x in Fig. 5 and Fig. 6.

Table 5 compares the performance of two object detection models, Detectron2 and YOLOv8 Nano, on the task of classifying the type of wall cracks (horizontal or vertical). The metrics include mAP50, Recall, and Average Precision (AP) for each crack

Table 4 Damage Level Models Results

Metric	Detectron2 (155900 iteration)	YOLOv8x (200 epoch)
mAP50	55.44	68.9
Recall	-	63.3
AP (Heavy Damage)	38.47	82.4
AP (Minor Damage)	20.25	52.3
AP (Moderate Damage)	26.04	64.5
AP (Undamaged)	40.81	76.4

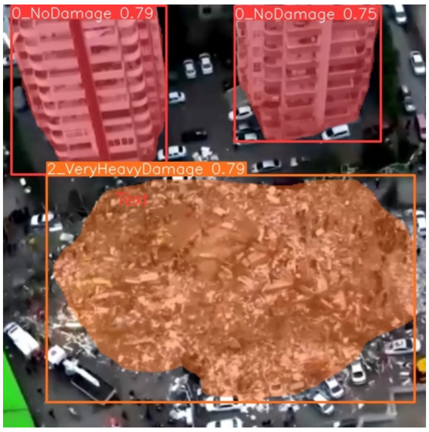

Fig. 5 Damage level detection results using YOLOv8x

Fig. 6 Damage level detection results using Detectron2

type. Vertical and horizontal cracks have distinct structural implications that require separate identification. YOLOv8 Nano distinguishes between these crack types by learning specific features associated with each orientation during training, utilizing its convolutional layers to capture and classify the orientation of cracks. Similarly, Detectron2's Faster R-CNN model leverages its region proposal and classification stages to accurately identify and differentiate crack orientations based on their spatial patterns and characteristics.

Table 5 Wall Damage Type Models Results

Metric	Detectron2 (22890 iterations)	YOLOv8 Nano (80 epoch)
mAP50	92.01	89.4
Recall	-	83.7
AP (Horizontal Crack)	73.16	82.3
AP (Vertical Crack)	70.43	96.5

To provide a clearer understanding of the performance differences between the two models in detecting wall damage types, we present visual results obtained from both Detectron2 and YOLOv8 Nano in Fig. 7 and Fig. 8.

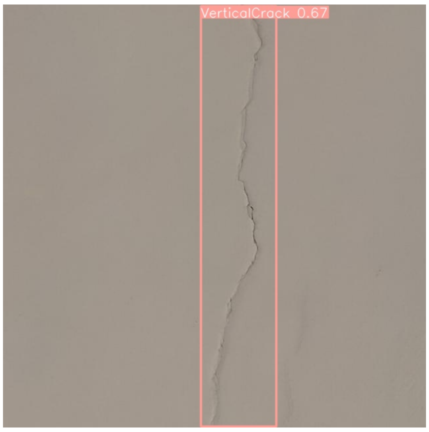

Fig. 7 Wall damage type detection results using YOLOv8 Nano

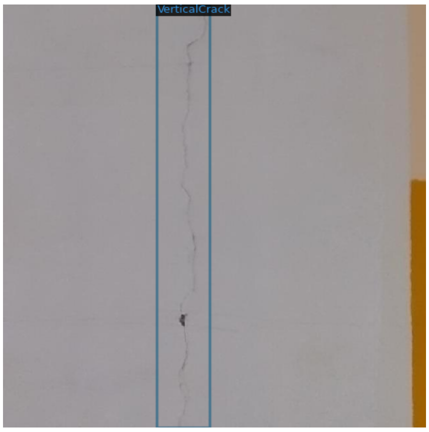

Fig. 8 Wall damage type detection results using Detectron2

The findings from this comparison offer valuable insights for future research in earthquake damage detection using drone imagery. The superior performance of YOLOv8 in this study highlights the potential benefits of leveraging single-stage object detection models for this task. YOLOv8's better performance in detecting damage types and wall damage types can be attributed to its efficient real-time processing capabilities, which excel in capturing and classifying damage features effectively. However, YOLOv8's lower performance in damage level classification can be explained by its single-stage architecture, which might struggle with the complexities involved in distinguishing between subtle differences in damage levels compared to multi-stage models. Detectron2's poorer performance in detecting both damage types and damage levels, in contrast to its better performance in wall damage type detection, can be attributed to the intricacies of its two-stage architecture. While Detectron2's region proposal and classification stages offer high accuracy in scenarios with well-defined and distinguishable features, such as wall cracks, they may face challenges in differentiating between complex damage types and levels. The model's performance for damage type and level tasks might be hindered by the varied and less distinct nature of damage features compared to the more discrete nature of wall cracks.

Specifically, for wall damage types, Detectron2's high mAP50 score indicates its effectiveness in overall detection accuracy. However, the model's performance for detecting vertical and horizontal cracks shows variability. This discrepancy could be due to the model's training focusing more on general crack detection, leading to less effective differentiation between crack orientations. Overall, the better mAP50 performance of Detectron2 compared to YOLOv8 may reflect its strength in achieving high overall detection accuracy. However, YOLOv8's superior performance in specific cases, such as damage type detection, suggests that its single-stage architecture might offer advantages in certain detection scenarios. The observed differences highlight the trade-offs between model architectures and underscore the importance of selecting the appropriate model based on the specific requirements of the detection task.

7 Implementation Details

This section details the approaches employed for processing videos using multiple deep learning models for earthquake damage detection. It also describes the integration of OpenVINO (Open Visual Inference and Neural Network Optimization toolkit) to optimize the model size and processing efficiency.

Extracting information from videos often necessitates processing each frame individually (Schultz & Stevenson, 1996). When multiple deep learning models were used for video analysis, two main strategies were used to ensure that they worked simultaneously. Frame splitting and threading. Frame splitting involves dividing each video frame into smaller segments and assigning these segments to different models for processing. This allows for parallelization, where each model can analyze its assigned frame sets concurrently. For instance, a video can be segmented into three parts, with each model tasked with analyzing a specific set of frames. While this approach offers a relatively straightforward implementation and enables faster processing compared to sequential analysis, it has limitations. The degree of parallelization is restricted by the number of models and the number of frames per segment. Additionally, splitting frames can lead to a loss of information between consecutive frames, potentially impacting the accuracy of tasks that rely on temporal continuity in the video, such as tracking moving objects or analyzing changes over time.

Fig. 9 illustrates the concept of single-threaded versus multi-threaded processes.

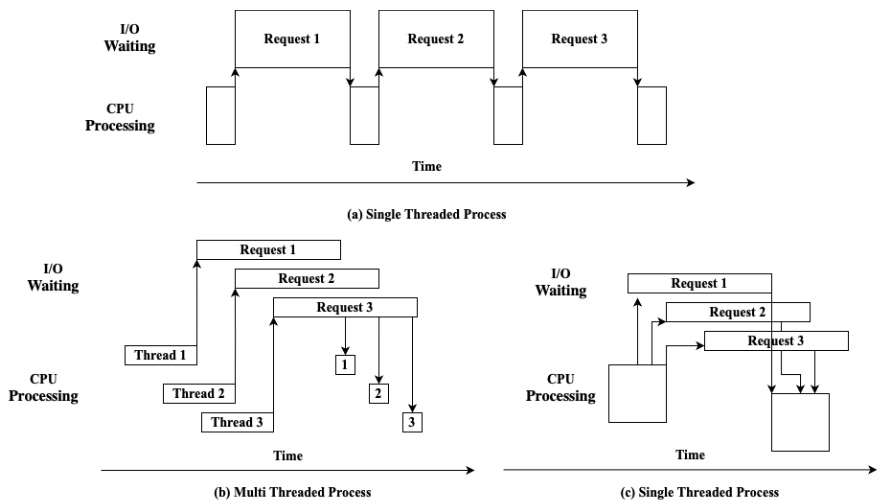

Fig. 9 The processes of single and multi-thread models

The second of these used threading. Threading offers an alternative approach that leverages the concept of threads within a computer program (Nguyen, 2018). A thread is a lightweight unit of execution that allows a program to execute multiple instructions concurrently, improving processing efficiency. In the context of model video processing, each deep learning model can be wrapped in a separate thread. These threads then execute concurrently, processing different video frames simultaneously. This approach offers a higher degree of parallelization compared to frame splitting, as the number of threads is not limited by the number of frames per segment. Threading can significantly reduce processing time by effectively utilizing multiple CPU cores or hardware threads. However, implementing multi-threaded video processing can be more intricate than frame splitting. Careful synchronization mechanisms are required to avoid race conditions and ensure data consistency between threads. Additionally, creating and managing threads introduces some overhead, which might negate the benefits for short videos or models with low processing demands.

The choice between these two approaches depends on the specific requirements of the application. Frame splitting might be suitable for simpler scenarios with a limited number of models while threading offers greater potential for efficiency when dealing with numerous models or processing large videos. Real-world deployment of deep learning models often necessitates efficient processing on resource-constrained platforms (Kamath & Renuka, 2023). To address this challenge, we employed OpenVINO. OpenVINO provides a comprehensive suite of tools designed for optimizing and deploying deep learning models for efficient inference (Demidovskij et al., 2020). OpenVINO offers tools for optimizing fine-tuned model by reducing their size and computational complexity. This can be achieved through techniques like quantization, which reduces the precision of model weights from floating-point to lower-bit integer representations. Quantization leads to smaller model footprints and faster inference on hardware platforms with limited resources, such as drones or edge computing hardware. OpenVINO facilitates the deployment of optimized models on various hardware platforms, including CPUs, GPUs, and specialized inference accelerators. This allows for efficient execution of the models on resource-constrained devices, which might be deployed for real-time earthquake damage assessment.

Fig. 10 depicts the inference time per image for various YOLOv8 models (YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, YOLOv8x) in different formats (PyTorch, TorchScript, ONNX, OpenVINO) running on a central processing unit (CPU). Smaller model sizes and optimized formats like OpenVINO generally lead to faster inference times.

In the context of real-time earthquake damage assessment, applying lightweight and model compression techniques is indeed crucial for enhancing the performance of the trained models. OpenVINO's optimization techniques, such as quantization, were employed to refine the trained models by reducing their size and computational complexity. This approach allows for significant improvements in inference speed while maintaining an acceptable level of accuracy. The results, as shown in Fig. 10, demonstrate that the optimized models, particularly those converted to the OpenVINO format, achieved faster detection speeds compared to their original versions without sacrificing detection accuracy. The inference times per image were notably reduced, which is essential for real-time assessment scenarios where rapid processing is critical. Thus, the application of lightweight methodologies and model compression techniques, facilitated by tools like OpenVINO, was not only appropriate but necessary to meet the demands of real-time deployment in resource-constrained environments.

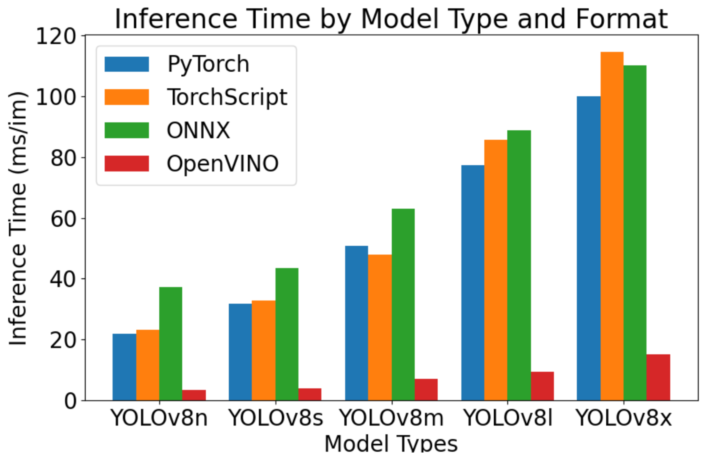

Fig. 10 The inference time per image by model size and format (Core CPU)

By integrating OpenVINO, we aimed to strike a balance between model accuracy and processing efficiency. While reducing model size through optimization techniques might lead to slight degradation in performance metrics, the gains in processing speed and reduced resource requirements can be crucial for real-world applications, particularly in scenarios where rapid damage assessment is essential. Additionally, this approach allows the models to be deployed on a wider range of hardware platforms, including those with limited computational resources, thereby enhancing the practicality and scalability of the system in real-world disaster response scenarios.

8 Discussion

This study investigated the effectiveness of deep learning models for earthquake damage detection using drone imagery. We extended the capabilities of pre-trained models like VGG16, traditionally used for image classification, by adapting them to the object detection problem through transfer learning techniques. This allowed us to leverage the powerful feature extraction capabilities of these models for our specific task. We then compared the performance of three models: YOLOv8, Detectron2 (using a Faster R-CNN R50-FPN backbone), and an adapted VGG16 architecture. Various evaluation metrics were employed to assess their accuracy in identifying damaged buildings. Our findings offer valuable insights and pave the way for future advancements in this domain.

The results demonstrate the superior performance of YOLOv8 for earthquake damage detection. YOLOv8 achieved consistently higher scores across all evaluation metrics, including mAP, mAP50, and recall. Specifically, YOLOv8 Nano outperformed Detectron2 with a mAP50 of 92.1% compared to 48.83% and a recall of 82.1% compared to no available recall data in the Damage Type Models. In the Damage Level Models, YOLOv8x achieved a mAP50 of 68.9% compared to 55.44% and a recall of 63.3%. YOLOv8 Nano also excelled in the Wall Damage Type Models with a mAP50 of 89.4% versus 92.01% for Detectron2, but it demonstrated a higher recall at 83.7% and better performance in detecting vertical cracks with an AP of 96.5%. Notably, its advantage in mAP50 highlights its stronger capability in detecting damaged buildings even with moderate bounding box overlap, a crucial factor for real-world applications. These findings suggest that YOLOv8, a single-stage object detection model, might be well-suited for this task due to its potential for both accuracy and efficiency. This study contributes significantly to the field of earthquake damage assessment by demonstrating the feasibility of using deep learning models, including adapted architectures like VGG16, for the detection of damaged structures from drone imagery. Rapid and accurate damage assessment is essential for effective disaster response, and our findings suggest that drone-based deep learning models hold promise for streamlining this process.

To enhance the feasibility of real-world deployment, we explored two primary strategies for enabling the simultaneous operation of multiple deep learning models for video processing: frame splitting and threading. Frame splitting divides video frames into segments for processing by different models while threading leverages multiple threads to execute models concurrently on the same video. The choice between these approaches depends on the specific requirements and the number of models involved. Furthermore, we employed OpenVINO to optimize the size and computational complexity of the deep learning models. This optimization process, often involving techniques like quantization, reduces the resource requirements of the models while maintaining acceptable accuracy. This optimization is crucial for enabling real-time processing on resource-constrained platforms such as drones.

9 Future Works

While our study yielded promising results, it is important to acknowledge limitations and areas for future exploration. Firstly, the training dataset employed in this research can be further expanded to encompass a wider variation in damage types, lighting conditions, and geographical locations. Additionally, exploring alternative deep learning architectures beyond YOLOv8, Detectron2, and adapted VGG16 could potentially lead to even more accurate and robust models. Furthermore, this study focused solely on object detection. Future research could investigate incorporating additional image analysis techniques, such as semantic segmentation, to provide more detailed information about the nature and extent of damage. Finally, real-world deployment necessitates considerations beyond model performance. Future efforts should explore integration with drone platforms, address latency constraints for real-time processing, and ensure robustness to variations in weather conditions during drone operation.

Future efforts should focus on developing and sharing large-scale, high-resolution, and well-annotated datasets specifically designed for earthquake damage detection using drone imagery. This would facilitate the training of more robust and generalizable models. Future research could explore incorporating domain knowledge specific to earthquake damage assessment into the deep learning models. This might involve integrating additional data modalities like LiDAR sensors alongside the imagery or developing specialized loss functions that prioritize the detection of specific damage patterns. While this study focused on model performance using evaluation metrics, real-world deployment of these models necessitates further considerations.Factors like ease of integration with drone platforms, latency constraints for real-time processing, and robustness to variations in lighting and weather conditions during drone operation need to be addressed for successful real-world implementation. While deep learning models offer promising advancements in damage detection, the critical nature of post-earthquake assessment tasks suggests the potential value of human-in-the-loop systems. Future research could explore methods for integrating these models with human expertise to leverage the strengths of both for improved decision-making in real-world scenarios.

10 Conclusion

Demonstrating the effectiveness of deep learning models, particularly YOLOv8, for damage detection from drone imagery, including models adapted from architectures like VGG16. Providing insights into the evaluation metrics crucial for assessing model performance in this specific task. Highlighting the potential of drone-based deep learning solutions for rapid and accurate earthquake damage assessment, along with methods for enabling model processing and model optimization for real-world feasibility. These findings can be applied in various disaster management scenarios. Real-time damage assessment facilitated by drone-based deep learning models can provide crucial information to emergency responders, enabling them to prioritize rescue efforts, allocate resources efficiently, and expedite recovery operations in the aftermath of earthquakes.

In conclusion, this study compared the performance of YOLOv8 and Detectron2 models for earthquake damage detection using drone imagery. The results demonstrate the effectiveness of YOLOv8 in this task, achieving superior performance across various evaluation metrics. The insights gleaned from this comparison pave the way for future research directions, including exploring efficiency-accuracy trade-offs, investigating alternative architectures, incorporating domain knowledge, and addressing real-world deployment considerations. By fostering further research and development in these areas, can contribute to the advancement of robust and practical drone-based solutions for rapid and accurate earthquake damage assessment. By addressing the limitations identified in this study and pursuing further research directions, this research can contribute to the development of robust and practical drone-based solutions for rapid and accurate earthquake damage assessment, ultimately improving disaster response capabilities and saving lives.

Acknowledgements

We extend our sincere gratitude to Engineer Research and Development Center for their generous support and funding.

References

Abdi, G., & Jabari, S. (2021). A multi-feature fusion using deep transfer learning for earthquake building damage detection. *Canadian journal of remote sensing*, *47*(2), 337–352.

Abri, S., Abri, R., Yarıcı, A., & Çetin, S. (2020). Multi-thread frame tiling model in concurrent real-time object detection for resources optimization in yolov3. *Proceedings of the 2020 6th International Conference on Computer and Technology Applications*, 69–73.

Adams, B. J. (2004). Improved disaster management through post-earthquake building damage assessment using multitemporal satellite imagery. *ISPRS XXth Congress*.

Adriano, B., Yokoya, N., Xia, J., Miura, H., Liu, W., Matsuoka, M., & Koshimura, S. (2021). Learning from multimodal and multitemporal earth observation data for building damage mapping. *ISPRS Journal of Photogrammetry and Remote Sensing*, *175*, 132–143.

Ali, A. A., Katta, R., Jasek, R., Chramco, B., & Krayem, S. (2022). Covid-19 detection from chest x-ray images using detectron2 and faster r-cnn. In *Springer computational methods in systems and software* (pp. 37–53).

Ayan, E., & Ünver, H. M. (2018). Data augmentation importance for classification of skin lesions via deep learning. *IEEE Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT)*, 1–4.

Bhowmick, S., Nagarajaiah, S., & Veeraraghavan, A. (2020). Vision and deep learning-based algorithms to detect and quantify cracks on concrete surfaces from uav videos. *Sensors*, *20*(21), 6299.

Dell'Acqua, F., & Gamba, P. (2012). Remote sensing and earthquake damage assessment: Experiences, limits, and perspectives. *Proceedings of the IEEE*, *100*(10), 2876–2890.

Demidovskij, A., Tugaryov, A., Suvorov, A., Tarkan, Y., Fatekhov, M., Salnikov, I., Kashchikhin, A., Golubenko, V., Dedyukhina, G., Alborova, A., et al. (2020). Openvino deep learning workbench: A platform for model optimization, analysis and deployment. *IEEE international conference on tools with artificial intelligence (ICTAI)*, 661–668.

Fernandez Galarreta, J., Kerle, N., & Gerke, M. (2015). Uav-based urban structural damage assessment using object-based image analysis and semantic reasoning. *Natural hazards and earth system sciences*, *15*(6), 1087–1101.

Fujita, A., Sakurada, K., Imaizumi, T., Ito, R., Hikosaka, S., & Nakamura, R. (2017). Damage detection from aerial images via convolutional neural networks. *IEEE IAPR international conference on machine vision applications (MVA)*, 5–8.

Gerke, M., & Kerle, N. (2011). Automatic structural seismic damage assessment with airborne oblique pictometry© imagery. *Photogrammetric Engineering & Remote Sensing*, *77*(9), 885–898.

Gong, L., Li, Q., & Zhang, J. (2013). Earthquake building damage detection with object-oriented change detection. *IEEE International Geoscience and Remote Sensing Symposium-IGARSS*, 3674–3677.

Hasanlou, M., Shah-Hosseini, R., Seydi, S. T., Karimzadeh, S., & Matsuoka, M. (2021). Earthquake damage region detection by multitemporal coherence map analysis of radar and multispectral imagery. *Remote Sensing*, *13*(6), 1195.

Henderson, P., & Ferrari, V. (2017). End-to-end training of object class detectors for mean average precision. *13th Springer Asian Conference on Computer Vision*, 198–213.

Hong, Z., Zhong, H., Pan, H., Liu, J., Zhou, R., Zhang, Y., Han, Y., Wang, J., Yang, S., & Zhong, C. (2022). Classification of building damage using a novel convolutional neural network based on post-disaster aerial images. *Sensors*, *22*(15), 5920.

Hoskere, V., Narazaki, Y., & Spencer Jr, B. F. (2022). Physics-based graphics models in 3d synthetic environments as autonomous vision-based inspection testbeds. *Sensors*, *22*(2), 532.

Isa, I. S., Rosli, M. S. A., Yusof, U. K., Maruzuki, M. I. F., & Sulaiman, S. N. (2022). Optimizing the hyperparameter tuning of yolov5 for underwater detection. *IEEE Access*, *10*, 52818–52831.

Joshi, A. R., Tarte, I., Suresh, S., & Koolagudi, S. G. (2017). Damage identification and assessment using image processing on post-disaster satellite imagery. *IEEE Global Humanitarian Technology Conference (GHTC).*

Kalantar, B., Ueda, N., Al-Najjar, H. A., & Halin, A. A. (2020). Assessment of convolutional neural network architectures for earthquake-induced building damage detection based on pre-and post-event orthophoto images. *Remote Sensing*, *12*(21), 3529.

Kamath, V., & Renuka, A. (2023). Deep learning based object detection for resource constrained devices: Systematic review, future trends and challenges ahead. *Elsevier Neurocomputing*, *531*, 34–60.

Khodaverdi zahraee, N., & Rastiveis, H. (2017). Object-oriented analysis of satellite images using artificial neural networks for post-earthquake buildings change detection. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *42*, 139–144.

Kim, R., Kim, G., Kim, H., Yoon, G., & Yoo, H. (2020). A method for optimizing deep learning object detection in edge computing. *International Conference on Information and Communication Technology Convergence (ICTC).*

Kim, S. J., Holub, C. J., & Elnashai, A. S. (2011). Experimental investigation of the behavior of rc bridge piers subjected to horizontal and vertical earthquake motion. *Engineering Structures*, *33*(7), 2221–2235.

Lee, J., & Hwang, K. I. (2022). Yolo with adaptive frame control for real-time object detection applications. *Multimedia Tools and Applications*, *81*(25), 36375–36396.

Li, Y., Fan, Q., Huang, H., Han, Z., & Gu, Q. (2023). A modified yolov8 detection network for uav aerial image recognition. *MDPI Drones*, *7*(5), 304.

Liu, C., Sepasgozar, S. M., Zhang, Q., & Ge, L. (2022). A novel attention-based deep learning method for post-disaster building damage classification. *Expert Systems with Applications*, *202*, 117268.

Liu, Y., Gross, L., Li, Z., Li, X., Fan, X., & Qi, W. (2019). Automatic building extraction on high-resolution remote sensing imagery using deep convolutional encoder-decoder with spatial pyramid pooling. *IEEE Access*, *7*, 128774–128786.

Lou, H., Duan, X., Guo, J., Liu, H., Gu, J., Bi, L., & Chen, H. (2023). Dc-yolov8: Small-size object detection algorithm based on camera sensor. *MDPI Electronics*, *12*(10), 2323.

Mangalathu, S., Sun, H., Nweke, C. C., Yi, Z., & Burton, H. V. (2020). Classifying earthquake damage to buildings using machine learning. *Earthquake Spectra*, *36*(1), 183–208.

Naito, S., Tomozawa, H., Mori, Y., Nagata, T., Monma, N., Nakamura, H., Fujiwara, H., & Shoji, G. (2020). Building-damage detection method based on machine learning utilizing aerial photographs of the kumamoto earthquake. *Earthquake Spectra*, *36*(3), 1166–1187.

Nguyen, Q. (2018). *Mastering concurrency in python: Create faster programs using concurrency, asynchronous, multithreading, and parallel programming*. Packt Publishing Ltd.

Park, H.-J., Kang, J.-W., & Kim, B.-G. (2023). Ssfpn: Scale sequence (s 2) feature-based feature pyramid network for object detection. *MDPI Sensors*, *23*(9), 4432.

Pi, Y., Nath, N. D., & Behzadan, A. H. (2021). Detection and semantic segmentation of disaster damage in uav footage. *American Society of Civil Engineers Journal of Computing in Civil Engineering*, *35*(2), 04020063.

Ramzi, E., Thome, N., Rambour, C., Audebert, N., & Bitot, X. (2021). Robust and decomposable average precision for image retrieval. *Advances in Neural Information Processing Systems*, *34*, 23569–23581.

Rao, A., Jung, J., Silva, V., Molinario, G., & Yun, S. H. (2023). Earthquake building damage detection based on synthetic-aperture-radar imagery and machine learning. *Natural Hazards and Earth System Sciences*, *23*(2), 789–807.

Reis, D., Kupec, J., Hong, J., & Daoudi, A. (2023). Real-time flying object detection with yolov8. *arXiv preprint arXiv:2305.09972*.

Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS One*, *10*(3), e0118432.

Salvi, M., Acharya, U. R., Molinari, F., & Meiburger, K. M. (2021). The impact of pre- and post-image processing techniques on deep learning frameworks: A comprehensive review for digital pathology image analysis. *Elsevier Computers in Biology and Medicine*, *128*, 104129.

Schultz, R. R., & Stevenson, R. L. (1996). Extraction of high-resolution frames from video sequences. *IEEE transactions on image processing*, *5*(6), 996–1011.

Turker, M., & Sumer, E. (2008). Building-based damage detection due to earthquake using the watershed segmentation of the post-event aerial images. *International Journal of Remote Sensing*, *29*(11), 3073–3089.

Ünlü, R., & Kiriş, R. (2022). Detection of damaged buildings after an earthquake with convolutional neural networks in conjunction with image segmentation. *The Visual Computer*, *38*(2), 685–694.

Vocaturo, E., Zumpano, E., & Veltri, P. (2018). Image pre-processing in computer vision systems for melanoma detection. *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2117–2124.

Wang, Y., Jing, X., Cui, L., Zhang, C., Xu, Y., Yuan, J., & Zhang, Q. (2023). Geometric consistency enhanced deep convolutional encoder-decoder for urban seismic damage assessment by uav images. *Engineering Structures*, *286*, 116132.

Wu, L., Xu, M., Wang, J., & Perry, S. (2019). Recall what you see continually using gridlstm in image captioning. *IEEE Transactions on Multimedia*, *22*(3), 808–818.

Xiong, C., Li, Q., & Lu, X. (2020). Automated regional seismic damage assessment of buildings using an unmanned aerial vehicle and a convolutional neural network. *Automation in Construction*, *109*, 102994.

Zhang, J., & Gong, L. (2013). Sar images before and after earthquake change detection based on object oriented method and damage evaluation. *IEEE International Geoscience and Remote Sensing Symposium-IGARSS*.

Zhang, X., Feng, X., Xiao, P., He, G., & Zhu, L. (2015). Segmentation quality evaluation using region-based precision and recall measures for remote sensing images. *Elsevier ISPRS Journal of Photogrammetry and Remote Sensing*, *102*, 73–84.

Znamenskaya, I., Doroshchenko, I., & Sysoev, N. (2023). Computer vision study of the flow generated by a sliding discharge. *Computer*, *15*(4), 1–11.

Furkan Kizilay graduated from Manisa Celal Bayar University with a degree in Computer Engineering. He specialized in AI, focusing on data science, NLP, and computer vision. As president of the IEEE Computer Society, he led various projects and activities. Furkan started his career as a Python Instructor at a British company. He completed internships in data science and computer vision, and later worked as a research assistant at Manisa Celal Bayar University. Currently, he is a Data Scientist in R&D, working on machine learning, NLP, and computer vision projects. He has several publications on transformer-based models, LLMs, and RAG systems.

Mina R. Narma n is a student at Acıbadem Mehmet Ali Aydınlar University, pursuing a degree in Computer Engineering. She has developed a wide array of academic interests, particularly in cutting-edge areas such as machine learning, natural language processing, and robotics. Her focus also includes programming languages, scientific computing, and security. Additionally, Mina is drawn to systems and networking, the theory of computation, and computer vision, reflecting her dedication to both theoretical and applied aspects of computer science. With a strong foundation in these fields, she aims to contribute to advancements in technology that address complex real-world challenges.

Hwapyeong Song is a PhD student studying computer science in Engineering at Marshall University. Song received the B.S. in Computer Science and M.S. in Cybersecurity from Marshall University. Her research interests include computer vision, image processing, machine learning, software architecture, authentication algorithm, and vulnerability analysis. She has been involved in interdisciplinary research, including a notable project on machine learning applications in civil engineering and bioinformatics. Her current research focuses on developing advanced algorithms and models to address complex real-world problems across various domains.

Husnu S. Narman is an Associate Professor in the Department of Computer Sciences and Electrical Engineering at Marshall University. He earned his Ph.D. in Computer Science from the University of Oklahoma and completed a post-doctorate fellowship at Clemson University. His research interests include distributed computing, cloud and edge computing, the Internet of Things, cyber-physical systems, and machine learning applications. He has organized numerous computer science-themed summer camps to increase awareness of computer science and cybersecurity. Dr. Narman is the recipient of several awards, including the outstanding Ph.D Student award, the 2021-2022 College of Engineering and Computer Sciences Weisberg Service Award and the 2020-2021 Marshall University Distinguished Artists and Scholars Junior Category Award.

Cumhur Cosgun is currently working at the Pennsylvania Department of Transportation as a Senior Civil Engineer - Research Project Manager. He previously held the position of Director of Labs for Education at NJIT and worked as a Structural Engineering Lab Manager at Marshall University. He earned his Ph.D. in Civil Engineering from İstanbul Technical University in 2014 and holds MS and BS degrees from İstanbul Kultur University. Additionally, he obtained a second MS degree in Information Systems Engineering and Management from Harrisburg University. He has served as an Assistant Professor at Istanbul Cerrahpasa University and a Visiting Professor at Wichita State University. His research interests include structural analysis, reinforced concrete design, performance-based structural analysis, experimental studies, and machine learning in structural engineering.

Ammar Alzarrad is an Assistant Professor in the Department of Civil Engineering at Marshall University. He graduated with dual bachelor's degrees in Civil Engineering and Business Administration from the University of South Alabama. He received his M.Sc. and Ph.D. in Civil Engineering from The University of Alabama. Before assuming his current position, he was an Assistant Professor in the Department of Civil Engineering and Construction at Bradley University. Prior to joining academia, Dr. Alzarrad was a Virtual Design and Construction (VDC) manager at an engineering design firm in Chicago, where he managed multi-million projects (i.e., Wrigley Field restoration and expansion project). Dr. Alzarrad is a PMP©, CPEM©, and the Director of The Engineering Management Graduate Program at Marshall University