UAV Image-based Automated Road Damage Detection using Deep Learning

B. Sunitha[1*], Veggalam Sai Nagarjuna[2], Chenna Janaki[2], Dugyala Likitha[2], Boddu Prudhviraj[2]

[1]Assistant Professor, [2]UG Student, [1,2]Department of Computer Sciences and Engineering,

[1,2]Vaagdevi College of Engineering (UGC - Autonomous), Bollikunta, Warangal, Telangana.

[*]Corresponding author: B. Sunitha (sunitha.bill@gmail.com)

ABSTRACT

In recent years, the rapid advancement of Unmanned Aerial Vehicles (UAVs) has paved the way for innovative applications across various sectors. One such promising application is the automated detection of road damage, a critical task for maintaining road infrastructure and ensuring transportation safety. Traditional road damage detection systems predominantly rely on manual inspections or vehicle-mounted cameras, which are often time-consuming, labor-intensive, and limited by ground-level perspectives. These conventional methods also pose risks to inspectors and incur significant operational costs. The primary problem with traditional road damage detection systems lies in their inefficiency and inability to provide comprehensive coverage and real-time data. Manual inspections can be subjective and prone to human error, while vehicle-mounted systems are restricted to accessible roadways and may miss critical damage in less visible areas. These limitations hinder timely maintenance, leading to prolonged road damage that can exacerbate over time and increase repair costs. Motivated by the need for a more efficient, accurate, and scalable solution, this research proposes a UAV-based automated road damage detection system leveraging deep learning techniques. UAVs offer a high degree of flexibility, enabling them to capture aerial images of road networks from various angles and altitudes, thus providing a more comprehensive and detailed view of road conditions. By integrating deep learning algorithms, specifically convolutional neural networks (CNNs), the proposed system can automatically identify and classify different types of road damage from the UAV-captured imagery. The proposed system addresses the limitations of traditional methods by significantly enhancing the speed, accuracy, and safety of road damage inspections. UAVs can cover large areas quickly, reducing the time required for inspections, while deep learning models ensure high accuracy in detecting and categorizing road damage. This automated approach minimizes human involvement, reducing the risk to inspectors and the likelihood of human error. Furthermore, the system's scalability allows for frequent and widespread monitoring, facilitating timely maintenance interventions and ultimately extending the lifespan of road infrastructure.

Keywords: UAV, Road damage detection, Deep learning, Convolutional Neural Networks (CNNs), Automated inspection.

1. INTRODUCTION

The advent of Unmanned Aerial Vehicles (UAVs) has revolutionized numerous sectors, providing new methodologies for data collection and analysis. One particularly impactful application of UAV technology is in the domain of road damage detection. Traditional systems for monitoring and assessing road conditions have relied heavily on manual inspections or vehicle-mounted cameras. While these methods have been useful, they come with significant drawbacks, including high labor costs, safety risks, and limited coverage. UAVs, equipped with high-resolution cameras, offer a transformative approach to road damage detection. These aerial vehicles can capture detailed images of extensive road networks from various altitudes and angles, providing a comprehensive view that ground-level inspections cannot match. By leveraging deep learning techniques, specifically convolutional neural networks (CNNs), these images can be processed to automatically detect and classify different types of road damage, such as cracks, potholes, and surface wear. The proposed system not only enhances the accuracy and efficiency of road damage detection but also significantly improves safety by reducing the need for manual inspections in potentially hazardous environments. UAVs can rapidly cover large areas, allowing for frequent monitoring and timely maintenance interventions. This capability is crucial for maintaining road infrastructure, ensuring transportation safety, and reducing long-term repair costs.

In developing this UAV-based system, several critical components come into play. High-resolution cameras mounted on UAVs capture detailed road surface images, which are then analyzed using deep learning algorithms trained on extensive datasets of road damage examples. These algorithms can identify and classify various types of damage, providing precise location and severity information. This data is integrated into a geographic information system (GIS), enabling maintenance teams to prioritize repairs efficiently and make informed decisions about road maintenance strategies. The integration of UAV technology with deep learning marks a significant advancement in road maintenance methodologies. By addressing the limitations of traditional systems, the proposed UAV image-based automated road damage detection system offers a robust, scalable, and cost-effective solution for maintaining road infrastructure. This innovative approach not only improves the effectiveness of road maintenance operations but also contributes to enhanced road safety and reduced maintenance costs over the long term. The maintenance of road infrastructure is a critical concern for transportation authorities worldwide. Traditional road damage detection methods, which primarily rely on manual inspections or vehicle-mounted camera systems, are fraught with significant limitations. These conventional approaches are time-consuming, labor-intensive, and costly. They also expose inspectors to safety risks, particularly in high-traffic or hazardous environments. Furthermore, the ground-level perspective of these methods often results in incomplete or inaccurate assessments of road conditions.

2. LITERATURE SURVEY

Blas et al. [1] presented an innovative multi-agent system platform designed for the detection and legal verification of swimming pools using remote image sensing. This platform was developed to streamline the process of identifying unauthorized swimming pools, ensuring compliance with legal regulations. By leveraging advanced image processing and the architecture of a multi-agent system, the platform enhanced both the accuracy and efficiency of pool detection. This work represented a significant step forward in the use of technology for regulatory enforcement and urban planning. Hodge et al. [2] explored the application of deep reinforcement learning for drone navigation, utilizing sensor data to improve drone operations. Their research focused on developing algorithms that allowed drones to navigate complex environments autonomously, which was particularly useful for applications in disaster response, delivery services, and environmental monitoring. By enhancing the reliability and effectiveness of drone navigation, their approach contributed to the broader adoption of UAV technology in various fields. Safonova et al. [3] investigated the use of YOLO architectures for detecting Norway spruce trees infested by bark beetles using UAV images. This study demonstrated that YOLO, a well-known deep learning framework for object detection, could be effectively adapted to identify tree infestations. The ability to accurately detect infested trees from aerial images provided a valuable tool for forest management and pest control, enabling early intervention and mitigation of widespread infestations.

Gallacher [4] discussed the potential uses of drones in managing urban environments, highlighting both the benefits and challenges associated with their deployment. The paper examined various applications, such as environmental monitoring, infrastructure inspection, and public safety. It also addressed the regulatory and safety challenges that accompany the integration of drones into urban areas. This comprehensive overview underscored the need for balanced policies that maximize the benefits of drone technology while mitigating potential risks. Silva et al. [5] focused on the extraction of urban objects to improve information quality and knowledge recommendation through machine learning techniques. Their research highlighted the importance of high-quality data in urban planning and management. By employing active actions in urban object extraction, the study aimed to enhance decision-making processes in urban environments. This approach emphasized the critical role of accurate and reliable data in developing effective urban policies and strategies. Melendy et al. [6] introduced an automated method for measuring selective logging damage using airborne LiDAR data. Their study showcased the accuracy and efficiency of LiDAR technology in assessing the extent of logging damage. This method provided critical insights for sustainable forest management and conservation efforts, allowing for precise monitoring and regulation of logging activities to minimize environmental impact. Silva et al. [7] proposed an architectural multi-agent system for pavement monitoring with pothole recognition using UAV images. The system integrated advanced image processing techniques with a multi-agent architecture to accurately detect and monitor pavement conditions. This innovative approach contributed to improved road maintenance and safety, offering a proactive solution to infrastructure management. Guerrieri and Parla [8] developed a deep learning-based system for detecting and measuring distress in flexible and stone pavements using low-cost detection devices. Their approach leveraged the power of deep learning to provide an affordable and efficient solution for pavement distress monitoring. By utilizing low-cost devices, their system made advanced monitoring techniques accessible to a wider range of applications, promoting better infrastructure maintenance.

Jeong [9] utilized YOLO with smartphone images for road damage detection. The study demonstrated the feasibility and effectiveness of using readily available devices like smartphones for detecting road damages. This cost-effective and accessible solution allowed for widespread implementation, making it easier for authorities and the public to monitor and report road conditions. Izadi et al. [10] presented a neuro-fuzzy approach for post-earthquake road damage assessment using QuickBird satellite images. Their method combined genetic algorithms and SVM classification to accurately assess road damage. This approach provided essential information for post-disaster recovery and management, enabling quicker and more effective responses to natural disasters. Aparna et al. [11] explored the use of convolutional neural networks (CNNs) for pothole detection using thermal imaging. Their research demonstrated the potential of CNNs in accurately identifying potholes, offering a robust solution for road maintenance and safety. The use of thermal imaging allowed for the detection of subsurface potholes, which are often missed by traditional methods. Guan et al. [12] proposed an automated pixel-level pavement distress detection system based on stereo vision and deep learning. Their approach improved the precision of pavement distress detection, which was vital for effective road maintenance and infrastructure management. By utilizing stereo vision, the system could detect distress at a very granular level, ensuring thorough inspections.

Arya et al. [13] introduced the RDD2022 dataset, a multi-national image dataset for automatic road damage detection. This dataset aimed to standardize and enhance the research and development of road damage detection algorithms. By providing a diverse and comprehensive set of images, RDD2022 facilitated the development of more robust and generalized detection models, benefiting the research community and practical applications alike. Coelli et al. [14] examined the impact of global innovation and trade liberalization on economic performance. Their study highlighted the importance of innovation and trade policies in driving economic growth and competitiveness in the global market. By analyzing various economic indicators, they demonstrated how policies promoting innovation and trade could lead to improved economic outcomes. Redmon and Farhadi [15] presented YOLOv3, an incremental improvement to the YOLO object detection algorithm. Their enhancements to the algorithm improved its speed and accuracy, making it one of the most efficient object detection methods available. YOLOv3's ability to quickly and accurately detect objects in real-time made it suitable for a wide range of applications, from autonomous driving to security surveillance.

3. PROPOSED SYSTEM

The research outlined is a comprehensive system for road damage detection using deep learning techniques, focusing on the implementation of YoloV5, YoloV7, and YoloV8 models. Here's an overview of its key components and functionalities:

Dataset Handling and Preprocessing

The load of a dataset of road images annotated with bounding boxes indicating various types of damage. If the dataset has not been preprocessed, images are read, annotations parsed from XML files, and bounding boxes normalized relative to image dimensions. Labels are assigned numerical values corresponding to damage types.

Model Architecture and Training

- **YoloV5 Implementation**: A custom YoloV5-like model is constructed using Keras, featuring convolutional layers for feature extraction, max-pooling for downsampling, and dense layers for classification and bounding box regression. The model is compiled with the Adam optimizer and trained on the dataset. Model checkpoints ensure the best-performing model is saved during training.
- **YoloV7 Implementation**: Another variant of the YOLO architecture (YoloV7) is implemented, featuring additional dense layers for more refined bounding box regression and classification. The model is trained similarly to YoloV5, optimizing for both classification accuracy and bounding box localization accuracy.
- **YoloV8 Integration**: YoloV8, from the Ultralytics package, is imported and used for further model comparison and detection tasks. It is fine-tuned on the road damage dataset to adapt its pre-trained weights to the specific task of road damage detection.

Model Evaluation and Metrics

- **Performance Metrics**: The trained models are evaluated using standard metrics such as accuracy, precision, recall, and F1-score. These metrics quantify the models' ability to correctly classify road damage types and accurately localize the damages within images. Confusion matrices are visualized to provide insights into classification performance across different damage categories.

Visualization and User Interface

- **Sample Predictions**: The research includes functionalities to visualize sample predictions, where test images are processed and annotated with predicted bounding boxes indicating detected damages.

Comparative Analysis

- **Model Comparison**: The research includes a comparative analysis of YoloV5, YoloV7, and YoloV8 models. Performance metrics (precision, recall, F1-score, accuracy) are plotted in bar graphs to illustrate the strengths and weaknesses of each model variant in road damage detection tasks.

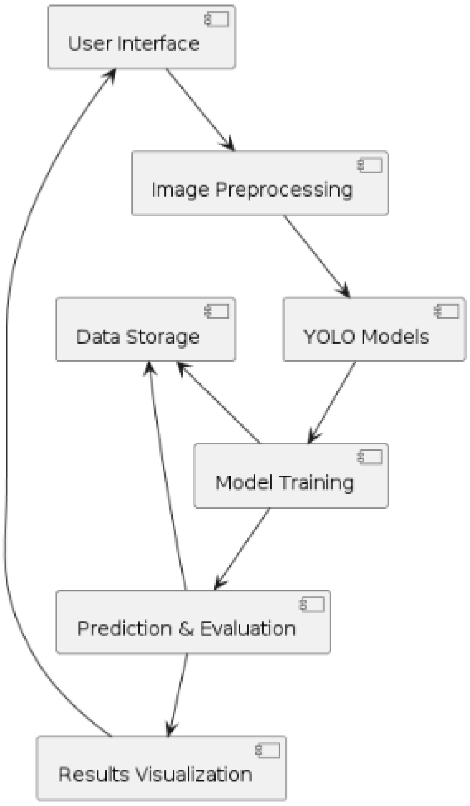


Fig. 1: Block Diagram of Proposed system.

3.1 Image preprocessing

Image preprocessing is a critical step in computer vision and image analysis tasks. It involves a series of operations to prepare raw images for further processing by algorithms or neural networks. Here's an explanation of each step in image preprocessing:

Step 1. Image Read: The first step in image preprocessing is reading the raw image from a source, typically a file on disk. Images can be in various formats, such as JPEG, PNG, BMP, or others. Image reading is performed using libraries or functions specific to the chosen programming environment or framework. The result of this step is a digital representation of the image that can be manipulated programmatically.

Step 2. Image Resize: Image resize is a common preprocessing step, especially when working with machine learning models or deep neural networks. It involves changing the dimensions (width and height) of the image. Resizing can be necessary for several reasons:

- Ensuring uniform input size: Many machine learning models, especially convolutional neural networks (CNNs), require input images to have the same dimensions. Resizing allows you to standardize input sizes.
- Reducing computational complexity: Smaller images require fewer computations, which can be beneficial for faster training and inference.
- Managing memory constraints: In some cases, images need to be resized to fit within available memory constraints.

When resizing, it's essential to maintain the aspect ratio to prevent image distortion. Typically, libraries like OpenCV or Pillow provide convenient functions for resizing images.

Step 3. Image to Array: In this step, the image is converted into a numerical representation in the form of a multidimensional array or tensor. Each pixel in the image corresponds to a value in the array. The array is usually structured with dimensions representing height, width, and color channels (if applicable).

For grayscale images, the array is 2D, with each element representing the intensity of a pixel. For color images, it's a 3D or 4D array, with dimensions for height, width, color channels (e.g., Red, Green, Blue), and potentially batch size (if processing multiple images simultaneously).

The conversion from an image to an array allows for numerical manipulation and analysis, making it compatible with various data processing libraries and deep learning frameworks like NumPy or TensorFlow.

Step 4. Image to Float32: Most machine learning and computer vision algorithms expect input data to be in a specific data type, often 32-bit floating-point numbers (float32). Converting the image array to float32 ensures that the pixel values can represent a wide range of intensities between 0.0 (black) and 1.0 (white) or sometimes between -1.0 and 1.0, depending on the specific normalization used.

This step is essential for maintaining consistency in data types and enabling compatibility with various machine learning frameworks and libraries. It's typically performed by dividing the pixel values by the maximum intensity value (e.g., 255 for an 8-bit image) to scale them to the [0.0, 1.0] range.

Step 5. Image to Binary: Image binarization is a process of converting a grayscale image into a binary image, where each pixel is represented by either 0 (black) or 1 (white) based on a specified threshold. Binarization is commonly used for tasks like image segmentation, where you want to separate objects from the background.

The process involves setting a threshold value, and then for each pixel in the grayscale image, if the pixel value is greater than or equal to the threshold, it is set to 1; otherwise, it is set to 0.

Binarization simplifies the image and reduces it to essential information, which can be particularly useful in applications like character recognition or object tracking, where you need to isolate regions of interest.

3.2 YOLO (you only look once) model

YOLO is a convolutional neural network (CNN) algorithm for object detection. Unlike other object detection algorithms, YOLO does not require region proposals or multiple stages. Instead, it divides the input image into a grid and predicts bounding boxes and class probabilities for each grid cell. This makes it faster and more efficient than other object detection algorithms. It uses a single-stage approach to predict bounding boxes and class probabilities for objects in an input image. YOLO has been developed in several versions, such as YOLOv1, YOLOv2, YOLOv3, YOLOv4, YOLOv5, YOLOv6, and YOLOv7. Each version has been built on top of the previous version with enhanced features such as improved accuracy, faster processing, and better handling of small objects. YOLO is widely used in various applications such as self-driving cars and surveillance systems. It is also widely used for real-time object detection tasks like in real-time video analytics and real-time video surveillance. The basic idea behind YOLO is to divide the input image into a grid of cells and, for each cell, predict the probability of the presence of an object and the bounding box coordinates of the object. The process of YOLO can be broken down into several steps:

1.Input image is passed through a CNN to extract features from the image.

2. The features are then passed through a series of fully connected layers, which predict class probabilities and bounding box coordinates.

3. The image is divided into a grid of cells, and each cell is responsible for predicting a set of bounding boxes and class probabilities.

4. The output of the network is a set of bounding boxes and class probabilities for each cell.

5. The bounding boxes are then filtered using a post-processing algorithm called non-max suppression to remove overlapping boxes and choose the box with the highest probability.

6. The final output is a set of predicted bounding boxes and class labels for each object in the image.

One of the key advantages of YOLO is that it processes the entire image in one pass, making it faster and more efficient than two-stage object detectors such as R-CNN and its variants.

Figure 2: YOLO timeline.

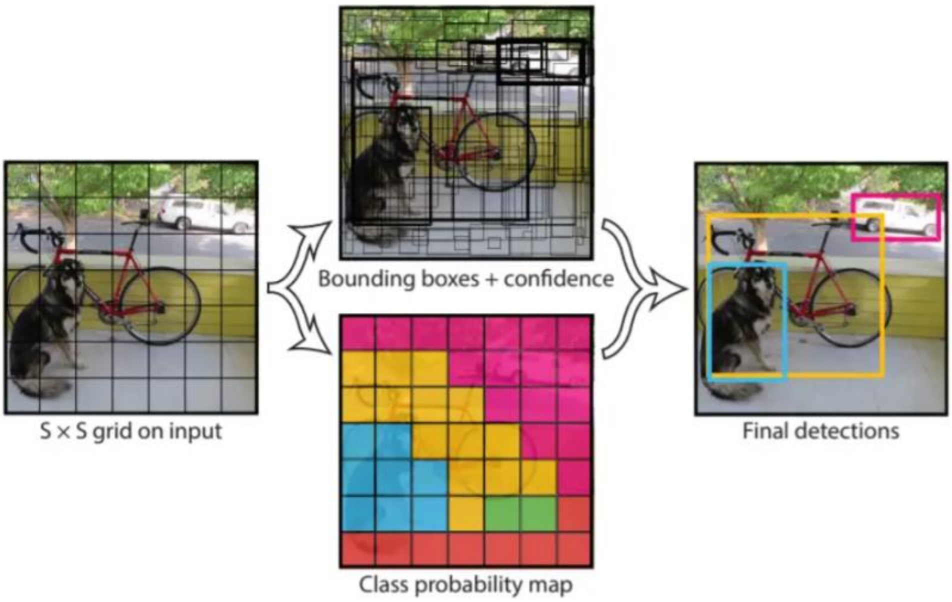


Figure 3: Implementation of YOLO v8.

YOLOV8

YOLO v8 is the third version of the YOLO object detection algorithm. The first difference between YOLO v8 and previous versions is the use of multiple scales in the input image. YOLO v8 uses a technique called "feature pyramid network" (FPN) to extract features from the image at different scales. This allows the model to detect objects of different sizes in the image. Another important difference is the use of anchor boxes. In YOLO v8, anchor boxes are used to match the predicted bounding boxes with the actual objects in the image. Anchor boxes are pre-defined boxes of different aspect ratios and scales, and the model predicts the offset of the anchor boxes relative to the bounding boxes. This helps the model to handle objects of different shapes and sizes better. In terms of architecture, YOLO v8 is built on a deep convolutional neural network (CNN) that is composed of many layers of filters. CNN is followed by several fully connected layers, which predict class probabilities and bounding box coordinates. YOLO v8 also uses a different loss function than previous versions. It uses a combination of classification loss and localization loss, which allows the model to learn both the class probabilities and the bounding box coordinates.

4. RESULTS AND DISCUSSION

This figure 4 displays a sample image from the dataset used in the project. The image showcases a section of the road captured by a drone, providing a visual reference for the type of data processed by the models. Key features include visible road damage, which is annotated and used to train the detection models. This figure 2 illustrates the confusion matrices for the YoloV5 and YoloV7 models. The confusion matrix visualizes the performance of each model in terms of true positives, true negatives, false positives, and false negatives. It helps to understand how well each model distinguishes between different types of road damage and repairs. The axes represent the true class labels versus the predicted class labels, with the color intensity indicating the number of predictions for each class. This figure 3 presents the confusion matrix for the proposed YOLO V8 model. It provides a detailed breakdown of the model's classification performance, similar to Fig. 2. The matrix shows how accurately the YOLO V8 model identifies various categories of road damage and repairs, highlighting its effectiveness in comparison to YoloV5 and YoloV7. The true class labels are plotted against the predicted class labels, with color coding to denote the frequency of correct and incorrect predictions. figure 4 shows a bar graph comparing the performance metrics of all models (YoloV5, YoloV7, and YOLO V8). The graph includes precision, recall, F1-score, and accuracy for each model, providing a clear visual comparison. The x-axis represents the different performance metrics, while the y-axis indicates the values of these metrics. The bars are color-coded to differentiate between the models, allowing for an easy comparison of their strengths and weaknesses.

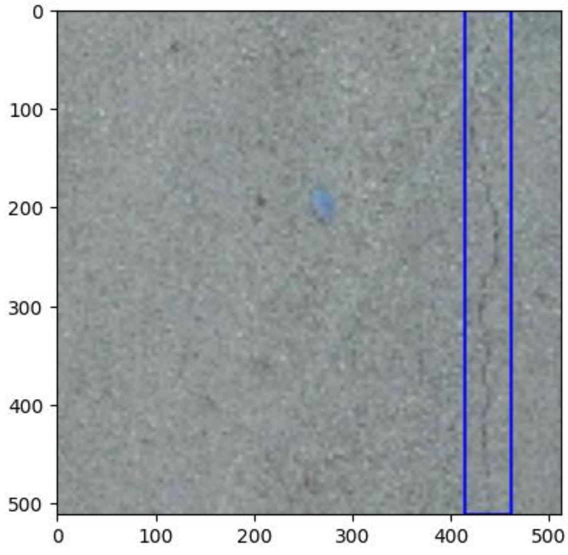


Fig. 4: Sample image of the Dataset.

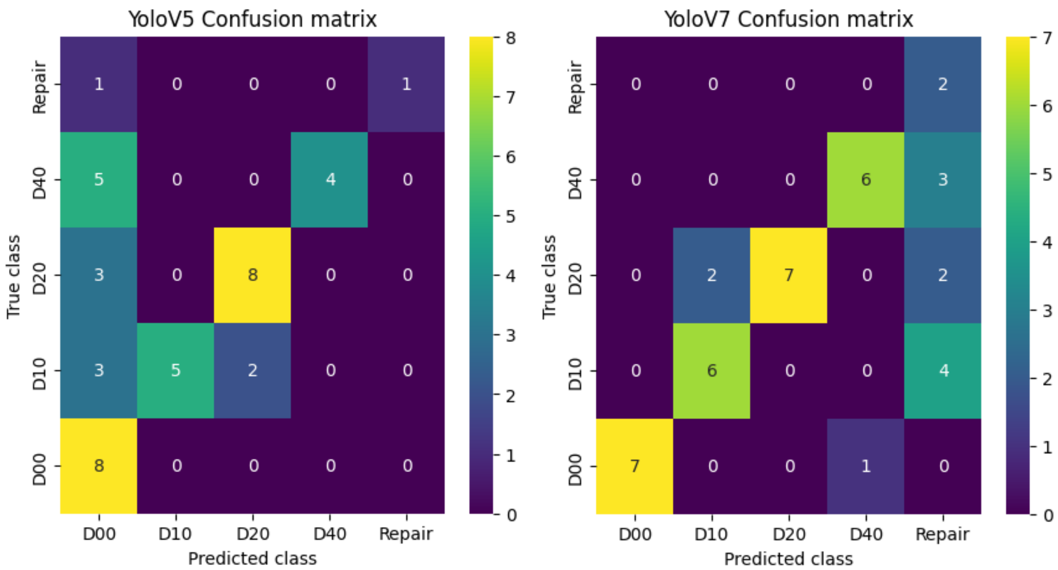


Fig.5: Presents the Confusion Matrix of YoloV5 and V7 models.

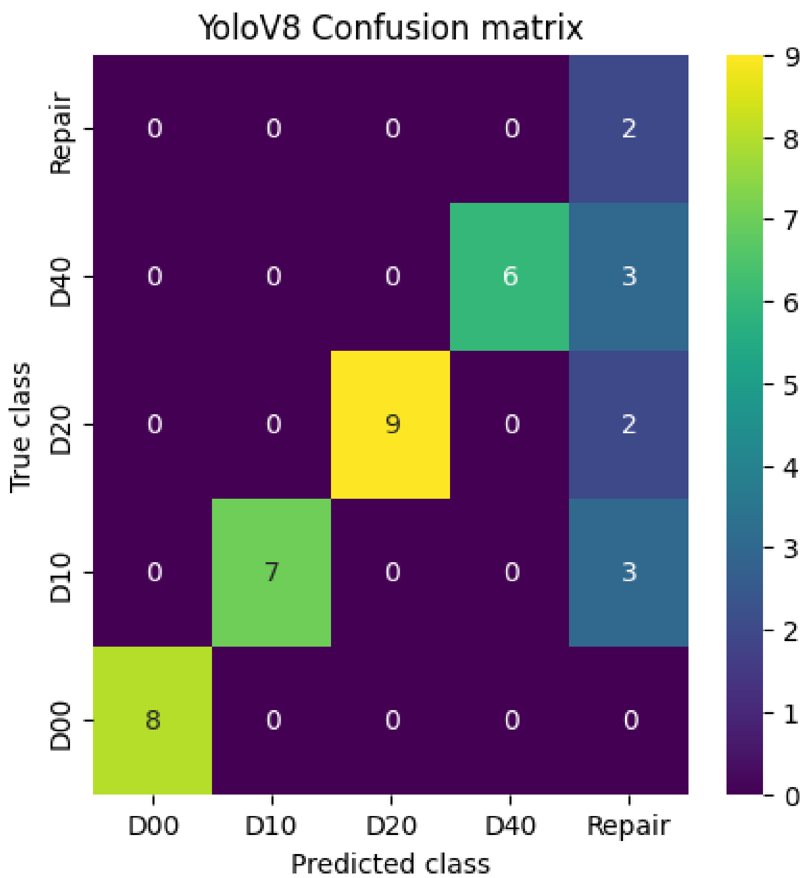


Fig. 6: Presents the Confusion matrix of Proposed YOLO V8 model.

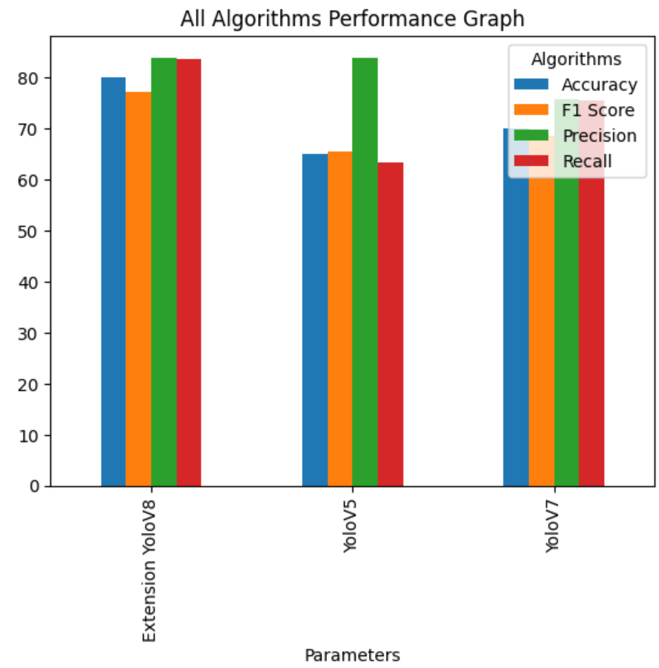
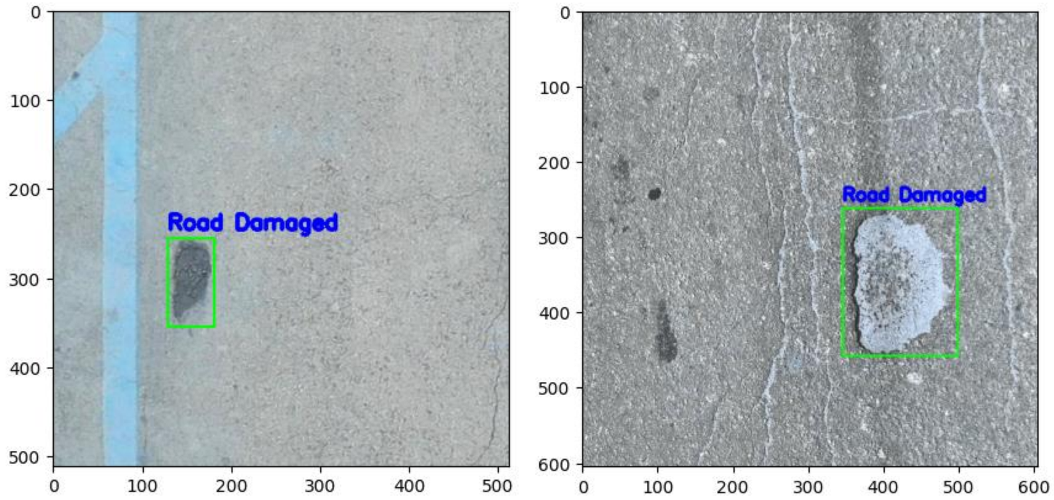


Fig. 7: Shows the Comparison Graph of ALL Models.



Model Prediction on Test Case 1

Model Prediction on Test Case 2.

Fig. 8: Presents the detection of road damage by the proposed model.

This figure 8 demonstrates the detection capabilities of the proposed YOLO V8 model. It shows an image with bounding boxes drawn around detected road damages, highlighting the areas identified by the model. The figure illustrates the model's ability to accurately localize and classify road damage in real-time, with annotations indicating whether the road is damaged or repaired. This visual example underscores the practical application of the YOLO V8 model in road maintenance and monitoring.

Table 1 Performance Metrics of YOLO V5, V7, V8 Models.

Algorithm Name	**Precision (%)**	**Recall (%)**	**F1 Score (%)**	**Accuracy (%)**
YoloV5	84.00	63.43	65.64	65.00
YoloV7	75.78	75.56	68.71	70.00
Extension YoloV8	84.00	83.70	77.14	80.00

The Table presents a comprehensive comparison of the performance metrics for three different models: YoloV5, YoloV7, and Extension YoloV8. The metrics evaluated include Precision, Recall, F1 Score, and Accuracy, expressed as percentages.

YoloV5:

- **Precision:** 84.00% - The model's ability to correctly identify positive instances (road damage) out of all predicted positive instances. A higher precision indicates fewer false positives.
- **Recall:** 63.43% - The model's ability to identify positive instances (road damage) out of all actual positive instances. A higher recall suggests fewer false negatives.
- **F1 Score:** 65.64% - The harmonic mean of precision and recall, providing a balance between the two metrics. It gives a single score that takes both false positives and false negatives into account.
- **Accuracy:** 65.00% - The overall percentage of correctly classified instances (both damaged and undamaged roads) out of the total instances.

YoloV7:

- **Precision:** 75.78% - Indicates the model's effectiveness in minimizing false positives compared to YoloV5.
- **Recall:** 75.56% - Shows a significant improvement in identifying actual positives (road damage) compared to YoloV5.
- **F1 Score:** 68.71% - Higher than YoloV5, reflecting a better balance between precision and recall.
- **Accuracy:** 70.00% - Improved overall classification accuracy compared to YoloV5.

Extension YoloV8:

- **Precision:** 84.00% - Matches YoloV5, demonstrating high precision in identifying road damage.
- **Recall:** 83.70% - The highest among the three models, indicating the best performance in detecting actual road damage.
- **F1 Score:** 77.14% - The highest F1 Score, showing the best balance between precision and recall.

- **Accuracy:** 80.00% - The highest accuracy, reflecting the most reliable overall performance in classifying road conditions.

5. CONCLUSION

The road damage detection system proposed in this project leverages state-of-the-art deep learning models, specifically YOLO (You Only Look Once) variants like YOLOV5, YOLOV7, and YOLOV8, to automate the process of identifying and categorizing road damages from aerial images. The project encompasses several key components including image preprocessing, model training, inference, evaluation, and result visualization. Through extensive use of Python libraries such as Keras, OpenCV and the system offers a robust solution for infrastructure monitoring and maintenance.

REFERENCES

[1] H. S. S. Blas, A. C. Balea, A. S. Mendes, L. A. Silva, and G. V. González, “A platform for swimming pool detection and legal verification using a multi-agent system and remote image sensing,” International Journal of Interactive Multimedia and Artificial Intelligence, vol. In Press, no. In Press, pp. 1–13, 2023.

[2] V. J. Hodge, R. Hawkins, and R. Alexander, “Deep reinforcement learning for drone navigation using sensor data,” Neural Computing and Applications, vol. 33, no. 6, pp. 2015–2033, Jun. 2020.

[3] A. Safonova, Y. Hamad, A. Alekhina, and D. Kaplun, “Detection of Norway spruce trees (Picea abies) infested by bark beetle in UAV images using YOLOs architectures,” IEEE Access, vol. 10, pp. 10384–10 392, 2022.

[4] D. Gallacher, “Drones to manage the urban environment: Risks, rewards, alternatives,” Journal of Unmanned Vehicle Systems, vol. 4, no. 2, pp. 115–124, Jun. 2016.

[5] L. A. Silva, A. Sales Mendes, H. Sánchez San Blas, L. Caetano Bastos, A. Leopoldo Gonçalves, and A. Fabiano de Moraes, “Active actions in the extraction of urban objects for information quality and knowledge recommendation with machine learning,” Sensors-basel, vol. 23, no. 1, p. 138, Dec. 2022.

[6] L. Melendy, S. Hagen, F. Sullivan, T. Pearson, S. Walker, P. Ellis, Kustiyo, A. K. Sambodo, O. Roswintiarti, M. Hanson, A. Klassen, M. Palace, B. Braswell, and G. Delgado, “Automated method for measuring the extent of selective logging damage with airborne LiDAR data,” ISPRS J. Photogramm. Remote Sens., vol. 139, pp. 228–240, May 2018.

[7] L. A. Silva, H. Sanchez San Blas, D. Peral García, A. Sales Mendes, and G. Villarubia González, “An architectural multi-agent system for a pavement monitoring system with pothole recognition in UAV images,” Sensors-basel, vol. 20, no. 21, p. 6205, Oct. 2020.

[8] M. Guerrieri and G. Parla, “Flexible and stone pavements distress detection and measurement by deep learning and low-cost detection devices,” Eng. Fail. Anal., vol. 141, p. 106714, Nov. 2022.

[9] D. Jeong, “Road damage detection using YOLO with smartphone images,” in 2020 IEEE International Conference on Big Data (Big Data). IEEE, Dec. 2020, pp. 5559–5562.

[10] M. Izadi, A. Mohammadzadeh, and A. Haghighattalab, “A new neuro-fuzzy approach for post-earthquake road damage assessment using GA and SVM classification from QuickBird satellite images,” J. Indian Soc. Remote Sens., vol. 45, no. 6, pp. 965–977, Mar. 2017.

[11] Aparna, Y. Bhatia, R. Rai, V. Gupta, N. Aggarwal, and A. Akula, “Convolutional neural networks based potholes detection using thermal imaging,” Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 3, pp. 578–588, Mar. 2022.

[12] J. Guan, X. Yang, L. Ding, X. Cheng, V. C. Lee, and C. Jin, "Automated pixel-level pavement distress detection based on stereo vision and deep learning," Automat. Constr., vol. 129, p. 103788, Sep. 2021.

[13] D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, and Y. Sekimoto, "RDD2022: A multi-national image dataset for automatic road damage detection," 2022.

[14] F. Coelli, A. Moxnes, and K. H. Ulltveit-Moe, "Better, faster, stronger: Global innovation and trade liberalization," National Bureau of Economic Research, Tech. Rep., Sep. 2016.

[15] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," Tech. Rep.