International Journal of Computational and Experimental Science and Engineering (IJCESEN)

Vol. 11-No.3 (2025) pp. 4732-4744

http://www.ijcesen.com

ISSN: 2149-9144

Research Article

A Novel Deep Learning Approach for Enhanced Roadway Pothole Detection Using YOLOv8 Instance Segmentation Algorithms

Kathiravan M[1]*, Gangadevi G[2], Suresh Balakrishnan T[3], Saravanan SK[4], Justindhas Y[5], Moorthy Agoramoorthy[6]

[1]Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences (SIMATS), Saveetha University, Thandalam, Chennai-602 105. India
* **Corresponding Author Email:** kathiravanm.sse@saveetha.com- **ORCID:** 0000-0002-5377-7871

[2]Department of computer science and engineering, SRMIST Ramapuram
Email: ganga2007mtech@gmail.com- **ORCID:** 0000-0003-2339-3989

[3]Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences (SIMATS), Saveetha University, Thandalam, Chennai-602 105. India
Email: sbalakrishna30@gmail.com- **ORCID:** 0009-0001-1658-1642

[4]Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences (SIMATS), Saveetha University, Thandalam, Chennai-602 105. India
Email: saravanansk.sse@saveetha.com- **ORCID:** 0000-0001-7775-9275

[5]Department of Computer Science and Engineering (AI&ML)Easwari Engineering College, Ramapuram, Chennai
Email: justindhasy@gmail.com- **ORCID:** 0000-0003-0296-2206

[6]Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences (SIMATS), Saveetha University, Thandalam, Chennai-602 105. India
Email: moorthya.sse@saveetha.com- **ORCID:** 0000-0002-5377-78710000-0002-7061-2788

Article Info:

DOI: 10.22399/ijcesen.3285
Received : 29 April 2025
Accepted : 01 July 2025

Keywords

Instance Segmentation
YOLOv8 Architecture
Roadway Pothole Detection
Deep Learning Algorithms,
Autonomous Vehicle Navigation.

Abstract:

Potholes pose a major risk to roadway safety and vehicle durability, necessitating timely detection and repair. With advancements in artificial intelligence, deep learning has become crucial for automating pothole detection and segmentation. Previous studies using CNN and Haar cascade methods have achieved accuracies up to 98.2%. This paper presents a novel approach leveraging the YOLOv8 architecture for instance segmentation, enhancing detection accuracy by capturing contextual and spatial relationships. The process involves data collection, annotation, preprocessing, and model training using datasets from Roboflow Universe and Kaggle. The model's performance is assessed through sensitivity, precision, recall, F1 score, and mean average precision (mAP). Experimental results indicate a significant improvement, achieving a 99.2% mAP in pothole detection and segmentation. These findings highlight the potential of YOLOv8 in advancing automated road maintenance, ensuring safer and more efficient transportation systems.

1. Introduction

The Indian Road network stands as a crucial infrastructure component, stretching an impressive 6,331,791 kilometres as of December 3033, making it the second-largest globally, surpassed only by the United States. With a road density of 1.94 kilometres per square kilometre, comparable to Hong Kong and significantly higher than that in the United States, China, and Russia, it plays an integral role in the nation's transport system, handling over 71% of freight and approximately 85% of passenger traffic [29].

However, the vast network faces significant challenges, notably road damages that contribute to a substantial number of accidents. According to a 2021 survey, potholes are responsible for nearly 0.8% of all road accidents and deaths and 0.6% of injury cases reported [30]. These issues not only impact public safety but also have economic implications, especially in a developing country like India.

Potholes, primarily caused by weather and other physical factors, lead to traffic congestion and the deterioration of roadway infrastructure, evoking public frustration and dissatisfaction. Addressing these issues requires precise and efficient methodologies. Traditional manual inspections are costly and time-consuming, prompting the need for advanced technological solutions.

In this context, deep learning, particularly through instance segmentation, offers a cutting-edge approach. By optimising neural networks and advanced computer vision algorithms, this approach accurately identifies roadway anomalies. For predicting and finding potholes in real time [1], Haar feature-based cascade classifiers with AdaBoost layers and CNN have been put in place. Additionally, real-time detection technologies, such as OAK-D using Raspberry Pi as the host computer, have shown promise in object detection frameworks, including various versions of the YOLO algorithm [2].

Instance segmentation, a complex area in machine vision and computer vision research, employs methods like Mask RCNN, enhancing segmentation accuracy and efficiency. This approach ensures precise localization and recognition of object instances within images, effectively distinguishing a wide range of real-world scenarios [31]. The implementation of YOLOv8 and other advanced frameworks in Python further strengthens the ability to accurately detect and segment potholes, utilising a custom dataset for evaluation based on metrics like F1, precision, mAP, and recall.

2. Literature Work

The advancement in computer vision, particularly in the realm of instance segmentation, is evidenced by extensive research and methodologies. Deep learning techniques, now considered state-of-the-art for object detection, have shown remarkable efficiency in feature and instance representation in various computer vision applications. These methods effectively tackle challenges such as limited training datasets, multi-sensor data, and complex backgrounds.

The Haar feature-based cascade classifier, complemented by a sequence of 9 Ada boost adaptive boosting, has proven effective in detecting and identifying potholes on Indian roads. Furthermore, the integration of Convolutional Neural Networks (CNN) for real-time pothole prediction enhances the instance segmentation algorithm used to mask and quantify pothole regions, marking a significant contribution to current project development [1]. The OAK D, a single-board Raspberry Pi computer, serves as an edge computing device for pothole detection. Its real-time performance, compared with other object detection frameworks such as various versions of the YOLO algorithm (YOLOv4, Tiny YOLOv4, YOLOv5), showed a high mean average precision of 80.04%, 85.48%, and 95% respectively [2].

Distributional instance segmentation with distributional expressiveness has been proposed to address the inherent ambiguity in real-world scenarios. This approach introduces latent codes that predict multiple hypotheses of object masks, achieving a double pick rate of 82.3% [3]. Additionally, the U Net Backbone, a fully convolutional neural network with a modified architecture, is tailored for working with fewer training images and yields more precise segmentation, significantly improving the MaskIoU head of the Mask scoring RCNN model [4].

The Grad-CAM (Gradient Weighted Class Activation Mapping) method, which utilizes the gradients of the classification score relative to the final convolutional feature map, aids in identifying image parts that most impact the classification score. This method achieved an overall mAP score of 88.9% [5].

Instance segmentation using multimodal input data compared in RGB and RGB+ HSV formats showed that training and evaluation in HSV format improved precision from 89.4% to 97.5%, compared to using RGB data alone [6]. Furthermore, the use of data augmentation techniques to expand road datasets using Mask RCNN and image inpainting has been pivotal. This method involves using Mask RCNN to detect vehicles on the road and employing image inpainting to remove and augment these vehicles for training models related to roadways [7].

Instance segmentation enhances segmentation accuracy and efficiency, ensuring accurate localization and recognition of object instances within images/frames. This signifies that instance segmentation effectively distinguishes a wide range of real-world scenarios and is efficient in localizing and recognizing object instances [8].

3. Material and Methods

The methodology of this project is centred on the implementation of instance segmentation within the realm of computer vision, particularly for detecting and segmenting roadway potholes. Instance segmentation is a sophisticated computer vision task that integrates two fundamental objectives: object detection and semantic segmentation. This approach is instrumental in recognizing and precisely delineating individual objects within an image while also differentiating between multiple instances of the same class [9].

In this project, the focus is on applying instance segmentation to the specific challenge of roadway pothole detection - a critical issue in road maintenance and vehicular safety. The methodology involves the use of the YOLOv8 framework, a state-of-the-art model in object detection known for its exceptional speed and accuracy. Adapting this model to the nuances of pothole detection involves training it on a diverse and comprehensive dataset of roadway images [10]. These images are curated to cover a wide range of pothole scenarios, ensuring robustness and reliability in real-world conditions.
The broader application of this methodology extends to the field of autonomous driving technology. By equipping self-driving vehicles with the capability to identify and avoid potholes, the project contributes to enhancing road safety and the overall efficiency of autonomous transportation systems.

3.1 YOLOv8 Framework for Pothole Localization

YOLO, abbreviated as "You Only Look Once", is an object instance detection algorithm recognized for its efficiency and accuracy. It operates on a single neural network capable of simultaneously predicting the bounding boxes and class probabilities for each object within an image. YOLO has evolved through various iterations, with the latest being YOLOv8. This version represents a significant upgrade over its predecessors, incorporating advanced features that enhance its performance in object detection and segmentation.

YOLOv8 distinguishes itself with several key modifications. It introduces spatial attention, which focuses on relevant features within the image by creating an attention map in the feature map or a tensor's cross-section. This attention to detail allows for more accurate identification and localization of objects. Additionally, YOLOv8 employs feature fusion to extract higher-order discriminative information from multiple input features. This process not only increases the detection accuracy but also eliminates redundant information, streamlining the detection process.
Another critical aspect of YOLOv8 is its emphasis on context aggregation. By integrating information from various parts of the image, YOLOv8 can understand the scene comprehensively, leading to more effective object detection.

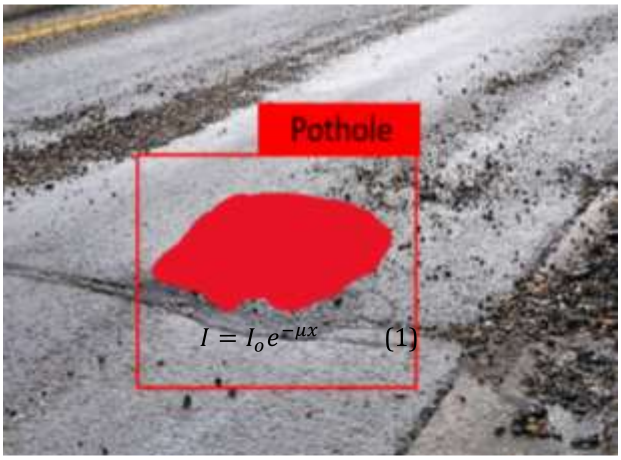

Figure 1. *Instance Segmentation*

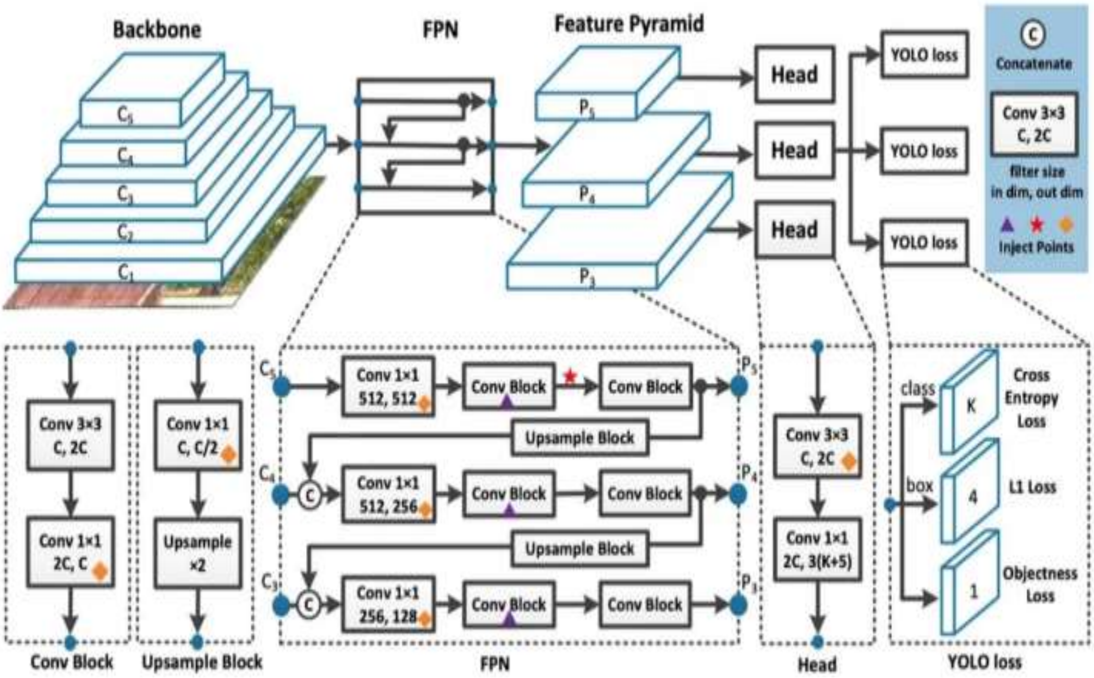

Figure 2: *Object detection Model of YOLO*

At the core of YOLOv8's architecture lies CSPDarknet53, a convolutional neural network backbone that comprises 53 layers. This framework is an enhanced adaptation of DarkNet-53, employing the CSPNet approach. CSPDarknet53 segments the feature map of its base layer into two distinct parts, merging them through a cross-stage hierarchy. This split-and-merge strategy enhances the gradient flow within the network, thus boosting the overall efficacy and efficiency of object detection in the YOLOv8 model. With these advancements, YOLOv8 emerges as a key player in the field of computer vision and object detection research.

In the YOLOv8 model, the head section is a key component, consisting of multiple convolutional layers followed by fully connected layers. These layers are instrumental in the model's ability to perform essential tasks such as predicting bounding boxes, determining object confidence scores, and calculating class probabilities for the objects detected in an image. This structure is crucial for the accurate identification and classification of objects, making it a cornerstone of YOLOv8's functionality.

A significant innovation in YOLOv8 is the integration of a self-attention mechanism within the head section. This feature enables the model to dynamically focus on various regions of the image, adjusting the emphasis on different features based on their importance to the object detection task. This self-attention mechanism is particularly effective in enhancing the model's detection capabilities, as it allows for a more nuanced and context-aware analysis of the visual data.

Furthermore, YOLOv8 is notably proficient in multi-scaled object detection, a capability achieved through the implementation of a feature pyramid network. This network is composed of multiple layers, each designed to detect objects of different sizes and scales. The multi-tiered structure of the network ensures that YOLOv8 can accurately identify both large and small objects, showcasing its versatility and effectiveness in handling a diverse range of object detection scenarios.

Overall, the architecture of YOLOv8, particularly its head section with convolutional and fully connected layers, the self-attention mechanism, and the feature pyramid network, collectively contribute to its advanced object detection performance. This makes YOLOv8 a highly capable model for various applications, including those that require detecting and differentiating objects of varying sizes within complex visual environments.

3.1.1 CSPDarknet53

CSPDarknet53 stands as a prominent backbone architecture in YOLOv8, playing a significant role in enhancing the speed and accuracy of object detection. This framework, a pivotal feature in the Convolutional Neural Network (CNN) domain, is distinguished by its implementation of the Cross Stage Partial Connection, a feature integral to the YOLO (You Only Look Once) family known for its real-time object detection capabilities.

The essence of the Cross Stage Partial Connection lies in its ability to augment the flow of information across the neural network layers. This feature is engineered to enhance both the accuracy and efficiency of object detection and segmentation tasks. It achieves this by optimizing the way information is processed and shared within the network, leading to more accurate and efficient detection outcomes.

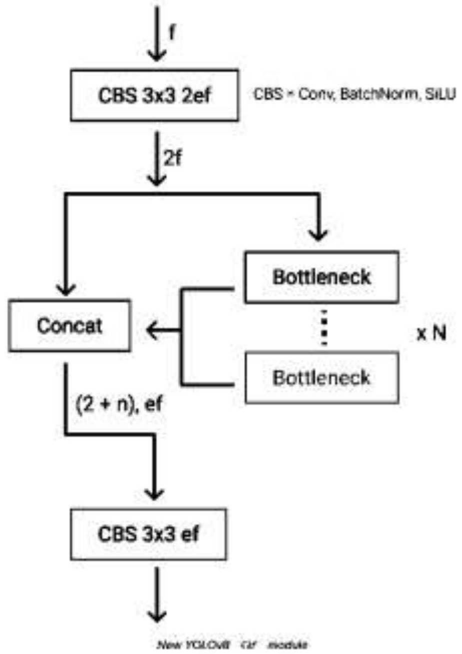

Figure 3: *New Convolutions of YO*

At the core of the CSPDarknet53 architecture are convolutional layers, residual connections, and specific architectural features common in deep learning models. With its 53 convolutional layers, CSPDarknet53 excels in capturing and processing intricate features present within images, thus enabling the detailed and accurate detection of objects.

f = "features"
e=" expansion rate"

The latest iteration of YOLOv8 sees a modification in the CSPDarknet53 backbone. Notably, the network's first 6x6 convolution has been replaced by a 3x3 convolution, and the main building block of the framework has shifted from C2f to C3. This change is highlighted in the framework's design, where the outputs from the bottleneck, containing 3x3 convolutions along with their residual connections, are concatenated in the C2f configuration. In contrast, the C3 configuration utilizes the output from the last bottleneck. A key similarity between YOLOv5 and YOLOv8 is the bottleneck structure, although YOLOv8 features a change in the first convolution's kernel size from 1x1 to 3x3, reflecting the continuous evolution and refinement in the YOLO series for improved object detection performance.

3.1.2 Evaluation of the YOLO family versions

Model	size (pixels)	mAP^{val} 50-95	Speed CPU (ms)	Speed T4 GPU (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	-	-	3.2	8.7
YOLOv8s	640	44.9	-	-	11.2	28.6
YOLOv8m	640	50.2	-	-	25.9	78.9
YOLOv8l	640	52.9	-	-	43.7	165.2
YOLOv8x	640	53.9	-	-	68.2	257.8

Figure 4: *Inference made by stating that the YOLOv8 coco accuracy is state of art for the compared models.*

Instance segmentation, a complex yet crucial task in computer vision, involves detecting object instances and generating their per-pixel segmentation masks or bounding boxes. This process can be approached using two primary methods: RCNN (Region-based Convolutional Neural Network) and FCN (Fully Connected Network).
A key characteristic of Convolutional Neural Networks (CNNs) is their translation invariance, meaning they can detect the class of an input regardless of its position in the image. However, this feature poses a challenge for instance segmentation. Unlike object detection where the focus is on identifying objects regardless of their specific pixels, instance segmentation requires segmenting each object instance at the pixel level. This is because the semantics of an image vary across different regions and pixels.
Mask RCNN, a State-Of-The-Art (SOTA) model for instance segmentation, effectively addresses this challenge. It produces three types of outputs: class labels, bounding boxes, and object masks. The object masks generated for each Region of Interest (ROI) are particularly significant as they maintain the spatial layout of the object. Unlike vector representations, these masks preserve spatial dimensions, providing a detailed and contextual understanding of each object's location and shape within the image.

In the context of this project, instance segmentation is applied to detect potholes in roadways. The YOLOv8 framework, known for its efficiency in object detection, is utilized for this purpose. The project leverages a customized dataset specifically tailored for pothole detection, as well as datasets available from the Roboflow universe. By focusing on instance segmentation using YOLOv8, the project aims to accurately identify and segment potholes, thereby contributing to the improvement of road maintenance and safety.

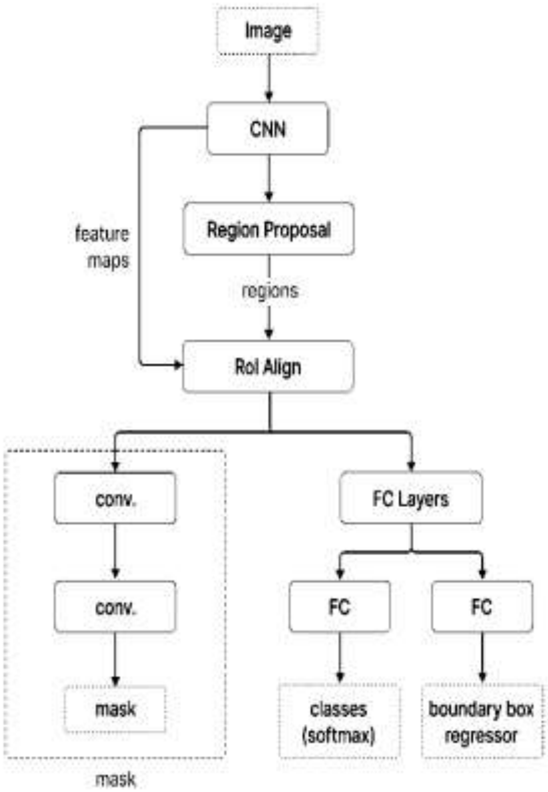

Figure 5: *Instance Segmentation Algorithm*

4. Implementations

For this project, a specialized pothole dataset was created using diverse sources like Google Images, YouTube, and Roboflow Universe to cover a wide range of pothole instances under various conditions. Roboflow, known for its robust platform in computer vision model development, played a pivotal role in dataset preparation. It provides AI-assisted labeling tools for bounding boxes, polygons, and instance segmentation, facilitating accurate and efficient dataset annotation. Roboflow's comprehensive capabilities include dataset uploading, labelling, generation, and training models for object detection and classification tasks [11], both single and multi-label, making it an invaluable asset in the development of this project.

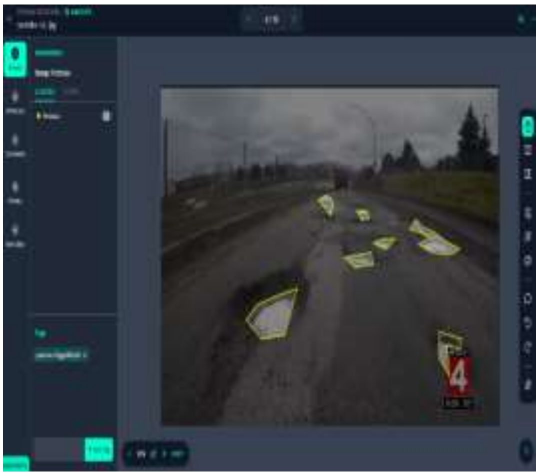

Figure 6: *Robo-flow annotation tool*

4.1 Data Creation and Annotation

The initial step involved creating a new project on the Roboflow dashboard, specifying the project type as instance segmentation for pothole detection [10]. The image data, comprising 130 images, was extracted from Roboflow Universe and YouTube, with videos sampled at a frame rate of one frame per second. Additional data were sourced from publicly available annotated datasets to enhance the model's training effectiveness.

4.2 Preprocessing Techniques

Roboflow's suite of preprocessing tools was employed to prepare the dataset for training:

Auto-Orient and Resize: This step involved stripping images of their EXIF data to maintain consistency in display and resizing the images to desired dimensions [7].

Advanced Preprocessing: Additional methods like grayscale conversion, auto-adjust, adaptive equalization, object isolation, static cropping, and filtering were used to refine the dataset further.

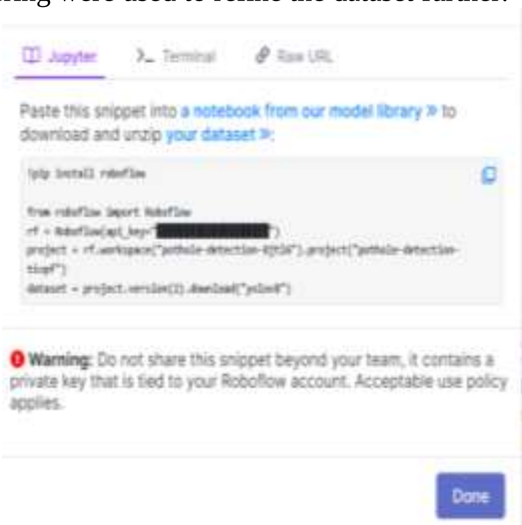

Figure 7*: API Download code*

4.3 Dataset Export and Training Preparation

Once preprocessing was complete, the dataset was exported in a format compatible with YOLOv8. A download API code snippet was generated from Roboflow, which was then utilized for training the model with the prepared dataset. This structured approach in dataset preparation and preprocessing ensured the development of a robust, well-annotated dataset, tailored for training the YOLOv8 model in accurately detecting and segmenting potholes in various roadway conditions.

4.4 State of Art YOLOv8

YOLOv8 represents the latest advancement in the YOLO (You Only Look Once) series, a prominent algorithm in the field of computer vision known for its speed and effectiveness in real-time object detection tasks. The evolution of YOLO, from its second version (v2) incorporating features like batch normalization, anchor boxes, and dimension clusters, to subsequent versions introducing multiple anchors, spatial pyramid pooling, and Mosaic data augmentation, signifies its continuous improvement.

Developed by Ultralytics, YOLOv8 stands as the successor to its preceding versions, bringing new features and improvements that elevate its performance, flexibility, and efficiency. This version extends its capabilities to a wide range of AI tasks, including detection, segmentation, pose estimation, tracking, and classification, supporting a 360-degree vision in AI applications.

4.4.1 YOLOv8 Accuracy

YOLOv8 demonstrates superior accuracy, as indicated by its Mean Average Precision (mAP)

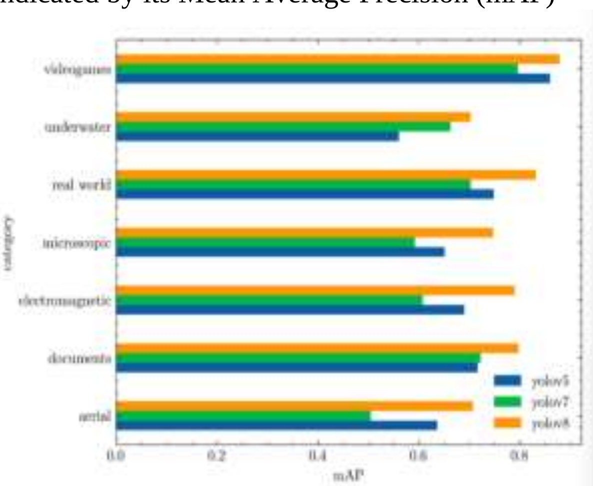

Figure 8: *mAP score predictions against RF100 categori*

against RF100 Categories. This metric reflects the model's ability to accurately detect and classify objects across a wide range of categories. YOLOv8's performance, in terms of data necessity and training time, is inferred to be more efficient and effective compared to its predecessors, marking it as a leading choice in object detection tasks. The advancements and refinements in YOLOv8 ensure that it remains at the forefront of object detection technology in computer vision.

4.5. Model Training for YOLOv8

The training of the YOLOv8 model was meticulously carried out over 30 epochs. An epoch, in this context, represents a complete cycle where the entire training dataset is passed forward and backward through the neural network once. The chosen image size for the training was 640 pixels, aligning with the model's requirements for optimal performance.

4.6 Training Metrics and Evaluation

Mean Average Precision (mAP): The model achieved a remarkable mAP score of 0.992 at an Intersection over Union (IoU) threshold of 0.5. This high mAP score is indicative of the model's exceptional ability in object detection tasks, particularly in accurately identifying objects with a high degree of overlap with the ground truth.

Confusion Matrix Analysis: The confusion matrix, a tool used to assess the performance of a classification model, showed a true positive rate of 0.093. This implies that approximately 93% of the instances were correctly identified as positive cases by the model, underlining its effectiveness in classification tasks.

Accuracy Measurement: An overall accuracy of 93% further confirms the model's proficiency in correctly identifying and classifying positive instances within the dataset.

4.7 Intersection over Union (IoU)

Definition and Importance: IoU is a crucial metric in object detection models, measuring the overlap between the predicted bounding box and the actual ground truth box. A higher IoU value suggests a closer alignment of the model's predictions with the actual object locations, which is critical for precise object detection.

Role in Model Evaluation: IoU plays a significant role in evaluating the accuracy of the model's predictions. In this project, the high IoU score achieved by the model underscores its ability to produce bounding boxes that closely match the actual objects in the image.

The training results, characterized by high mAP and IoU scores, along with a strong performance in the confusion matrix analysis, collectively demonstrate the model's robustness and accuracy in detecting objects. This level of precision, especially with a reasonably high IoU threshold, positions YOLOv8 as a highly reliable tool for object detection tasks, particularly in the context of detecting potholes in roadways.

The Intersection over Union (IoU) is a fundamental metric used in object detection models to evaluate the accuracy of predicted bounding boxes against the actual ground truth. The IoU calculation involves measuring the overlap between two bounding boxes: the predicted box and the ground truth box.

Let's consider boxes X and Y

X= (A1, B1, C1, D1)

X= (A2, B2, C2, D2)

Let's mention the coordinates the Ai and Bi as the top left corner coordinates of the bounding boxes and Ci & Di for the top right respectively.

Ai = max (A1, A2)

Bi = max (B1, B2,)

Ci = max (C1, C2)

If any value of the corner co-ordinates is equal to zero then the IoU value is also zero

X Y = (Ci- Ai) x (Di- Bi)

X = $(C_1 - A_1) \times (D_1 - B_1)$

| Y | = $(C_2 \; A_2) \times (D_2 - B_2)$

| XU | Y | = X + Y - XnY

Intersection over Union= TP/(TP+FN+FP)

The precision determines the measures of how well the model can find true positives (TP) out of all positive predictions. (TP+FP).

Recall determines measures of how well the model can find true positives (TP) out of all predictions (TP+FN). Out of all these, the mean Average Predictions are calculated.

mAP= $\frac{1}{N}\sum_{i=1}^{N} AP_i$. This metric provides an overall effectiveness of the model in object detection tasks across different categories.

Figure 9: *Stating the area of overlap and union*

4.8 Model Validation

The validation phase of the model plays a crucial role in establishing its reliability and accuracy. In this project, the YOLOv8 model underwent a validation process using specific weights and a set confidence threshold of approximately 0.25, or 25%. This confidence level was selected to ensure a balance between precision and recall, aiming to reduce false positives while still maintaining a high detection rate.

4.9 Performance Metrics

Processing Speed: The model demonstrated a swift processing speed, taking only 5 milliseconds per image. This rapid speed is indicative of the model's suitability for real-time applications, such as in-vehicle systems for pothole detection and avoidance.
Image Resolution: The validation was performed on images with a resolution of 640x640 pixels, which is adequate for capturing sufficient detail to accurately identify potholes in various conditions.
Sample Outputs
The outputs from the model implementation provided visual confirmation of its detection capabilities. The model successfully identified potholes with high confidence scores, indicated by the bounding boxes and associated confidence levels overlaid on the images. These scores, such as 0.98 and 1.0, reflect the model's high certainty in its predictions.
The sample outputs showcase the model's effectiveness in distinguishing potholes from the surrounding pavement, which is essential for any subsequent automated repair dispatch or navigational adjustment in autonomous driving scenarios. The high confidence levels aligned with the bounding boxes around the detected potholes confirm the model's precision in localizing and quantifying the extent of the roadway damage.
Overall, the model's validation results affirm its robustness and potential for practical deployment in smart city infrastructure management and autonomous vehicle systems. Its ability to process images swiftly and with high accuracy paves the way for enhanced road safety and maintenance.

***Figure 10**: Picture for different pothole*

5. Evaluation and Result

This work entailed a comprehensive evaluation and testing phase executed on the Google Cloud Platform, specifically using Colab on a system with Windows 11 and Python version 3.10.12. The Ultralytics YOLO version 8.0.28, a leading-edge algorithm within the YOLO family, was employed to train the pothole detection and segmentation model. This particular version is lauded for its high mean Average Precision (mAP) score and swift inference speed, both critical factors in object detection performance [12,13,14]. For computational support, PyTorch version 11.8 with built-in CUDA version 11.8 was utilized, taking advantage of NVIDIA's GPU acceleration capabilities through the Tesla T4 GPU [11]. This powerful combination of software and hardware facilitated efficient parallel processing, significantly enhancing the model's training and evaluation process. The training regimen for the model incorporated transfer learning, utilizing pre-trained weights as a starting point, which were then fine-tuned on a custom dataset comprised exclusively of annotated pothole images. This approach leverages learned features from large datasets to improve performance on the specialized task of pothole detection [15]. Evaluations focused on mAP scores ranging from 50-95 and inference speed—common benchmarks of success in object detection algorithms. The mAP metric is particularly prominent in object detection, providing an average precision across various IoU thresholds, thereby offering a balanced measure of the model's precision and recall capabilities. The precision of the model was assessed based on the number of true positives, which correspond to a high IoU with the ground truth, versus the false positives, where the IoU falls below the threshold. The model achieved an exceptional mAP score of approximately 99.2%, indicating an extremely high level of precision and recall [10,16]. This score suggests that the model is adept at accurately detecting and localizing object instances across a diverse array of images.

The overall evaluation employed various matrices to gauge the model's performance comprehensively, ensuring that the model's capabilities were thoroughly assessed and validated to meet the stringent requirements of object detection and segmentation tasks [17].

3. Results and Discussions

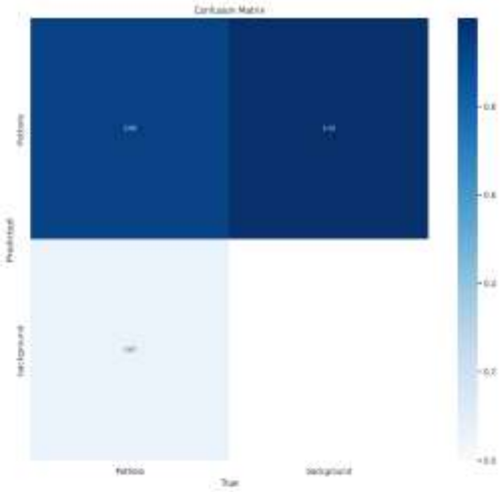

***Figure 11**: Confusion Matrix*

From the above confusion matrix, we can infer,
Precision is the ability of a classifier not to label a positive sample that is negative. In other words, it tells us how many of the potholes that the classifier detected were potholes [16,17]. The precision score can be calculated using the formula:

$$\frac{tp}{tp + fp}$$

In this case, TP is 1.00 and FP is 0.07. Therefore, the precision is:
Precision = 1.00 / (1.00 + 0.07) Precision = 0.93
The precision is 0.93, which means the classifier detected 93% of the potholes that were potholes.
Recall (Sensitivity) is the ability of a classifier to find all the positive samples [15]. In other words, it tells us how many of the actual potholes that the classifier detected were potholes. The recall score can be calculated using the formula:

$$\frac{tp}{tp + fn}$$

In this case, TP is 1.00 and FN is -0.6. Therefore, the recall is:
Recall = 1.00 / (1.00 + (-0.6)) Recall = 1.00
The recall is 1.00, which means the classifier detected 100% of the actual potholes that were potholes.

Precision Recall Curve

The provided figure showcases a Precision-Recall (PR) Curve, a graphical representation that measures the performance of a binary classification model. On the graph, precision is plotted on the vertical y-axis and recall, or sensitivity, on the horizontal x-axis [17,].
Observing the curve's position in the top right corner, we can deduce that the model consistently maintains high precision across the spectrum of recall values. The precision nears the perfect score of 1.0, implying an exceptional level of accuracy where virtually all pothole predictions by the model are true positives, with negligible false positives. Similarly, the recall approaches 1.0, indicating the model's adeptness at identifying the vast majority of actual potholes present in the dataset.
The curve's behavior is particularly informative for imbalanced datasets, where the positive class, in this case, potholes, is less common. It effectively communicates how the model's ability to make correct positive predictions (precision) and its success in capturing true positive cases (recall) evolves as the classification threshold is varied.
The legend provides additional context, noting that the model achieved a mean Average Precision (mAP) of 0.992 at an Intersection over Union (IoU) threshold of 0.5, a standard metric for object detection models. This high mAP score is indicative of the model's robust precision and recall, confirming its efficacy in pothole detection tasks.
In essence, the PR curve affirms the model's exemplary performance in pothole detection, signifying both a high hit rate and an impressive ability to avoid false alarms, thereby reinforcing its utility for roadway maintenance applications.

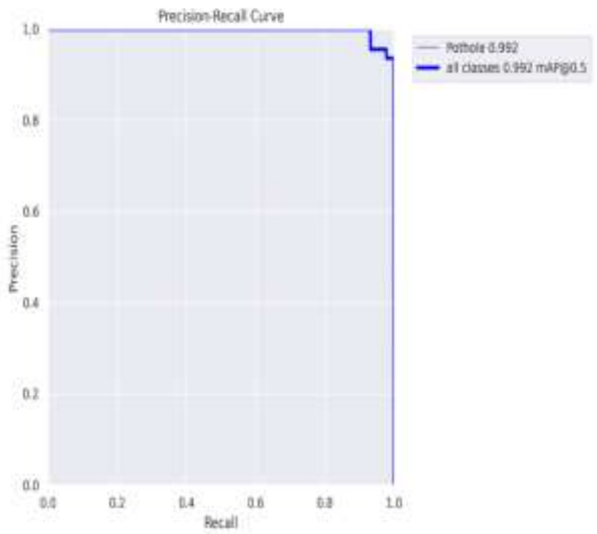

***Figure 12:** Precision Recall Curve Precision*

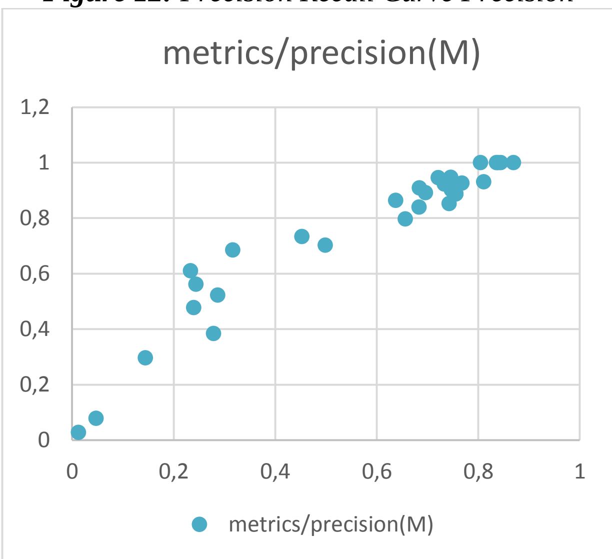

***Figure 13:** Precision plot*

The provided scatter plot is a graphical representation of a machine learning model's precision values, identified by "M", which could refer to a specific model or category related to pothole detection [16,18]. Precision, as depicted on the x-axis, measures the proportion of accurate positive predictions by the model and is expressed on a scale from 0 to 1.

Each point on the plot corresponds to the model's precision at various evaluation points or thresholds. A notable clustering of points towards the higher end of the scale, approaching 1.0, signifies instances where the model precisely identified potholes with minimal false positives. Conversely, points scattered towards the lower end could reflect reduced accuracy, potentially due to complex data or atypical conditions during those tests.

In evaluating model performance, such a scatter plot is essential for assessing precision across different operational contexts. A high concentration of points with precision values above 0.7, particularly those clustering near 1.0, indicates robust model accuracy. Moreover, if these high precision scores are consistently achieved across epochs, it suggests the model is effectively learning and stabilizing its predictive capabilities.

Overall, the plot suggests the model demonstrates strong performance in pothole detection with an average precision around 87%, indicating it as a potentially optimal solution. However, the presence of lower scores in certain instances underscores the typical variability encountered in real-world applications, emphasizing the importance of comprehensive testing across diverse conditions.

mAP metrics

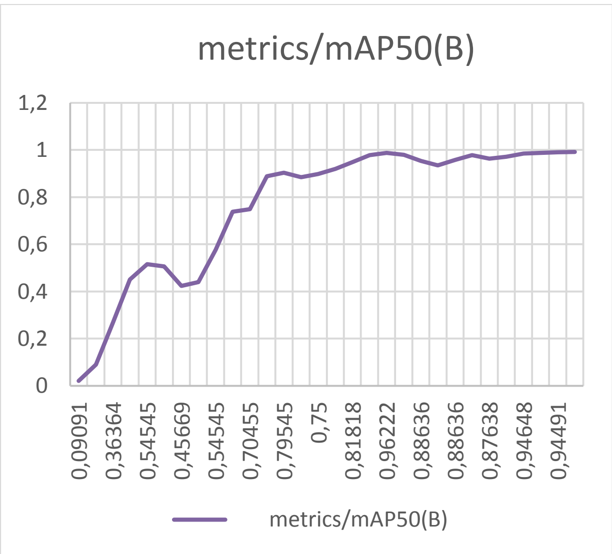

***Figure 14**: mAP score evaluation*

The line graph presented illustrates the mean Average Precision (mAP) at an Intersection over Union (IoU) threshold of 0.5, referred to as mAP50, for a specific model, denoted as 'B'. The x-axis indicates sequential evaluation points, likely corresponding to different epochs in the model's training progression [14,15]. The y-axis quantifies the mAP50 score, which is normalized between 0 and 1—a conventional scoring range for such evaluations.

Observing the trajectory of the line, an initial upward trend in mAP50 score is evident, signifying an enhancement in the model's precision for object detection with a moderate ground truth overlap. Throughout the training, the mAP50 score displays minor variances but predominantly sustains above the 0.8 mark, denoting a robust precision level in the model's predictions.

The graph's latter section, characterized by stable and elevated mAP50 values, implies that the model has reached a reliable and consistent detection performance. Assuming 'B' symbolizes a model specialized in pothole detection, this stability suggests a successful learning outcome, with the model adeptly recognizing the essential features for pothole identification without significant performance deviation.

In essence, the model exemplifies a well-calibrated training process, achieving and consistently upholding a high precision level as indicated by the mAP50 score. This performance is further substantiated by an overall mAP score around 99.2%, underscoring the model's efficacy in pothole detection. Such a standardized metric like mAP is crucial for comparing different object detection models and serves as a reliable benchmark in research and practical applications.

Loss Function: The loss function is an essential component in the training of a machine learning model, serving as a critical indicator of performance [12,13]. It quantifies the model's effectiveness by measuring the discrepancy between the predicted outputs and the actual target values. During training, the model's parameters are adjusted to minimize the loss, which in turn enhances the model's ability to make accurate predictions.

A key function of the loss metric is to guide the learning process. By iteratively reducing loss, the model learns to fine-tune its internal parameters, leading to more accurate and reliable decision-making. Additionally, the loss function plays a pivotal role in evaluating the model's performance, providing insight into how well the model generalizes from the training data to unseen data. This aspect of generalization is vital, as it determines the model's predictive power in real-world scenarios.

Furthermore, loss functions are designed to penalize outliers and anomalies, which helps in improving the overall accuracy of the model. In the context of the YOLO algorithm, the loss function is multifaceted, encompassing three fundamental components that each address different aspects of the detection task, from localizing objects within the image to classifying them accurately. These tailored loss functions collectively ensure that the YOLO algorithm performs optimally across various object detection challenges.

Localization: The loss eradicates the error in the prediction of the bounding box coordinates using the Mean Squared error.

$$\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{obj}\left[(x_i-\hat{x}_i)^2+(y_i-\hat{y}_i)^2\right]$$

$$+\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{obj}\left[\left(\sqrt{w_i}-\sqrt{\hat{w}_i}\right)^2+\left(\sqrt{h_i}-\sqrt{\hat{h}_i}\right)^2\right]$$

Confidence loss: The main significance of the confidence loss is to measure the recall/ confidence in the objectness of the bounding box. The confidence loss contains Object confidence loss & non object confidence loss. The object confidence loss basically eradicates if the model identifies the object as there is no object and for non-object confidence loss, it's the vice versa.

Object Confidence loss:

$$\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{obj}\left(C_i-\hat{C}_i\right)^2$$

Non object confidence loss:

$$\lambda_{noobj}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{noobj}\left(C_i-\hat{C}_i\right)^2$$

Class Loss : The class loss is the measure of how much errors are present in the class predictions.

$$\sum_{i=0}^{S^2}\mathbb{1}_{i}^{obj}\sum_{c\in classes}(p_i(c)-\hat{p}_i(c))^2$$

The total loss is calculated by:
Total Loss (L_total) = λ_coord * L_coord + λ_conf * L_conf + λ_cls * L_cls

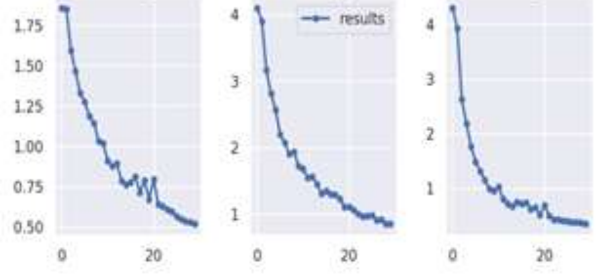

***Figure 15**: Results for train loss*

Objectness Loss: Objectness loss evaluates how well the model distinguishes between object pixels and background pixels. A lower score reflects the model's competence in identifying the presence of an object within its predicted bounding box. Segmentation Loss: Segmentation loss pertains to the model's ability to correctly segment the object, defining its shape and boundaries at the pixel level. A reduced segmentation loss signifies that the model effectively captures the contours of objects, correctly assigning class labels to each pixel, and making accurate pixel-wise predictions. When the segmentation loss is minimal during both training and validation, it suggests the model's strong generalization abilities—it does not merely memorize the training data but can extrapolate its learning to accurately predict on new, unseen data. In conclusion, the low values observed across all three loss components suggest that the YOLOv8-seg model has been trained effectively, with optimal performance in detecting and segmenting potholes. This level of performance indicates a well-generalized model capable of accurately delineating and classifying potholes across different environments. Deep learning has been applied in different fields [25-32].

6. Conclusion

The paper presents an efficient approach using the YOLOv8 algorithm for detecting and segmenting potholes, a critical need for Indian roads. Through the analysis of 700 annotated samples, with a significant portion dedicated to training and the rest to testing and validation, the study showcases the capability of the algorithm to deliver high performance in identifying potholes, which can greatly aid in road maintenance and the advancement of autonomous driving technologies.

However, the study acknowledges a major limitation in the form of dataset diversity, primarily focused on Indian roads, which may not encompass the wide variety of global road conditions. This specificity could potentially restrict the model's broader application. Additionally, the model's adaptability to varied environmental factors such as lighting and weather conditions remains to be thoroughly evaluated, which is crucial for real-world deployment.

Looking forward, enhancing the dataset to include a wider range of road types and conditions from different geographic areas would be instrumental in increasing the robustness and generalizability of the model. Further development is also needed to ensure the model's efficiency in real-time processing, especially for use in autonomous vehicles, and to assess its scalability for detecting other types of road anomalies or infrastructural defects. By addressing these areas, future research can build upon the current work to develop more comprehensive and universally applicable road condition assessment tools.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

[1] Pan, Q., Gao, M., Wu, P., Yan, J., & Li, S. (2021). A Deep-Learning-Based Approach for Wheat Yellow Rust Disease Recognition from Unmanned Aerial Vehicle Images. *Sensors*, 21(19), 6540. https://doi.org/10.3390/s21196540

[2] Wang, D., & Yang, S. X. (2022). Intelligent feature extraction, data fusion and detection of concrete bridge cracks: current development and challenges. *Intelligence & Robotics*, 2(4), 391-406. http://dx.doi.org/10.20517/ir.2022.25

[3] Xia, B., Luo, H., & Shi, S. (2022). Improved Faster R-CNN Based Surface Defect Detection Algorithm for Plates. *Computational Intelligence and Neuroscience*, 2022, 3248722. https://doi.org/10.1155/2022/3248722

[4] Xu, B., Wang, W., Falzon, G., Kwan, P., Guo, L., Chen, G., Tait, A., & Schneider, D. (2020). Automated cattle counting using Mask R-CNN in quadcopter vision system. *Computers and Electronics in Agriculture*, 171, 105300. https://doi.org/10.1016/j.compag.2020.105300

[5] Triki, A., Bouaziz, B., Gaikwad, J., & Mahdi, W. (2021). Deep Leaf: Mask R-CNN based leaf Detection and Segmentation from digitized herbarium specimen images. Pattern *Recognition Letters*, 150. https://doi.org/10.1016/j.patrec.2021.07.003

[6] Wan, S., & Goudos, S. (2019). Faster R-CNN for Multi-class Fruit Detection using a Robotic Vision System. *Computer Networks*, 168, 107036. https://doi.org/10.1016/j.comnet.2019.107036

[7] Padma, E., Rohith, D. N. V., & Sai Charan, E. V. (2022). Mask RCNN: Object Detection Approach using Machine Learning Techniques. Volume 13, Issue 03.

[8] Pan, Q., Gao, M., Wu, P., Yan, J., & Li, S. (2021). A Deep-Learning-Based Approach for Wheat Yellow Rust Disease Recognition from Unmanned Aerial Vehicle Images. *Sensors*, 21(19), 6540. https://doi.org/10.3390/s21196540

[9] Sharma, R., Saqib, M., Lin, C. T., et al. (2022). A Survey on Object Instance Segmentation. *SN Computer Science*, 3, 499. https://doi.org/10.1007/s42979-022-01407-3

[10] Satti, S. K., Devi, K. S., Dhar, P., et al. (2022). Detecting potholes on Indian roads using Haar feature-based cascade classifier, convolutional neural network, and instance segmentation. *Soft Computing*, 26, 9141–9153. https://doi.org/10.1007/s00500-022-07265-8

[11] Zimmermann, R. S., Siems, J. N. (2019). Faster training of Mask R-CNN by focusing on instance boundaries. Computer Vision and Image Understanding, 188, 102795. https://doi.org/10.1016/j.cviu.2019.102795

[12] Wang, W., Dai, J., Chen, Z., Huang, Z., Li, Z., Zhu, X., Hu, X., Lu, T., Lu, L., Li, H., & Wang, X. (2023). InternImage: Exploring Large-Scale Vision Foundation Models with Deformable Convolutions. *In IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 14408-14419). https://doi.org/10.1109/CVPR52729.2023.01385

[13] M. J. Carmel Mary Belida, A. Begum, S. Alex David, E. Kannan, K. Senthil and N. Ruth Naveena, "Predictive Modeling for Medical Insurance Malpractice Using Random Forest and XGBoost," *2024 International Conference on Communication, Computing and Internet of Things (IC3IoT)*, Chennai, India, 2024, pp. 1-5.

[14] Bai, R., Shen, F., Wang, M., et al. (2023). Improving Detection Capabilities of YOLOv8-n for Small Objects in Remote Sensing Imagery: Towards Better Precision with Simplified Model Complexity [Preprint]. *Research Square*. https://doi.org/10.21203/rs.3.rs-3085871/v1

[15] Ganesh, P., Volle, K., Burks, T. F., & Mehta, S. (2019). Deep Orange: *Mask R-CNN based Orange Detection and Segmentation. IFAC-PapersOnLine.*

[16] Liu, Y., Mishra, N., Abbeel, P., & Chen, X. (2023). Distributional Instance Segmentation: Modeling Uncertainty and High Confidence Predictions with Latent-MaskRCNN. *In IEEE International Conference on Robotics and Automation (ICRA*) (pp. 7069-7075). London, United Kingdom. https://doi.org/10.1109/ICRA48891.2023.1016081 2

[17] R. M. Naidu and V. Nagaraju, "Efficient Online Medical Store Finding and Availability of Medicines Using SVM Compared over CNN with Improved Accuracy," *2023 Eighth International*

Conference on Science Technology Engineering and Mathematics (ICONSTEM), Chennai, India, 2023, pp. 1-5.

[18] Bučko, B., Lieskovská, E., Zábovská, K., Zábovský, M. (2022). Computer Vision Based Pothole Detection under Challenging Conditions. *Sensors.*

[19] Amo-Boateng, M., Sey, N. E. N., Amproche, A. A., & Domfeh, M. K. (2022). Instance segmentation scheme for roofs in rural areas based on Mask R-CNN. *The Egyptian Journal of Remote Sensing and Space Science*, 25(2), 569-577. https://doi.org/10.1016/j.ejrs.2022.03.017

[20] Zhang, Q., Chang, X., & Bian, S. B. (2020). Vehicle-Damage-Detection Segmentation Algorithm Based on Improved Mask RCNN. *IEEE Access*, 8, 6997-7004. https://doi.org/10.1109/ACCESS.2020.2964055

[21] Knopp, L., Wieland, M., Rättich, M., & Martinis, S. (2020). A Deep Learning Approach for Burned Area Segmentation with Sentinel-2 Data. *Remote Sensing*, 12(15), 2422. https://doi.org/10.3390/rs12152422

[22] S. Ashokkumar, S. M. Kumar, R. Rajaraman, D. Sugumaran and N. N. Saranya, "Comparative Analysis of Deep Learning Algorithms for Image Recognition in Medical Imaging," *2024 Second International Conference on Advances in Information Technology (ICAIT)*, Chikkamagaluru, Karnataka, India, 2024, pp. 1-6

[23] Bagheri, F., Tarokh, M. J., Ziaratban, M. (2021). Skin lesion segmentation from dermoscopic images by using Mask R-CNN, Retina-Deeplab, and graph-based methods. *Biomedical Signal Processing and Control*, 67, 102533. https://doi.org/10.1016/j.bspc.2021.102533

[24] Khan, M. A., Akram, T., Zhang, Y.-D., & Sharif, M. (2021). Attributes based skin lesion detection and recognition: A mask RCNN and transfer learning-based deep learning framework. *Pattern Recognition Letters*, 143, 58-66. https://doi.org/10.1016/j.patrec.2020.12.015

[25] R. Sundar, M. Ganesan, M.A. Anju, M. Ishwarya Niranjana, & T. Surya. (2025). A Context-Aware Content Recommendation Engine for Personalized Learning using Hybrid Reinforcement Learning Technique. *International Journal of Computational and Experimental Science and Engineering*, 11(1). https://doi.org/10.22399/ijcesen.912

[26] Ibeh, C. V., & Adegbola, A. (2025). AI and Machine Learning for Sustainable Energy: Predictive Modelling, Optimization and Socioeconomic Impact In The USA. *International Journal of Applied Sciences and Radiation Research* , 2(1). https://doi.org/10.22399/ijasrar.19

[27] Johnsymol Joy, & Mercy Paul Selvan. (2025). An efficient hybrid Deep Learning-Machine Learning method for diagnosing neurodegenerative disorders. *International Journal of Computational and Experimental Science and Engineering*, 11(1). https://doi.org/10.22399/ijcesen.701

[28] Kumari, S. (2025). Machine Learning Applications in Cryptocurrency: Detection, Prediction, and Behavioral Analysis of Bitcoin Market and Scam Activities in the USA. *International Journal of Sustainable Science and Technology*, 1(1). https://doi.org/10.22399/ijsusat.8

[29] Sivananda Hanumanthu, & Gaddikoppula Anil Kumar. (2025). Deep Learning Models with Transfer Learning and Ensemble for Enhancing Cybersecurity in IoT Use Cases. *International Journal of Computational and Experimental Science and Engineering*, 11(1). https://doi.org/10.22399/ijcesen.1037

[30] Olola, T. M., & Olatunde, T. I. (2025). Artificial Intelligence in Financial and Supply Chain Optimization: Predictive Analytics for Business Growth and Market Stability in The USA. *International Journal of Applied Sciences and Radiation Research* , 2(1). https://doi.org/10.22399/ijasrar.18

[31] G Nithya, R, P. K., V. Dineshbabu, P. Umamaheswari, & T, K. (2025). Exploring the Synergy Between Neuro-Inspired Algorithms and Quantum Computing in Machine Learning. *International Journal of Computational and Experimental Science and Engineering*, 11(3). https://doi.org/10.22399/ijcesen.2484

[32] Chui, K. T. (2025). Artificial Intelligence in Energy Sustainability: Predicting, Analyzing, and Optimizing Consumption Trends. *International Journal of Sustainable Science and Technology*, 1(1). https://doi.org/10.22399/ijsusat.1