

INFORMATICS INSTITUTE OF TECHNOLOGY In

Collaboration with

ROBERT GORDON UNIVERSITY ABERDEEN

**CM4606 – Machine Vision**

Assessment Document by

Kavindya Koralegei – 20210575

Module leader:

Prof. Nihal Kodikara

Submitted in partial fulfillment of the requirements for the B.Sc. (Hons) in Artificial Intelligence

and Data Science degree at Robert Gordon University.

**March 2025**

© The copyright for this project and all its associated products resides with the Informatics

Institute of Technology

# Table of Content

1 . Introduction	2
2 . Literature Review	2
3 . Dataset Information	5
3.1 General Information	5
3.2 Data Preprocessing and Preparation	6
3.2.1 Resizing and Augmentation	6
3.2.2 Data Preprocessing	6
3.3 Dataset Structure and Configuration	7
3.4 Exploratory Data Analysis	7
4 . Methodology	9
4.1 High Level Diagram	9
4.2 Model Information	10
4.3 Hyperparameter Tuning	10
4.4 Real-Time Road Damage Assessment	11
5 . Evaluation	12
5.1 Evaluation Metrics	12
5.2 Training and Validation	13
5.2.1 Overall Loss Trends	13
5.2.2 Component-Specific Learning Curves	14
5.2.3 Confidence-Based Curves (Bounding Box and Mask)	15
5.2.4 Precision-Recall Curves (Bounding Box and Mask)	16
5.2.5 Confusion Matrix	17
5.3 Testing	17
5.3.1 Additional Evaluation Metrics	19
6. Limitations and Future Work	20
Reference list	21

# Road Pothole Detection and Segmentation

## 1 . Introduction

Road infrastructure is a crucial element of urban and rural transportation systems, and maintaining its quality is essential for ensuring public safety and efficient mobility. Potholes, which develop due to weather conditions, traffic loads, and inadequate road maintenance, pose serious hazards to vehicles and pedestrians, leading to accidents, increased vehicle repair costs, and disruptions in transportation networks. Traditional pothole detection methods rely on manual inspections and sensor-based approaches, which are labor-intensive, time-consuming, and often inefficient in covering large road networks. With advancements in artificial intelligence and machine vision, automated pothole detection has emerged as a promising solution to enhance road maintenance efficiency and safety.

This coursework focuses on developing an artificial intelligence-based solution for pothole detection and segmentation using deep learning techniques. By leveraging computer vision and neural networks, the proposed system aims to automatically detect, and segment potholes in road images and videos. The project will explore an applicable deep learning model to achieve high accuracy and real-time processing capabilities. Additionally, the study will talk about preprocessing techniques, exploratory data analysis (EDA), model training, hyperparameter tuning, and evaluation metrics to ensure the robustness and reliability of the system. The ultimate goal is to provide a scalable, efficient, and automated approach for road damage assessment, contributing to smarter and safer transportation infrastructure.

## 2 . Literature Review

Pothole detection is a crucial aspect of road maintenance and infrastructure management, as deteriorating road conditions pose significant risks to vehicular and pedestrian safety. Traditional approaches such as manual inspections and sensor-based methods, while effective to some extent, suffer from inefficiencies in scalability, time consumption, and human error. With advancements in artificial intelligence, deep learning-based approaches have emerged as a viable alternative, offering higher accuracy, scalability, and real-time capabilities. These approaches integrate computer vision, machine learning, and segmentation techniques to enable automated pothole detection and classification in varying environmental conditions. For example, the application of Convolutional Neural Networks (CNNs), Region-Based CNNs (R-CNNs), and YOLO (You Only Look Once) models has significantly improved pothole detection accuracy and real-time deployment feasibility.

Deep learning methods for pothole detection can be categorized into two main categories; one-stage object detection models and two-stage models. One-stage models, such as YOLOv8 and SSD (Single Shot MultiBox Detector), offer fast inference and high accuracy, making them ideal for real-time pothole monitoring. In contrast, two-stage models like Faster R-CNN provide higher accuracy but require more computational resources, limiting their suitability for embedded and real-time systems (Wang et al., 2024).

Camilleri and Gatt (2020) and Ping, Yang and Gao (2020) conducted comparative studies on deep learning models for pothole detection, evaluating trade-offs between accuracy and computational efficiency. Camilleri and Gatt (2020) found that while YOLOv3-SPP achieved the highest mAP (68.83%), it was computationally demanding, making it less suitable for embedded platforms. They demonstrated that SSD Lite MobileNet V2, when optimized for TensorFlow Lite (TFLite), provided the best speed-accuracy trade-off, achieving an inference time of 0.085 seconds per image on an Android device and 1.68 seconds on a Raspberry Pi. Similarly, Ping, Yang and Gao (2020) compared YOLOv3, SSD, HOG with SVM, and Faster R-CNN, concluding that YOLOv3 achieved the best balance between speed and accuracy, outperforming SSD and Faster R-CNN in real-time detection tasks. These findings highlight that while Faster R-CNN provides the highest detection accuracy, it requires significant computational resources, whereas YOLO-based models offer a more practical balance for real-time applications, particularly in embedded systems and mobile platforms. Ahmed (2021) introduced a modified Faster R-CNN with a dilated MVGG16 backbone, improving inference speed while maintaining high detection accuracy (81.4% mAP). Their study compared multiple YOLOv5 variants and Faster R-CNN backbones, concluding that while YOLOv5 models offer faster inference, Faster R-CNN with ResNet50 and MVGG16 achieves superior detection performance, making them more suitable for applications where accuracy is prioritized over speed. The study by R. Swathika et al. (2024) also compared different deep learning models such as Faster RCNN, YOLOv4, and YOLOv8, demonstrating that YOLOv8 outperformed its predecessor (YOLOv4) and Faster RCNN in terms of detection accuracy and real-time processing. Similarly, R S et al. (2024) trained YOLOv8 on a dataset comprising 3,074 annotated road images, achieving a box mAP50 of 82.1% and a mask mAP50 of 80.5%, proving its effectiveness in real-world applications. A more recent improvement is the POT-YOLO model, an advanced version of YOLOv8, which integrates edge segmentation techniques for enhanced detection precision. The study on POT-YOLO reported 99.1% accuracy, 97.6% precision, and 93.52% recall, outperforming Faster and Masked R-CNN and SSD models (Bhavana et al., 2024). These findings emphasize the potential of YOLOv8-based architectures as the most effective solutions for real-time pothole detection due to its balance between speed and precision.

In addition to object detection that uses bounding boxes to localize potholes, segmentation-based approaches have been widely explored to enhance pothole localization by providing a pixel-level understanding of pothole size, shape, and severity. Instance segmentation techniques, such as

Mask R-CNN, DeepLabv3+ and the recently introduced DSASNet (Dual Self-Attention Segmentation Network), provide pixel-level classification, allowing for more precise pothole boundary delineation (Wang et al., 2024). DSASNet has demonstrated superior performance over traditional segmentation models, achieving an F1-score of 93.65% and an Intersection over Union (IoU) score of 0.881, indicating its robustness in handling complex road conditions. R. Swathika et al. (2024) explored instance segmentation within YOLOv8, allowing potholes to be detected with improved precision by defining their exact contours. Similarly, R S et al. (2024) integrated real-time segmentation techniques into YOLOv8, improving detection in autonomous road monitoring systems. Sowbarnika Thiruppathiraj, Kumar and Swapnil Buchke (2020) implemented Mask R-CNN and U-Net, demonstrating that Mask R-CNN significantly outperformed U-Net, achieving 80% accuracy and an F1-score of 0.85. However, U-Net suffered from low recall (34%) in complex road conditions. In contrast, Feng et al. (2022) proposed MAFNet, a multimodal fusion network that integrates RGB and disparity images for enhanced segmentation. Their model outperformed AA-RTFNet and DeepLabV3+, achieving an mIoU of 87.29% and an F1-score of 92.90%, particularly excelling in edge segmentation which highlights the importance of multimodal fusion and attention mechanisms in improving pothole segmentation accuracy.

The evaluation of pothole detection and segmentation models typically involves multiple performance metrics, including accuracy, precision, recall, F1-score, IoU, and mean Average Precision (mAP). Among these, YOLOv8 variants have been extensively benchmarked, with YOLOv8x achieving the highest mAP of 51.4% when tested on a diverse road dataset (Kumari et al., 2023). Comparative studies have highlighted that POT-YOLO improves overall detection accuracy by 12.3% compared to traditional deep learning approaches, reinforcing the importance of edge segmentation in enhancing detection performance (Bhavana et al., 2024). YOLOv3, tested in multiple studies, consistently outperformed SSD and Faster R-CNN in real-time detection, achieving 82% accuracy in a comparative evaluation (Ping, Yang and Gao, 2020). On the other hand, Camilleri and Gatt (2020) highlighted the computational trade-offs in model selection, showing that while YOLOv3-SPP achieved a mAP of 68.83%, it required significant processing power, whereas SSD Lite MobileNet V2, optimized for TensorFlow Lite, provided a more efficient speed-accuracy balance for embedded applications. These findings underscore the ongoing challenge of designing pothole detection models that maximize accuracy while maintaining real-time efficiency, particularly for autonomous road monitoring and embedded systems.

## 3 . Dataset Information

### 3.1 General Information

This study utilizes two publicly available datasets specifically designed for pothole detection and segmentation. Both datasets are structured according to the YOLOv8 format, ensuring compatibility with state-of-the-art deep learning models for object detection and instance segmentation.

The first dataset was obtained from Kaggle, where it has received significant engagement from the machine learning community, reflected by numerous upvotes and a high usability rating. According to its metadata, the dataset was originally sourced from Roboflow.com and comprises a total of 780 annotated images. These annotations are formatted in the YOLOv8 structure, making them directly usable for training deep learning models. The dataset initially included training and validation subsets, making it a valuable resource for pothole detection research.

The second dataset was obtained from a research study by R. Swathika et al. (2024) which also sourced the data from Roboflow. This dataset contains a total of 4,376 images, divided into three subsets: 3,036 images for training, 1,043 images for testing, and 300 images for validation. The dataset provides a more extensive collection of pothole images, which contributes to improving model generalization.

Furthermore, both datasets have undergone preprocessing and data augmentation techniques to enhance model robustness while also offering high-quality annotations and a well-structured format, making them suitable for training deep learning models to detect and segment potholes effectively. The combination of these datasets allows for a more comprehensive evaluation of the proposed model's performance in real-world scenarios.

- Dataset link 1:  
<https://www.kaggle.com/datasets/farzadnekouei/pothole-image-segmentation-dataset>
- Dataset link 2:  
<https://universe.roboflow.com/thien-bk-1dkcf/pothole-detection-system-rnoh4/dataset/2>

Initial Dataset image counts:

```
Total training images: 3636
Total validation images: 360
Total test images: 1083
```

## 3.2 Data Preprocessing and Preparation

### 3.2.1 Resizing and Augmentation

Both the original datasets have undergone preprocessing and augmentation before being used for model training. Preprocessing steps included auto-orientation of pixel data, ensuring consistent image orientation by stripping EXIF metadata, and resizing all images to  $640 \times 640$  pixels using a stretch transformation. Resizing was necessary to standardize input dimensions, improve computational efficiency, and ensure compatibility with the YOLOv8 model, which is optimized for processing images of this size.

Data augmentation was applied exclusively to the training data to enhance model generalization and robustness by artificially increasing dataset diversity. Each source image was augmented to generate three additional versions with the following transformations:

- 50% probability of horizontal flip – Helps the model learn spatial invariance.
- Random cropping (0 to 20%) – Removes small portions of the image, making the model more resilient to occlusions.
- Random rotation (-15° to +15°) – Simulates different viewing angles.
- Random shearing (-5° to +5°) horizontally and vertically – Distorts the image slightly to improve robustness.
- Random brightness adjustment (-25% to +25%) – Helps the model handle variations in lighting conditions.
- Random exposure adjustment (-25% to +25%) – Simulates different illumination scenarios.

### 3.2.2 Data Preprocessing

Before utilizing the combined dataset for model training, several preprocessing steps were necessary to ensure consistency and accuracy in the annotations. One key issue in the second dataset was the presence of two object classes: "Pothole" and "Object." However, since the objective of this study is to detect and segment potholes, and the "Object" class contained relatively few images, all images and annotations corresponding to this specific class were removed.

```
After deletion:  
Total training images: 3581  
Total validation images: 355  
Total test images: 1057
```

Following this, a thorough verification was conducted of annotation consistency across the training, validation, and testing subsets. This step was crucial to detect and correct annotation

mismatches, ensuring that all images were correctly labeled. After resolving any inconsistencies, the dataset distribution was finalized.

```
After deletion:
Total training images: 3579
Total validation images: 355
Total test images: 1056
```

Lastly, the YAML configuration file was updated to reflect the modifications, ensuring that the dataset contained only a single class, "Pothole." The revised YAML file was then saved, marking the completion of the dataset preprocessing and preparation stage.

```
names:
- Pothole
nc: 1
roboflow:
  license: CC BY 4.0
  project: pothole-detection-system-rnoh4
  url: https://universe.roboflow.com/thien-bk-1dkcf/pothole-detection-system-rnoh4/dataset/2
  version: 2
  workspace: thien-bk-1dkcf
test: ../test/images
train: ../train/images
val: ../valid/images
```

### 3.3 Dataset Structure and Configuration

The dataset was organized into train, test, and validation directories, each containing two subdirectories: images and labels. Each label file provides information about the pothole instances, including the class index and normalized segmentation coordinates. The annotations follow the format:

[class-index] [x1] [y1] [x2] [y2] ... [xn] [yn],

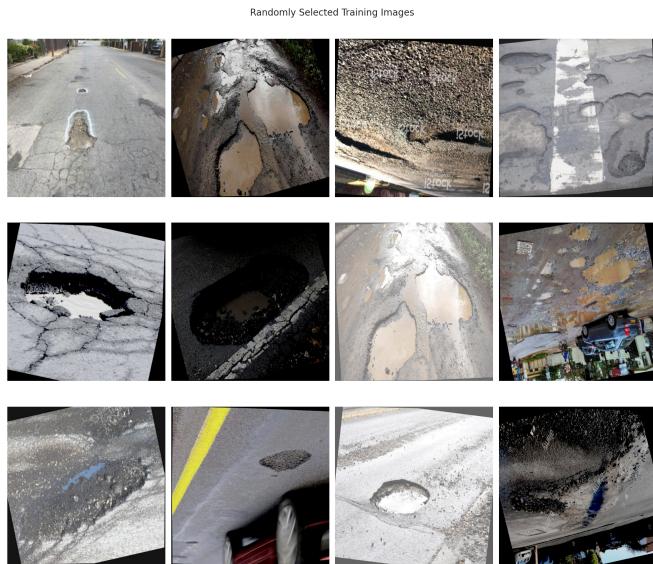
where [class-index] represents the class identifier for the object, and [x1] [y1] [x2] [y2] ... [xn] [yn] define the segmentation mask's bounding coordinates, separated by spaces. The new\_data.yaml file serves as the Ultralytics YOLO dataset configuration file, ensuring proper integration with the YOLOv8 model. This file specifies the paths to the training, testing and validation datasets, defines the number of object classes (1), and assigns the class label "Pothole". It plays a critical role in structuring the dataset and enabling accurate model training.

### 3.4 Exploratory Data Analysis

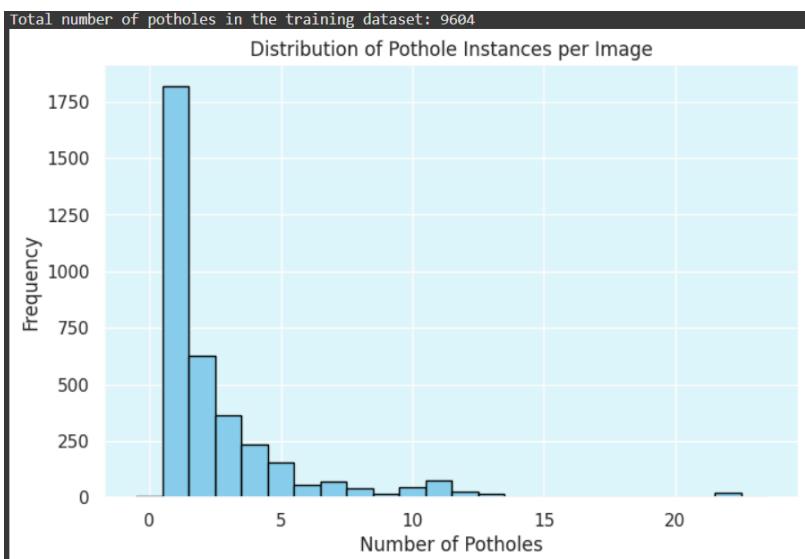
To gain a better understanding of the dataset, several exploratory analyses were conducted. First, the number of images in each directory (training, validation, and testing) was verified to ensure the dataset was properly split. Additionally, all images were checked for uniformity in size, confirming that each image was resized to 640×640 pixels, as specified during preprocessing.

```
Every training image is of size: (640, 640)
Every validation image is of size: (640, 640)
Every test image is of size: (640, 640)
```

To visually inspect the dataset, 12 random training images were selected and displayed in a  $3 \times 4$  grid layout. This step helped in assessing the diversity of the images, their quality, and how well potholes were represented in different scenarios, such as varying lighting conditions, angles, and road textures.



A quantitative analysis of the dataset was performed by examining the label files. The total number of pothole instances in the training set was calculated by iterating through the annotation files and counting the number of labeled potholes in each image. A histogram was then plotted to visualize the distribution of pothole instances per image, providing insight into how frequently multiple potholes appear within a single image. This analysis helped in understanding the dataset's composition and the potential challenges in detecting and segmenting potholes accurately.



To further investigate the quality of annotations, a single image was selected, and its corresponding label file was parsed to extract the segmentation mask of a pothole. The polygon coordinates provided in the annotation were converted from normalized values to actual pixel positions, enabling the visualization of the pothole's location within the image. Using these coordinates, a binary mask was created to highlight the pothole region.

For deeper analysis, the extracted pothole region was processed using Canny edge detection to highlight the pothole's boundaries. The results were displayed in a three-panel visualization showing:

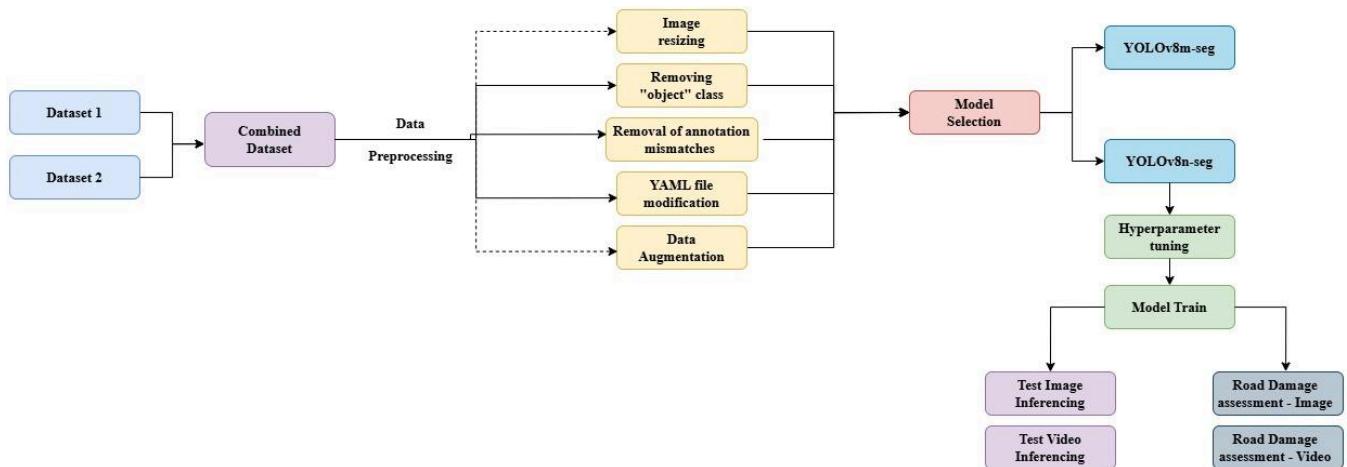
1. The original image
2. The pothole region extracted using the mask
3. The detected edges of the pothole using Canny edge detection



These steps provided a clearer understanding of the dataset, verified annotation correctness, and assessed the potential challenges in segmenting potholes from road images.

## 4 . Methodology

### 4.1 High Level Diagram



## 4.2 Model Information

The **YOLOv8n-seg** model was selected for this project due to its optimal balance between accuracy, computational efficiency, and real-time processing capabilities. As the "nano" (n) variant of the YOLOv8 segmentation models, it is designed for lightweight performance, making it an ideal choice for prompt pothole detection and segmentation in road images. In real-world applications such as autonomous driving, road maintenance, and intelligent transportation systems, high-speed inference is crucial, and YOLOv8n-seg excels in delivering fast processing with lower memory requirements. Unlike larger variants such as YOLOv8m-seg or YOLOv8x-seg, which offer slightly improved accuracy but demand significantly higher computational power, YOLOv8n-seg provides a practical trade-off that makes it well-suited for deployment in resource-constrained environments like embedded systems and edge devices. This model ensures efficient pothole detection while maintaining real-time processing capabilities, making it a viable choice for applications that require both speed and accuracy.

The model architecture consists of 151 layers with 3,409,968 trainable parameters, offering a streamlined yet effective structure for segmentation tasks. It integrates convolutional layers, bottleneck blocks, and feature fusion techniques, which help in learning both local and global spatial features crucial for identifying potholes in various road conditions. The Conv2D layers perform feature extraction, while Batch Normalization and SiLU activation functions enhance stability and learning efficiency.

## 4.3 Hyperparameter Tuning

During the hyperparameter tuning process, various configurations were tested to optimize the YOLOv8n-seg model for pothole detection using the kaggle dataset initially. Experiments were conducted by adjusting several parameters, including dropout rate, learning rate, optimizer, and batch size, both on CPU and GPU. The final best-performing configuration was achieved with the following settings:

- Epochs: 150 (with an early stopping patience of 15 epochs)
- Image Size: 640×640 pixels
- Batch Size: 16
- Optimizer: AdamW
- Initial Learning Rate: 0.0001 with a final learning rate multiplier of 0.01
- Momentum: 0.9
- Dropout Rate: 0.25
- Device: GPU (index 0)
- Seed: 42

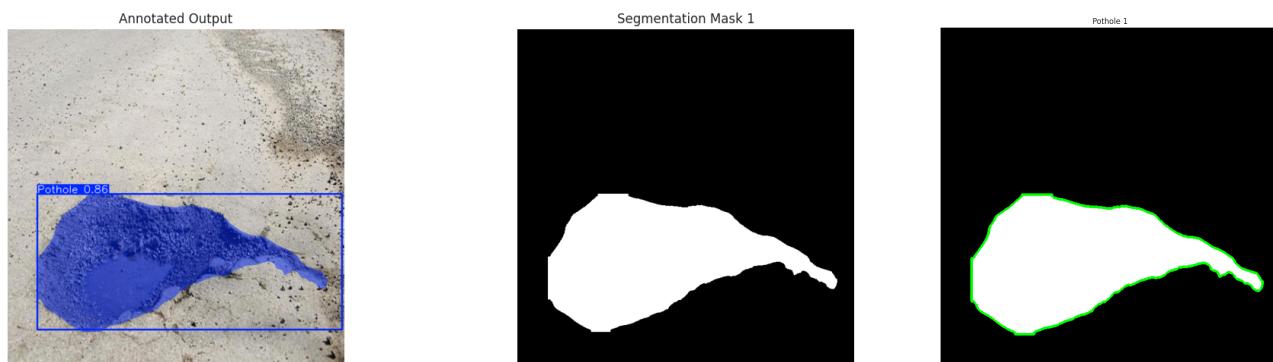
During the tuning phase, different values were explored: dropout values of 0.3 and 0.5 were evaluated, along with higher learning rates (0.001 and 0.002), alternative optimizers such as SGD, and batch sizes of 16 versus 32. The combination detailed above yielded the most promising results. The same optimized hyperparameters were then used for the final combined and preprocessed dataset.

Additionally, experiments with the YOLOv8s-seg variant were performed; however, it was significantly slower and did not offer comparable performance. This tuning process ensured that the chosen hyperparameters not only improved detection accuracy but also maintained efficient training and inference times, especially critical for real-time applications.

## 4.4 Real-Time Road Damage Assessment

### **Image-Based Road Damage Assessment:**

In this approach, the YOLO segmentation model analyzes an image to detect potholes and generate segmentation masks outlining the damaged areas. For each detected pothole, the model calculates the area by summing the pixel areas of the mask contours, and then computes the damage percentage by comparing the pothole area to the total image area. For example, in one analysis, a single pothole was detected with an area of 77,170.50 pixels, constituting 18.84% of the image's 409,600 pixels. The severity of the damage is then classified using predefined thresholds: if the damage covers less than 5% of the image, it is considered "Minor"; between 5% and 15% is classified as "Moderate"; and any damage exceeding 15% is deemed "Severe." In this case, with an 18.84% damage level, the road condition is classified as Severe. This quantification method provides a clear and objective metric for assessing road damage on a per-image basis.



Pothole 1 Area: 77170.50 pixels (18.84% of image)
-----
Total Pothole Area: 77170.50 pixels
Total Image Area: 409600 pixels
Overall Road Damage: 18.84%
Overall Damage Severity: Severe

## Real-Time Video Road Damage Assessment:

Extending the same methodology to video, each frame of the video undergoes a similar analysis where the YOLO segmentation model detects and segments potholes. The area of each pothole mask is calculated and expressed as a percentage of the frame's total pixel count. To enhance interpretability and ensure a more distinguishable range of damage values, the computed percentage is scaled by a factor of 1000 rather than 100. This scaling allows for finer granularity in damage differentiation, making it easier to detect small variations in pothole severity, as supported by R. Swathika et al. (2024). To further counteract fluctuations and achieve a more stable measure, the system applies a smoothing mechanism by averaging the damage percentages over the last three frames (smoothing window = 3). This averaged damage percentage is then used to classify the overall road damage into "Minor," "Moderate," or "Severe" based on the same thresholds used in the image-based assessment. The annotated frames display both the damage percentage and the severity level, providing continuous, real-time feedback on road conditions, which is particularly useful for automated maintenance planning and real-time monitoring applications.

## 5 . Evaluation

### 5.1 Evaluation Metrics

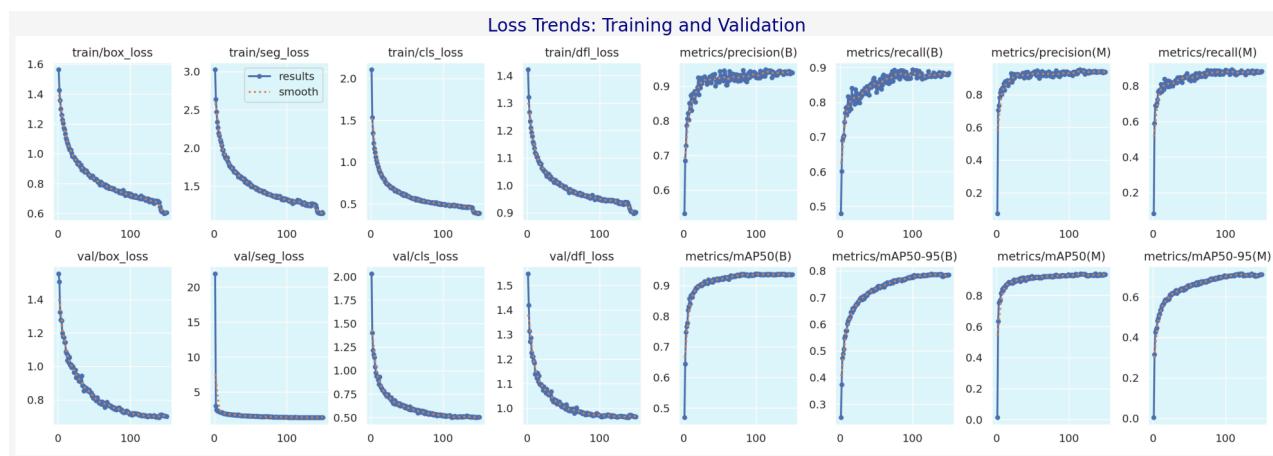
The following are the key evaluation metrics used for object detection and segmentation in this study. These metrics were used to assess the model's performance and generate visual plots for analysis.

1. **Precision** - Measures the accuracy of the model's positive detections. Represents the proportion of correctly identified potholes out of all predicted potholes. A higher precision value indicates fewer false positives.
2. **Recall** - Assesses the model's ability to detect all actual potholes. Indicates the proportion of correctly identified potholes out of all existing potholes. A higher recall value means fewer potholes are missed.
3. **F1 score** - A balance between precision and recall, providing a single measure of overall performance. A high F1 score signifies that the model is both accurate (high precision) and detects most potholes (high recall).
4. **Intersection over Union (IoU)** - Measures the overlap between the predicted segmentation mask and the actual ground truth mask. Higher IoU values indicate better segmentation accuracy.

5. **Mean Average Precision at IoU 0.5 (mAP@50)** - Evaluates detection performance using an IoU threshold of 50%. A higher mAP@50 value indicates strong detection capabilities, though it may not imply precise segmentation.
6. **Mean Average Precision at IoU 50-95 (mAP@50-95)** - Assesses detection and segmentation accuracy across multiple IoU thresholds (50% to 95%). A higher mAP@50-95 score suggests that the model produces well-defined segmentation boundaries.

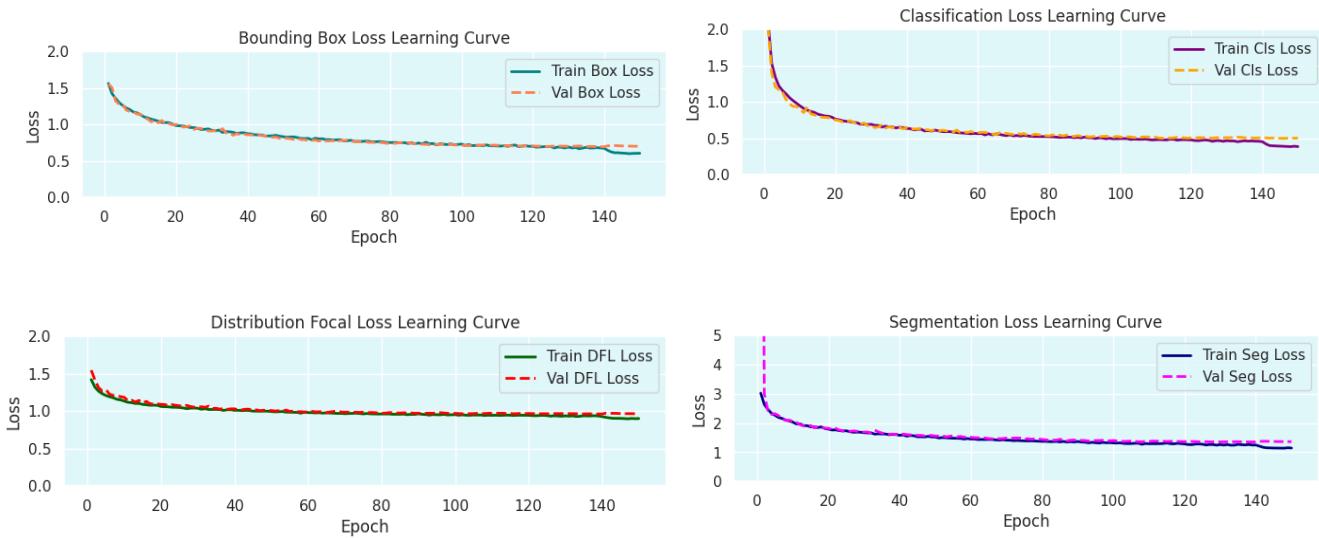
## 5.2 Training and Validation

### 5.2.1 Overall Loss Trends



From the Loss Trends plot, we can see that all loss components (bounding box loss, classification loss, segmentation loss, and distribution focal loss) generally decrease over the course of training for both the training and validation sets. This indicates that the model is progressively learning to better localize and classify potholes, as well as produce more accurate segmentation masks. The fact that the validation losses track the training losses without diverging significantly suggests that the model is not overfitting. While the training losses are typically lower, the validation losses do not rise sharply, which would be a red flag for overfitting. Instead, they remain relatively close to the training losses, implying a stable learning process and reasonable generalization to unseen data.

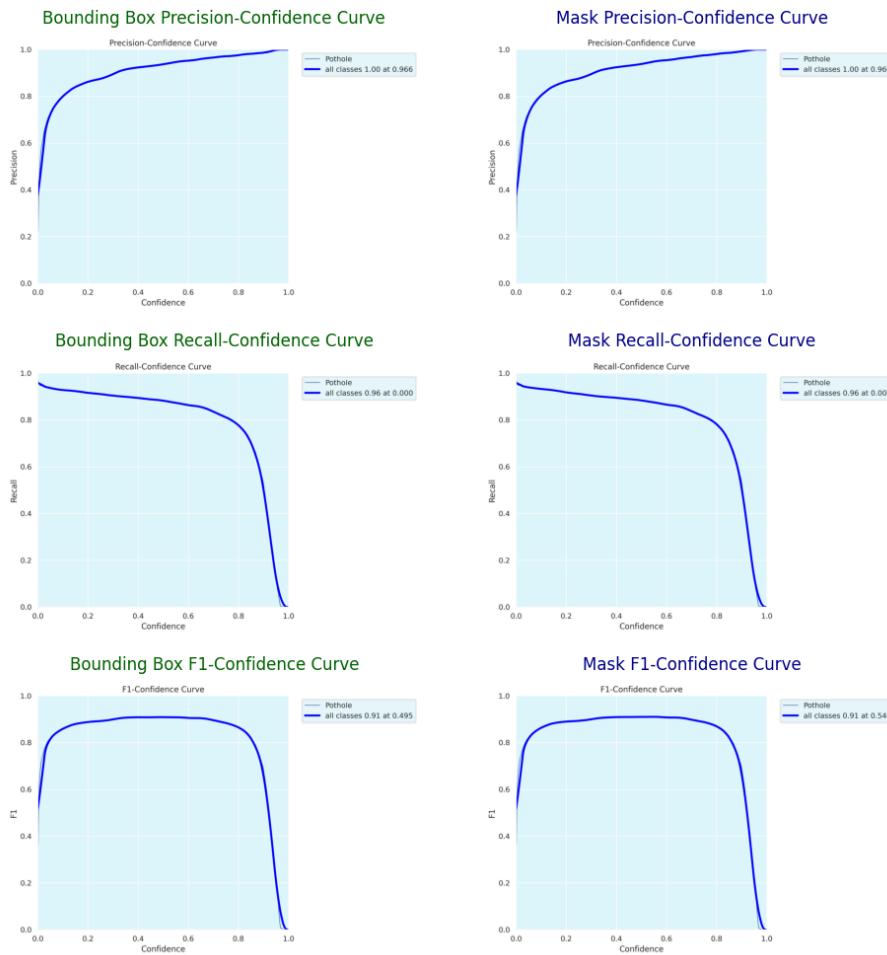
### 5.2.2 Component-Specific Learning Curves



Breaking down the losses into box\_loss, seg\_loss, cls\_loss, and dfl\_loss provides a more detailed view of how each aspect of the model's learning evolves:

- **Box Loss:** This loss measures how well the model is predicting the coordinates of the bounding boxes around potholes. The gradual decline in both training and validation curves indicates the model quickly learns basic localization and reflects fine-tuning of the bounding box boundaries.
- **Segmentation Loss:** This curve shows how effectively the model learns the exact shape and coverage of each pothole. Early on, the segmentation loss is high, especially on the validation set and then declines steadily. The close alignment of training and validation segmentation loss curves suggests consistent performance on both seen and unseen images.
- **Classification Loss:** This indicates how accurately the model distinguishes potholes from background. Both training and validation classification loss decrease substantially in the early epochs, then taper off. This pattern is typical as the model learns the distinguishing features for the “Pothole” class.
- **Distribution Focal Loss (DFL):** YOLOv8 uses DFL to improve bounding box regression by focusing on the distribution of potential box coordinates. The training and validation DFL curves start higher and then converge, mirroring the box loss trend. This consistency indicates that the model is refining its bounding box predictions over time.

### 5.2.3 Confidence-Based Curves (Bounding Box and Mask)



These plots illustrate how the model's precision, recall, and F1-score change as the confidence threshold varies from low to high. For bounding boxes and masks alike, precision tends to increase at higher confidence thresholds because the model makes fewer predictions but is more certain about them. Conversely, recall declines at higher thresholds, as more true positives are filtered out alongside false positives. The F1-score, which balances precision and recall, typically peaks at an intermediate threshold. This suggests there is a sweet spot where the model achieves a good trade-off between catching most potholes (high recall) and avoiding too many false positives (high precision).

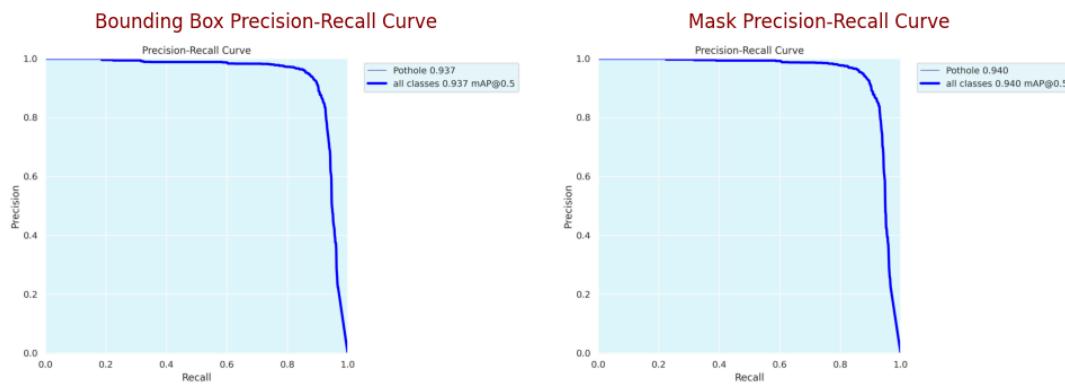
- Bounding Box Precision-Confidence Curve: Shows that as confidence increases, bounding box precision rises, but the model may detect fewer potholes overall.
- Bounding Box Recall-Confidence Curve: Indicates recall starts higher at low thresholds and decreases as the threshold grows.

- Bounding Box F1-Confidence Curve: Combines both metrics, usually peaking at a moderate confidence level.

The Mask plots follow the same pattern, but they focus on segmentation quality. A high confidence threshold yields more accurate segmentation masks but potentially misses some true pothole regions, while a low threshold catches more potholes but may also include more noise or false positives.

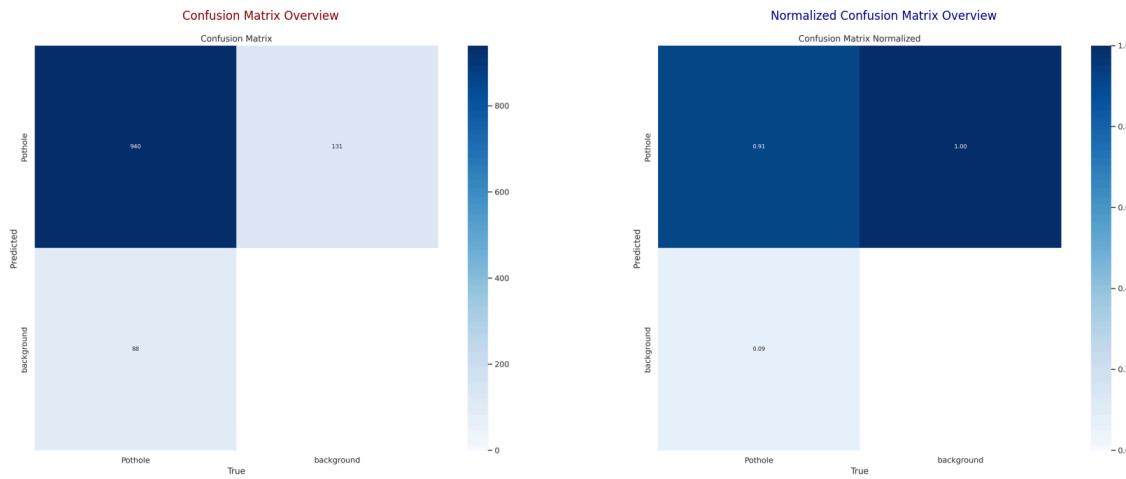
#### 5.2.4 Precision-Recall Curves (Bounding Box and Mask)

The precision-recall curves demonstrate how the model's precision changes as it attempts to increase recall across a range of confidence thresholds. At lower recall, precision is typically higher, meaning the model is very selective and confident in its predictions. As recall increases (catching more potholes), the precision tends to drop because it inevitably includes more false positives.



- Bounding Box Precision-Recall Curve: For the “Pothole” class, the curve shows a great balance between precision and recall across different thresholds, reflected in a reported average precision (AP) around 0.937. This suggests the model can reliably localize potholes over a wide range of recall levels.
- Mask Precision-Recall Curve: Mirrors the bounding box performance but for segmentation masks. The model achieves a similar or slightly higher AP (e.g., 0.940), indicating that it not only locates potholes effectively but also segments them with comparable accuracy.

### 5.2.5 Confusion Matrix



The confusion matrix shows that 91% of actual potholes are correctly identified (true positives) while 9% are classified as background (false negatives). Additionally, the model currently classifies all background instances as potholes, indicating that it tends to over-predict potholes. Although the model demonstrates a significant ability to detect potholes, there is room for improvement in distinguishing between actual pothole features and background regions. Refinements such as augmenting and balancing the dataset with a greater emphasis on diverse background examples can help improve this differentiation and overall model performance.

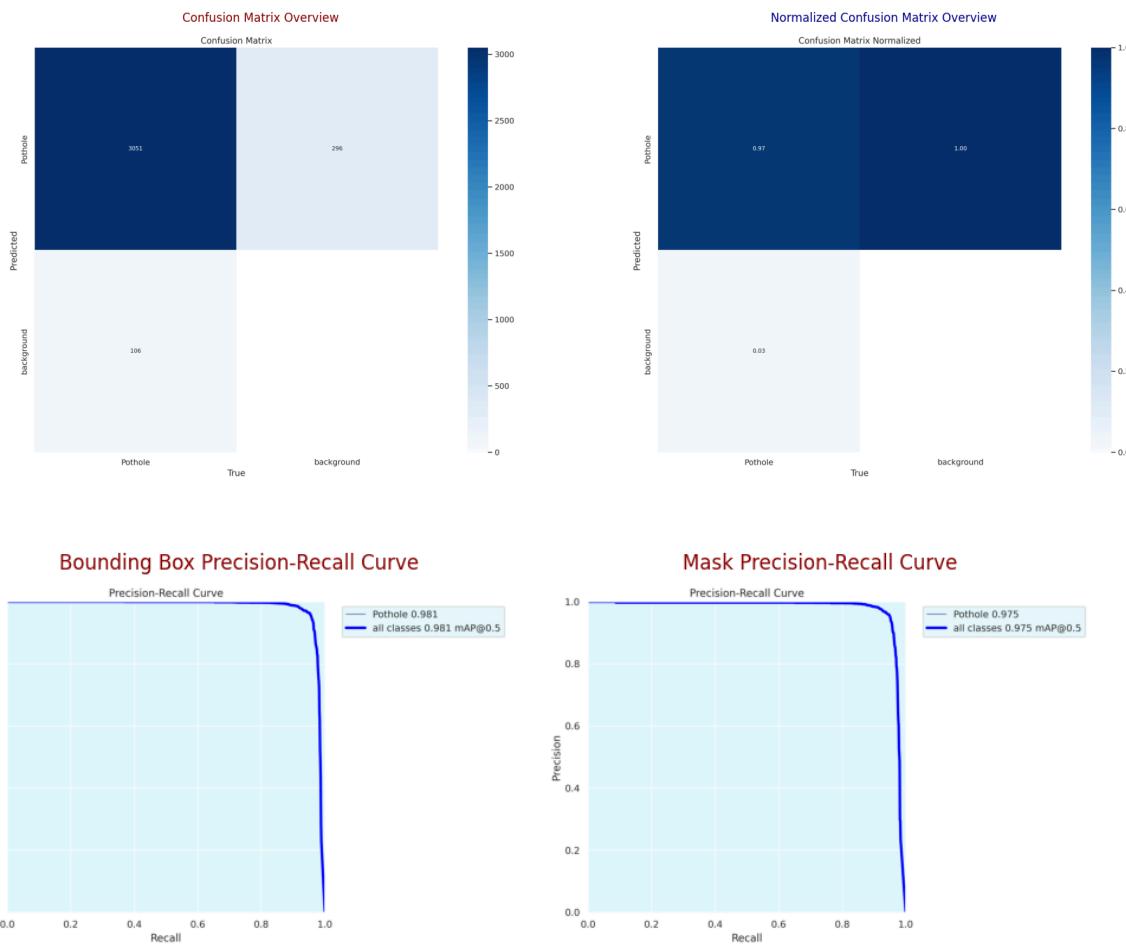
### 5.3 Testing

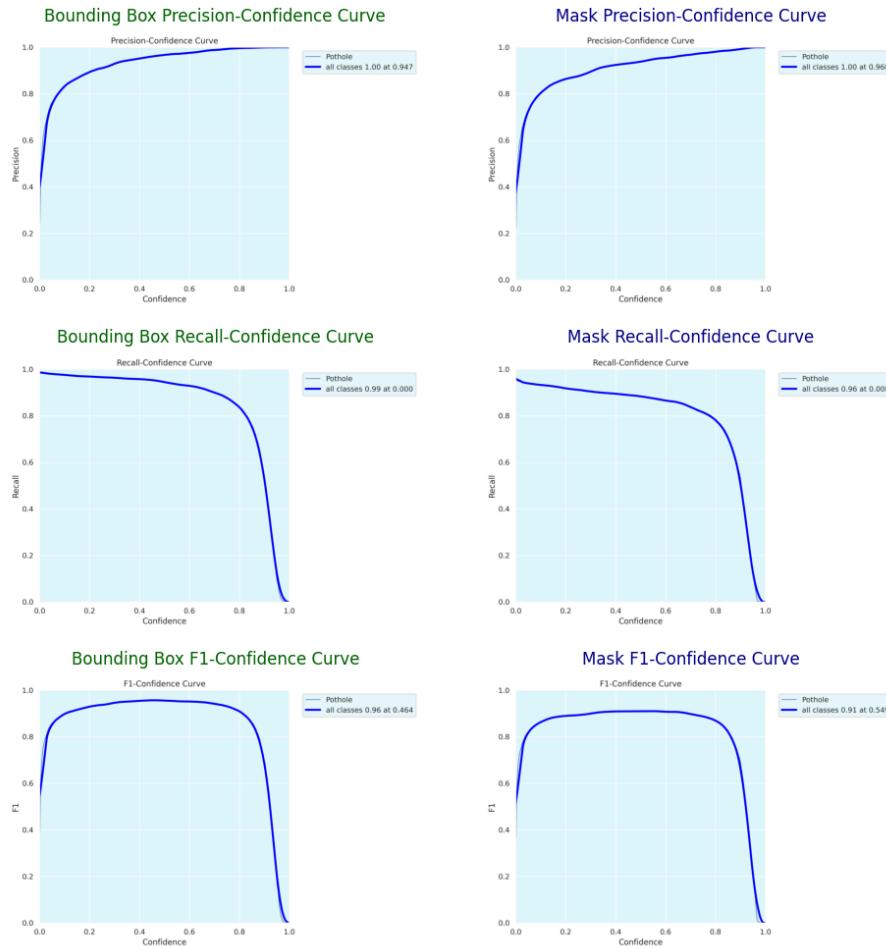
During the evaluation on the testing dataset which comprised 1056 images containing 3157 pothole instances the YOLOv8n-seg model demonstrated solid performance in both detection and segmentation tasks. For bounding boxes, the model achieved a precision of 0.963 and a recall of 0.953, meaning that about 96% of the predicted boxes were correct and it successfully detected approximately 95% of the true potholes. The mean Average Precision (mAP) at an IoU threshold of 0.5 for bounding boxes was 0.981, while the mAP averaged over a range of IoU thresholds from 0.5 to 0.95 was 0.852.

Similarly, for segmentation masks, the model recorded a precision of 0.960 and a recall of 0.944, with mAP50 at 0.975 and mAP50-95 at 0.756. These values indicate that the model is slightly better at producing accurate bounding boxes than segmentation masks when evaluated at a fixed IoU of 0.5, though performance drops slightly under stricter overlap criteria (0.5–0.95 IoU). Collectively, these evaluation results highlight the model's effectiveness in detecting and segmenting potholes.

Metric	Value
metrics/precision(B)	0.963
metrics/recall(B)	0.953
metrics/mAP50(B)	0.981
metrics/mAP50-95(B)	0.852
metrics/precision(M)	0.960
metrics/recall(M)	0.944
metrics/mAP50(M)	0.975
metrics/mAP50-95(M)	0.756

For the testing evaluation, the same set of performance plots as for the training and validation phases were generated including the confusion matrix, confidence-based curves for both bounding boxes and masks, and precision-recall curves for both modalities. Notably, these plots show a marked improvement over those from training and validation. The testing confusion matrix demonstrates a more balanced distribution of true positives and reduced misclassifications, while the confidence-based and precision-recall curves indicate higher precision and recall across various thresholds. This suggests that the model generalizes better to unseen data, achieving more robust and reliable detection and segmentation performance on the testing set.





### 5.3.1 Additional Evaluation Metrics

In addition to standard evaluation metrics, the model's performance was further assessed using Intersection over Union (IoU) and F1-score for both bounding box detection and mask segmentation. IoU measures the overlap between predicted and ground truth regions, while the F1-score evaluates the balance between precision and recall. The evaluation was conducted on the test dataset, where the model achieved a mean Box IoU of 0.890 and a Box F1-score of 0.948, indicating high accuracy in detecting potholes with bounding boxes. For segmentation, the model obtained a mean Mask IoU of 0.845 and a Mask F1-score of 0.900, demonstrating its effectiveness in delineating potholes at the pixel level. These metrics further validate the robustness of the model in both localization and segmentation tasks.

Box IoU: 0.890, Box F1: 0.948  
 Mask IoU: 0.845, Mask F1: 0.900

## 6. Limitations and Future Work

One of the primary limitations of this project was the dataset size and diversity. Initially, only the Kaggle dataset was used, which contained a total of 780 images split into training and validation subsets. However, the model's performance was suboptimal due to the limited number of training examples. To improve the results, the dataset was expanded by incorporating an additional dataset from the research study by R. Swathika et al. (2024) increasing the total number of images significantly. This enhancement led to improved detection and segmentation accuracy.

Despite this improvement, the dataset still lacked sufficient diversity. A major limitation was the absence of background-only images (images without potholes), which are crucial for training the model to correctly differentiate between potholes and non-pothole regions. This lack of negative samples could contribute to false positives, where the model incorrectly detects potholes in non-damaged areas.

Another significant challenge was computational constraints. Initially, the model was trained using Google Colab's CPU, which severely limited training efficiency allowing only 10 epochs over 2 hours. When utilizing the built-in GPU, training efficiency improved significantly, enabling 150 epochs in just 1 hour and 32 minutes. However, Google Colab's GPU resources are quickly exhausted, and acquiring additional GPU time can take days, slowing down the experimentation process.

Additionally, managing a large dataset through Google Drive posed challenges in both storage and accessibility. As the dataset size increased, loading and retrieving large batches of images during training became slower, further limiting model scalability. Future work should focus on acquiring a larger, more diverse dataset that includes background images and exploring alternative computing solutions such as more powerful GPUs, cloud-based computing, or distributed training to enhance both scalability and model accuracy.

### Acknowledgment

I acknowledge the use of Generative AI tools (ChatGPT) to support my learning and understanding of concepts related to model evaluation and architecture. It was also used to enhance the clarity and coherence of the report writing. All content has been critically reviewed and adapted to reflect my own understanding and academic integrity.

## Reference list

- Ahmed, K.R. (2021). Smart Pothole Detection Using Deep Learning Based on Dilated Convolution. *Sensors*, 21(24), p.8406. doi:<https://doi.org/10.3390/s21248406>.
- Bhavana, N., Kodabagi, M.M., Kumar, B.M., Ajay, P., Muthukumaran, N. and Ahilan, A. (2024). POT-YOLO: Real-Time Road Potholes Detection Using Edge Segmentation-Based Yolo V8 Network. *IEEE Sensors Journal*, 24(15), pp.24802–24809. doi:<https://doi.org/10.1109/jsen.2024.3399008>.
- Camilleri, N. and Gatt, T. (2020). *Detecting road potholes using computer vision techniques*. [online] IEEE Xplore. doi:<https://doi.org/10.1109/ICCP51029.2020.9266138>.
- Feng, Z., Guo, Y., Liang, Q., Usman, M., Wang, H., Liu, M. and Sun, Y. (2022). MAFNet: Segmentation of Road Potholes With Multimodal Attention Fusion Network for Autonomous Vehicles. *IEEE Transactions on Instrumentation and Measurement*, 71, pp.1–12. doi:<https://doi.org/10.1109/tim.2022.3200100>.
- Kumari, S., Gautam, A., Basak, S. and Saxena, N. (2023). YOLOv8 Based Deep Learning Method for Potholes Detection. doi:<https://doi.org/10.1109/cvmi59935.2023.10465038>.
- Ping, P., Yang, X. and Gao, Z. (2020). A Deep Learning Approach for Street Pothole Detection. *International Conference on Big Data*. doi:<https://doi.org/10.1109/bigdataservice49289.2020.00039>.
- R S, S.D., A, J.S., S, S. and Van S K, T. (2024). Pothole Detection and Instance Segmentation Using Yolo V8. *2024 International Conference on IoT Based Control Networks and Intelligent Systems (ICICNIS)*, pp.1185–1190. doi:<https://doi.org/10.1109/icicnis64247.2024.10823139>.
- R. Swathika, N. Radha, Parthiban D and Sriram J (2024). Pothole Detection and Road Damage Assessment System Utilizing Image Processing Techniques. pp.269–278. doi:<https://doi.org/10.1109/icici62254.2024.00052>.
- Sowbarnika Thiruppathiraj, Kumar, U. and Swapnil Buchke (2020). Automatic pothole classification and segmentation using android smartphone sensors and camera images with machine learning techniques. pp.1386–1391. doi:<https://doi.org/10.1109/tencon50793.2020.9293883>.
- Wang, A., Lang, H., Chen, Z., Peng, Y., Ding, S. and Jian John Lu (2024). The Two-Step Method of Pavement Pothole and Raveling Detection and Segmentation Based on Deep Learning. *IEEE transactions on intelligent transportation systems*, pp.1–16. doi:<https://doi.org/10.1109/tits.2023.3340340>.