School of computer Science and Engineering,

Vellore Institute of Technology, VIT University, Chennai

POTHOLE DETECTION USING YOLOV8+SQUEEZE AND EXCITATION BLOCK

Dr.PandiyaRaju V[1], Dhanushkumar V[2],Maniesh Pukhraj [3]

1st pandiyaraju.v@vit.ac.in

2nd dhanushkumar.v2022@vitstudent.ac.in

3rd maniesh.pukhraj2022@vitstudent.ac.in

Abstract:

This paper establishes a very accurate system for pothole detection built using the latest YOLOv8 object-detection model and along with this includes an SEBlock for enhancement of the model's detection and robustness. YOLOv8 acts as the core of the real-time detection because it gives out bounding boxes and segmentation masks for the potholes; hence, selection of refinement by the SEBlock in making selective-channel-importance adjustments during inference makes the model pay attention to the most important features.

The augmentation strategy readies the dataset to generalize better. It contains a severity classification mechanism that classifies potholes as mild, moderate, and severe with color-coded bounding boxes and severity scores. The first purpose of this hybrid system is to efficiently work in real conditions with any kind of obstacle created due to varying illumination, cluttered environments, and different road conditions. This model not only detects potholes but also generates actionable insights on the prioritization of road repairs, hence a practical solution to urban management infrastructure.

I. Introduction:

India is the second most populated country in the world, with over 1.5 billion people. It has world's second largest network of roads that come to an impressive 6.3 million kilometers, closely following only the United States. The road network comprises 179,000 kilometers of state highways and 146,000 kilometers of national highways. It has to be built 3-4 times in the near future; therefore, maintenance and monitoring of roads efficiently is an area of great concern.

Road defects, particularly potholes, are very significant issues: a vehicle may suffer serious damage, conditions of driving pose hazards, and, in extreme cases, lead to fatal accidents. Enjoymen t of driving is also compromised along with the safety of the driver. Such problems should be addressed promptly enough to maintain a safe and efficient transportation system.

On the other hand, sheer volumes of India's road network make it absolutely impossible to inspect this in person. The procedure would quite be time-consuming and labor-intensive. What is required is a transition from this extremely outdated manual process into one that is faster, more efficient, and perhaps even automated on a sufficiently large scale as well as allowing timely intervention to enhance road safety and durability.

II. Literature review:

This paper [1] is the development of a mobile application for live pothole detection for visually impaired users. It focuses on the latest detection model, YOLOv5, known for its efficiency and processing speed. The accuracy level achieved is 82.7% for the static images detected, and this model processes live video at 30 frames per second, ensuring real-time detection.This application allows using it in a portable and friendly manner, giving the handicapped visually safe tool navigation. It focuses on real-time pothole detection in high ways, which makes it very beneficial but also has some limitations: the range of detection is relatively short, meaning less time for advance warning. The application is specialized for pothole detection, doesn't currently help with other hazards like debris or obstacles.This research showcases the potential of integrating developed computer vision models like YOLOv5 into assistive technologies, particularly in improving the mobility and safety of visually impaired people.

Detection and evaluation of potholes on a roadway using smartphone accelerometer and gyroscope data, augmented by machine learning algorithms, is explored in this study[2]. Analyzing sensor data, the system provides immediate road condition feedback, with an accuracy of 93% in classifying the varieties of roadway surface and identifying potholes.A major advantage to this approach would be the ability to crowdsource, thus allowing for easy and broad coverage of data to be obtained for a whole roadway mapping system. Such a system could lead to detailed and updated maps of road conditions, improving urban planning and transportation authorities' operations.However, its performance ultimately depends on the quality of smartphone sensors, which can vary dramatically among devices. Second, the success of crowd-sourcing is dependent on the active and consistent participation of users in an effort to maintain the accuracy and relevance of collected data. These notwithstanding, the study goes to demonstrate how utilizing everyday technology can solve some of the most critical infrastructure issues under effective conditions.

This research [3] would introduce "PotSpot" as a participatory sensing system capable of real-time pothole detection and mapping using a custom Convolutional Neural Network. The system is an implemented Android application attached to Google Maps, which will allow users to identify and map potholes during their travel. With the custom CNN accuracy being 97.5% for pothole detection, it is a highly reliable solution.PotSpot will provide real-time mapping, benefiting city authorities in prioritizing road repairs based on current data. With scalability, hence cloud integration, it will more accurately collect much data while holding it centrally, which is perfect for high-density installation within an urban environment.Some limitations are, however, present for the system. The system relies mainly on volunteers for data collection, which may lead to some inconsistencies in the data and may also affect the reliability of the data. In addition, the accuracy of detection is determined by the camera resolution of the user device, which can be vastly different. As a prototype, PotSpot does face issues concerning dependability and efficiency but shows incredible potential in improving road maintenance through community-driven efforts and advanced machine learning techniques.

This work [4] is targeted towards pothole detection, with deep learning approach combining the MVGG16 network, YOLOv5, and then employing it within a Faster R-CNN framework. Hence, with these diluted convolutions, there's better detection accuracy at hand, so the model may obtain high accuracy with a balanced trade-off of inference speed and performance.The proposed system is designed to work in real-time and is therefore suitable for harsh road conditions and various sizes of potholes. Its adaptability will ensure proper detection over different environments, which is essential for practical deployment.There are limits to this approach, however. High computational requirements mean that developing algorithms for deployment on low-power devices, such as smartphones or edge devices, is a bit more complicated. Further, the Micro YOLOv5 models are faster in processing but also have lower precision and recall compared to larger variants. Optimal performance also depends on high-quality input images that might not always be available in practical environments. However, challenges notwithstanding, there is clear evidence of significant advancement towards pothole detection based on the state-of-the-art deep learning techniques.

In this work [5], the authors explore using SegFormer, a transformer-based architecture, for high-resolution pothole detection and segmentation, setting it up to be a promising alternative to U-Net. Results for the segmentation of potholes show high accuracy and greatly reduced false positives and negatives. Its strong performance on various conditions in the environment enables it to be highly adaptable for real-world settings.This system is also cost-efficient because it requires quite low processing power in contrast to other high-resolution segmentation models, enabling broader application. Nevertheless, its use of high-resolution scans for optimal accuracy means its functionality is constrained in poor-quality imaging scenarios. Although efficient, SegFormer requires substantial processing power, limiting its potential to be deployed on mobile devices or low-resource systems. The model also has undergone limited field testing, which means further adaptation and evaluation need to happen for the model to be applicable in different and large environments. However, SegFormer is scalable and will be efficient for detecting and segmenting potholes.

This paper [6] introduces a Multi-feature View-based Shallow Convolutional Neural Network architecture to efficiently carry out road segmentation tasks in real-time applications like driver assistance. Contrary to typical Deep Convolutional Neural Networks, where DCNNs call for a huge amount of computational power and labeled data, MVS-CNN incorporates gradient information as extra input channels to boost feature learning. The model attains 2.7% greater accuracy than a baseline CNN with RGB inputs while outperforming the SegNet in terms of accuracy and efficiency. With training on the datasets KITTI and Cityscapes, the MVS-CNN demonstrates higher segmentation accuracy and faster processing, hence perfect for application to autonomous vehicle systems.

This paper [7] presents a system for the detection of ships and oil spills using side-looking airborne radar images. The proposed method utilizes a two-stage architecture that consists of three pairs of convolutional neural networks (CNNs). For each couple of networks, two steps are followed: the first network conducts a coarse detection, and then it was the role of a second specialized CNN to obtain the precise localization of the pixels belonging to each class (ship, oil spill, and coast). After classification, a postprocessing stage is applied through the application of a morphological opening filter to eliminate small look-alikes as well as the removal of oil spills and ships surrounded by a minimum amount of coast. Data augmentation is also performed to increase the number of samples, since collecting a sufficient number of correctly labeled SLAR images is difficult. The approach is evaluated and compared with that of a single multiclass CNN architecture and previous state-of-the-art methods, using accuracy, precision, recall, F-measure, and intersection over union. The results demonstrate that the proposed method is efficient and competitive and outperforms the approaches previously applied to this task.

This research paper [8] explains us blind people will face challenges such as uncharted routes and potholes that may lead to injuries and impaired mobility. This research suggests the use of YOLO in real-time pothole detection through a smartphone application, where the application may provide auditory or haptic feedback for safe clearing. The dataset is trained on a model of 82.7% accuracy in images and 30 FPS in live video. Although it is very sensitive to detect potholes close at hand, the system currently only detects potholes and can be a portable aid for visually-impaired users.

The paper [9] is on computer vision and deep learning for pothole detection. As part of it, a modified VGG16 network-MVGG16 is proposed to reduce the computational costs and maximize the accuracy-thus, becoming the backbone for the Faster R-CNN network. The paper further explores the performance comparison of various models such as YOLOv5 (Large, Medium, and Small), ResNet101, among others, for real-time detection of potholes. Results indicate that the YOLOv5 Small model, Ys, is better suited for real-time detection because it is fast, and the MVGG16 in Faster R-CNN offers better precision along with faster inferences compared to other models. Thus, this approach balances the accuracy and speed for pothole detection effectively.

This research [10] improves pothole and traffic sign detection for advanced driver assistance systems and autonomous vehicles. Current methods lack detection of water-filled, illuminated, or tree-covered potholes and in distorted or nighttime traffic signs. The proposed approach combines a cascade classifier for pattern recognition with a deeper analysis stage by a Vision Transformer, to improve identification capability over such challenging cases. Trained on ICTS, GTSRDB, KAGGLE, and CCSAD datasets, it surpasses state-of-the-art methods like YOLOv3, YOLOv4, Faster R-CNN, and SSD by attaining 97.14% mAP for traffic sign detection and 98.27% mAP for pothole detection; more accuracy and faster response capability are available.

III. Proposed Model for Pothole Detection:

It brings along deep learning techniques for making an effective and automated pothole detection system. Thus, the proposed system will break free across the limits of manual inspection in terms of scalability, accuracy, and real-time processing. This model contains the following:

YOLOv8 for Real-Time Detection

YOLOv8 is a sophisticated algorithm in object detection, which can be thought of as the backbone for the task of detection.It attains high accuracy with an additional advantage of fast bounding box predictions. This algorithm is extremely suitable for real-time applications.It can feel potholes and will give information in its locality.

Integration of SE Block

An SEBlock was appended to supplement the produced YOLOv8 feature maps.A dynamic channel-wise weights implementing SEBlock makes features important while suppressing unimportant ones.This proposed method enhances the detection robustness as the adverse conditions are due to changing illumination and cluttered road surfaces.

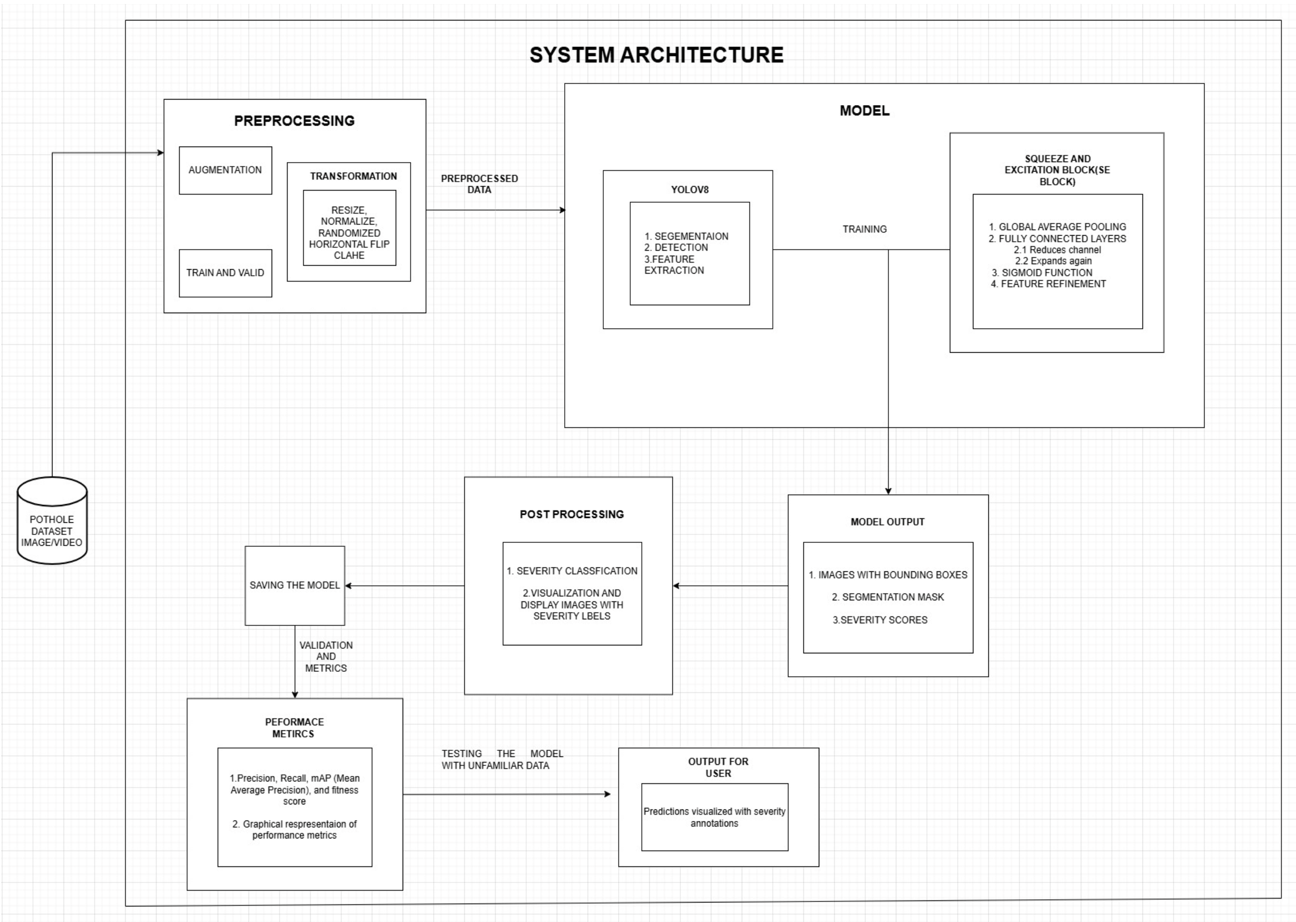
IV. Proposed work:

4.1 Overview:

Our proposed work introduces a robust and efficient pothole detection system that integrates SEB with YOLOv8, enhanced by fine-tuning and data augmentation strategies. Unlike the traditional models which, at the time of feature extraction, are nearly reliant on CNN architectures, including ResNet, VGG16 and VGG19, all of which work pretty well in terms of learning a local feature but carry an insufficient global context, our proposed model relies on more powerful techniques from SEB and YOLOv8 to overcome such flaws.This SEB is integrated into feature extraction to refine the process. The channel-wise feature responses are then adapted through recalibration, which pushes more relevant features into detection. This avoids the distracting influences that the model might receive from the environment, therefore allowing it to instead concentrate on the most information-rich area in the image.Even though CNNs are very strong and efficient at detecting local patters,they fail to understand the higher spatial information,which is very important for accuracy and efficiency.The SEB module bridges this gap by providing a more global understanding about the image, enhancing the model's ability to predict potholes even in a complex or cluttered environment.

Then, the real-time object detection model, namely, state-of-the-art YOLOv8 really quickly and with less computing detect potholes within live video feeds or images. One of the most desirable alternatives in terms of speed and computational efficiency for fast and efficient detection in high-performance, real-time systems is YOLOv8. Using a specialized dataset for fine-tuning YOLOv8 and data augmentation with multiple techniques enhances the model's robustness and the ability of the model to generalize across different conditions of roads and environments.Besides detecting the potholes, our system is capable of calculating the severity score for each based on its size. The severity score given to each pothole is an actionable metric for determining the urgency and the potential impact of every pothole found. Based on the size of the detected pothole, we will be able to give a severity score that will help in prioritizing the maintenance in place, ensuring that the dangers are rectified first.Overall, our method hence integrates the strength of feature refinement with SEB, the real-time detection capability of YOLOv8, and severity scoring for actionable insight, which other models could not achieve. The ability of such advanced techniques to integrate into a system that is both more accurate and efficient enough for large-scale, real-time deployment and hence suitable for applications in smart city infrastructure and road maintenance.

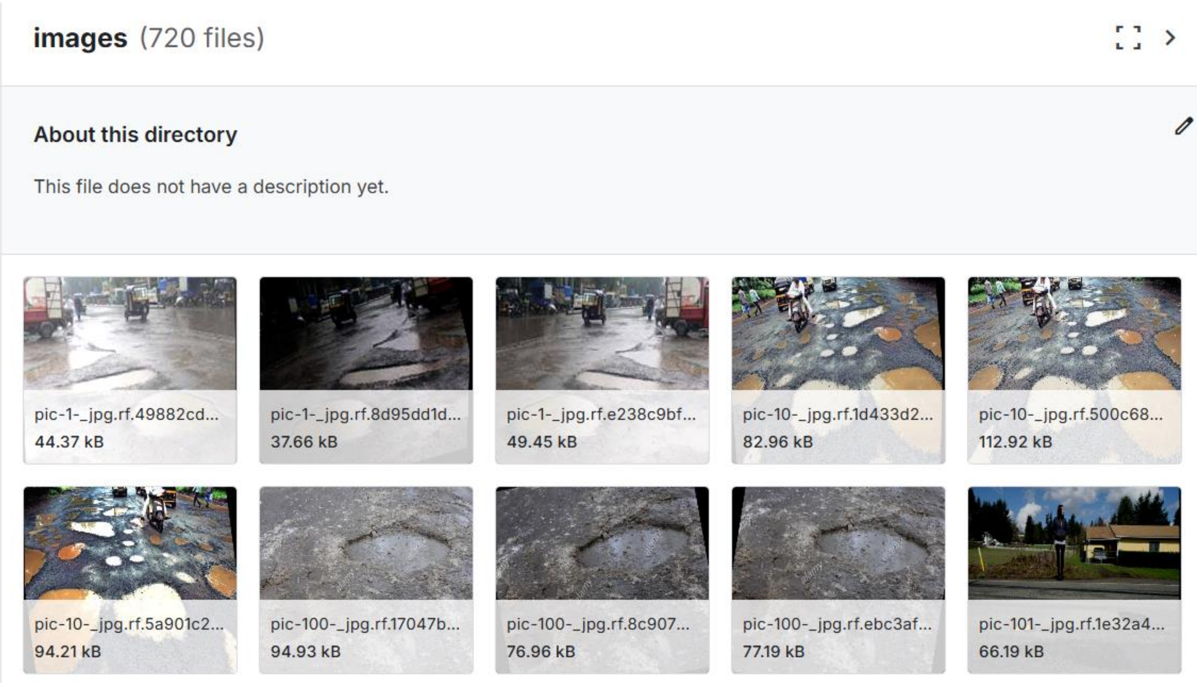
4.2 Overall proposed architecture:



Preprocessing Module: It works to prepare the dataset for the model. Pre-processing procedure begins with augmentation where the dataset is enhanced by the application of setting transformations in rotation, scaling and flipping so as to make model more robust. Under transformation, processes of resizing of images and pixel values normalization, image randomization, horizontal flips, and Contrast Limited Adaptive Histogram Equalization (CLAHE) which improve feature while clumping data are all standardization procedures that are aimed at accomplishing. The last step in this phase is the split of dataset into training and validation subsets, this provides room for developing and testing models effectively.

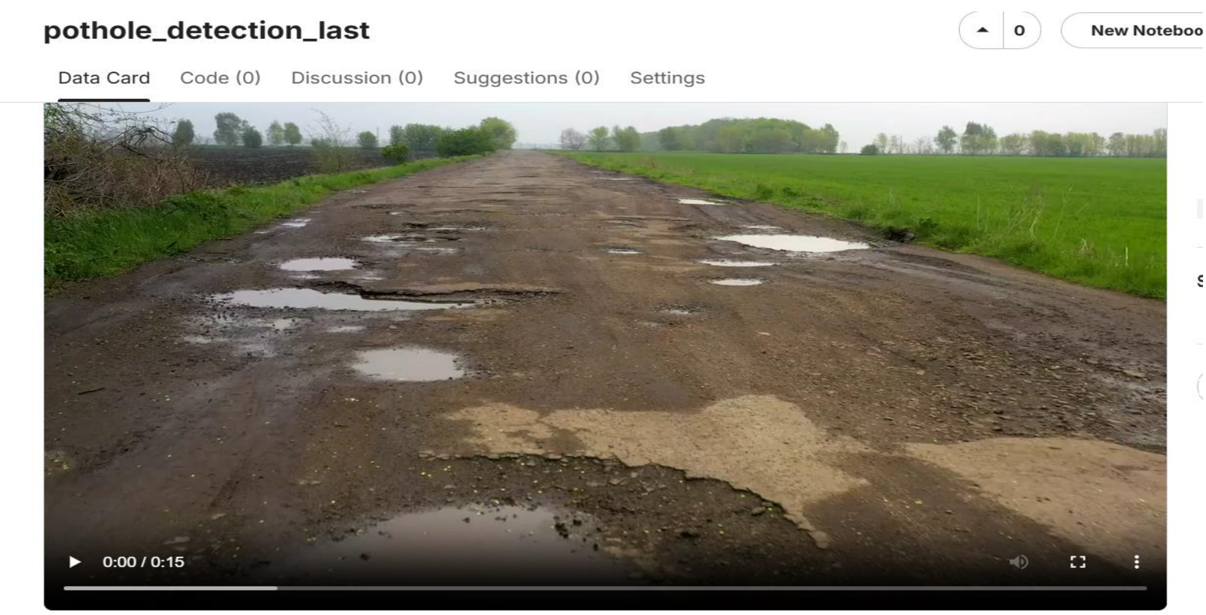
Model module consists of the YOLOv8 and Squeeze and Excitation block that has been configured to make the system efficient in detecting potholes and extracting features. Segmentation is accomplished by YOLOv8 to identify and cut out potholes, detection to pinpoint the coordinates of these potholes in the images, and feature extraction which entails, pulling useful information that would be needed for the identification and classification to be accurate. This SE block enhances the ability of the model to due to average global pooling, After shrinking the dimensions of space into a number where it only assesses the important features only. So that after that fully connected layers increase and decrease size of the feature space to fine the output. The important features are then marked using the sigmoid of to facilitate the maximizing of accuracy.

4.3 Dataset exploration:



Training and Validation Dataset link: https://www.kaggle.com/code/farzadnekouei/pothole-segmentation-for-road-damage-assessment

Testing Dataset: https://drive.google.com/drive/folders/1CPzyhQq5FI28d0mqm0p-sGkYrWraaDKY



Our dataset aims to provide sufficient training and evaluation of the pothole detection model. It consists of 720 images with annotations that will be used in the training set, and 60 images with labels belonging to the validation set. Each image is also annotated with YOLO protocol, which marks each object of interest (potholes in this case) with its bounding box and class, allowing the model during training to learn to spot objects in precise locations. The images include various road conditions and pothole types so that the model does not only learn a single pattern or feature.

Apart from still images, a video for real-time pothole detection is also contained within the dataset. This enables us to effectively put the model to the test using various real-time moving situations and evaluate how well the model performs in situation where pothole detection is deployed in real-time continuous videofeed but with dynamic environments and inconsistent light conditions.

Moreover, for evaluation purposes, we possess a set of 10 random images, which were not utilized in either the training or validation. These images would show different kinds of roads and settings which will be perfect for testing generalization. We can evaluate the model based on this data unseen to the model and, therefore, find how effective its performance is in identifying potholes in instances it has not been trained on to measure its reliability and dependability on accuracy in the real world.

4.4 Modules of the proposed work:

1. Augmentation Pipeline:

This step is crucial in augmenting the model's generalization capability by artificially enlarging the size of the training dataset. Below are the kinds of augmentations that were applied to the augmentation process:

RandomCrop:
The image is randomly cropped into a size of 640 x 640. This creates several viewports for a pothole given different positions within the image, allowing the model to learn to detect potholes in various locations.

Rotate:
The image is randomly rotated up to 10 degrees. This simulates the minor deviations that happen in the direction of potholes and road conditions, helping the model generalize better for different perspectives.

HorizontalFlip:
With a probability of 50%, the image is flipped horizontally. This increases the size of the database by generating mirror images of the potholes and, therefore, enhances the model's ability to detect potholes from any direction on the road.

CLAHE(Contrast Limited Adaptive Histogram Equalization):
CLAHE increases the visibility of potholes under varying lighting conditions. It locally adjusts contrasts in the image, thus improving the quality of potholes, especially under poor lighting or high-contrast situations.

2. SE Block (Squeeze and Excitation Block):

The SEBlock aims to enhance the feature representation by recalibrating the feature map. Here's how it works:

Global Average Pooling:
The SEBlock uses global average pooling, which returns a single feature vector for every channel.

This captures the global context of the image, which is crucial when distinguishing between a pothole and the background or surrounding objects.

Fully Connected Layers (fc1 & fc2):
The first fully connected layer reduces the number of channels by a factor of 16, and the second restores the original number of channels. These layers help learn which features are most important by reweighting them.

Sigmoid Activation:
A channel-wise attention map is generated using a sigmoid function, which is then multiplied by the input feature map. This allows the model to emphasize pothole-like features while diminishing less relevant features, such as road textures or surrounding vehicles.

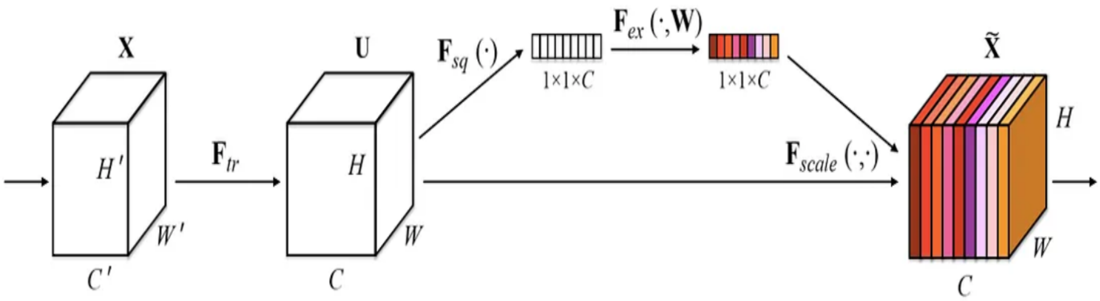


FIGURE 1 :MEDIUM

3. Configuration of the YOLOv8 Model:

Pre-trained YOLOv8 Model:
The YOLOv8 model is configured with pre-trained weights, which have learned general features from a large dataset. These weights can be fine-tuned for specific tasks like pothole detection.

Training Configuration:
The model is trained on a custom-defined dataset located in the data.yaml file. This dataset consists of images of potholes and their corresponding annotations in the YOLO format.

Hyperparameters:
The model is trained for 150 epochs with a batch size of 16. It uses the AdamW optimizer, with a learning rate of 0.0001 and a learning rate scheduler (lrf=0.01).

Patience:
The patience is set to 15, which means that if the model cannot improve beyond this threshold for 15 epochs, training will stop early.

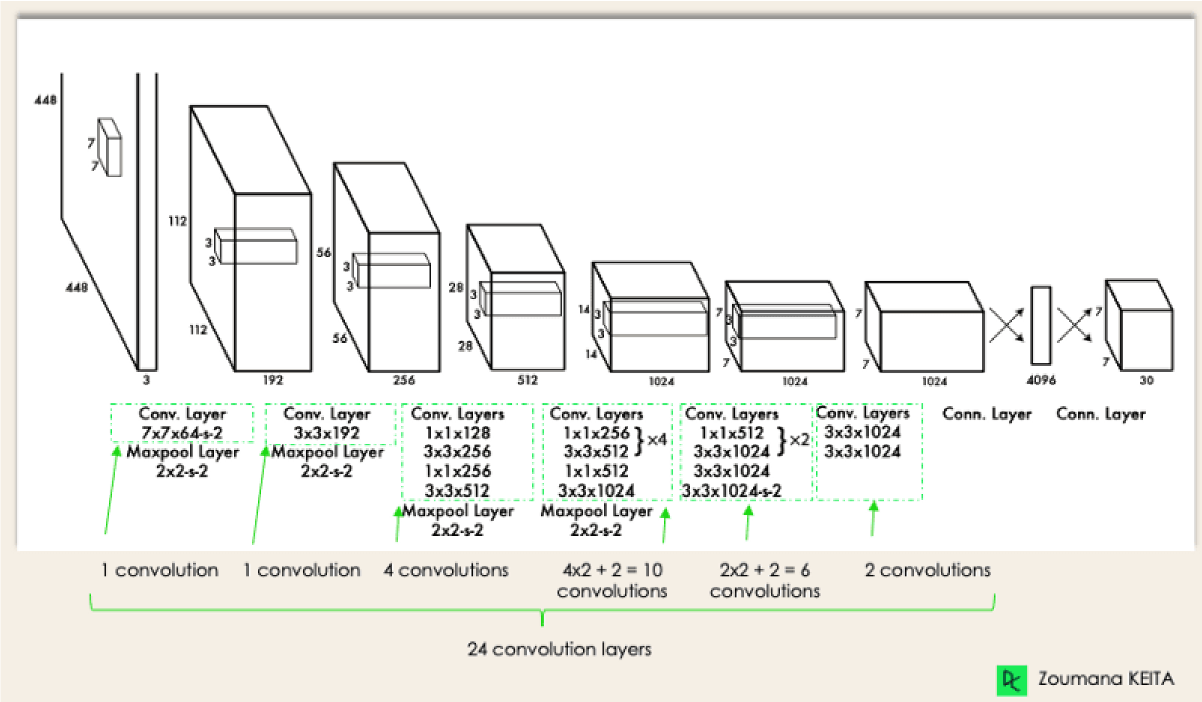


FIGURE: 2 FROM DATACAMP

4. SEBlock At Inference:

Wrapper Class for Inference:
During inference, the wrapper class YOLOWithSEInference is used to integrate the SEBlock. While YOLOv8 performs well for object detection, the SEBlock refines the feature maps, improving the precision of pothole detection.

Prediction Process:
The wrapper class accepts an input image and passes it through the YOLOv8 model to obtain the primary predictions. After that, the SEBlock enhances these predictions by rebalancing the feature maps.

5. YOLO Format Annotations to Bounding Box Conversion

Iterating through the YOLO format of annotations in (center x, center y, width, height) and transforms YOLO format into the (x1, y1, x2, y2) coordinate format where it refers to the boundary box's top-left and bottom-right corner coordinate.
Conversion Formula:
For an image of width W and height H:
YOLO to pixel coordinates:

$$x_{center} = x_{yolo} \times W$$

$$y_{center} = y_{yolo} \times H$$

$$box\_width = width_{yolo} \times W$$

$$box\_height = height_{yolo} \times H$$

Bounding Box Conversion:

$$x1 = x_{center} - \frac{box\_width}{2}$$

$$y1 = y_{center} - \frac{box\_height}{2}$$

$$x2 = x_{center} + \frac{box\_width}{2}$$

$$y2 = y_{center} + \frac{box\_height}{2}$$

YOLOv8:

YOLO is an object detection model that predicts bounding boxes and class probabilities in one pass.It divides the image into grids and predicts multiple bounding boxes with class label.

Localisation loss:

$$L_{loc} = \sum 1_{\{i \in Object\}} [(\bar{x}_i - x_i)^2 + (\bar{y}_i - y_i)^2 + (\bar{w}_i - w_i)^2 + (\bar{h}_i - h_i)^2]$$

Confidence loss:

$$L_{loc} = \sum 1_{\{i \in Object\}} [\bar{c}_i - c_i]^2$$

Classification loss:

$$L_{loc} = \sum 1_{\{i \in Object\}} [\bar{p}_i - p_i]^2$$

6. Evaluation:

Precision (B) for bounding boxes:

$$Precision(B) = \frac{TP}{TP + FP}$$

Recall (B) for bounding boxes:

$$Recall(B) = \frac{TP}{TP+FN}$$

mAP50 (B) for bounding boxes:

$$mAP50(B) = \frac{1}{N}\sum_{i=1}^{N} AP50\,(i)$$

mAP50-95 (B) for bounding boxes:

$$mAP50-95(B) = \frac{1}{N}\sum_{i=1}^{N} AP50-95\,(i)$$

Precision (M) for masks:

$$Precision(M) = \frac{TP}{TP+FP}$$

Recall (M) for masks:

$$Recall(M) = \frac{TP}{TP+FN}$$

mAP50 (M) for masks:

$$mAP50(M) = \frac{1}{N}\sum_{i=1}^{N} AP50\,(i)$$

mAP50-95 (M) for masks:

$$mAP50-95(M) = \frac{1}{N}\sum_{i=1}^{N} AP50-95\,(i)$$

Fitness (overall model score):

Fitness=Combined performance metric

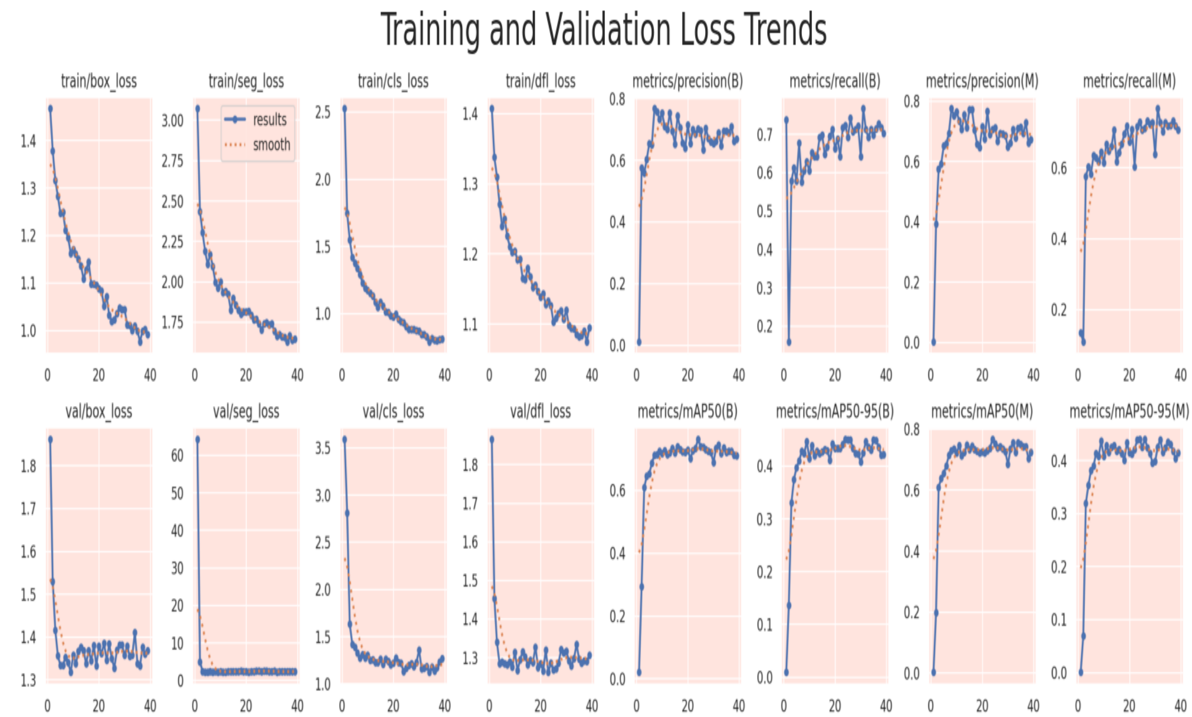
Where:

- TP = True Positives
- FP = False Positives
- FN = False Negatives
- AP50 = Average Precision at IoU 50%
- AP50-95 = Average Precision across multiple IoU thresholds

V. Experimental Results and Discussion:

PROCESSOR	Intel i5 13^{th} gen
RAM	30GB
Environment	Kaggle notebook
Frameworks	PyTorch, TensorFlow,Ultralytics,cv2,numpy,deque,PIL,tqdm,YOLO,m

	atplot,pandas,seaborn,random, yaml
Model	YOLOv8,Squeeze and Excitation block



Loss Metrics:

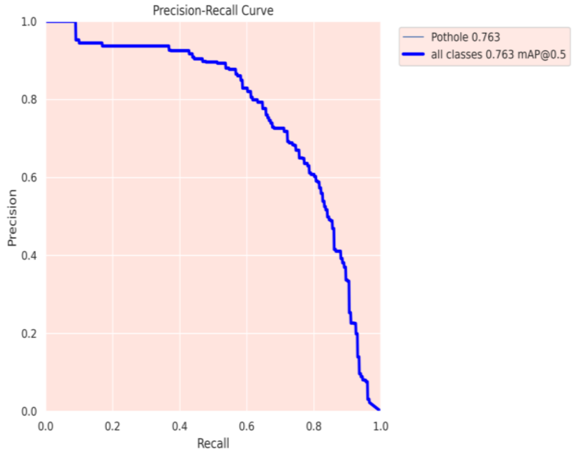
Box Loss: The Accuracies at box loss localization can be visualized. Train/box_loss and val/box_loss both decline, which indicates better location of objects through training.

Segmentation Loss: It monitors segmentation performance; the train/seg_loss and val/seg_loss losses explains the decreasing behavior consistently, which indicates better pixel-wise classification.

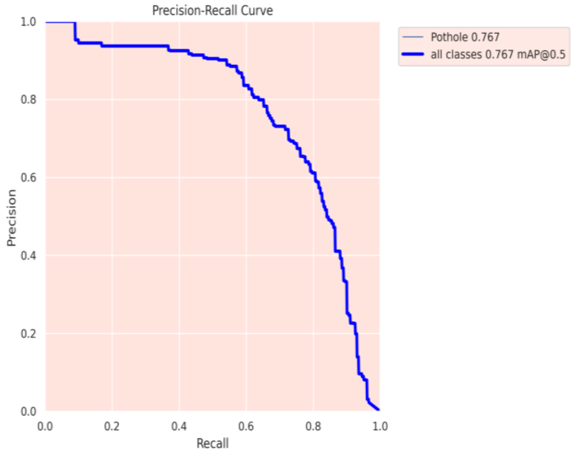
Classification Loss: It drops for both training and validation (train/cls_loss and val/cls_loss), meaning better object classification.

Distance Focal Loss: This loss(train/ dfl_loss and val/dfl_loss) also decreases indicating better precision in bounding box placement.

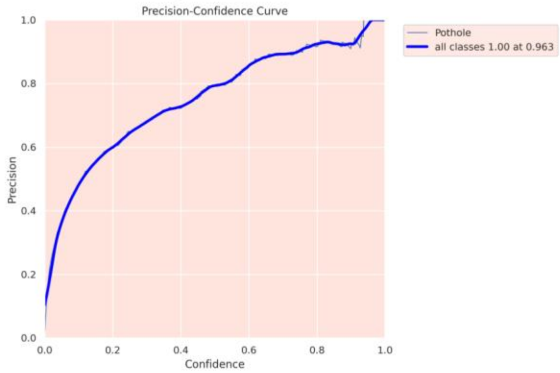
Bounding Box Precision-Recall Curve



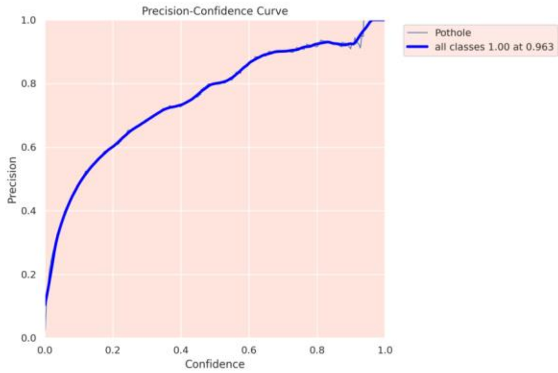
Mask Precision-Recall Curve



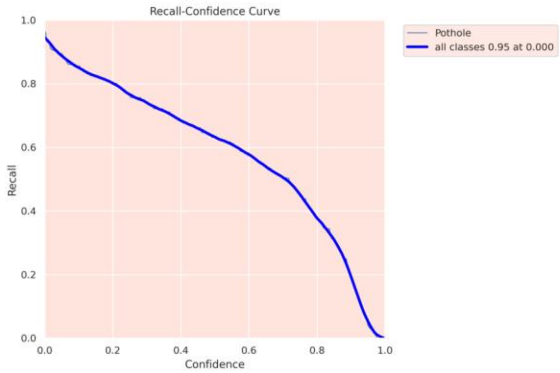
Bounding Box Precision-Confidence Curve



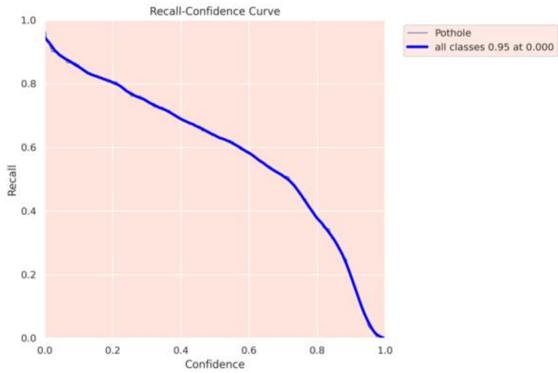
Mask Precision-Confidence Curve



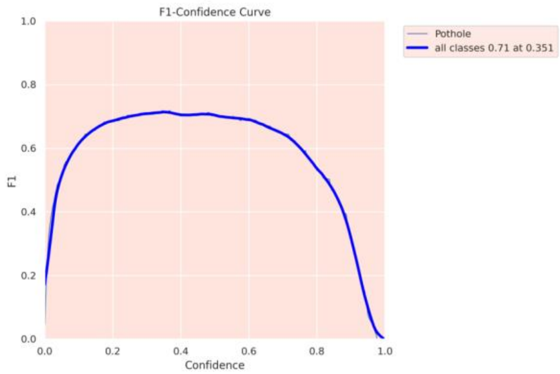
Bounding Box Recall-Confidence Curve



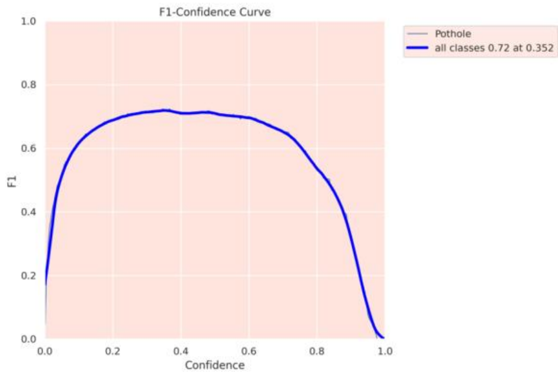
Mask Recall-Confidence Curve



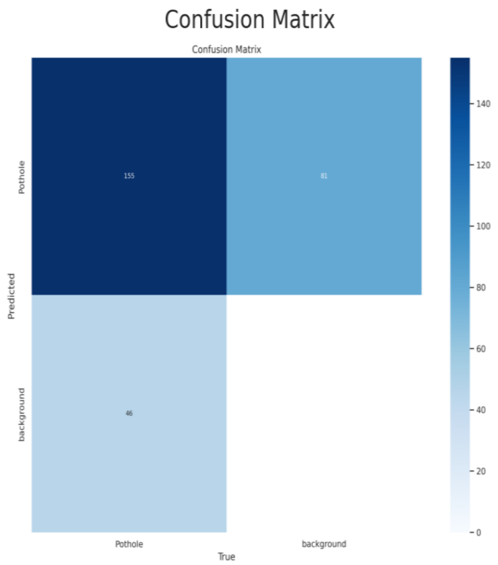
Bounding Box F1-Confidence Curve

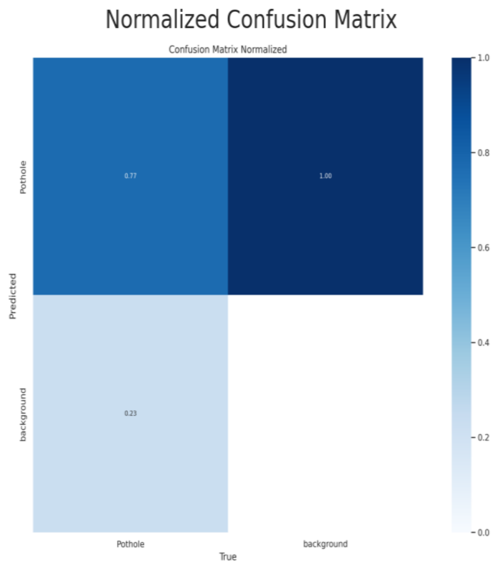


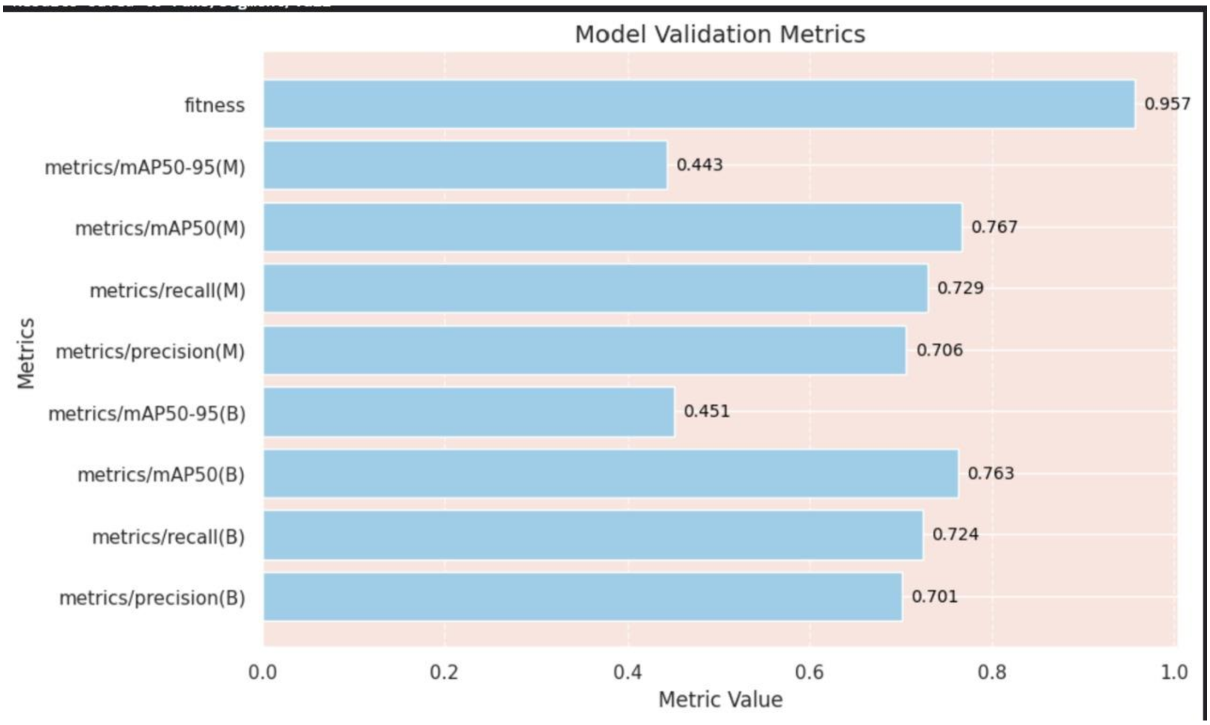
Mask F1-Confidence Curve



The graphs represent the performance analysis of the pothole detection and segmentation model. Bounding Box Precision-Confidence Curve and Mask Precision-Confidence Curve illustrate that the precision is highly higher for high-confidence ranges, which shows that the model predicts correct results at higher levels of confidence. Bounding Box Recall-Confidence Curve and Mask Recall-Confidence Curve represent that there is a decrease in recall as thresholds are increased, implying fewer true positives detected at stricter thresholds. Bounding Box F1-Confidence Curve and Mask F1-Confidence Curve The F1 score is plotted versus optimal confidence thresholds as a curve and it peaks at the thresholds that balance precision and recall. The best trade-off between the two metrics is given. Bounding Box Precision-Recall Curve and Mask Precision-Recall Curve The trade-offs between precision and recall are presented as this trade-off is achieved by monotonic decreases in precision with increasing recall. The average mean precision at a threshold of 0.5 IoU is 0.763 for the detected bounding boxes and 0.767 for the segmentation mask, suggesting effective performance in detection and segmentation while balancing appropriate amounts of false positives and false negatives between the tasks.







5.1 Evaluation Metrics:

	Metric Value
metrics/precision(B)	0.701
metrics/recall(B)	0.724
metrics/mAP50(B)	0.763
metrics/mAP50-95(B)	0.451
metrics/precision(M)	0.706
metrics/recall(M)	0.729
metrics/mAP50(M)	0.767
metrics/mAP50-95(M)	0.443
fitness	0.957

Performance Metrics:

Precision & Recall for Bounding Box (B):We got a high and stable precision & recall for bounding box detection with the model avoiding false positives and false negatives object detection.

Precisión & Recall for Segmentation (M): We can see that the precision and recall levels also stabilize at high values for segmentation, provides a proper segmentation with minimal errors.

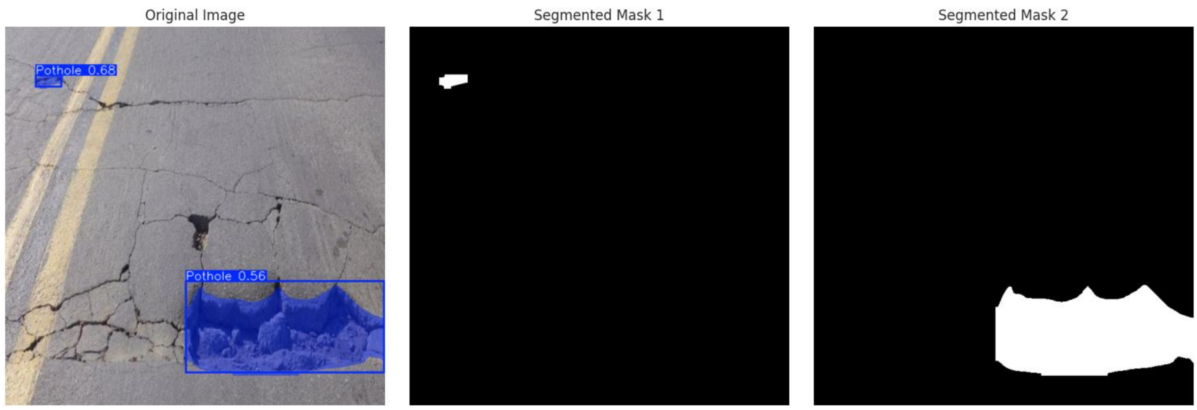
Mean Average Precision (mAP):

Bounding Box (B) and Segmentation (M) (mAP50 and mAP50-95): These two metrics check how good the model detects objects under various IoU thresholds. High mAPs mean that it is performing well at most precision thresholds to detect and segment the objects.

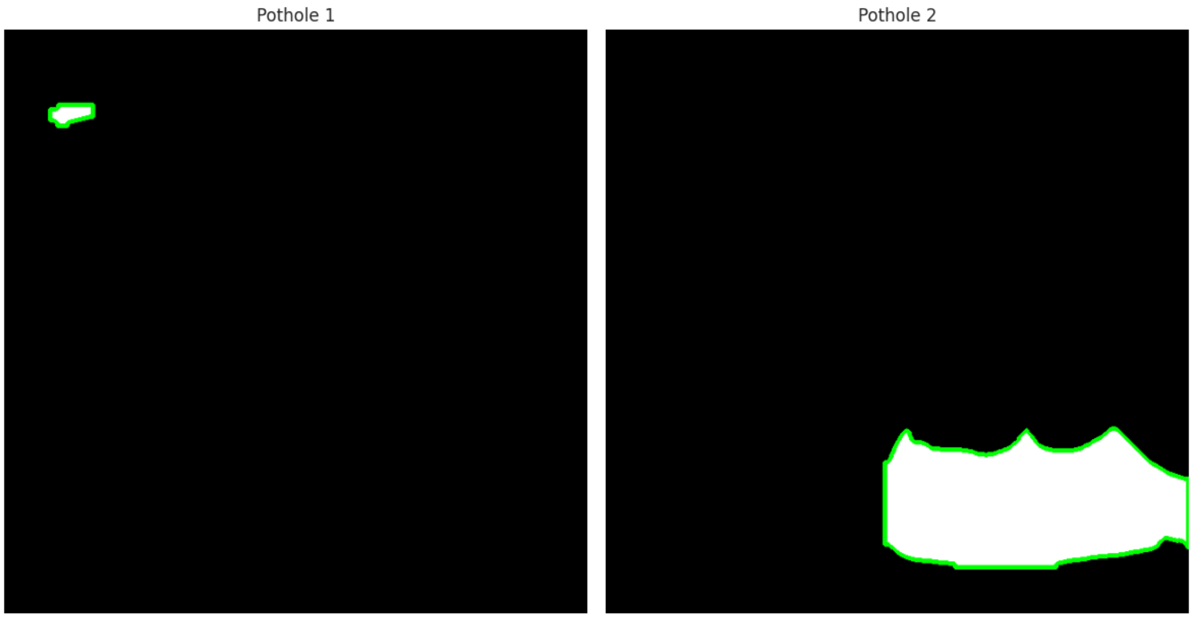
1. Precision: Measures correct predictions out of total positive predictions.
2. Recall: Identifies true positives out of actual positives.

3. F1-Score: Balances Precision and Recall for a single performance measure.
4. Precision-Recall Curve: Shows the trade-off between Precision and Recall.
5. Confidence Curves: Analyze model performance across confidence thresholds.
6. mAP (mean Average Precision): Evaluates overall accuracy at IoU 0.5 for detection and segmentation.

5.3 Results and Output



Predicting the road damage



Area of Pothole 1: 798.0 pixels

Area of Pothole 2: 39975.5 pixels

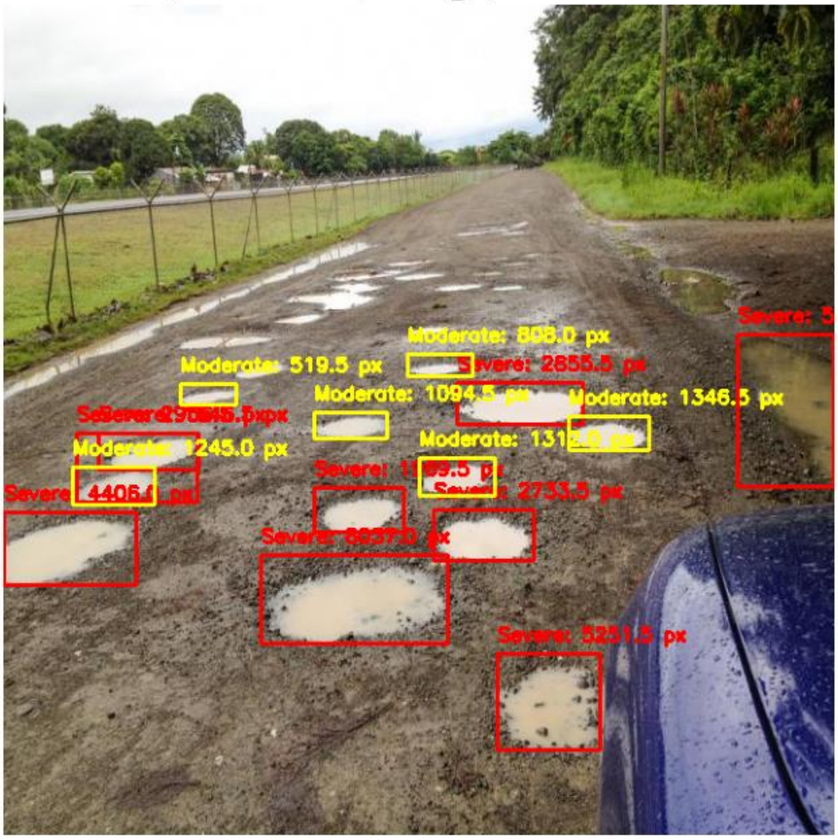
Total Damaged Area by Potholes: 40773.5 pixels

Total Pixels in Image: 409600 pixels

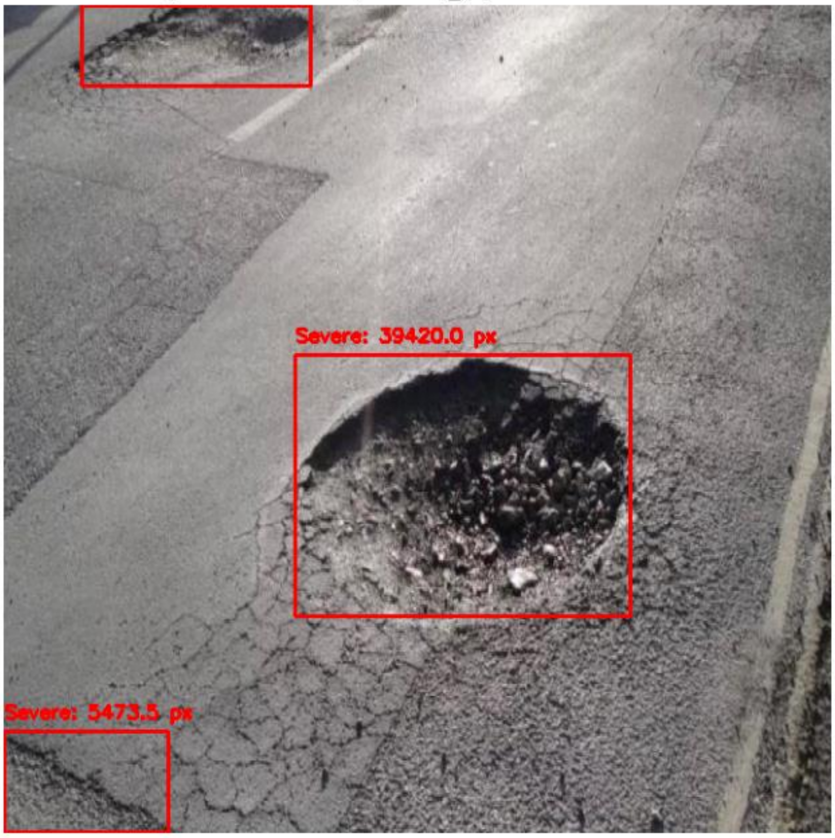
Percentage of Road Damaged: 9.95%

Classifying the severity of the pothole using the pixel area detected:

Pothole Detection with Severity Classification - pic-157-_jpg.rf.2247b000d655232fbf8a58b5add102ca.jpg



Pothole Detection with Severity Classification - pic-151-_jpg.rf.45a82558c583efe9dda4ea02e294595a.jpg

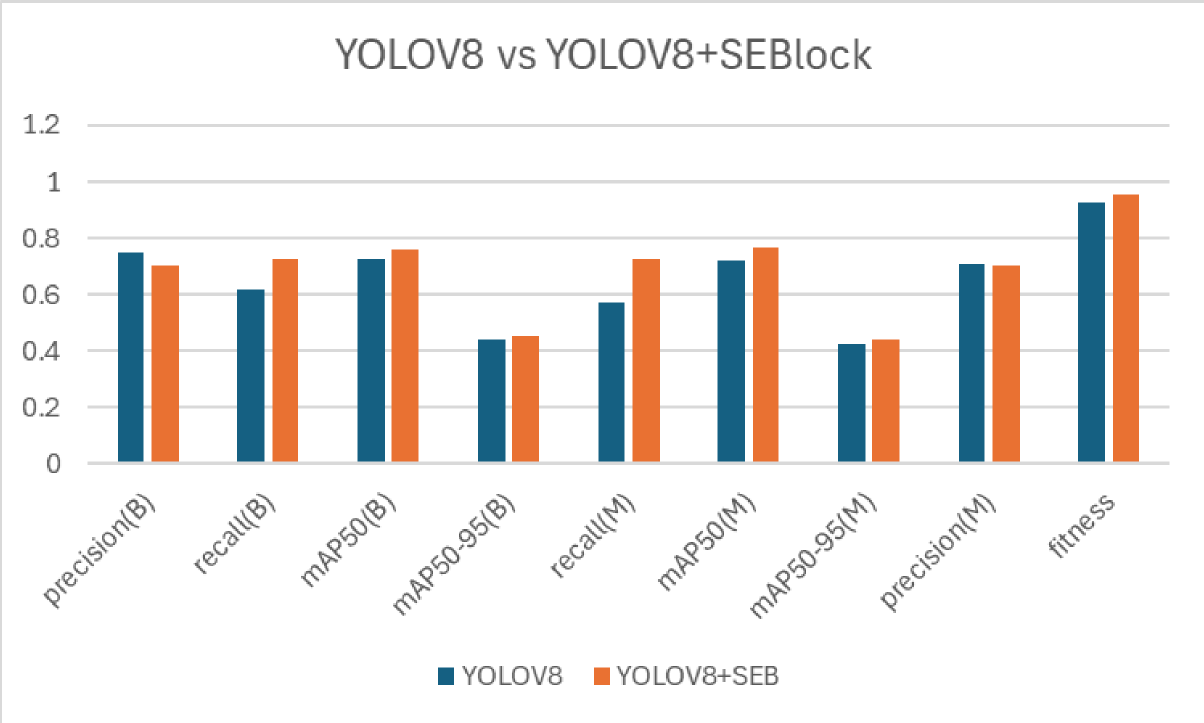


Pothole Detection with Severity Classification - 2.jpg

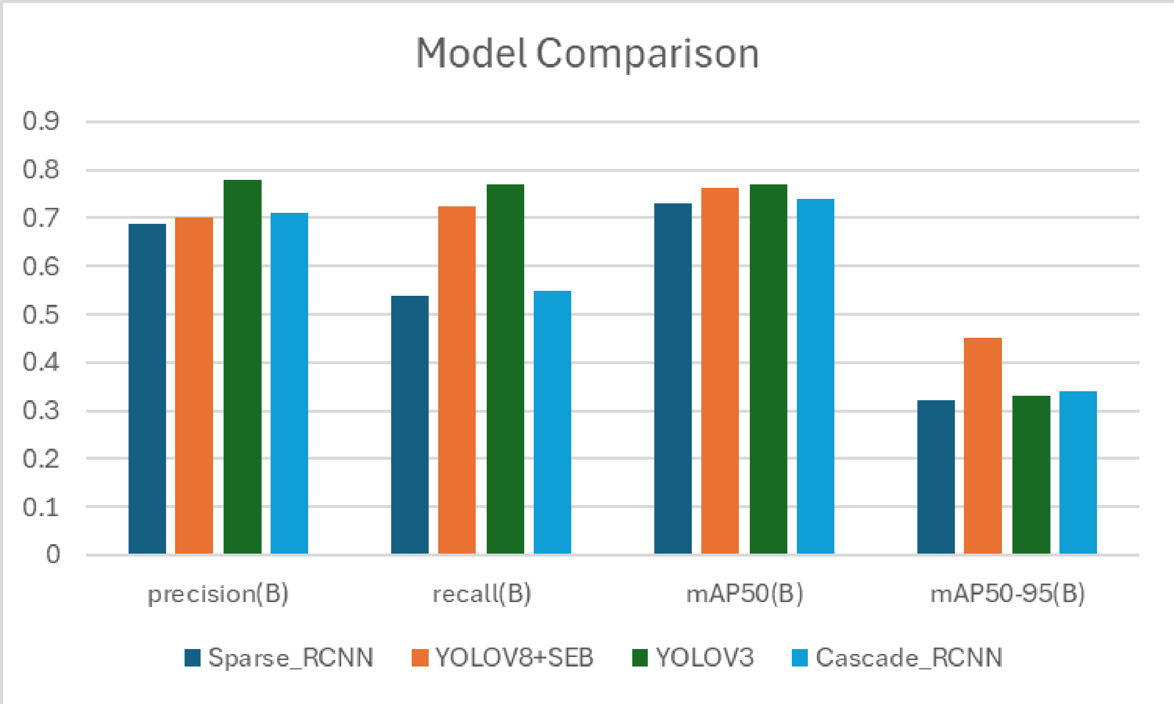
5.2 Comparative Analysis:

YOLOv8 VS YOLOv8+SEB:

Comparing the relative strengths across all the different performance metrics, YOLOv8 and YOLOv8+SEBlock show minimal differences in precision and accuracy while achieving slightly greater precision. There is almost no effect on the metrics when the SEBlock is integrated. On the other hand, by integrating the SEBlock in YOLOv8+SEBlock, sensitivity or the recall is greatly improved along with achieving mAP50 and mAP50-95 values better than YOLOv8.For models of medium size, YOLOv8+SEBlock achieves a much higher recall for mAP50 and mAP50-95 while achieving superiority in object recognition and localization. While YOLOv8 itself maintained the lead in precision, the highest fitness score of YOLOv8+SEBlock demonstrates a better tradeoff between precision and recall.YOLOv8+SEBlock would be more suitable where higher sensitivity and generalized performance are desired, but YOLOv8 would be more favored where precision is favored. In a nutshell, YOLOv8+SEBlock offers a well-balanced detection model, particularly where the target task is recall and mAP privileged, and YOLOv8 is sound when precision is quite critical.



Interestingly, comparison of all four models, Sparse_RCI, YOLOv8+SEBlock, YOLOv3, and Cascade_RCI, along several metrics gives insight. For Precision (B), YOLOv3 leads the race with the highest precision, 0.78, that it performs quite well on true positives; the other two models are pretty close: YOLOv8+SEBlock is at 0.701 and Cascade_RCI at 0.71 while Sparse_RCI trails at 0.69. For Recall (B), the YOLOv3 has 0.77, then the YOLOv8+SEBlock have 0.724. Thus, these are best suited for use in applications where high recall has to be achieved, besides the efficient true positive detection. Sparse_RCI and Cascade_RCI possess much lower recall values with 0.54 and 0.55 respectively, thus hampering their utility in such applications.In mAP50 (B) for detecting or localizing objects with an IoU threshold of 50%, YOLOv3 at 0.77 was almost of the same strength, closely matched to YOLOv8+SEBlock at 0.763. Decent results were also shown by Cascade_RCI (0.74) and Sparse_RCI (0.73), but these trailed behind the best models.Notably, the mAP50-95 (B) metric, which generalizes over an extremely large range of IoU thresholds, clearly points to YOLOv8+SEBlock as the best of all at 0.451, which demonstrates superior robustness and adaptability. Compared to YOLOv3 at 0.33, Cascade_RCI at 0.34 and Sparse_RCI at 0.32, it seems underperforming in terms of generalization.In conclusion, YOLOv3 is used when there is a necessity of high precision since it always minimizes the number of false positives. On the other hand, YOLOv8+SEBlock offers the best balance of precision, recall, and generalization as well, hence more suitable for sensitive applications requiring broad detection capabilities. Although Sparse_RCI and Cascade_RCI possess a good performance in terms of precision and mAP50, its lower recall, limited generalization make it less competitive in general.



VI. FUTURE WORK:

Augmentation of Light Adaptability:

The model will be aimed at enhancing its performance with respect to different light conditions specifically low light, twilight lighting conditions, bright ambient lighting and in different weather conditions (rain, fog, etc.).

Real-Time Pothole Detection:

It is possible to perform a combination of computer vision modules and appropriate sensors (for example, LiDAR, ultrasonic sensors) in order to implement real time pothole detection systems. This system would also use audio beeps and/or vibrations to alert drivers of the presence of potholes in real time. The alerts will be determined by the level of damage caused by the pothole.

Integration with Autonomous Vehicles:

The development of a self-driving feature that allows the vehicle to maneuver around potholes ‘seen’ by the system avoiding any physical damage or putting passengers in danger.

The use of such detection data for road maintenance report mapping for municipalities and other local governing bodies.

VII. CONCLUSION:

We implemented a hybrid model to pothole in an image using YOLOv8+SEB. Severity-based analysis with YOLOv8 has a new way of detecting pothole. Our model gave good results with an impressive scores, precision 0.704, recall 0.724, mAP@50 equal to 0.73, and mAP@95 0.451. These metrics indicate the ability of our model in identifying the presence of potholes as well as the given severity levels. However, these positive results notwithstanding, there is still more that can be done. In the future work will be dedicated to performance enhancement of the models achieved by, amongst other things, optimization of the data preprocessing, more advanced augmentation techniques study, and other features that may help predict severity much better. This research input in road safety will therefore go a long way in providing a basis for the development of efficient real time pothole detection systems in various terrains.

References:

1. Pothole detection using YOLOv5 by atharva parikh
2. Wu, C., Wang, Z., Hu, S., Lepine, J., Na, X., Ainalis, D., & Stettler, M. (2020). An automated machine-learning approach for road pothole detection using smartphone sensor data. *Sensors*, *20*(19), 5564.
3. Patra, S., Middya, A. I., & Roy, S. (2021). PotSpot: Participatory sensing based monitoring system for pothole detection using deep learning. *Multimedia Tools and Applications*, *80*(16), 25171-25195.
4. Ahmed, K. R. (2021). Smart pothole detection using deep learning based on dilated convolution. *Sensors*, *21*(24), 8406.
5. Katsamenis, I., Sakelliou, A., Bakalos, N., Protopapadakis, E., Klaridopoulos, C., Frangakis, N., ... & Kalogeras, D. (2023, July). Deep transformer networks for precise pothole segmentation tasks. In *Proceedings of the 16th International Conference on PErvasive Technologies Related to Assistive Environments* (pp. 596-602).
6. Junaid, M., Ghafoor, M., Hassan, A., Khalid, S., Tariq, S. A., Ahmed, G., & Zia, T. (2020). Multi-feature view-based shallow convolutional neural network for road segmentation. *IEEE Access*, *8*, 36612-36623.
7. Nieto-Hidalgo, M., Gallego, A. J., Gil, P., & Pertusa, A. (2018). Two-stage convolutional neural network for ship and spill detection using SLAR images. *IEEE Transactions on geoscience and remote sensing*, *56*(9), 5217-5230.
8. Paramarthalingam, A., Sivaraman, J., Theerthagiri, P., Vijayakumar, B., & Baskaran, V. (2024). A deep learning model to assist visually impaired in pothole detection using computer vision. *Decision Analytics Journal*, *12*, 100507.
9. Ahmed, K. R. (2021). Smart pothole detection using deep learning based on dilated convolution. *Sensors*, *21*(24), 8406.
10. Satti, S. K., Rajareddy, G. N., Mishra, K., & Gandomi, A. H. (2024). Potholes and traffic signs detection by classifier with vision transformers. *Scientific reports*, *14*(1), 2215.
11. Sathya, R., Saleena, B., & Prakash, B. (2023, February). Pothole Detection Using YOLOv3 Model. In *2023 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)* (pp. 1-7). IEEE.
12. Yu, G., & Zhou, X. (2023). An improved YOLOv5 crack detection method combined with a bottleneck transformer. *Mathematics*, *11*(10), 2377.
13. Heo, D. H., Choi, J. Y., Kim, S. B., Tak, T. O., & Zhang, S. P. (2023). Image-based pothole detection using multi-scale feature network and risk assessment. *Electronics*, *12*(4), 826.
14. Frnda, J., Bandyopadhyay, S., Pavlicko, M., Durica, M., Savrasovs, M., & Banerjee, S. (2024). Analysis of Pothole Detection Accuracy of Selected Object Detection Models Under Adverse Conditions. *Transport and Telecommunication Journal*, *25*(2), 209-217.