EMITTER International Journal of Engineering Technology
Vol. 12, No. 1, June 2024, pp. 62~89
DOI : 10.24003/emitter.v12i1.837

A Novel Technology Stack for Automated Road Quality Assessment Framework using Deep Learning Techniques

Prabavathy Balasundaram, Pradeep Ganesh, Pravinkrishnan K, Rahul Kumar Mukesh

Department of Computer Science and Engineering, Sri Sivasubramaniya Nadar College of Engineering, Chennai, Tamil Nadu, India
Corresponding Author: prabavathyb@ssn.edu.in

Received February 28, 2024; Revised April 9, 2024; Accepted May 20, 2024

Abstract

Road infrastructure plays a pivotal role in supporting societal, economic, and cultural progress. The capacity of a road refers to its ability to handle vehicular volume. Inadequate road capacity and the presence of defects like potholes and cracks result in suboptimal travel conditions and pose significant safety risks for drivers, cyclists, and pedestrians. The regular evaluation of these road quality aspects is essential for effective maintenance. However, current methods for assessing road capacity are time-consuming, subjective, and heavily reliant on manual labor. Moreover, existing deep learning-based approaches for detecting road defects often lack accuracy. To overcome these challenges, a fully automated and accurate system for evaluating road quality is imperative. Thus, the objective of this research work is to propose a novel technology stack for a comprehensive Automated Road Quality Assessment (ARQA) framework designed to assess road quality. The experimental findings demonstrate that the suggested vehicle detection and pothole detection methods work effectively and exhibit enhancements of 18% and 6%, respectively, in comparison to existing approaches.

Keywords: Vehicle Tracking, LOS computation, Pothole Detection, Crack Segmentation, YOLO Algorithms, Segmentation

1. INTRODUCTION

With the rapid development and urbanization of cities, the need to maintain the quality of transportation systems has increased dramatically. Congestion on urban roads in developing countries often exceeds tolerable levels during peak hours, slowing down the economic growth of cities. Poor road conditions caused by defects on roads like potholes and cracks can also pose a threat to the safety of motorists, cyclists, and pedestrians. Millions of people around the world lose their lives or suffer injuries each year due to road accidents, making them a major cause of death and injury. Road accidents can be attributed to poor road conditions such as potholes, cracks, and inadequate capacity of roads. Assessing the quality of roads, considering the capacity of the road and defects like potholes and cracks, has become critical in this context.

The existing works [1, 2, 3, 4] have detected vehicles on roads as part of the capacity measurement of the roads. However, the Level of Service (LOS) computation from the detection is not automated. The existing works [5, 6, 7, 8] carry out the LOS computation manually. Further, the existing models on pothole detection [9, 10, 11] are used to detect the presence of potholes on roads using classification methods independent of the road conditions, such as highways and muddy roads. However, the road may have multiple potholes. Hence, it is necessary to localize and classify them. The work [12], which utilized object detection methods, did not consider the variability of the potholes. Subsequently, the work on road cracks [13, 14] utilizes a deep learning classification model to detect road cracks. The other works utilized [15] the object detection method for road cracks, which coarsely spots multiple objects in images by drawing boundaries around them.

Hence, the purpose of this study is to propose a novel technology stack to build an automated and comprehensive solution framework, namely, an Automated Road Quality Assessment (ARQA) framework comprising the best-suited deep learning object detection techniques for vehicle detection, pothole detection, and road crack detection.

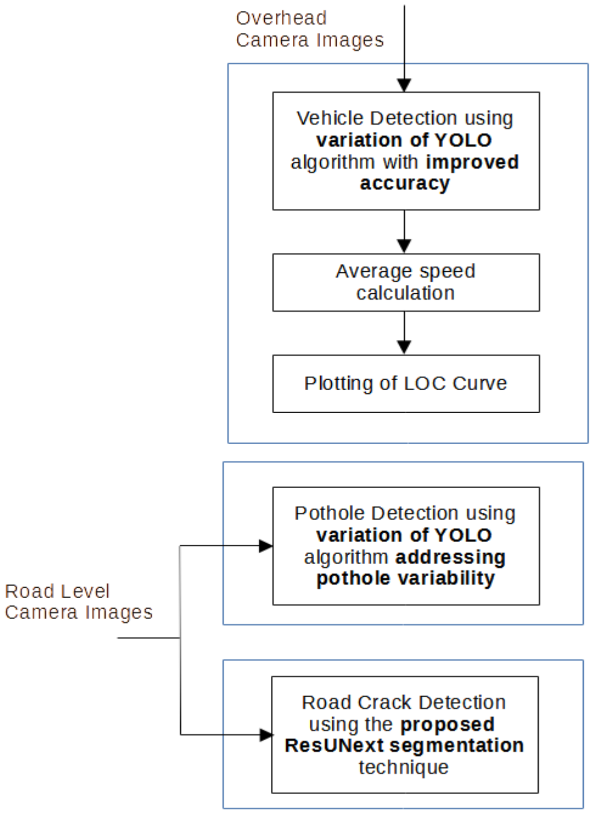

Figure 1. Working principle of ARQA Framework

Figure 1 explains the working principles of the proposed novel technology stack for the ARQA framework. Overhead camera images of the road will be fed into the framework to detect the vehicle using a variant of the YOLO algorithm with improved accuracy. Further, road-level camera images will be fed into the framework to detect potholes with a variant of the YOLO algorithm to address pothole variability. Subsequently, the same images will be used to detect road cracks with the proposed ResUNext segmentation technique.

The rest of the paper is organized as follows: Section 2 discusses research related to road quality assessment techniques. Sections 3 and 4 describe the originality of the work and the design of the proposed framework. Section 5 discusses the experimentation and analysis of the results. Section 6 concludes the work and discusses future work.

2. RELATED WORKS

Road quality assessment involves evaluating the condition and performance of roads to determine their safety, durability, and overall usability which shown in Table 1.

It is a crucial process for transportation authorities, road maintenance agencies, and infrastructure planners to ensure that roads are well-maintained and provide a smooth and safe travel experience for motorists. There are several methods and factors involved in road quality assessment.

The LOS introduced by the Highway Capacity Manual (HCM) denotes the quality of the road. HCM proposes designated letters that show the range of operating conditions of a facility.

The existing works [1, 2, 3, 4] have detected vehicles on roads as part of the capacity measurement of the roads. Otherwise, the LOS computation was done manually [5, 6, 7, 8]. Hence, there is a gap in the computation of the average speed and volume of the vehicles to automatically compute the LOS of the road. The detection of potholes using computer vision techniques involves several challenges. Firstly, the shape of potholes keeps varying constantly, making it a huge hurdle to develop a generalised pothole detection system. Secondly, the size of potholes can vary anywhere from the size of a small rock to the size of a boulder. The existing works [9, 10, 11, 12] did not consider the variability of potholes. Hence, this is considered in the proposed work. Subsequently, the existing works [13, 14, 15] on road cracks have utilized either classification or objection detection methods. However, every pixel of the crack must be identified to accurately predict road cracks. Hence, the proposed ResUNext architecture was utilised to segment the cracks.

This research work contributes to a comprehensive automated ARQA framework with an efficient technology stack, automated LOS computation to measure the capacity of the road, consideration of pothole variability, and accurate detection of road cracks.

Table 1. Comparison of Related Research

Research	LOS Computation	Pothole Detection	Road Crack Detection
[1-8, 17, 18, 23, 24]	Vehicle Detection using YOLO and Mobilenet V2 LOS Computation is manual or Semi-automated	-	-
[9, 10, 11, 12, 26]	-	CNN based Networks or Yolo Network for the detection of potholes	-
[13, 14, 15, 19, 20, 21, 22, 25, 27]	-	-	Object detection methods for identification of road cracks
Proposed ARQA Framework	YOLOV8 in conjunction with the SORT algorithm, is used to detect and track vehicles. With the help of tracking, IDs are assigned. By using the IDs, the velocity of the vehicles is computed, which helps to compute the LOS.	YOLOX with robust augmentation techniques, namely Mosaic and Mixup, is used to address the pothole variability.	In the Proposed ResUNext architecture, convolution layers in the UNet architecture are replaced with ResNeXt blocks accurately detect road cracks.

3. ORIGINALITY

A survey of currently available literature regarding road quality assessment reveals a lack of accurate and comprehensive solutions that can effectively assess the quality of roads. Most of the current methods for estimating road capacity rely on manual or semi-automated techniques. Further, from the current works, there is a lack of a single comprehensive system that can fully assess the quality of a given road. Subsequently, current systems for pothole detection do not consider the variability in the shape and size of potholes. However, it is essential to address the variability of potholes, as in real-life situations, potholes are of unique shapes and sizes, and it is not possible to generalize the features of a pothole based on images from a dataset. This research work addresses this issue with the help of an object detection technique, namely, YOLOX, which utilizes the best augmentation technique to introduce variability in the pothole images to build a more generalized detection module. In addition, the performance analysis of existing segmentation models reveals them to be primitive. and underwhelming. Hence, this work proposes a novel architecture, ResUNext, which is a combination of ResUNet and ResNeXt models that improves the performance of road crack analysis.

4. SYSTEM DESIGN

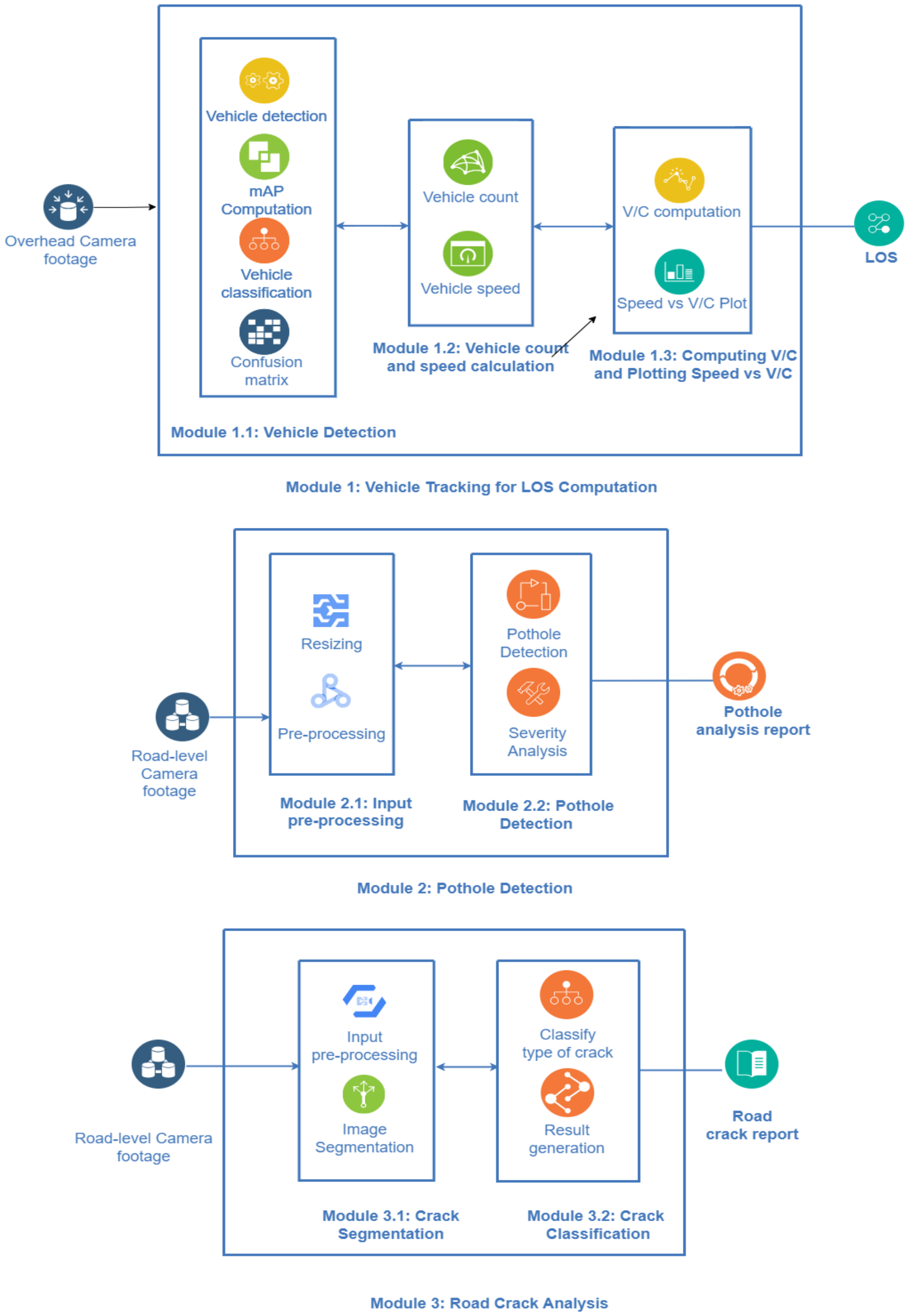

Figure 2. Design of proposed ARQA framework

The proposed ARQA framework for measuring the quality of roads involves the integration of various computer vision and deep learning techniques, including object detection, image processing, object classification, and image segmentation. The ARQA framework will have the capability to process large amounts of data in real time and the flexibility to adapt to changes in traffic patterns and road conditions.

The system consists of three modules, shown in Figure 2, each of which requires user input. The user input consists of a vehicle flow video from an overhead view and images of the road from a road-level view. The first module, namely, LOS Computation, uses the overhead view as input to perform vehicle tracking for computing LOS. The second module, namely, Pothole Severity Analysis, performs pothole detection by taking the road-level images as input and presenting the number of potholes and their location in the input image. The third module, namely, Road Crack Analysis, performs road crack analysis using the road-level images to segment and classify the cracks present in the image. The output from each of these modules is combined to produce the overall assessment of the quality of the road in the input data.

4.1 Vehicle Tracking for LOS Computation

LOS stands for "Level of Service", which is a measure of the capacity of a road. It is a qualitative measure of how drivers perceive the conditions on the road. In other words, it gives an idea about how easy or difficult it is for drivers to travel on a particular road. The task of computing LOS is divided into three steps:

- Vehicle Tracking
- Average Speed Calculation
- Plotting a Speed-Volume Graph

4.1.1 Vehicle Tracking using YOLOv8 Architecture

Module 1.1 of Figure 1 is responsible for tracking vehicles. A deep learning object detection model, YOLOv8, is used to identify and categorize vehicles in real-time. It is engineered to identify and categorize vehicles in the video input through a single forward pass of the network, eliminating the need for multiple regions of interest.

Ultralytics YOLOv8 is the latest version of the YOLO object detection and image segmentation model. As a cutting-edge, state-of-the-art model, YOLOv8 builds on the success of previous versions, introducing new features and improvements for enhanced performance, flexibility, and efficiency. YOLOv8 is designed with a strong focus on speed, size, and accuracy, making it a compelling choice for various vision AI tasks. It outperforms previous versions by incorporating innovations like a new backbone network, a new anchor-free split head, and new loss functions. These improvements enable YOLOv8 to deliver superior results while maintaining a compact size and exceptional speed. Additionally, it also supports a full range of vision AI tasks, including detection, segmentation, pose estimation, tracking, and classification. This versatility allows users to leverage YOLOv8's capabilities across diverse applications and domains. Figure 3 shows the architecture of YOLOv8.

4.1.2 Average Speed Calculation

Once vehicles have been detected, the next step is to calculate the average speed. This is calculated through the identification of unique objects in each frame and the length of the road segment captured in the video.

4.1.3 Plotting Speed-Volume Graph

In the Speed-Volume plot shown in Figure 4, the volume of vehicles is plotted on the x-axis, and the average speed of these vehicles is plotted on the y-axis. Each curve on the plot corresponds to a specific LOS rating, such as the six LOS classes (A to F) as defined in the HCM.

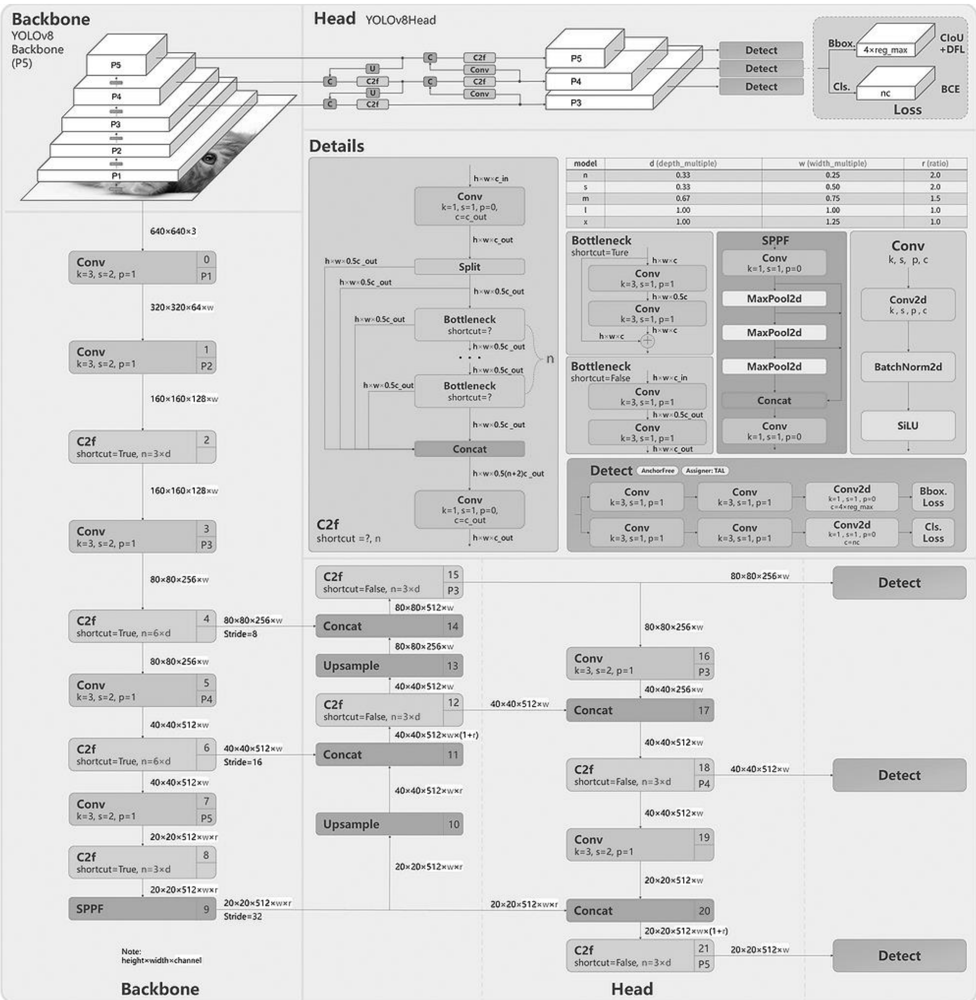

Figure 3. Architecture of YOLOv8

To determine the LOS of a road using the *speed-volume* plot, information regarding the volume and speed of vehicles travelling on the road is collected and plotted on the Speed-Volume plot to find the curve on the plot that best matches the data. The LOS is then determined by identifying the LOS rating that corresponds to this curve. For example, if the curve that matches the collected data is the curve for LOS D, then the LOS of the road is D.

In summary, the *speed-volume* plot estimates the LOS of a road and helps to qualitatively measure the driving experience of the users of that road. Consequently, roads that need improvement can be identified, and resources can be effectively allocated to improve transportation infrastructure.

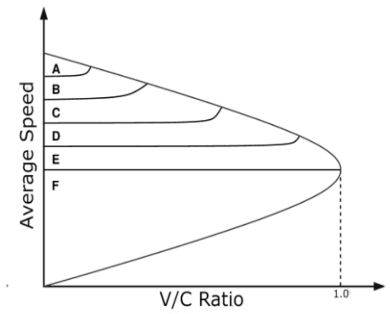

Figure 4. Average Speed vs V/C plot for LOS computation

4.2 Pothole Detection using YOLOX Architecture

The detection of potholes using computer vision techniques involves several challenges. Firstly, the shape of potholes keeps varying constantly. This is a huge hurdle to developing a generalised pothole detection system. Secondly, there is a large variability in the size of potholes.

Module 2.2 utilizes YOLOX object detection methods to classify potholes with high accuracy. The YOLOX algorithm shown in Figure 5 has introduced several significant improvements to the previous versions of YOLO. Firstly, it uses a decoupled head instead of a coupled one, which separates the classification and localization tasks, resulting in increased model performance. Secondly, robust data augmentation approaches, such as Mosaic and MixUp, have been integrated into the algorithm, which has further improved its performance. Additionally, the anchor mechanism, which increases the inference time, has been removed in YOLOX, thus reducing the number of predictions per image, and significantly improving the inference time. Finally, the algorithm uses SimOTA, a more robust label assignment strategy, instead of the intersection of unions (IoU) approach. This strategy not only reduces training time but also avoids extra hyperparameter issues, resulting in an improved detection mAP of 3%. Overall, these improvements make YOLOX a more efficient and effective object detection algorithm.

Automatic detection of potholes enhances the maintenance and repair of roads in a timely manner. This, in turn, can significantly reduce accidents caused by potholes and prevent costly repairs to vehicles. Furthermore, deep learning models can be continuously trained with new data, allowing them to adapt to changing road conditions and detect new types of potholes that may emerge over time. This can provide an effective means of monitoring and maintaining the condition of roads, ensuring the safety and efficiency of transportation systems for years to come.

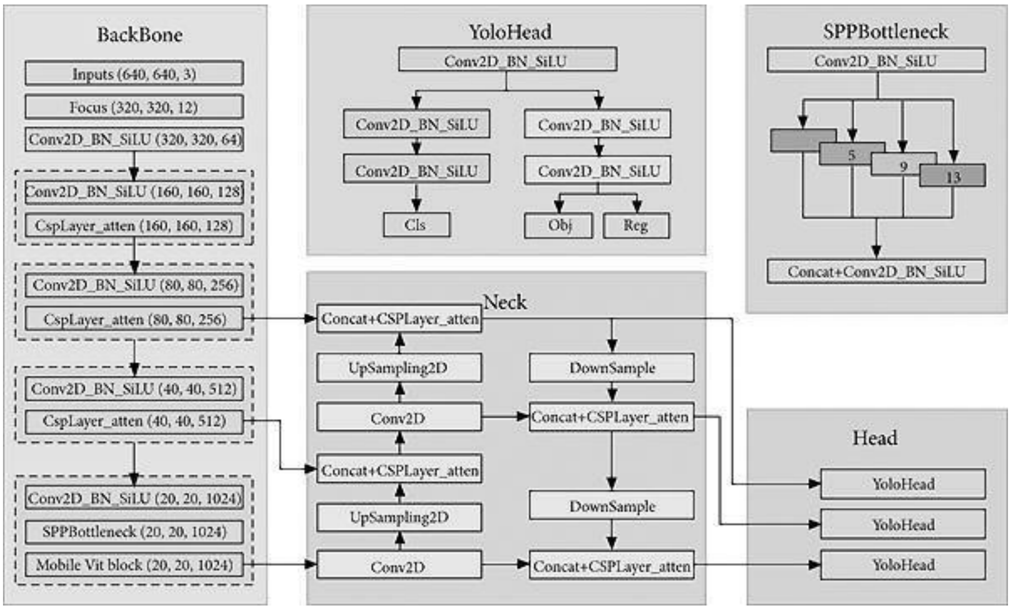

Figure 5. Architecture of YOLOX

4.3 Road Crack Analysis using Proposed ResUNext Architecture

Road crack analysis is an essential part of road maintenance and management. Computer vision and segmentation techniques have proven to be effective in automating this process. This process comprises several stages. First, the data must be prepared, which involves collecting images of the road surface and labelling them to identify the cracks. Module 3.1 utilizes the proposed hybrid architecture, namely, ResUNext for segmentation of cracks.

ResUNet is an improved version of the popular UNet architecture that utilizes residual connections to enhance its performance for segmentation tasks. The updated ResUNet architecture is represented in Figure 6. Residual connections were introduced in the ResNet architecture to alleviate the vanishing gradient problem that can occur in very deep neural networks. In ResUNet, residual connections are used to learn residual functions that are easier to optimize. These residual functions enable information to flow directly from the input to the output layers of the network, improving its training speed and accuracy. By incorporating residual connections, ResUNet is able to produce more accurate segmentation results than UNet, especially for complex images.

Moreover, ResUNet employs a deep supervision strategy that involves adding auxiliary classifiers at different stages of the network. This strategy enables the network to learn multi-scale representations of the input image, which can capture both global and local features more effectively than UNet. The deep supervision strategy also helps to address the problem of information loss during the encoding process, where the input image is progressively downsampled. The auxiliary classifiers provide additional supervision signals that help preserve information and maintain spatial resolution during the decoding process. By incorporating deep supervision, ResUNet is able to produce more precise segmentation results, especially for small objects.

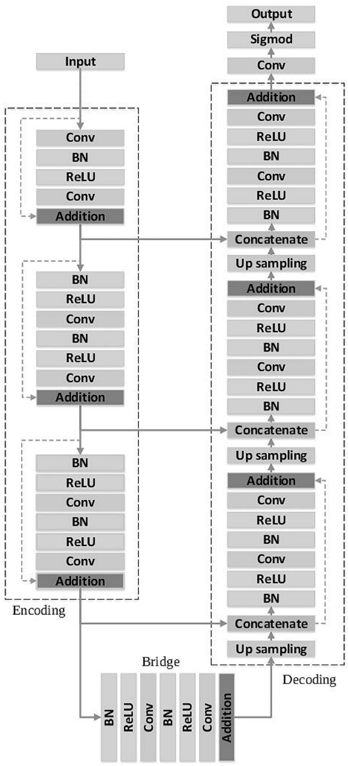

Figure 6. Architecture of ResUNet

The proposed ResUNeXt is an improvement over the ResUNet architecture. The ResUNeXt makes use of state-of-the-art ResNeXt blocks, as seen in Figure 7. It introduces a new module called *cardinality* that allows for increased model capacity and improved feature representation. Cardinality is a hyperparameter that controls the number of parallel paths within a block of convolutional layers. By increasing the cardinality, ResNeXt can learn more diverse and discriminative features. The proposed hybrid ReUNeXt architecture uses the existing ResUNet architecture and replaces the ResNet blocks with the ResNeXt blocks.

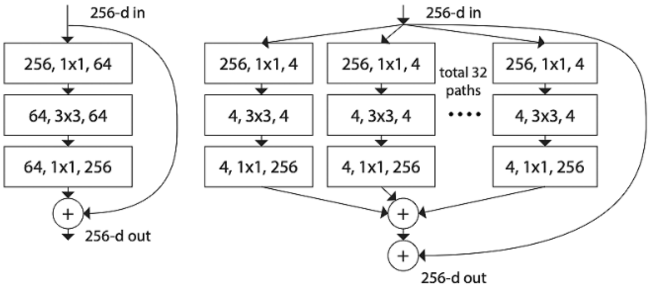

Figure 7. Architecture of ResUNext

Analysis of road cracks can help improve the efficiency and accuracy of road maintenance and management processes, ultimately leading to safer and more reliable transportation infrastructure.

5. EXPERIMENT AND ANALYSIS

The ARQA framework's three modules were implemented by investigating and testing with various deep learning architectures to identify the best performing networks. The implementation was carried out using Google Colab and the PyTorch and Tensorflow libraries. During the model training process, we recorded the validation and training losses, as well as the mean Average Precision (mAP) for detection problems and the dice coefficient for segmentation.

While the training loss indicates the goodness of model fitting on training data, the validation loss provides the same input on new data. These losses were used to measure the capability of the built model.

Mean Average Precision is a metric used to predict the performance of the object detection model by leveraging Intersection over Union (IoU). The IoU calculates the overlap between the predicted and ground truth bounding boxes. The IoU computes the overlap between the predicted and ground truth bounding boxes. An IoU of 1 indicates complete overlap, whereas an IoU of 0 indicates no overlap. The Mean Average Precision at 50 IoU (mAP 50) indicates that the predicted and ground truth bounding boxes overlap by at least 50%.

The *Dice coefficient* is used to compare a predicted segmentation to its related ground truth image on a pixel-by-pixel basis. The Dice coefficient is 0 to 1, with 1 reflecting a perfect match and 0 indicating no overlap.

5.1. Vehicle Tracking for LOS Computation

The deep learning networks chosen for this module are YOLOv5 and YOLOv8.

5.1.1. Dataset

The custom vehicle detection dataset contains images of vehicles on a 3-lane road, collected from a foot-over bridge in Anna Nagar West, Chennai, Tamil Nadu. The dataset includes a pedestrian class and six vehicle classes, namely two-wheelers, four-wheelers, motorised three-wheelers, Light Commercial Vehicles (LCV), Heavy Commercial Vehicles (HCV) and buses. The dataset was created by extracting frames from a 20-minute video, resulting in a collection of over 2,500 images, with every 10th frame chosen for inclusion. The distribution of images across the vehicle classes is shown in Figure 8.

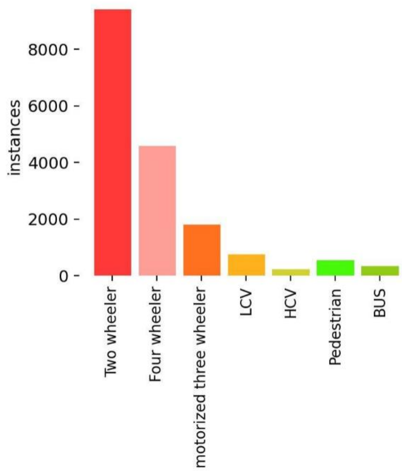

Figure 8. Class distribution in the dataset

5.1.2 Implementation using YOLOv5

The custom dataset was used to train the YOLOv5 deep learning network. The model was trained for 14 epochs, and training and validation losses were recorded after each epoch. Figure 8 shows the plot of the change in the loss for every epoch. It is evident from Figure 9 that the loss value declines with each epoch and starts to become constant after around 12 epochs. From Figure 10, it can be inferred that YOLOv5 performs best at identifying 2-wheelers, 4-wheelers, and motorised 3-wheelers with more than 94% accuracy. However, the detection of pedestrians and HCV is the least accurate, with 70% and 59% accuracy, respectively. This is because pedestrians and HCV account for only 10% of the total object instances in the dataset. The least accurate class is HCV. About 23% of HCV gets misclassified as bus, while 9% of HCV instances are misclassified as LCV. This misclassification is because the front view of these classes of vehicles looks like each other.

The dataset contains images of vehicles on a 3-lane road, collected from a foot-over bridge in Anna Nagar West, Chennai, Tamil Nadu. The dataset includes a pedestrian class and six vehicle classes, namely two-wheelers, four-wheelers, motorised three-wheelers, Light Commercial vehicles (LCV), Heavy Commercial Vehicles (HCV) and buses.

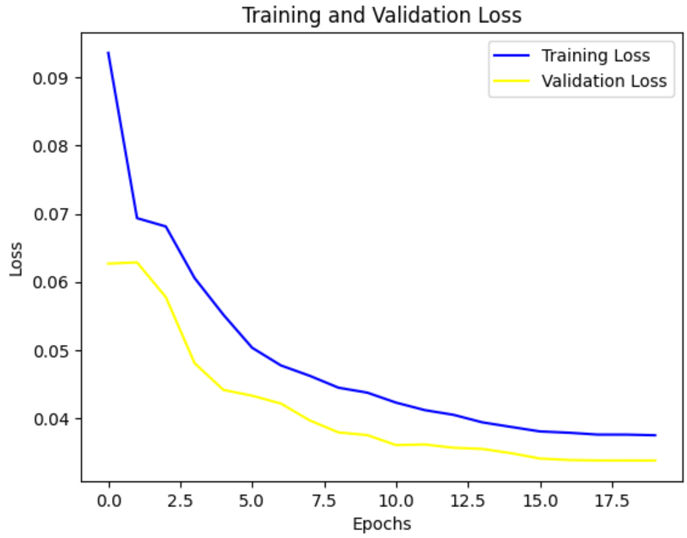

Figure 9. Loss vs Epoch Curve – YOLOv5

5.1.3 Implementation using YOLOv8

The custom dataset was used to train the YOLOv8 deep learning network. The model was trained for 20 epochs, and the training and validation losses were recorded for each epoch. Figure 10 shows the change in training and validation losses after each epoch. It is evident from Figure 11 that the loss value declines with each epoch and starts to converge after 15 epochs. It can be inferred from Figure 12 that YOLOv8 produces high accuracy for each of the different vehicle classes. The HCV class is identified with the highest accuracy of 100%, while the least accurately identified class is that of the pedestrian, with 91% accuracy.

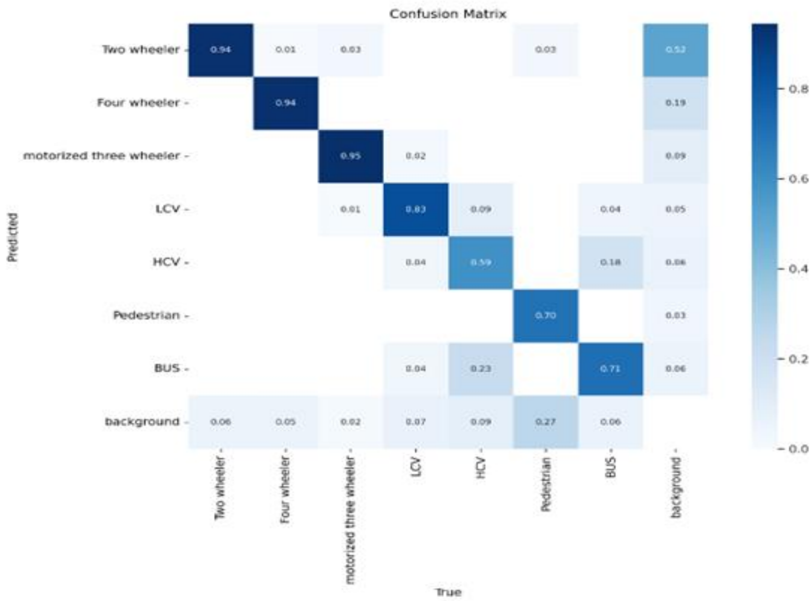

Figure 10. Confusion matrix – Vehicle Detection using YOLOv5

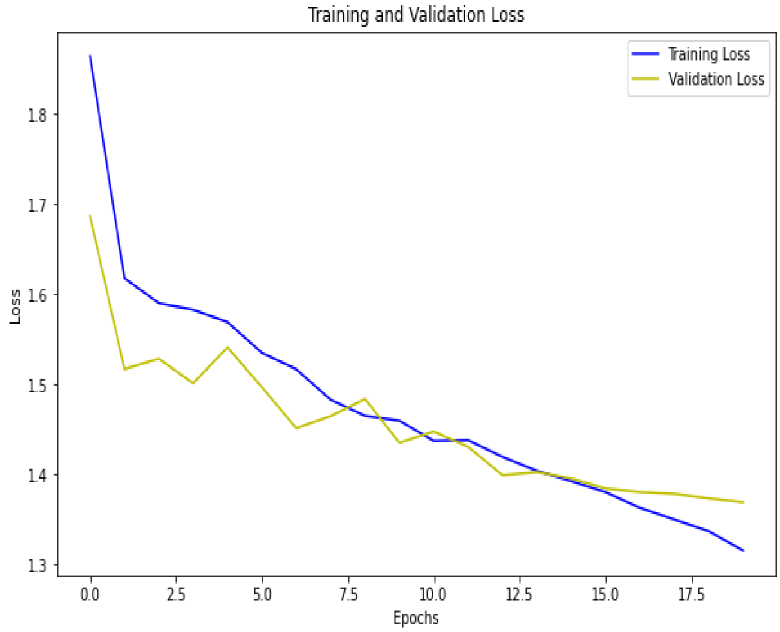

Figure 11. Loss vs Epoch Curve – YOLOv8

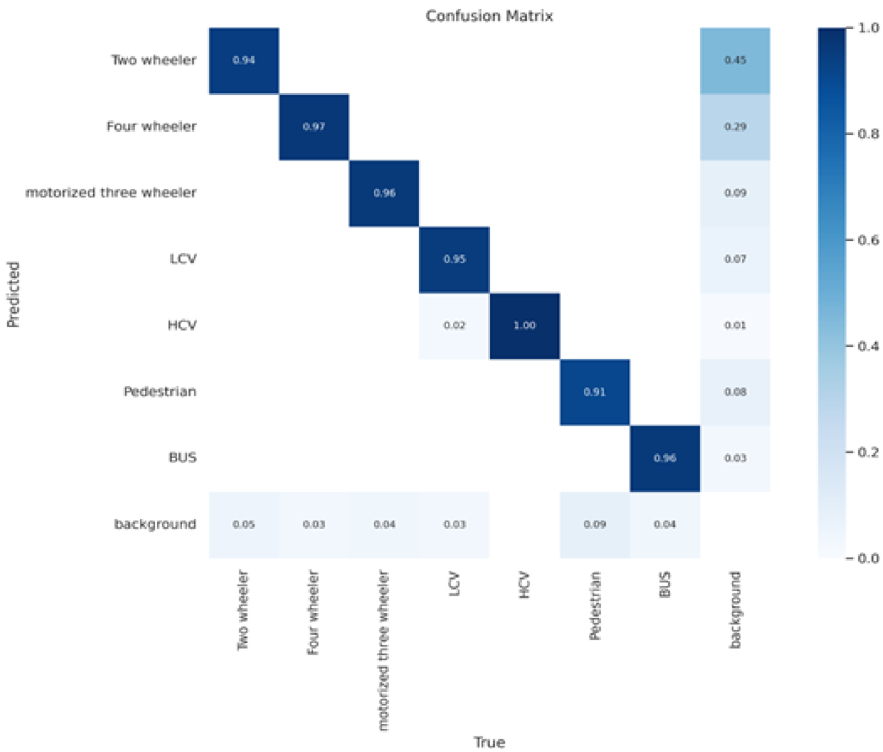

Figure 12. Confusion matrix – Vehicle Detection using YOLOv8

5.1.4 Result Comparison

The YOLOv5 model achieved a mAP of approximately 0.881, while the YOLOv8 model achieved a significantly higher mAP of 0.9575. The results, shown in Figure 13, indicate that YOLOv8 outperforms YOLOv5 in the task of vehicle detection.

The performance improvement of YOLOv8 can be attributed to several factors. First, YOLOv8 uses a new backbone network that enables more efficient feature extraction from the input images. This new backbone network provides YOLOv8 with a higher level of abstraction and better representation of image features, which leads to improved accuracy in vehicle detection.

Second, YOLOv8 utilizes an anchor-free detection head, which eliminates the need for predefined anchor boxes. This improves the flexibility of the model in detecting vehicles of different shapes and sizes. The anchor-free detection head also simplifies the training process and reduces the computational cost of the model.

Finally, YOLOv8 incorporates a new loss function, which enhances the model's ability to distinguish between foreground and background objects. This loss function improves the model's ability to learn and recognize vehicle features in complex scenes.

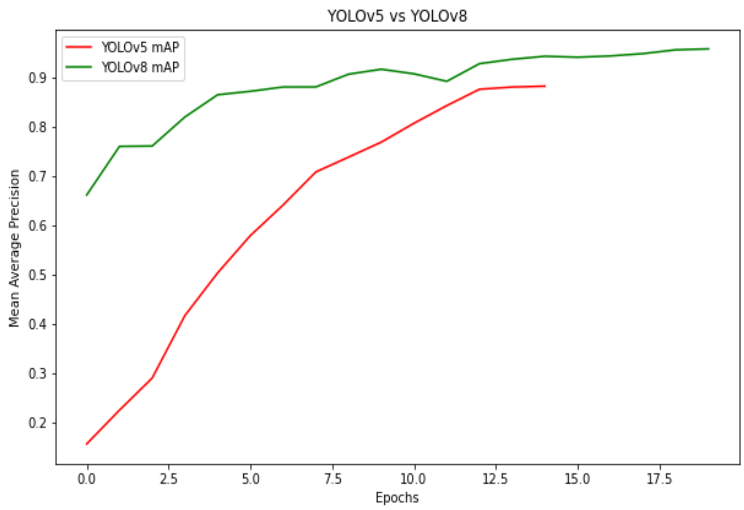

Figure 13. Comparison of YOLOv5 vs YOLOv8 – Training Results

The YOLOv5 and YOLOv8 models were tested for mAP by gradually increasing the number of images. The number of images ranges from 1 to 500. From Figure 14, it is evident that the YOLOv8 outperforms the YOLOv5 by a huge margin for vehicle detection. The better-performing YOLOv8 was used for implementing the system.

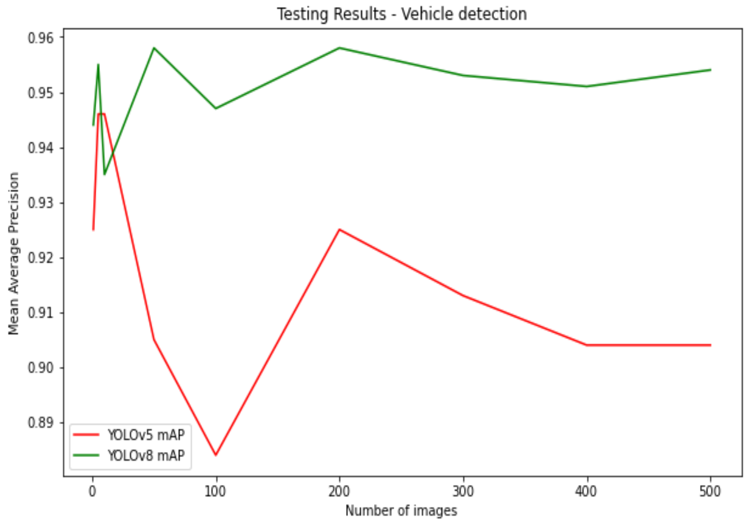

Figure 14. Comparison of YOLOv5 vs YOLOv8 – Testing Results

5.1.5 Vehicle Tracking and Plotting Speed-Volume Graph

The model trained for vehicle detection is used in conjunction with SORT algorithms for tracking the objects. This tracker assigns unique IDs to the vehicles throughout the time they are on the screen, allowing for efficient and accurate tracking. By using these IDs, the velocity of the vehicle can be computed.

Table 2. HCM defined conditions for LOS computation

LOS	Condition
A	Average Speed > 70 and Volume < 0.22
B	Average Speed > 50 and Volume < 0.45
C	Average Speed > 40 and Volume < 0.63
D	Average Speed > 28 and Volume < 0.81
E	Average Speed > 15
F	Otherwise

The entry and exit frames of the vehicles are maintained using the IDs assigned to each object through the vehicle tracking module. From this information, the total time taken to cover the distance is obtained, and the average speed of the vehicle is computed. This process is repeated for all the vehicles, and the overall average speed is obtained. The volume of the road is computed in each frame, which depends on the count and type of vehicle. The average speed and average volume are computed. The *speed-volume* graph is plotted using the computed average speed and volume. As per the HCM, LOS can be computed based on the conditions outlined in Table 2.

5.2. Pothole Detection

The dataset implementing this module was obtained from Kaggle. The pre-annotated dataset consists of 665 images, which were split and used for training (532), validation (66), and testing (67). The annotation files were obtained in Pascal VOC format and converted to YOLO format for training the YOLOv5 model. The wide variety of types of pothole images and their corresponding annotations provided in this dataset make it suitable for training the chosen algorithms that are to be implemented in the system. The deep learning networks chosen for this module are YOLOv5 and YOLOX.

5.2.1 Implementation using YOLOv5

The pothole dataset was used to train the YOLOv5 deep learning network. The model was trained for 50 epochs, and the training and validation losses were recorded for every epoch. Figure 15 shows the change in training and validation losses after each epoch. It is evident from the figure that the loss value declines with each epoch and starts to become constant after around 45 epochs. From Figure 16, it is inferred that the YOLOv5 model achieves a true positive rate of 81%.

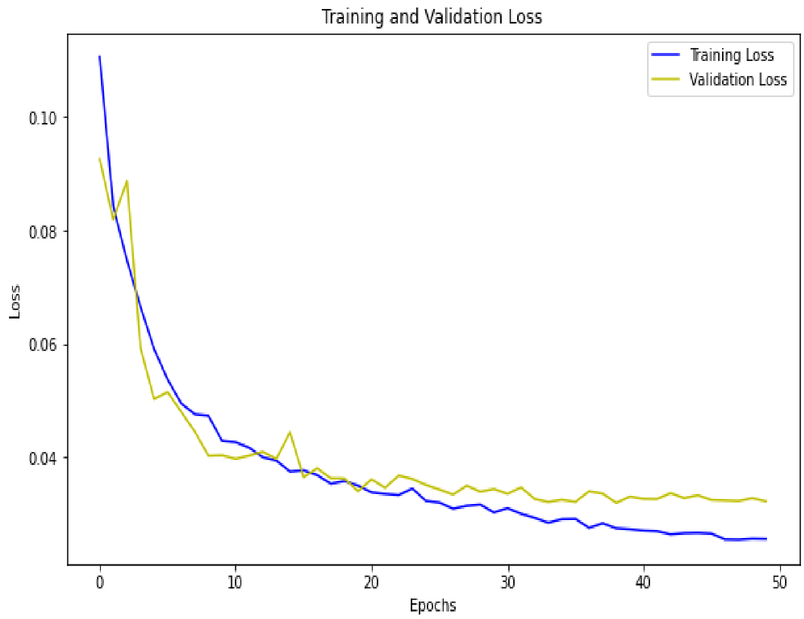

Figure 15. Loss vs Epoch Curve – YOLOv5

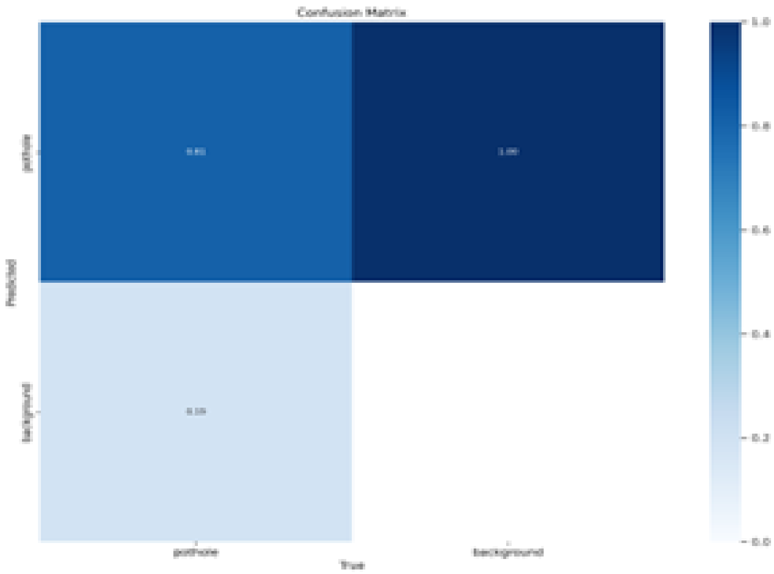

Figure 16. Confusion matrix – Vehicle Detection using YOLOv5

5.2.2 Implementation using YOLOX

The YOLOX deep learning network was trained on the pothole dataset for 150 epochs. The training and validation losses were recorded after each epoch. From Figure 17, it is inferred that the training and validation losses decrease with progress over epochs. From Figure 18, it is inferred that the YOLOX model achieves a true positive rate of 80%.

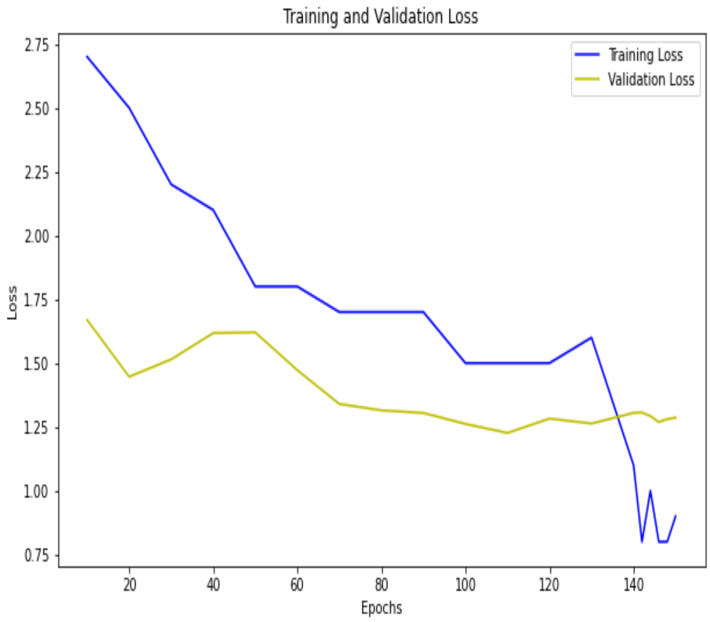

Figure 17. Loss vs Epoch Curve – YOLOX

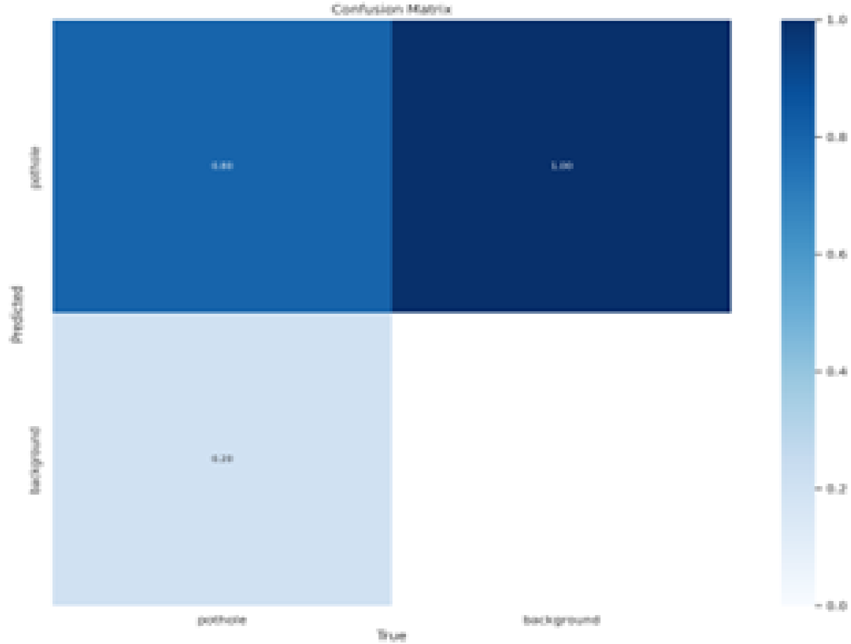

Figure 18. Confusion matrix – Vehicle Detection using YOLOX

5.2.3 Result Comparison

In real-life implementation of pothole detection, the shapes of potholes are very diverse, and this variability cannot be captured from the dataset alone. YOLOX's Mixup and Mosaic augmentation techniques are highly effective in creating variability in pothole images during training. The model's ability to handle diverse scenarios is enhanced by these techniques, leading to better performance in real-world pothole detection applications. Figure 19 shows the plot of mAP vs. epoch between the YOLOv5 and YOLOX deep learning models. It shows the models' learning and generalization to the training data is reflected by the increasing mAP during training.

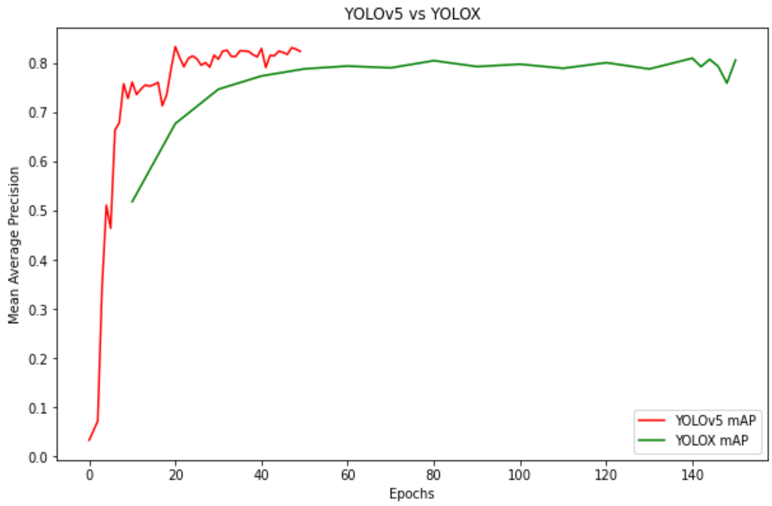

Figure 19. YOLOv5 vs YOLOX for pothole – Training Results

The YOLOv5 and YOLOX models were tested for mAP by gradually increasing the number of images. The number of images ranges from 1 to 300. Testing mAP across epochs is an experiment shown Figure 20 is for assessing the generalization and performance stability of object detection models. T ensures the effectiveness of the model on new unseen data and robust in real-world scenarios. From Figure, both YOLOv5 and YOLOX perform well in pothole detection. However, due to its robust data augmentation module and decoupled head, which allow better feature extraction, YOLOX was preferred for implementing this module.

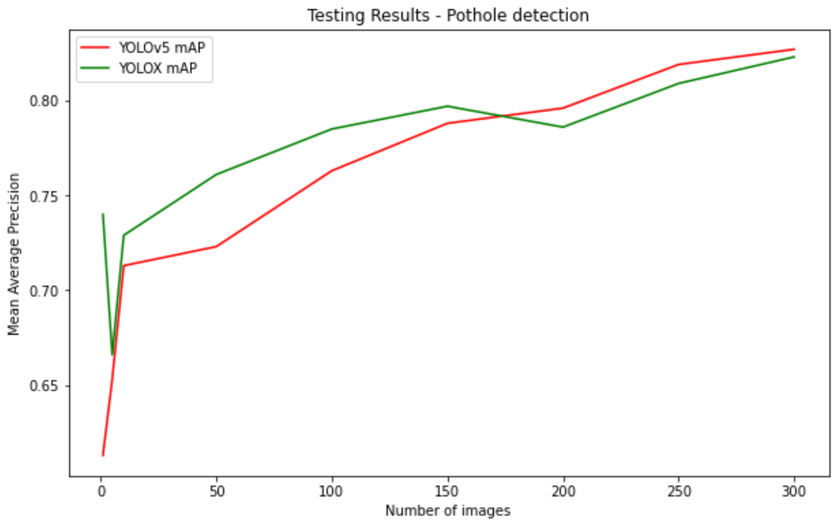

Figure 20: Number of Images vs mAP – Testing Results

5.3. Road Crack Detection

The crack detection dataset comprises 9,603 images with accompanying masks. The dataset images are RGB and resized to 448 x 448 pixels, while the corresponding masks are grayscale images. This image segmentation dataset is specifically designed for crack detection, a critical task in the maintenance and safety of infrastructure. The high volume of images and corresponding masks provided in this dataset makes it suitable for training deep learning models that can accurately identify cracks in infrastructure.

5.3.2. Implementation using UNet

The UNet architecture was implemented using TensorFlow and trained on the road crack dataset for 100 epochs. The training time of UNet in comparison with the other networks of study for crack segmentation was observed to be less. From Figure 21, it can be inferred that both the training and validation losses are stagnant until the 35th epoch, after which they spike up and gradually decrease. This suggests overfitting in the model.

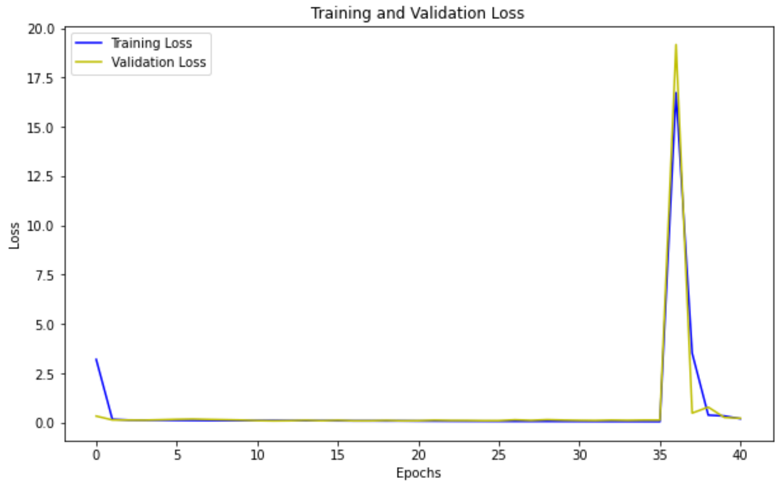

Figure 21. Loss vs Epoch Curve – UNet

5.3.3. Implementation using ResUNet

The ResUNet architecture has been implemented using the TensorFlow framework and trained on the crack dataset for 100 epochs. The ResUNet architecture was implemented by replacing the convolution blocks of the UNet architecture with ResNet blocks. The training time was observed to be slightly higher than the UNet and less than the ResUNeXt architecture. From Figure 22, it can be inferred that training and validation loss both decrease with an increase in epochs. This suggests that the model has trained well without overfitting and will generalize well to the testing data.

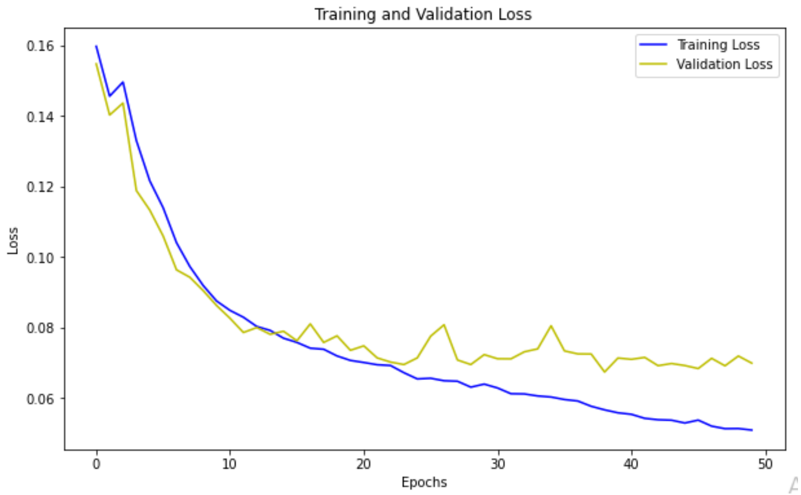

Figure 22. Loss vs Epoch Curve – ResUNet

5.3.4. Implementation using Proposed ResUNext

The ResUNeXt architecture has been implemented using TensorFlow. The convolution layers in the UNet architecture are replaced with ResNeXt blocks. Cardinality is taken as input from the user. The current model is trained with a cardinality of 3. From Figure 23, it can be inferred that training and validation loss both decrease with epoch. This suggests that the model has trained well without overfitting, has minimal loss, and will generalize well for the testing data.

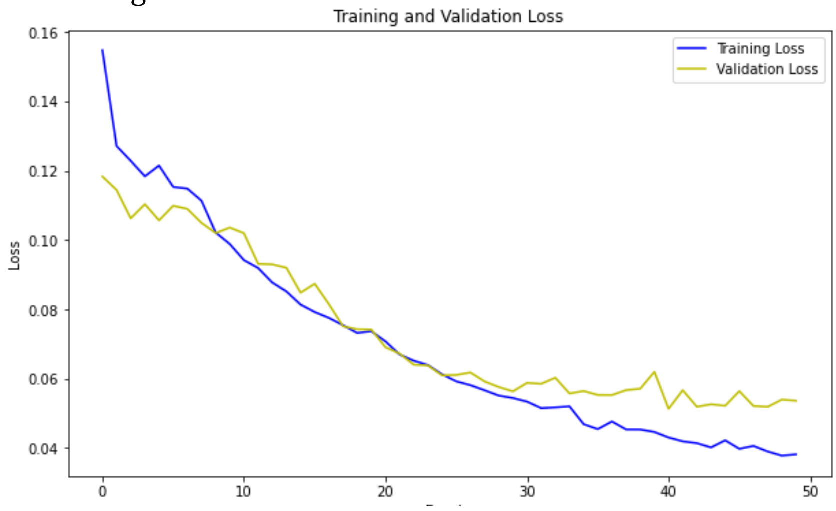

Figure 23. Loss vs Epoch Curve – ResUNext

5.3.5. Result Comparison

UNet and ResUNet are popular deep learning models for image segmentation. ResUNeXt, which incorporates residual connections and a deep supervision strategy, outperforms UNet and ResUNet in various segmentation tasks by capturing more spatial information from the input image. Residual connections help address the vanishing gradient problem and enable the network to learn residual functions, which are easier to optimize. While both models are effective, ResUNeXt's architecture makes it a more powerful and versatile model for segmentation tasks. From Figure 24, it can be clearly seen that the proposed ResUNeXt outperforms UNet and ResUNet. The Dice coefficient of ResUNeXt is seen to increase steadily and peaks at a higher value when compared to ResUNet.The Dice coefficient of Unet is not stable and sees a steep drop off at the 35th epoch, suggesting overfitting.

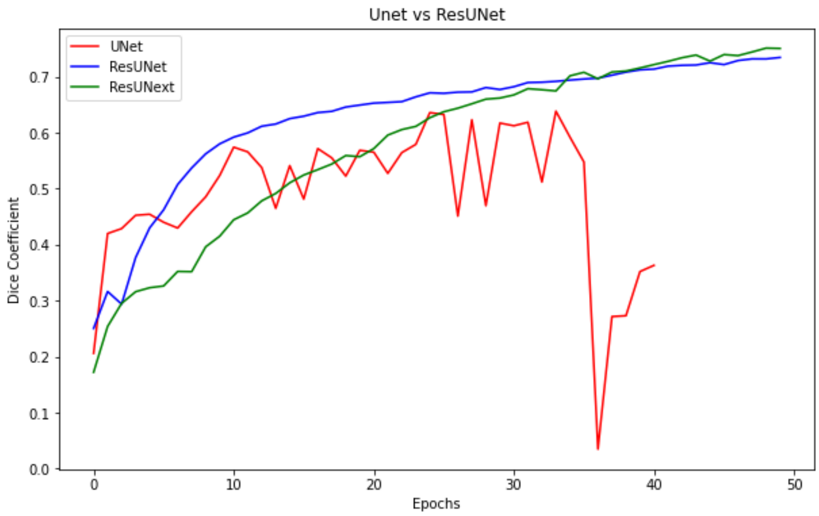

Figure 24. Comparison of UNet, ResUNet and ResUNext – Training Results

The UNet, ResUNet, and ResUNeXt models were tested for Dice Coefficient by gradually increasing the number of images. The number of images ranges from 1 to 500. From Figure 25, it can be seen that the proposed ResUNeXt model outperforms the existing UNet and ResUNet models. Thus, the proposed ResUNeXt model was used to build the module.

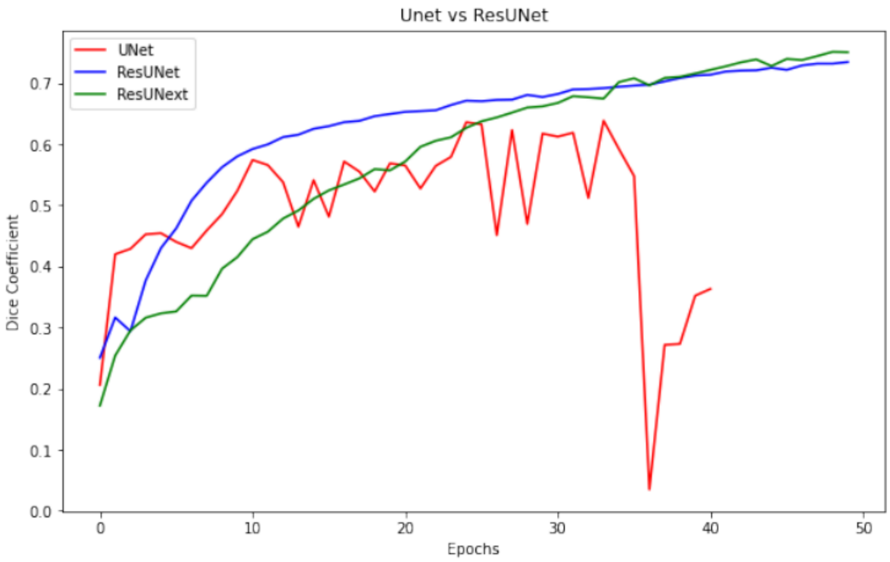

Figure 25. Comparison of UNet, ResUNet and ResUNext – Testing Results

5.4. Technologies used for RQA Framework

Table 3 shows the testing results that were measured for the test images using mAP for object detection and the Dice Coefficient for image segmentation. Through the inferences obtained from the testing results, the proposed ARQA framework uses YOLOv8 for vehicle detection, YOLOX for pothole detection, and ResUNext for crack segmentation, thus providing an optimal technology stack for the purpose of road quality assessment.

We have utilized variants of object detection algorithms such as YOLOv5 and YOLOX to detect potholes. On the other hand, existing variants of segmentation algorithms such as ResUNet, ResNeXt, and the proposed ResUNeXt are used to segment the road cracks.

YOLO and UNet can be used complementarily in a system. While YOLO can quickly identify the presence and locations of potholes, ResUNeXt can provide detailed segmentation of road cracks, offering a comprehensive analysis of road surface conditions that increases the effectiveness of the system.

Table 3. Consolidated Testing Results

Module	Model	Performance Metric Used	Result
Vehicle Detection	YOLOv5	mAP50 (0.0-0.1)	0.904
	YOLOv8		0.954
Pothole Detection	YOLOv5	mAP50 (0.0-0.1)	0.812
	YOLOX		0.816
Crack Segmentation	UNet	Dice Coefficient (0.0-1.0)	0.480
	ResUNeT		0.900
	ResUNexT		0.940

5.5. Web Application for ARQA Framework

Apache Flask is a widely adopted Python web framework for developing web applications, providing a range of tools for routing, templating, and database integration to create a robust platform. In this study, Flask was leveraged to build an application that uses HTML and CSS for the front end and Python for the back end. The user is prompted to upload the input files, which consist of a video for LOS computation, an estimate of the road length captured, a picture of potholes on the road for detection, and a picture of a road crack for classification. These files are passed as input to the trained machine learning models on the back end, and the resulting outputs are displayed in a report on the front end of the application.

To facilitate active learning, a separate folder was created to store the pothole images that the user gives as input. These images will be periodically sent to an expert for annotation and appended to the training data. Subsequently, the ML models can be retrained using the augmented dataset and redeployed for improved accuracy.

In summary, the Flask-based application provides an efficient and versatile approach to road condition monitoring with the potential for continual improvement through active learning. Figure 26 shows the visual representation of the fully developed ARQA framework.

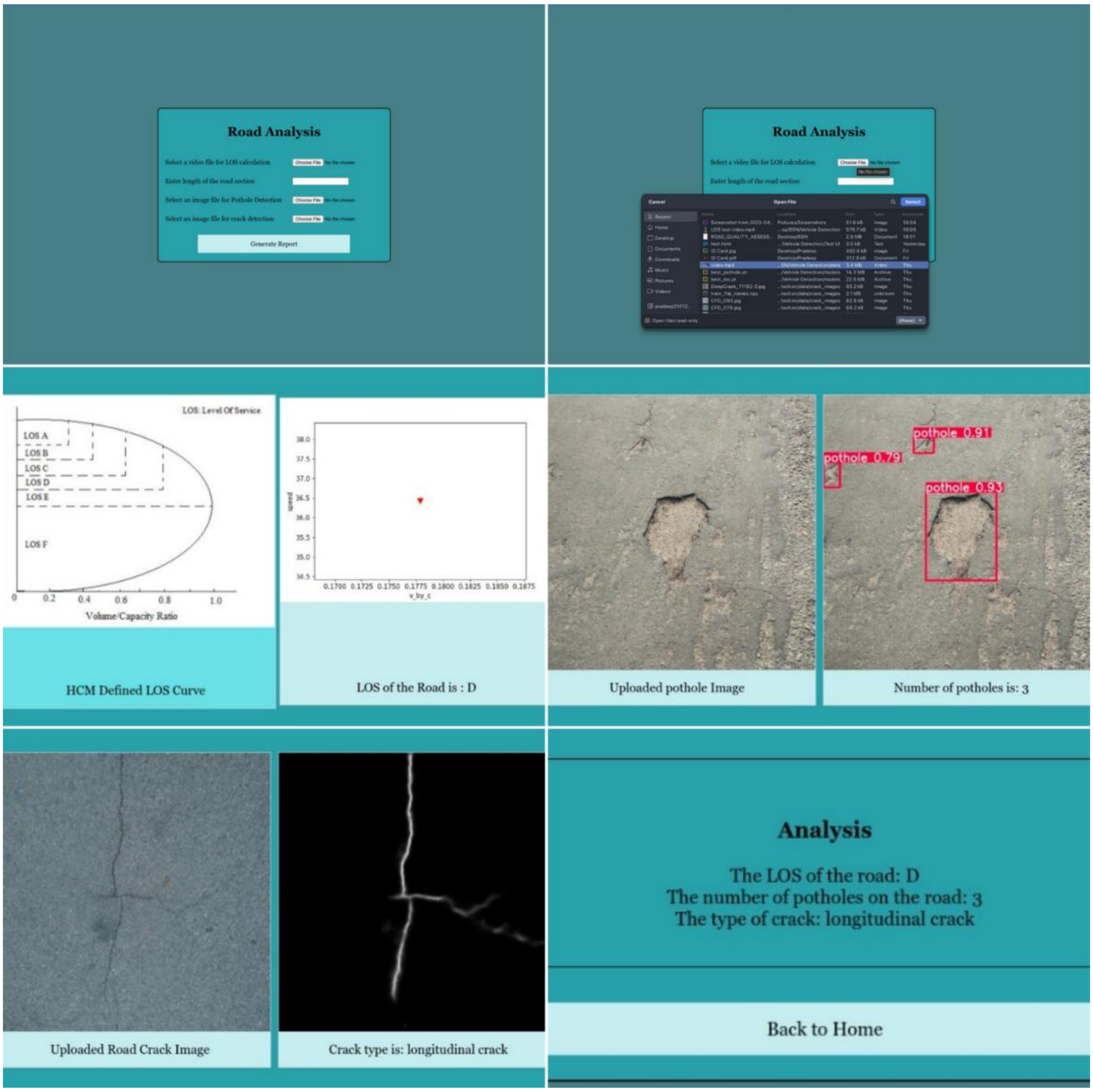

Figure 26. ARQA Framework Application UI Design

6. CONCLUSION

In today's age of rapid urbanization and increasing road traffic, it is essential to analyze road quality quickly and accurately. Both urban and rural road development authorities can easily identify roads that need to be upgraded comprehensively and accurately with assistance from the proposed ARQA framework. This framework involves vehicle detection, pothole detection and road crack segmentation.

The suitability of different object detection, segmentation, and proposed ResUNeXt algorithms in each module has been evaluated through experiments and tests. The first step in detecting vehicles was the implementation of YOLOv8 and YOLOv5 algorithms. From the experiments, the YOLOv8 model achieved a mAP of 0.954, which is 18% improvement over the existing work [2] that used YOLOv3.

Secondly, through the pothole detection module, this work has analysed and solved the problem of pothole variability, which was a hindrance to the real-world application of such systems. Detecting potholes was carried out using YOLOv5 and YOLOX algorithms. Our experiments revealed that the performance of the existing work [16] that uses the YOLOv5-based model is 0.77 mAP, which is inferior to the 0.816 mAP of the proposed system.

Lastly, this work has proposed a new segmentation model, ResUNeXt, that surpasses existing and widely used network architectures like UNet and ResUNet. The proposed ResUNeXt has a better Dice Coefficient value of 0.94 outperforming both UNet and ResUNet which have between 0.48 and 0.9 respectively.

As a part of future work, the ARQA framework can be extended to consider other factors that affect road quality, like weather conditions and the quality of materials used for constructing the road. Furthermore, the ARQA framework can be built as a mobile app with a minimalist interface, providing an uncluttered user experience so that novice users can use it with ease.

REFERENCES

[1] J. Azimjonov, and A. Ozmen, **A real-time vehicle detection and a novel vehicle tracking systems for estimating and monitoring traffic flow on highways**, Journal of Advanced Engineering Informatics, Vol. 50, pp. 1-12, 2021.

[2] R. Dhatbale and BR. Chilukuri, **Deep Learning Techniques for Vehicle Trajectory Extraction in Mixed Traffic**. Journal of Big Data Analytics in Transportation, Vol. 3, pp. 141-57, 2021.

[3] Y. He, Z. Pan, L. Li, Y. Shan, D. Cao, and L. Chen, **Real-time vehicle detection from short-range aerial image with compressed mobilenet**, Proceeding of the International Conference on Robotics and Automation, Montreal, Canada, pp. 8339-8345, 2019.

[4] S. Huang, Y. He, and XA. Chen, **M-YOLO: A Nighttime Vehicle Detection Method Combining Mobilenet v2 and YOLO v3**, Journal of Physics: Conference Series, Vol. 1883, pp. 1-6, 2021.

[5] K. Dudekula, R. Kiran, M. Reddy, R. Shet, R. Rohan, and JS. Vishwas, **Determination of Level of Service and Traffic Flow Characteristics for a Selected Arterial Roads in Bangalore City**, pp. 1-5, 2021.

[6] M. Khanam, and M. Marufuzzaman, **Determination of the Level of Service at Major Intersection: A Case Study on Traffic More Intersection Area, Pabna, Bangladesh**, Proceeding of the International Conference on Planning, Architecture and Civil Engineering, Bangladesh, pp. 1-6, 2019.

[7] J. Raj, and P. Vedagiri, **Evaluation of Perception and Non Perception based Approaches for Modeling Urban Road Level of Service**, Journal of The Institution of Engineers: Series A, Vol. 103, pp. 467-80, 2022.

[8] S. Volosenko and A. Laurinavicius, **Level of service evaluation methods: possible adaptation for Lithuania**, The Baltic journal of road and bridge engineering, Vol. 15, pp.145-57, 2020.

[9] AK. Pandey, R. Iqbal, T. Maniak, C. Karyotis, S. Akuma, and V. Palade, **Convolution neural networks for pothole detection of critical road infrastructure**, Journal of Computers and Electrical Engineering, Vol. 99, pp. 1-12, 2022.

[10] B. Varona, A. Monteserin, and A. Teyseyre, **A deep learning approach to automatic road surface monitoring and pothole detection**, Journal of Personal and Ubiquitous Computing, Vol. 24, pp. 519-34, 2022.

[11] C. Saisree, and U. Kumaran, **Pothole Detection Using Deep Learning Classification Method,** Procedia Computer Science. Vol. 218, pp. 2143-52, 2023.

[12] A. Dhiman, and R. Klette, Pothole **Detection Using Computer Vision and Learning**, Journal of IEEE Transactions on Intelligent Transportation Systems, Vol. 21, pp. 3536-3550, 2020.

[13] SJ. Wang, JK, Zhang, and XQ, Lu, **Research on Real-Time Detection Algorithm for Pavement Cracks Based on SparseInst-CDSM**, Mathematics. Vol. 11, pp. 1-20, 2023.

[14] Z. Lv, C. Cheng, and H. Lv, **Automatic identification of pavement cracks in public roads using an optimized deep convolutional neural network model**, Philosophical Transactions of the Royal Society A., Vol. 381, pp. 1-15, 2023.

[15] L. Deng, A. Zhang, J. Guo, Y. Liu, **An Integrated Method for Road Crack Segmentation and Surface Feature Quantification under Complex Backgrounds,** Remote Sensing, Vol. 10, pp. 1-18, 2023.

[16] YM Kim, YG Kim, SY Son, SY Lim, BY Choi, DH Choi, **Review of Recent Automated PotholeDetection Methods**, Journal of Applied Sciences, Vol. 12, pp. 1-15, 2022.

[17] J Raj, P VedagiriP**, Evaluation of Perception and Nonperception based Approaches for Modeling Urban Road Level of Service**, Journal of The Institution of Engineers (India): Series A. Vol. 103, No. 2, pp. 467-80, 2022.

[18] S Volosenko, A Laurinavicius, **Level of service evaluation methods: possible adaptation for Lithuani**a. The Baltic journal of road and bridge engineering. Vol. 15, No. 2, pp. 145-57, 2020.

[19] R Fan, MJ Bocus, Y Zhu, J Jiao, L Wang, F Ma, S Cheng, M Liu, **Road crack detection using deep convolutional neural network and adaptive thresholding**. IEEE Intelligent Vehicles Symposium, pp. 474-479. 2019.

[20] H Maeda, T Kashiyama, Y Sekimoto, T Seto, H Omata, **Generative adversarial network for road damage detection**. Computer-Aided Civil and Infrastructure Engineering, Vol. 36, No. 1, pp. 47-60, 2021.

[21] S Naddaf-Sh, MM Naddaf-Sh, AR Kashani, H Zargarzadeh. **An efficient and scalable deep learning approach for road damage detection**. IEEE International Conference on Big Data (Big Data), pp. 5602-5608, 2020.

[22] AK Pandey, R Iqbal, T Maniak, C Karyotis, S Akuma, V Palade, **Convolution neural networks for pothole detection of critical road infrastructure**. Computers and Electrical Engineering, Vol. 99, pp. 107725, 2022.

[23] K Dudekula, R Kiran, M Reddy, RR Shet, JS Vishwas, R Rohan, **Determination of Level of Service and Traffic Flow Characteristics for a Selected Arterial Roads in Bangalore City,** pp. 1- 5, 2021

[24] S Huang, Y He, XA Chen, M-YOLO: **A Nighttime Vehicle Detection Method Combining Mobilenet v2 and YOLO v3,** Journal of Physics: Conference Series, Vol. 1883, No. 1, pp. 012094), 2021.

[25] N Abulizi, A Kawamura, K Tomiyama, S Fujita, **Measuring and evaluating of road roughness conditions with a compact road profiler and ArcGIS**, Journal of Traffic and Transportation Engineering (English Edition), Vol. 3, No. 5, pp. 398-411, 2016.

[26] B Varona, A Monteserin, A Teyseyre, **A deep learning approach to automatic road surface monitoring and pothole detection**, Personal and Ubiquitous Computing, Vol. 24 No. 4, pp. 519-34, 2020.

[27] FX Ferdinandus FX, EI Setiawan, EM Yuniarno, MH Purnomo, **3D Visualization for Lung Surface Images of Covid-19 Patients based on U-Net CNN Segmentation,** EMITTER International Journal of Engineering Technology. Vol. 10. No. 2, pp. 320-337, 2022.