Journal of Advanced College of Engineering and Management

YOLOv11-s-Based Pothole Detection Model with Integrated Traffic Data for Maintenance Decision Support

[1*]S. Sivanraj, [2]D.N.L.S Uduwage, [3*]R.K. Shiwakoti, [4]M. Thripathi

[1,2]*Department of Building Economics, University of Moratuwa, Sri Lanka*

[3]*Institute of Engineering, Pulchowk Campus, Tribhuvan University, Nepal*

[4]*School of ICT, Thammasat University, Thailand*

[*]Corresponding Author: [*]ranju.shiwakoti@ioe.edu.np

DOI: 10.3126/jacem.v11i1.84529

Abstract

A pothole is a structural failure in a road that is a major cause of disturbances to traffic flow, particularly in developing countries where infrastructure maintenance is often delayed or inefficient. Traditional manual inspection methods for identifying potholes and planning repairs can be time-consuming. This study presents an automated pothole detection model using the latest YOLOv11-s object detection algorithm. Unlike previous research that focused on driver alerts and pothole classification based on the size of the pothole, this model integrates pothole detection with a traffic data API to prioritise road repair works based on traffic volumes. The dataset used for training, validation and testing was obtained from the RoboFlow platform, while images from Sri Lankan roads were used for deployment validation. The model achieved a precision of 0.851, a recall of 0.676, and mean average precision (mAP) scores of 0.802 (IoU@0.5) and 0.510 (IoU@0.5:0.95), demonstrating strong detection capabilities. The developed model can assist maintenance authorities in prioritising road repairs based on traffic volume. It can be further enhanced by integrating pothole characteristics, such as type and size, with traffic data to improve decision-making on the priority of repair works.

Keywords—*Object detection, Pothole, Road safety, Traffic volume, YOLOv11-s*

1. INTRODUCTION

Transportation systems play a significant role in economic and social development worldwide. However, ensuring the long-term quality and functionality of these systems can lead to significant challenges, particularly in the construction and maintenance of road infrastructure. In many regions, poor road design, material degradation, and inadequate maintenance contribute to surface defects such as potholes, which undermine the reliability and efficiency and quality of the transport network [1], [2].

Potholes lead to a range of operational issues, including reduced travel comfort, vehicle damage, increased travel time, and disruption to traffic flow [3]. These issues affect everyday road users and disrupt the traffic flow, and impose financial and logistical burdens on transportation authorities. In countries like India and Sri Lanka, where extensive road networks are heavily utilised, the timely detection and repair of potholes are crucial to maintaining road quality and supporting mobility. However, the urgency and priority of pothole repairs vary depending on contextual factors such as road type, usage intensity, traffic volume and nature of potholes.

In Sri Lanka, the road network is extensive and diverse, with about 150,000 km of roads categorised primarily by their function, volume, and administrative responsibility. The main categories include Expressways (E), national roads (A and B class roads managed by the Road Development Authority), provincial roads, and a large proportion classified as rural low-volume roads. The impact of potholes on these types of roads will vary according to the volume of traffic [4]. Therefore, it is necessary to analyse the traffic volume to determine the order in which the roads should be repaired.

Considering the large networks of roads, manual inspection and decision making on the order in which the roads should be repaired can be time-consuming and less cost-effective. Consequently, researchers have turned to automated pothole detection using cameras, sensors and deep learning techniques [5], [6]. Several studies have also explored the use of computer vision models for automated pothole detection with convolutional neural networks (CNNs) and object detection algorithms. For instance, Lakmal and Dissanayake [7] implemented image segmentation techniques to identify potholes on Sri Lankan roads, achieving a high detection accuracy. Similarly, various global studies have employed various You Only Look Once (YOLO) object detection algorithms to detect potholes and integrated them into mobile phone applications to notify presence of potholes to drivers [8]. Recent studies have shown that more advanced YOLO models, such as YOLOv8 [9] and YOLOv11-s [10] outperform earlier versions in both detection speed and accuracy. Various studies have also consistently shown that YOLO-based models are efficient and precise compared to other deep learning alternatives for pothole detection [11], [12].

While these systems have shown considerable effectiveness in performance, their primary focus was on real-time alerting for driver safety rather than maintenance planning. Therefore, there is a gap in research applying the latest YOLO architecture, YOLOv11-s, within the Sri Lankan context, where the traffic volume plays a critical role in prioritising maintenance works. Several studies have attempted to classify potholes based on size and depth, but they often overlook the importance of road classification based on traffic volume for determining repair urgency. Therefore, this study proposes a lightweight and efficient pothole detection model using the latest YOLO algorithm, YOLOv11-s. This model aims to determine the priority order of pothole repairs based on the traffic volume.

2. LITERATURE REVIEW

A. Classification of Potholes

Pothole classification and severity assessment are significant in determining the priority of road maintenance works. Potholes are generally classified based on features such as size, depth and surface area [13]. Studies have often used machine learning and image processing techniques to classify potholes according to such features. [14] developed a model that classifies pothole severity using colour disparity maps. The model used a severity scale determined by human-automated features that enables authorities to prioritise the severity of the potholes. Other approaches have classified potholes such as 'small' or 'large' based on surface area with high accuracy using CNN [15], [16]. Some studies have also attempted to consider the presence of water in potholes as a separate category in pothole severity classification [17].

However, existing models have overlooked the contextual importance of traffic volumes to prioritise the urgency of repair works. The impact of a pothole can cause significantly different risks depending on whether it appears on a high-traffic area or a low-traffic rural road. In high-speed zones, even minor potholes can lead to severe impacts in terms of traffic flow disturbances. Despite this, the classification of roads has not been integrated into pothole detection models. Therefore, incorporating traffic volume as a factor in severity assessment is critical for effective road maintenance planning.

B. Object Detection for Potholes

CNNs have been widely used for pothole detection in recent years due to their strong performance in object detection tasks [8], [18]. CNN-based object detection models can be generally classified into two categories: single-stage detectors and multi-stage detectors [19]. Single-stage detectors such as YOLO algorithms perform object classification and localisation simultaneously in a single step, which makes them ideal for real-time applications for object detection. In contrast, multi-stage detectors such as Fast Regional Convolutional Neural Networks (Fast R-CNN) and Mask Regional Convolutional Neural Network (Mask R-CNN), separate the object classification and localisation process. Multi-stage detectors often have high accuracy but slower inference speed in object detection.

Several studies have demonstrated the effectiveness of YOLO architectures in detecting potholes under diverse road conditions. For instance, Hiremath et al. [8] developed a mobile application that uses a YOLO-based model to detect potholes and notify drivers in real time. Other studies have explored the use of newer YOLO variants to improve detection accuracy and classify different pothole types. Bhavana et al. [9] proposed a model based on YOLOv8 to detect and categorise potholes, while other researchers have used YOLOv7 for enhancing traffic safety through real-time road surface monitoring [20]. Performance comparisons between different YOLO versions have also been explored in various studies. For example, Park (2021) conducted a study comparing YOLOv4 and YOLOv5, highlighting differences in detection speed and accuracy. More recently, Fortin and Llantos [10] evaluated the performance of the latest YOLO iterations such as YOLOv9, YOLOv10, and YOLOv11. The study revealed that YOLOv11 delivered the best overall performance for pothole detection tasks. The YOLOv11 algorithm is the latest version of the YOLO series, which offers significant improvements in object detection and classification (Fortin & Llantos, 2025; He et al., 2024).

The YOLO architecture typically consists of three main components: (1) the Backbone, responsible for feature extraction, (2) the Neck, which enables the model to detect objects of different sizes, and (3) the Head performs the final object classification and bounding box regression (He et al., 2024). According to Rasheed, object detection applications can be designed to focus on the object size to enhance resource efficiency. According to Kishor 2024, there are variants of YOLOv11 such as YOLOv11-n (nano), YOLOv11-s (small), YOLOv11-m (medium), YOLOv11-I (intermediate), and YOLOv11-x (extreme). Depending on the requirements, the suitable variant can be selected. This study uses the YOLOv11-s algorithm to develop the pothole detection model due to its ability to balance between speed and accuracy, ideal for medium-scale objects

C. Significance and Transferability

This study introduces an object detection model based on the YOLOv11-s algorithm to detect potholes and prioritise maintenance requirements according to the traffic volume on the road. While several studies have focused on pothole detection and severity classification based on the size and nature of the pothole, the outcomes have been used to provide real-time alerts to drivers and enhance road safety. None of the studies have attempted to classify potholes considering the traffic volume on the roads, despite their importance in decision-making for road maintenance works. This study addresses this gap by proposing a system that integrates object detection with a decision-making model to support maintenance planning based on road traffic volume data. The model was trained, tested and validated using an online dataset. To assess the model for deployment, pothole images from Sri Lankan roads were used. Although the focus of this study is on Sri Lankan roads, the model approach is transferable. By integrating the traffic volume data specific to other locations, the model can be generalised and implemented in other countries facing similar road infrastructure challenges.

3. METHODOLOGY

A. Dataset Collection

For the detection of the potholes, it was necessary to use a dataset for training, testing and validation. The dataset for this study was obtained from an online platform called RoboFlow. The platform provides a collection of high-quality images and metadata specifically designed for training computer vision models. The images were standardised to a resolution of 640×640×3 (RGB scale) to ensure the consistency of the dataset throughout the model training, testing and validation processes. The selected dataset consisted of 1,595, with 80% allocated for training, 10% for validation and 10% for testing. The split ensures that the model has a large enough dataset for training, validation and testing.

The YOLOv11-s algorithm has various hyperparameters that determine the performance and efficiency of the model [22]. These can be tuned before training the model and remain unchanged throughout the training process. In YOLOv11-s-based models, genetic algorithm used optimized the hyperparameters tuning through mutation process. The mutation process in the YOLOv11-s models introduces small random changes in the hyperparameters of the model to improve its accuracy without manual tuning. In such models, batch size, learning rate, momentum, weight decay, number of epochs and patience levels are the frequently tuned hyperparameters [23] . However, in this study, all hyperparameters were kept at their default values, which have already been optimised by the mutation process, with the exception of the number of epochs and patience level, which were adjusted to improve training efficiency and prevent overfitting.

The number of epochs refers to the number of times the entire dataset passes through the training. Studies have shown that models trained with 300 epochs have shown high performance and accuracy [24]. Therefore, in this study, the number of epochs was set at 300. However, object detection models can face issues such as overfitting due to a smaller dataset and a high number of epochs. This can be mitigated by using early stopping techniques [25], [26] and data augmentation [27]. Early stopping value can be set using the patience level, which defines the number of epochs the model will continue to train for without any performance improvements. Once the patience level is reached, the training will be halted. Data augmentation allows for generating new data from existing data through methods like geometric transformations, colour adjustments, and noise addition [28]. In the developed model, an early stopping with a patience level of 20 epochs was set, where the model will stop training when there is no improvement in performance observed for 20 consecutive epochs.

B. Model Training

As illustrated in Figure 1, the initial stage involved training the pothole detection models using 80% of the pothole dataset. For the training, two types of inputs were used such as plain image and the metadata (annotations) that provided information about the image and the objects within it. In the pothole dataset, the annotation defined the bounding box around the pothole. The bounding box coordinates consist of 5 components (X, Y, W, H and confidence score) as shown in Figure 2. The X and Y values represent the normalised coordinates of the centre of the bounding box, while the W and H represent the normalised height and width of the bounding box. The confidence score is also referred to as the Intersection over Union (IoU), which represents the ratio between the prediction and the ground truth, as shown in Figure 3.

IoU measures the ratio of the overlapping area between the prediction and the ground truth relative to the total area covered by both. Figure 3 displays three IoU value representations, where 4A indicates relatively low overlap and imprecise localisation, resulting in an IoU of 0.475. 4B depicts a better prediction with an IoU of 0.741, while 4C demonstrates the highest detection accuracy with an IoU of 0.891.

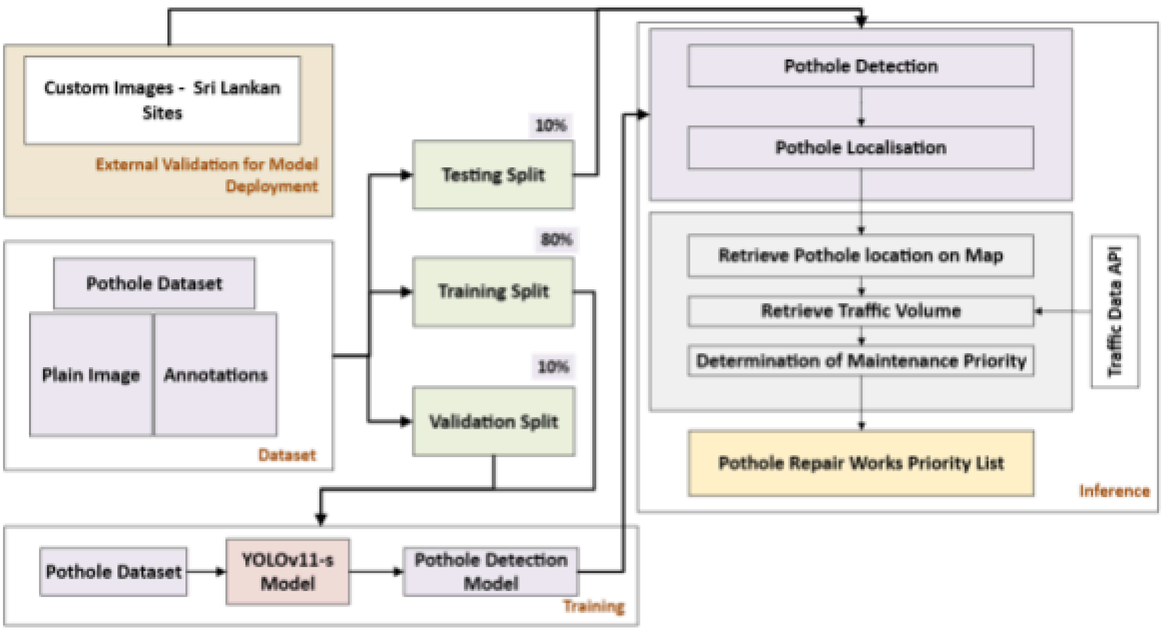


*Figure 1 SEQ Figure * ARABIC 1: Overall Architectural Diagram for Developed Model*

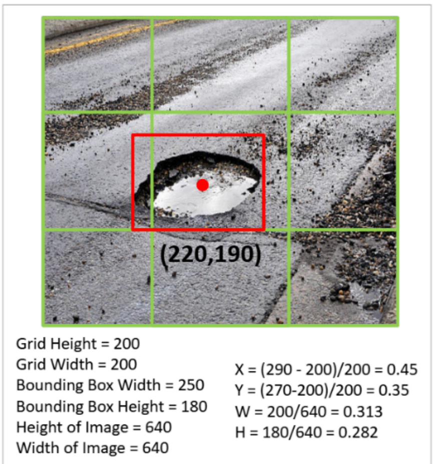


Figure 2: Bounding Box Coordinates

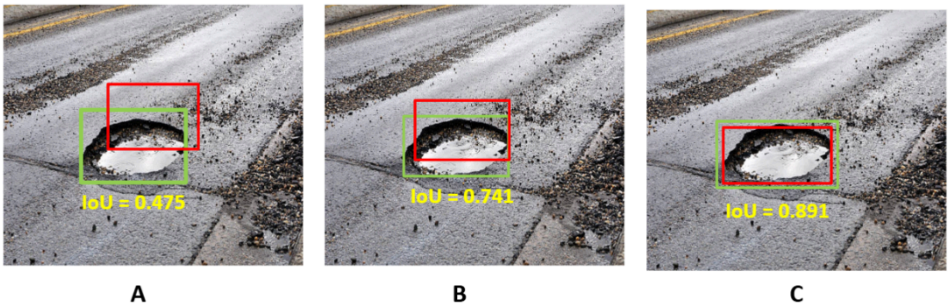


Figure 3: Calculation of IoU values

C. Model Validation (Internal)

The performance of the model was monitored throughout the training process using the validation dataset split (10% of the used dataset). For every training epoch, the model was evaluated using the validation dataset to assess the improvements in performance. This phase ensured that the optimum level of training was given to the model so that it was able to detect potholes from unseen data.

D. Model Testing

Upon the completion of internal validation, the model was tested using the remaining 10% of the dataset through the inference pipeline. Inference is the process of using a trained model to make predictions on unseen data. As illustrated in Figure 1, each test image is passed through the model to detect and localise potholes. The performance of the model was evaluated using metrics such as precision values, recall values, precision-recall curves, mean Average Precision (mAP) scores and loss curves.

E. Integration of Traffic Data API

Once a pothole is detected, the model retrieves its location using embedded GPS metadata and identifies the corresponding road segment. The traffic volume for each road is then obtained through a traffic data Application Programming Interface (API) (See Figure 4), which provides real-time or average daily vehicle count data. Based on this information, the model assigns a repair priority to each pothole, giving higher priority to those found on roads with consistently high traffic volumes.

This priority-based ranking system enables road maintenance teams to focus on repairing potholes in the most critical and heavily used areas first, improving the efficiency and safety of maintenance efforts. Furthermore, the model was enhanced to continuously retrieve traffic flow data in specified areas, enabling it to identify traffic patterns and determine the urgency of repairs. Roads consistently showing high traffic volume are flagged as high-priority zones, ensuring that maintenance decisions are guided by both object detection and traffic data.

```
def fetch_traffic_data():
    url = (
        f"https://api.tomtom.com/traffic/services/4/flowSegmentData/relative0/10/json?"
        f"point={LAT}%2C{LON}&unit=KMPH&key={API_KEY}"
    )
    response = requests.get(url)
    if response.status_code != 200:
        print("Error:", response.status_code, response.text)
        return None
    return response.json()
```

Figure 4: Retrieval of Traffic Volume Data

F. Model Validation (External)

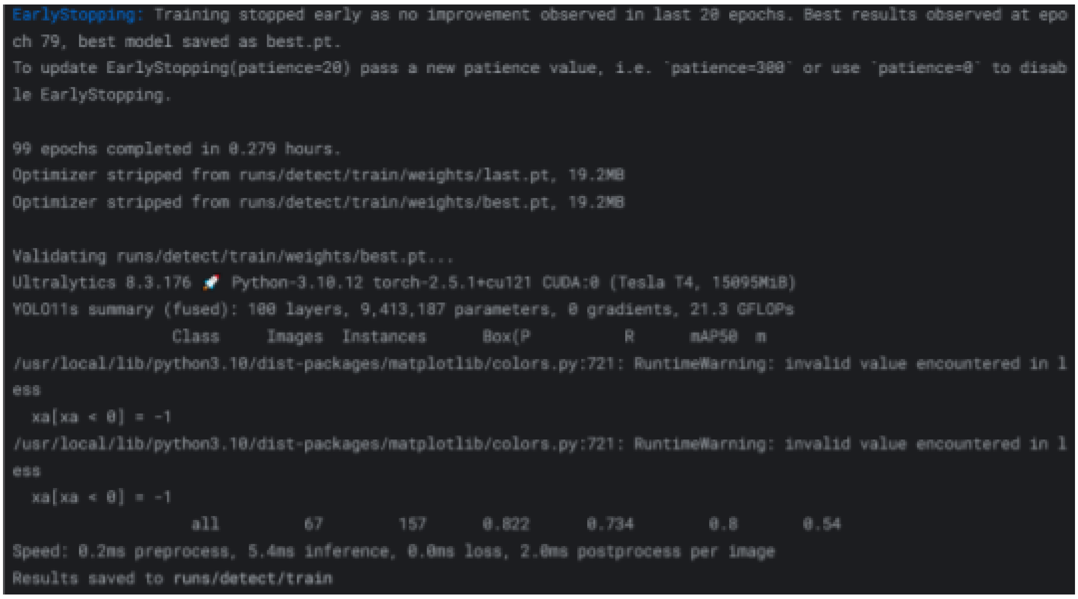
The model was validated for deployment using images captured from various Sri Lankan roads. These images were passed through the model's inference pipeline to evaluate its performance in detecting potholes and determining repair priorities by considering the traffic volume.

4. RESULTS AND PERFORMANCE EVALUATION

The YOLOv11-s model was trained with a total of 300 epochs set as the upper limit. However, due to the implementation of the early stopping hyperparameter, training was halted after 99 epochs, which were completed in approximately 0.279 hours. Early stopping prevents overfitting by terminating the training once the model's performance plateaus on the validation set. During this training period, the model achieved its best performance. The inference speed was recorded at 5.4ms (see Figure

5), indicating the time the model takes to process an image and output the detection results.

To evaluate the performance of the object detection model, several standard metrics were used, including precision, recall, mean average precision (mAP), and loss curves [23], [29], [30]. This section discusses the major performance evaluation metrics used to analyse the performance of the developed model.


```
EarlyStopping: Training stopped early as no improvement observed in last 20 epochs. Best results observed at epo
ch 79, best model saved as best.pt.
To update EarlyStopping(patience=20) pass a new patience value, i.e. `patience=300` or use `patience=0` to disab
le EarlyStopping.

99 epochs completed in 0.279 hours.
Optimizer stripped from runs/detect/train/weights/last.pt, 19.2MB
Optimizer stripped from runs/detect/train/weights/best.pt, 19.2MB

Validating runs/detect/train/weights/best.pt...
Ultralytics 8.3.176 🚀 Python-3.10.12 torch-2.5.1+cu121 CUDA:0 (Tesla T4, 15095MiB)
YOLO11s summary (fused): 100 layers, 9,413,187 parameters, 0 gradients, 21.3 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  m
/usr/local/lib/python3.10/dist-packages/matplotlib/colors.py:721: RuntimeWarning: invalid value encountered in l
ess
  xa[xa < 0] = -1
/usr/local/lib/python3.10/dist-packages/matplotlib/colors.py:721: RuntimeWarning: invalid value encountered in l
ess
  xa[xa < 0] = -1
                   all         67        157      0.822      0.734        0.8       0.54
Speed: 0.2ms preprocess, 5.4ms inference, 0.0ms loss, 2.0ms postprocess per image
Results saved to runs/detect/train
```

Figure 5: Model Performance Values

A. Precision

Precision refers to the proportion of correctly identified potholes (true positives) among all instances that the model predicted as potholes (true positives + false positives). A high precision score implies that the model makes fewer false positive detections, meaning it is less likely to wrongly identify background features as potholes. According to the confusion matrix, the developed model achieved a precision score of 0.822, indicating strong accuracy in distinguishing true potholes.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad (Equation\ 1)$$

B. Recall

Recall measures the model's ability to detect actual potholes in the dataset. It is the ratio of true positive detections to all actual positive cases. The model obtained a recall score of 0.734, demonstrating its effectiveness in detecting the relevant instances, though some were missed.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad (Equation\ 2)$$

C. Precision-Recall Curve

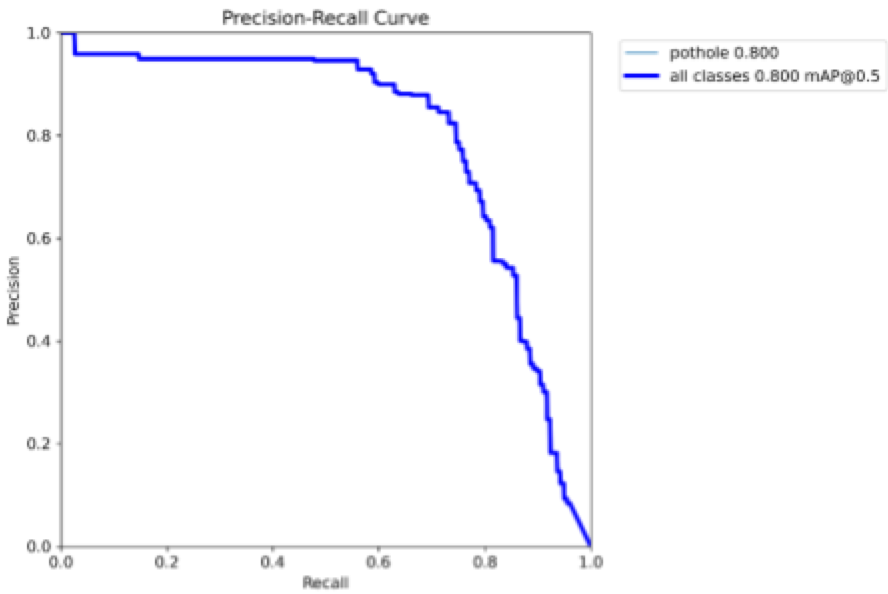


Figure 6: Precision-recall curve

The precision-recall (PR) curve visualises the trade-off between precision and recall at different threshold levels. A high precision with low recall indicates that while most of the model's predictions are correct, it fails to detect all instances. Conversely, high recall with low precision means the model identifies most true positives but includes more false positives. An ideal model offers a balance between these two metrics. As shown in Figure 6, the precision-recall curve for the developed model demonstrates a reasonable balance, supporting its suitability for pothole detection in real-world conditions [31].

D. Mean Average Precision

Mean Average Precision (mAP) is a widely used metric that combines both precision and recall to evaluate the accuracy of object detection models [32]. It measures how well the predicted bounding boxes overlap with the ground truth boxes. The mAP value is calculated as shown in Equation 3. In this study, since the model only performs detection (not multi-class classification), the mAP score is equivalent to the Average Precision (AP). The model's performance was evaluated at two Intersection over Union (IoU) thresholds: mAP@0.5 and mAP@0.5:0.95. The developed model achieved a mAP@0.5 of 0.800, indicating high accuracy in detecting potholes with a moderate overlap threshold. However, the model achieved a score of 0.540 for mAP@0.5:0.95. Although this is a relatively lower score, it demonstrates the model's ability to generalise well for higher thresholds.

Journal of Advanced College of Engineering and Management

$$mAP = \frac{1}{number\ of\ classes\ (n)} \sum_{i=1}^{n.} Average\ Precision\ \ (Equation\ 3)$$

E. Training and Validation Loss

The training process was further assessed using training and validation loss curves, which indicate the model's ability to learn and generalise. YOLOv11-s uses a unified loss function comprising three components:

- Bounding Box Loss (box_loss) measures the accuracy of the predicted bounding boxes compared to the ground truth.
- Classification Loss (cls_loss) evaluates the model's performance in correctly identifying the object class.
- Distribution Focal Loss (dfl_loss) addresses class imbalance and improves localisation accuracy, especially for smaller objects [33].

As shown in Figure 7, both the training and validation losses decreased steadily with increasing epochs, indicating effective learning and proper model optimisation. The consistent reduction in loss reflects the model's capacity to improve detection performance while avoiding overfitting.

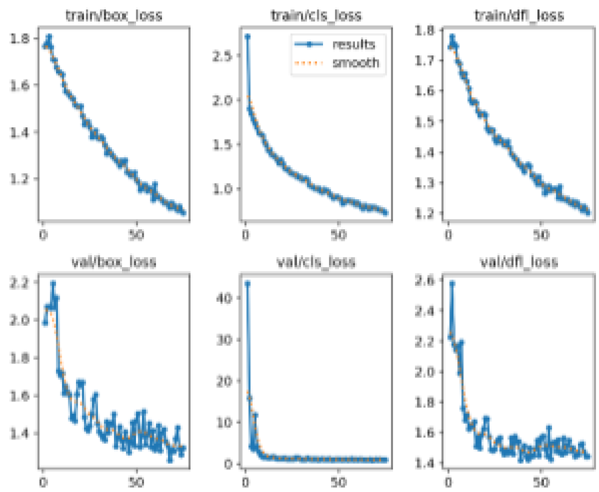


Figure 7: Training and Validation loss

F. Model output

Based on the labelled dataset, the YOLOv11-s was employed to detect potholes in the target road sections. The detection outcomes of the model are presented in Figure 8.

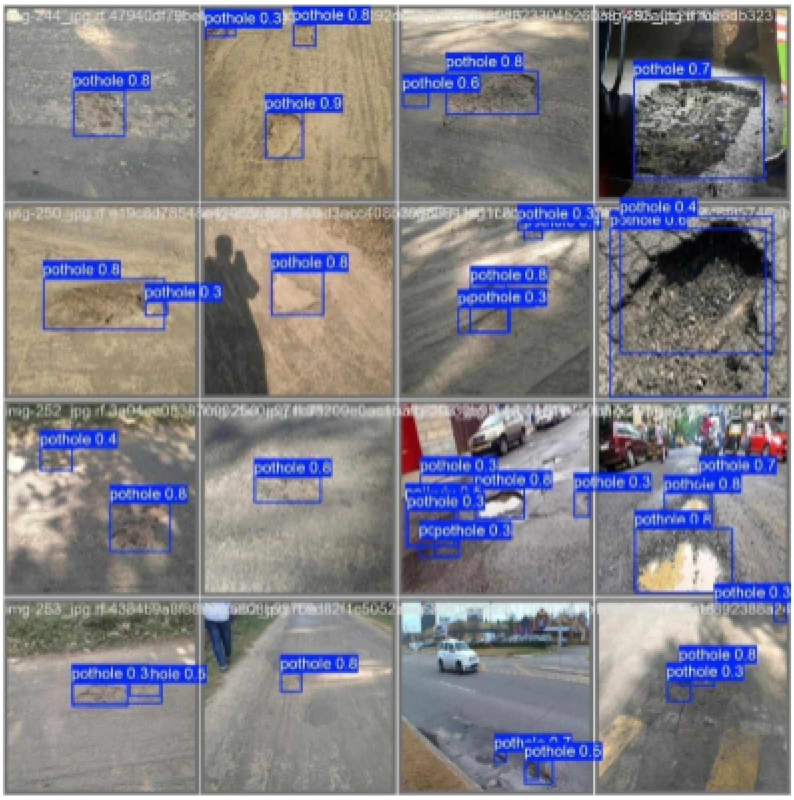


Figure 8: Results of pothole detection

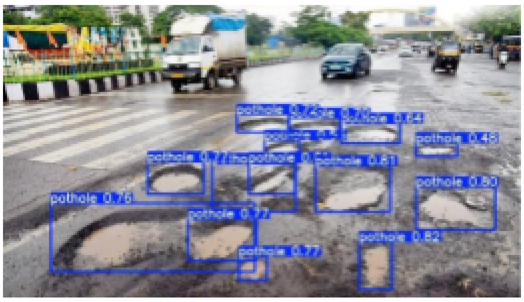


Figure 9: Pothole Detections on Sri Lankan Roads 1

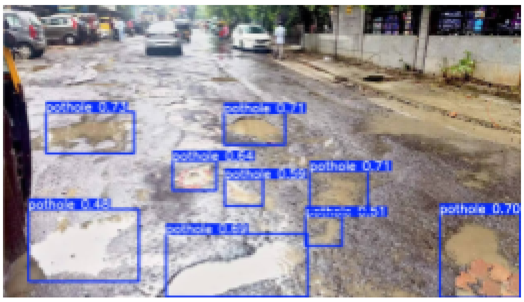


Figure 10: Pothole Detections on Sri Lankan Roads 2

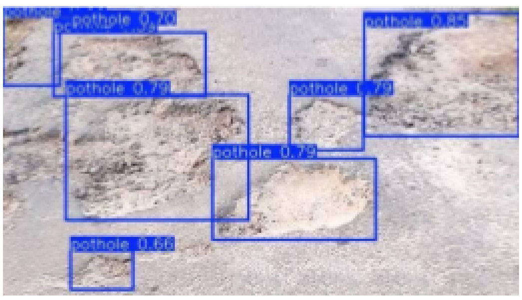


Figure 11: Pothole Detections on Sri Lankan Roads 3

The results of the deployment validation carried out using pothole images from Sri Lankan Roads have been provided in Figures 9, 10 and 11. The model accurately recognised and localised potholes in the input images using bounding boxes, following the trained detection framework. In addition to detection, the model utilised traffic volume data retrieved through the traffic data API to assess the urgency and severity of each pothole.

The severity was inferred by considering the traffic volume, such as the road type and its traffic flow. Potholes detected on roads with high traffic volume as per the traffic data API, were classified as high priority due to the increased safety risks and potential for traffic disruption. Conversely, potholes on low-volume rural or residential roads were assigned lower priority for maintenance. The final output consisted of annotated images showing the detected potholes, their bounding boxes, and associated confidence scores. Each detection was also associated with its location, road classification, and assigned repair priority. The model's output enables road maintenance authorities to schedule and allocate resources efficiently.

5. CONCLUSION AND RECOMMENDATION

Previous studies on pothole detection have explored a range of computer vision tools, including image segmentation and object detection, to identify potholes and classify their severity based on the physical characteristics of the potholes. These approaches have been used to alert drivers in real-time and improve road safety. However, none of the studies have incorporated traffic volume into the severity assessment of the potholes despite its relevance in decision-making for road maintenance. The findings of this study address that gap by demonstrating that the developed model can support maintenance authorities in prioritising pothole repairs based on actual road usage.

To address this gap, this study developed a deep learning-based pothole detection model using the YOLOv11-s algorithm to identify potholes and prioritise their repair based on traffic volume. The model was trained using an online pothole image dataset and integrated with a traffic data API to retrieve the traffic volume flow information. Once potholes were detected, the model determined the urgency of repair by assigning higher priority to potholes located on high-traffic roads. This approach enables road maintenance authorities to focus on the most critical areas first.

The model's performance was evaluated using key metrics such as precision, recall, and mean Average Precision (mAP). It achieved a high precision of 0.851 and competitive mAP scores of 0.802 at IoU 0.5 and 0.510 at IoU 0.5:0.95, demonstrating its effectiveness in accurately detecting and localising potholes. Validation with images from Sri Lankan roads confirmed the model’s practical applicability and adaptability to local conditions. The developed model can be further enhanced by integrating pothole characteristics, such as type and size, with traffic data to improve decision-making on the urgency of repair works.

REFERENCES

[1] H. Lu, M. Chen, and W. Kuang, "The impacts of abnormal weather and natural disasters on transport and strategies for enhancing ability for disaster prevention and mitigation," *Transp Policy (Oxf)*, vol. 98, pp. 2–9, Nov. 2020, doi: 10.1016/j.tranpol.2019.10.006.

[2] N. Stojanovic, B. Boskovic, M. Petrovic, I. Grujic, and O. I. Abdullah, "The impact of accidents during the transport of dangerous good, on people, the environment, and infrastructure and measures for their reduction: a review," *Environmental Science and Pollution Research*, vol. 30, no. 12, pp. 32288–32300, Feb. 2023, doi: 10.1007/s11356-023-25470-2.

[3] D. R. Rajan, M. K. Faizan, R. Kundelu, N. Nandal, and V. S. Gunda, "Deep learning based pothole detection," in *2023 International Conference on Emerging Smart Computing and Informatics (ESCI)*, IEEE, Mar. 2023, pp. 1–5. doi: 10.1109/ESCI56872.2023.10100020.

[4] A. P. Senasinghe, A. De Barros, S. C. Wirasinghe, and R. Tay, "Factors affecting crash severity on two major intercity roads in western Sri Lanka: A random parameter logit approach," *Journal of South Asian Logistics and Transport*, vol. 4, no. 2, pp. 41–65, Sep. 2024, doi: 10.4038/jsalt.v4i2.91.

[5] R. Madli, S. Hebbar, P. Pattar, and V. Golla, "Automatic detection and notification of potholes and humps on roads to aid drivers," *IEEE Sens J*, vol. 15, no. 8, pp. 4313–4318, Aug. 2015, doi: 10.1109/JSEN.2015.2417579.

[6] F. Ali, Z. H. Khan, K. S. Khattak, and T. A. Gulliver, "Evaluating the effect of road surface potholes using a microscopic traffic model," *Applied Sciences*, vol. 13, no. 15, p. 8677, Jul. 2023, doi: 10.3390/app13158677.

[7] H. K. I. S. Lakmal and M. B. Dissanayake, "Pothole detection with image segmentation for advanced driver assisted systems," in *2020 IEEE International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (WIECON-ECE)*, IEEE, Dec. 2020, pp. 308–311. doi: 10.1109/WIECON-ECE52138.2020.9398036.

[8] R. Hiremath, K. Malshikare, M. Mahajan, and R. V. Kulkarni, "A Smart App for Pothole Detection Using Yolo Model," in *Lecture Notes in Networks and Systems*, Springer Science and Business Media Deutschland GmbH, 2021, pp. 155–164. doi: 10.1007/978-981-15-8354-4_16.

[9] N. Bhavana, M. M. Kodabagi, B. M. Kumar, P. Ajay, N. Muthukumaran, and A. Ahilan, "POT-YOLO: Real-time road potholes detection using edge segmentation-based Yolo V8 network," *IEEE Sens J*, vol. 24, no. 15, pp. 24802–24809, 2024, doi: 10.1109/JSEN.2024.3399008.

[10] L. V. Fortin and O. E. Llantos, "Performance analysis of YOLO versions for real-time pothole detection," *Procedia Comput Sci*, vol. 257, pp. 77–84, 2025, doi: 10.1016/j.procs.2025.03.013.

[11] E. S. T. K. Reddy and V. Rajaram, "Pothole detection using CNN and YOLO v7 algorithm," in *2022 6th International Conference on Electronics, Communication and Aerospace Technology*, IEEE, Dec. 2022, pp. 1255–1260. doi: 10.1109/ICECA55336.2022.10009324.

[12] M. Jakubec, E. Lieskovská, B. Bučko, and K. Zábovská, "Comparison of CNN-Based Models for Pothole Detection in Real-World Adverse Conditions: Overview and Evaluation," May 01, 2023, *MDPI*. doi: 10.3390/app13095810.

[13] W. Shin, K. Kang, and S. Moon, "Advancing Road Infrastructure Management and Safety Through Pothole Classification Standards and Technology: A South Korean Perspective," 2024, doi: 10.6106/ICCEPM.2024.1035.

[14] S. Bhatia and R. Greer, "Classifying pothole severity with convolutional neural networks," in *2023 IEEE MIT Undergraduate Research Technology Conference (URTC)*, IEEE, Oct. 2023, pp. 1–4. doi: 10.1109/URTC60662.2023.10534954.

[15] A. A. Hasanaath, A. S. Mohammed, L. Alotaibi, M. Alabdulrahman, and A. Moinuddeen, "Pothole severity classification using structural features extracted from segmented images," in *2024 Arab ICT Conference (AICTC)*, IEEE, Feb. 2024, pp. 52–58. doi: 10.1109/AICTC58357.2024.10735016.

[16] C. F. Ahmad *et al.*, "Classification of Potholes Based on Surface Area Using Pre-Trained Models of Convolutional Neural Network," 2023.

[17] D. Wang, Z. Liu, X. Gu, W. Wu, Y. Chen, and L. Wang, "Automatic detection of pothole distress in asphalt pavement using improved convolutional neural networks," *Remote Sens (Basel)*, vol. 14, no. 16, p. 3892, Aug. 2022, doi: 10.3390/rs14163892.

[18] A. Aparna, Y. Bhatia, R. Rai, V. Gupta, N. Aggarwal, and A. Akula, "Convolutional neural networks based potholes detection using thermal imaging," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 3, pp. 578–588, Mar. 2022, doi: 10.1016/j.jksuci.2019.02.004.

[19] S. S. Sumit, J. Watada, A. Roy, and D. R. A. Rambli, "In object detection deep learning methods, YOLO shows supremum to Mask R-CNN," in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Jun. 2020. doi: 10.1088/1742-6596/1529/4/042086.

[20] M. Sathvik, G. Saranya, and S. Karpagaselvi, "An Intelligent Convolutional Neural Network based Potholes Detection using Yolo-V7," in *2022 International Conference on Automation, Computing and Renewable Systems (ICACRS)*, IEEE, Dec. 2022, pp. 813–819. doi: 10.1109/ICACRS55517.2022.10029263.

[21] Z. He, W. Kang, T. Fang, L. Su, R. Chen, and X. Fei, "Comprehensive Performance Evaluation of YOLOv11, YOLOv10, YOLOv9, YOLOv8 and YOLOv5 on Object Detection of Power Equipment," 2024.

[22] P. P. Ippolito, "Hyperparameter Tuning," 2022, pp. 231–251. doi: 10.1007/978-3-030-88389-8_12.

[23] M. A. R. Alif, "YOLOv11 for Vehicle Detection: Advancements, Performance, and Applications in Intelligent Transportation Systems," Oct. 2024, [Online]. Available: http://arxiv.org/abs/2410.22898

[24] J. Ieamsaard, S. N. Charoensook, and S. Yammen, "Deep Learning-based Face Mask Detection Using YoloV5," in *2021 9th International Electrical Engineering Congress (iEECON)*, IEEE, Mar. 2021, pp. 428–431. doi: 10.1109/iEECON51072.2021.9440346.

[25] D. Liao, J. Zhang, Y. Tao, and X. Jin, “ATBHC-YOLO: aggregate transformer and bidirectional hybrid convolution for small object detection,” *Complex and Intelligent Systems*, vol. 11, no. 1, Jan. 2025, doi: 10.1007/s40747-024-01652-4.

[26] M. L. Ali and Z. Zhang, “The YOLO Framework: A Comprehensive Review of Evolution, Applications, and Benchmarks in Object Detection,” Dec. 01, 2024, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/computers13120336.

[27] T. Kumar, R. Brennan, A. Mileo, and M. Bendechache, “Image Data Augmentation Approaches: A Comprehensive Survey and Future Directions,” *IEEE Access*, vol. 12, pp. 187536–187571, 2024, doi: 10.1109/ACCESS.2024.3470122.

[28] B. Li, Y. Hou, and W. Che, “Data augmentation approaches in natural language processing: A survey,” *AI Open*, vol. 3, pp. 71–90, 2022, doi: 10.1016/j.aiopen.2022.03.001.

[29] M. Tripathi, “Analysis of Convolutional Neural Network based Image Classification Techniques,” *Journal of Innovative Image Processing*, vol. 3, no. 2, pp. 100–117, Jun. 2021, doi: 10.36548/jiip.2021.2.003.

[30] S. Sivanraj, N. L. S. Uduwage Don, and M. Tripathi, “Real-Time Safety Detection on Construction Sites Using a Vision-Language and Nlp-Based Model,” 2025. doi: 10.2139/ssrn.5275809.

[31] B. Juba and H. S. Le, “Precision-Recall versus Accuracy and the Role of Large Data Sets,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 4039–4048, Jul. 2019, doi: 10.1609/aaai.v33i01.33014039.

[32] P. Henderson and V. Ferrari, “End-to-End training of object class detectors for mean average precision,” 2017, pp. 198–213. doi: 10.1007/978-3-319-54193-8_13.

[33] L. He, Y. Zhou, L. Liu, and J. Ma, “Research and Application of YOLOv11-Based Object Segmentation in Intelligent Recognition at Construction Sites,” *Buildings*, vol. 14, no. 12, Dec. 2024, doi: 10.3390/buildings14123777.