An Automated Approach for Detecting Road Surface Issues Using Deep Learning

Heba Fasihuddin hafasihuddin@uj.edu.sa
Department of Information systems and Technology
College of Computer Science and Engineering
University of Jeddah
Jeddah, Saudi Arabia

Corresponding Author: Heba Fasihuddin

Abstract

Road maintenance is crucial to facilitate safe and sustainable transportation systems. Conventional inspection methods, which are mainly based on human observations, are often slow and inefficient, however, particularly as expansion of urban environments continues. Contemporary advances in technology and smart systems and the expansion of deep learning capabilities in some networks has led to the automation of this process and the development of intelligent monitoring systems. Deep learning models enable the detections and classification of road defects issues in efficient ways, leading to the focus of this paper. The goal of this study was to develop a model to detect and classify various road issues using deep learning models. The datasets used to facilitate this included four classes representing four different road issues, namely cracks, potholes, water collection, and drain cover damage. Three models were evaluated across these datasets: YOLOv11X, Faster R-CNN, and RetinaNet, with similar Precision results emerging for all three models at 82.1%, 88.1%, and 82.4%, respectively. However, while Faster R-CNN achieved the highest precision value, taking overall performance and evaluation metrics into account showed YOLOv11X to be the most balanced model to offer satisfactory results. Overall, the findings of this study show this to be a promising approach that can be evolved and embedded into smart systems to generate real-time reporting for responsible authorities to ensure that the latter can develop safer and better managed and maintained urban infrastructures.

Keywords: Machine learning, Road surface, Artificial intelligence, Computer vision, Deep learning.

1. INTRODUCTION

Advancements in technology and machine learning have significantly impacted various fields of research and development. The improvement of road safety and traffic management is one such field, and this has garnered considerable attention from researchers and developers in recent years [1]. Numerous studies have explored the use of machine learning and deep learning to enhance

Citation: Heba Fasihuddin. An Automated Approach for Detecting Road Surface Issues Using Deep Learning. **Advances in Artificial Intelligence and Machine Learning**.2026. (Ahead of Print.) https://dx.doi.org/10.54364/AAIML.2026.63302

road safety management and maintenance [2]. To be more precise, the researchers at [3], revealed that monitoring and reporting road concerns in major cities is now a significant challenge, requiring prompt action to avoid potential risks.

This paper focuses on addressing the efficient monitoring and detection of common road defects, including cracks, water build-up, damaged drain covers, and potholes. Such issues are even more prevalent after natural disasters and adverse weather events and can endanger the safety of the public, including drivers, pedestrians, and cyclists. Moreover, it impacts the efficiency and robustness of infrastructure if not addressed. Nonetheless, traditional approaches to inspecting roads for such problems are often inefficient and time-consuming. Therefore, modern approaches are increasingly implementing technology-driven solutions in an effort to automate the process [4]. In turn, modern solutions include the deployment of road assessment vehicles that possess advanced imaging and detection technologies (i.e., sensors and cameras), with newer technologies (i.e., drones and real-time processing) resulting in the development of more innovative tools and substantial changes in road management. Machine and deep learning algorithms are applied in such technologies to accurately identify and report road defects, thus improving the management of urban infrastructure

This study thus aimed to exploit the benefits of deep learning algorithms to create a model that can detect a multitude of road defects. The main aim of this study was to develop a model that is able to classify road issues. This model can be embodied into a smart monitoring system to facilitate real-time reporting and ultimately promote a more efficient road management system. The addition of an intelligent system such as this will improve road safety and reduce potential infrastructure damage by ensuring that roads are sufficiently maintained. Thus, safer and more robust urban environments are achieved.

This paper is organized as follows: a discussion of relevant road defects is presented in Section 2 to offer some background to the work. This is followed by an overview of related work, presented in Section 3. Subsequently, in section 4, the research methodology will be discussed, with data collection and preprocessing practices being described in section 5 and model development being discussed in section 6. Section 7 will present the research results, which will be subject to a discussion in Section 8. Finally, section 9 will conclude the paper.

2. BACKGROUND

The safety of roads and the condition of vehicles are both largely impacted by faults in road infrastructure, including cracks, potholes, damaged drain covers, and rainwater buildup. Such issues tend to develop naturally over time as a result of environmental factors, natural changes, and constant vehicle pressure. Nonetheless, they can occur at a quicker rate after natural disasters such as flooding or torrential rain. In turn, these problems can reduce the safety of roads and cause traffic hazards. Traditional inspection methods, most of which involve human checks and reporting, are incapable of performing large-scale and continuous assessments, meaning that automated solutions are required, particularly in large industrial and economic cities.

Recent research has examined the use of machine learning and deep learning algorithms to detect road issues and surface defects [2, 5–8]. Such initiatives have allowed for infrastructure (particularly road surfaces) to be assessed in real-time, detecting issues such as cracks, potholes, and surface damage with high precision [6, 8]. In turn, advancements like this can significantly improve road safety and environmental management whilst also increasing maintenance efficiency, highlighting the importance of AI in the development of intelligent transportation systems

Section 3 presents an overview of some related works that have already applied different algorithms to the development of models to detect road defects and enhance road and infrastructure management.

3. RELATED WORK

Various studies have already implemented and assessed the accuracy of a range of models in the detection of road defects. This section highlights several studies that have made particularly remarkable achievements in this field.

Several studies have focused on the application of Convolutional Neural Networks (CNNs) to road surface problem detection, with their focuses covering a range of challenges such as potholes, cracks, rainwater collection, and damaged drain covers. In [5], the researchers presented a method for pothole detection using 1D-CNNs with different kernel sizes, achieving impressive classification results with a precision of 98%, a recall of 84%, and an F1-score of 91%. Their work focused on binary classification (pothole or no-pothole), developing a solid foundation for later expansion to multiple classes, as proposed in this paper. The researchers at [7], proposed a crack detection system based on a CNN-based model and adaptive image segmentation. This provided data about road surface cracks with 99.92% precision. Similarly, the researchers at [6], improved the field by developing a semi-supervised learning algorithm that could detect damage to road surfaces using data obtained by a car camera, with the semi-supervised approach appearing to be more effective than fully supervised approaches. Finally, the study at [9], proposed the use of CNNs that could produce very high-resolution aerial and remote sensing images to detect small objects (i.e., manhole covers), which was found to have high detection accuracy and to offer promising results when it comes to localizing manhole covers in intricate settings. When combined, these studies highlight the effectiveness of CNNs to address a multitude of road detection challenges, creating a framework for developing a comprehensive deep-learning-based system that can detect cracks, potholes, damaged manhole covers, and rainwater accumulation areas.

In addition to these key studies, other studies have supported the role of advanced object detection techniques using YOLO models in terms of addressing road surface issues such as potholes, cracks, and manhole covers. The study presented in [10], developed a pothole detection system using models such as SSD-TensorFlow, YOLOv3, and YOLOv4, with YOLOv3, achieving a mean Average Precision (mAP) of 65.65%. That system recorded pothole locations and visualized them using the Google Maps API. The researchers in [8], proposed a pothole detection system that also estimated the dimensions of potholes, determining in the process that YOLOv4 outperformed YOLOv3. While that approach was restricted to potholes, the dimension estimation technique proposed has since contributed to the further development of models across the field, particularly multi-classification models of road issues as proposed in this study.

Another study addressed manhole cover detection by applying several models, including MGB-YOLO, YOLOv5s, and Faster RCNN, to detect and classify qualified and unqualified covers [11].

Table 1: Summary of previous related work

Study	Dataset size	Used model	Model with best results
Study [5]	33,360	Various CNN models	1D-CNN, 2 hidden layers, Kernel size 5
Study [7]	40,000	Deep Learning with adaptive thresholding	CNN with ReLU
Study [6]	40,536	CNN (Semi-supervised)	Semi-supervised CNN
Study [9]	477,320	AlexNet variants	Customized AlexNet with cleaned database
Study [10]	1,087	SSD-TensorFlow, YOLOv3-Darknet53, YOLOv4 CSPDarknet53	YOLOv3-Darknet53
Study [8]	1,300	YOLOv3, YOLOv4	YOLOv4 at 4000 Iterations
Study [11]	6,000	MGB-YOLO, YOLOv5s, SSD, Faster RCNN, YOLOv7, YOLOv8s	MGB-YOLO
Study [12]	26,336	YOLOv7 with fine-tuning techniques	YOLOv7 with attention coordination

MGB-YOLO, which is an improved version of YOLOv5s, achieved a high accuracy of 96.6% in this process, demonstrating the effectiveness of augmenting YOLO to facilitate small object detection, which aligns with the proposed objective of this research in terms of the detection of multiple types of road issues. Similarly, the authors employed YOLOv7 in [12], enhanced with coordinate attention and label smoothing, to detect various road damage types, achieving an F1 score of 81.7%. They employed large datasets from a variety of regions, thus creating a strong foundation upon which to develop more robust models that can be employed in diverse environments.

Moreover, recent studies examined innovative approaches to the detection of road surface defects by implementing Recurrent Neural Network (RNN) and Visual Geometry Group (VGG) architectures. Various environmental conditions, including floodwater and moisture, were considered. For example, the researchers in [13], proposed a floodwater detection system that employed an image classification model to distinguish between dry and flooded road images to help with vehicle routing and traffic management. The dataset consisted of 491 images, and several feature extraction methods were employed. The results showed that this approach had major implications for flood detection, as well as road safety and traffic management improvements during periods of adverse weather.

TABLE 1 presents the key components and achievements of the aforementioned studies and approaches.

Table 2: Dataset classes

Num	Class name	Number of images
0	Cracks	1,223
1	Rainwater	1,325
2	drain covers	1,616
3	Potholes	1,225
Integrated classes	Total	5,389

4. METHODOLOGY

Data science methodology is implemented in this research, in which the following phases were implemented:

• Problem Domain Understanding: Understanding the study domain is a crucial phase for developing an efficient classification model. Understanding road issues such as cracks, potholes, drain covers, and rainwater collection, as well as identifying the key features of each issue, were very important steps toward achieving the aim of the study.

• Data Collection: The second phase was data collection ain which appropriate publicly available datasets were selected. The dataset selection ensured diversity in the images, conditions, and locations captured, to cover a wide variety of scenarios.

• Data preparation and processing: Considering the nature of datasets, this phase involved standardizing image sizes and resolutions to ensure consistency and appropriateness for model training. A number of other image augmentation tasks were also carried out to enhance the variety and robustness of the data.

• Data Modeling: This step involved developing a model that could extract features from images, detect road issues and ultimately classify them. The dataset was split into training, testing and validation sets, with each model being trained, tested, and assessed in terms of capturing and classifying road defects with high accuracy.

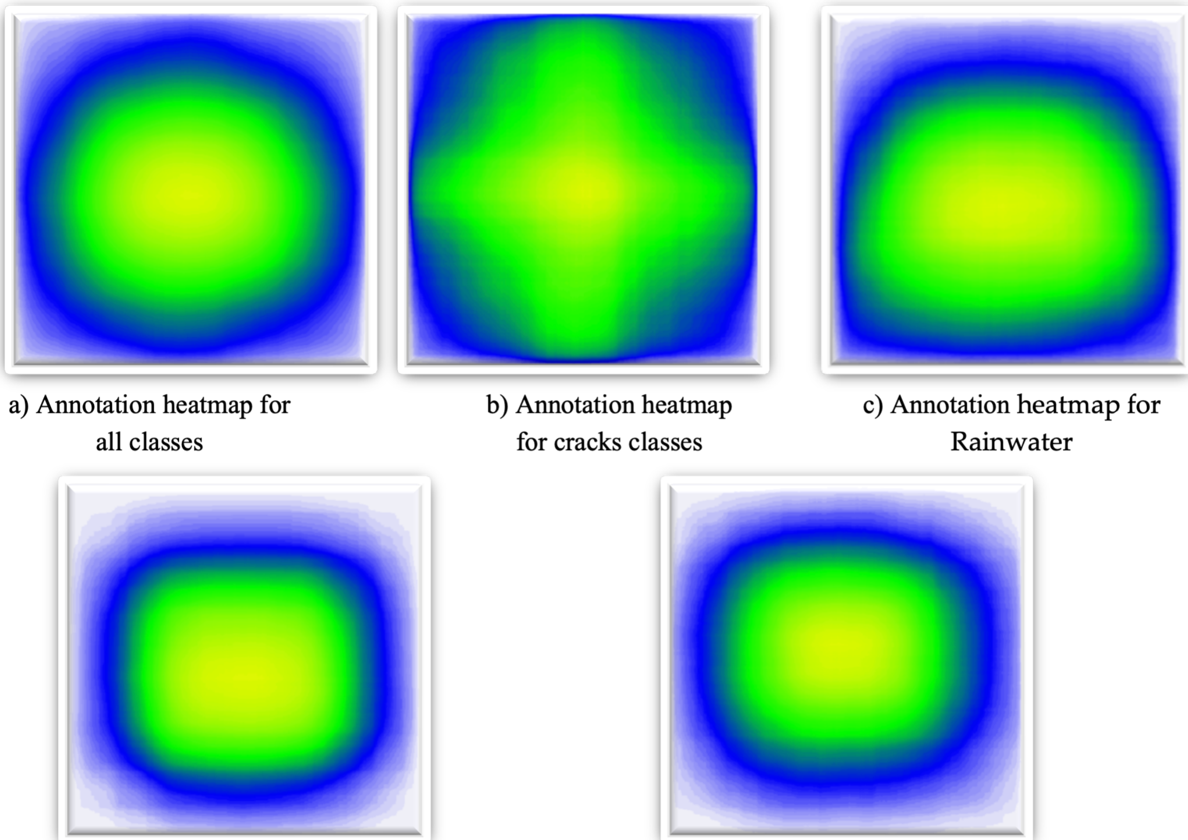
• Model Evaluation: The performance of each model was assessed at this stage to measure its efficiency in classifying road defects.

5. DATA COLLECTION AND PREPROCESSING

The datasets in this study were collected from two sources, Kaggle and Roboflow. Seven datasets with images of various road problems were combined [14–20], to create an integrated data set of over 5,000 images. TABLE 2 shows the dataset classes and distributions, while sample images are presented in FIGURE 1.

The dataset thus consisted of 5,389 images, with only two missing annotations, which ensured that almost all images had corresponding annotation files. A total of 9,260 annotations were distributed

Figure 1: Sample images



a) Annotation heatmap for all classes

b) Annotation heatmap for cracks classes

c) Annotation heatmap for Rainwater

d) Annotation heatmap for DrainCover classes

e) Annotation heatmap for pothole classes

Figure 2: Annotation density across classes

across the images, for an average of approximately 1.6 annotations per image. These annotations were thus classified across the four categories. The average image size in the dataset was 0.20 MP, with the smallest image being 0.02 MP and the largest being 16.15 MP. The average image was 512 x 458 pixels, with most images being roughly square while tending to landscape format.

The heatmap in FIGURE 2 visualizes both annotation density and distribution across all images in the dataset. Blue areas represent fewer annotations, suggesting that street damage is less frequent in this area. Green indicates a moderate number of annotations, with yellow indicating a higher number. Areas highlighted in red indicate the highest numbers of annotations.

A number of issues were highlighted in the initial exploration of the dataset, which needed to be addressed before developing the model. The first issue was class imbalance, with some classes containing much higher levels of data than others. Such an imbalance could lead to a model favoring common classes and making inaccurate predictions for less well-represented classes, which would reduce the performance and efficiency of the model. Additionally, image size inconsistencies, where

Table 3: Hyperparameter Selection Criteria for YOLOv11X

Hyperparameter	Description	Value
epochs	Number of training epochs	150
imgsz	Input image size (width × height) in pixels	640
batch	Batch size - number of samples per batch	16
lr0	Initial learning rate	0.0005

images vary in dimensions, can increase training costs and reduce accuracy, and these were prevalent in the data. Missing labels were also found for some images, causing them to lack classification, thereby limiting the model's ability to learn effectively. Lastly, several null entries were present, where either full data points were missing or those present lacked useful information. Again, this adversely impacts model training. It is essential to address these issues to achieve optimal model performance.

To address the concerns discussed above, necessary data processing was carried out. This involved cleaning the dataset and eliminating any null or missing values, whilst also resizing and standardizing images to a 640x640 format. Furthermore, any images that had not been annotated were manually annotated to identify objects. In addition, to improve model generalization and address dataset limitations, extensive data augmentation techniques were applied using Roboflow. The dataset was expanded from 5,389 to 20,652 images through applying horizontal and vertical flipping, 90° rotate (clockwise, counterclockwise, upside down), Cropping (0% minimum zoom, 20% maximum zoom), Random rotation ±15°, shear (10% horizontal and 10% vertical). These augmentations to simulate real-world variability such as lighting conditions, camera angles, and environmental noise, thereby improving model robustness.

6. MODEL DEVELOPMENT

Three models were developed in this research, and their accuracy was examined in terms of their ability to detect and classify road issues. The first model developed was YOLOv11X, an advanced version of the YOLO model, which is a high speed and accuracy model implementing deep learning approaches to detect defects like potholes. The second model was a more rapid R-CNN, an accurate model that can detect issues such as potholes and water accumulation by identifying and classifying problematic locations. The third was RetinaNet, a rapid model that focuses on detecting hard-to-spot issues such as potholes and cracks that is suitable for applications that require quick results with good accuracy.

Google Colab was used for Python coding and the Roboflow platform was used for building the selected models. TABLE 3 – TABLE 5 show the experimental setup and the hyperparameter values for each model. Different epochs values were applied in training the model by applying early stopping based on validation loss. This was to ensure that differences in performance are due to models' architecture rather than training duration.

Table 4: Hyperparameter Selection Criteria for Faster RCNN

Hyperparameter	Description	Value
num_classes	Number of classes detected by the model	4
iou_thresholds	IoU thresholds used for calculating the mAP metric	[0.5, 0.75]
optimizer	Type of optimizer used to update parameters during training	Adam
epochs	Number of training epochs	5
batch_size	Number of samples processed per batch	4
learning_rate (lr0)	Initial learning rate for the optimizer	0.0001
imgsz	Image size used during training	640

Table 5: Hyperparameter Selection Criteria for RetinaNet

Hyperparameter	Description	Value
num_classes	Number of classes detected by the model	4
iou_thresholds	IoU thresholds used for calculating the mAP metric	[0.5, 0.75]
optimizer	Type of optimizer used to update parameters during training	Adam
epochs	Number of training epochs	9
batch_size	Number of samples processed per batch	4
learning_rate (lr0)	Initial learning rate for the optimizer	0.0001
imgsz	Image size used during training	640

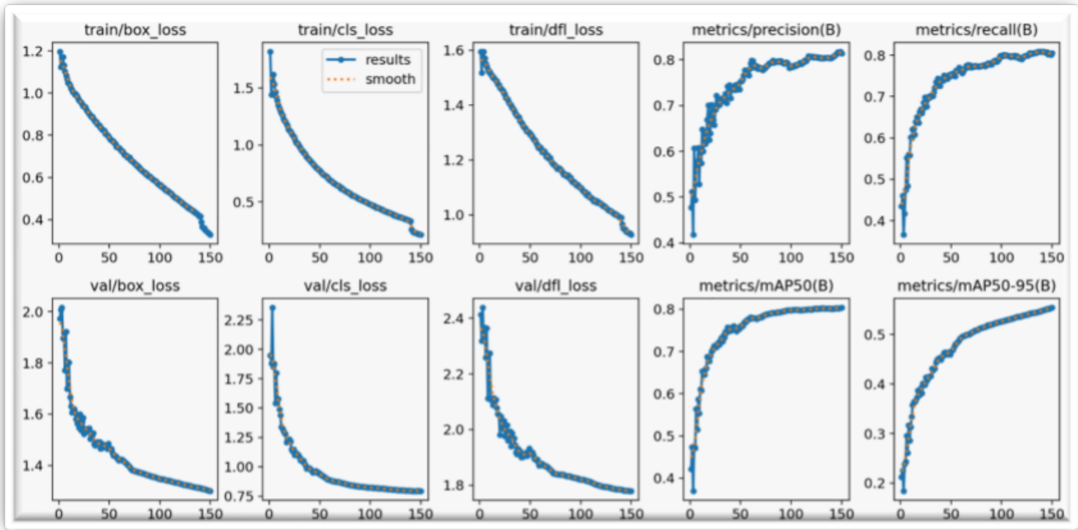


Figure 3: Training and Validation Losses with Performance Metrics Curves

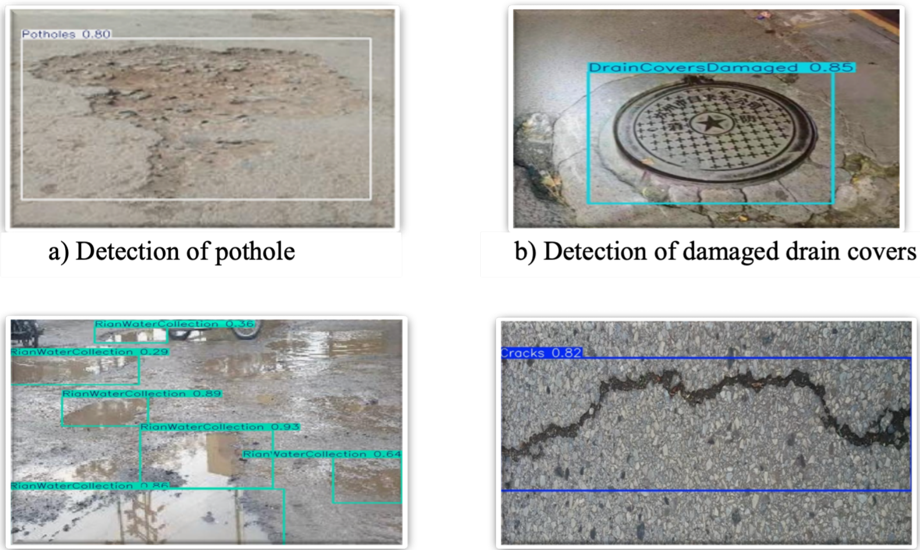
7. RESULTS

In this section, the results of object detection are presented for the selected models, YOLOvX11, Faster RCNN, and RetinaNet, alongside a performance evaluation for each. The initial data were divided in order to perform model evaluation, with 70% of the images available allocated to the training set, 20% to the validation set, and the remaining 10% reserved for the testing. The precision, recall, and MAP values were then used as evaluation metrics.

7.1 YOLOv11X

The settings for training YOLOv11X were conscientiously chosen to maintain efficiency and enhance performance. Moreover, to ensure that the model had enough learning time without causing overfitting, it was trained for 150 epochs, while the input image size was established at 640 × 640 pixels to create an equilibrium between speed and accuracy. It was decided that a batch size of 16 would be sufficient to facilitate efficient memory use and to ensure that all updates remained stable. Additionally, an initial learning rate of 0.0005 was established to ensure a steady learning progression whilst simultaneously minimizing the risk of divergence.

FIGURE 3 demonstrates that the loss measures during training (box, classification, DFL) gradually declined, and there is a pattern of clear convergence, suggesting that predictions and modeling have been improved. In terms of validation, there was a steady reduction in losses and no spikes, indicating effective generalization and minimal over-fitting. The performance measures also indicated significant progress over time, with accuracy reaching 0.82, recall at 0.794, and a mAP@50 of 0.804. In turn, this implies that the model performed very well across multiple domains. In FIGURE 4, several examples of validation results are presented, alongside their confidence scores, and per class results are given in TABLE 6.



a) Detection of pothole

b) Detection of damaged drain covers

c) Detection of water collection

d) Detection of cracks

Figure 4: YOLOv11x- Examples of validation results

Table 6: YOLOv11X per class results

Class	Precision	Recall	mAP@50
Potholes	0.88	0.86	0.85
Cracks	0.84	0.80	0.82
Drain Covers	0.81	0.77	0.79
Rainwater	0.79	0.73	0.75
Average	0.821	0.794	0.804

Table 7: Faster R-CNN per class results

Class	Precision	Recall	mAP@50
Potholes	0.93	0.90	0.70
Cracks	0.90	0.87	0.63
Drain Covers	0.87	0.83	0.56
Rainwater	0.82	0.76	0.51
Average	0.88	0.84	0.60

7.2 Faster R-CNN

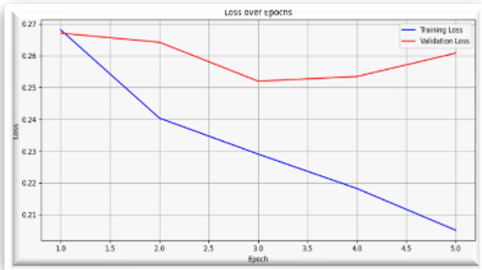
When training the Faster R-CNN model, a number of important hyperparameters were carefully chosen to enhance the model's performance to an optimal level. The model was configured so that it was able to detect five classes, with an Intersection over Union (IoU) threshold range of [0.5, 0.75] being implemented to compute the mean Average Precision (mAP). Furthermore, the Adam optimizer was used to update the model's parameters, and this optimizer was selected due to its well-known efficiency when training deep learning models. The model was trained for five epochs, with four samples being processed per batch, and an initial learning rate of 0.0001. In order to ensure accurate object detection and efficiency, the image size was set to 640 pixels throughout the training process.

In FIGURE 5, graphs showing training loss, precision, validation loss, recall, and mAP over five training epochs are presented. In (a), while the training loss (blue) steadily decreased, indicating effective learning, the validation loss (red) decreased initially and then increased after the third epoch, indicating potential overfitting. In (b), the training precision and recall (blue) steadily improved, however, demonstrating enhanced identification of true positives. In contrast, the validation precision and recall (red) fluctuated, with only slight improvement. In (c), the training mAP (blue) did show consistent improvement; however, the validation mAP (red) increased more gradually. By Epoch 5, the metrics were as follows: precision of 0.8810, recall of 0.8452, and mAP@.50 of 0.6032. FIGURE 6 shows some examples of validation results with their confidence scores, and per class results are given in TABLE 7.

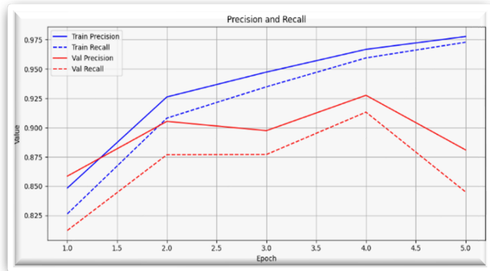
7.3 RetinaNet

During the training of the RetinaNet model, the main parameters were tuned to detect the five classes using the Adam optimizer, with an initial learning rate of 0.0001. The model was trained for nine epochs, with a batch size of four samples per batch. The IoU thresholds for calculating the mAP metric were set at 0.5 and 0.75. The image size used during training was set to 640 pixels, to ensure efficient object detection.

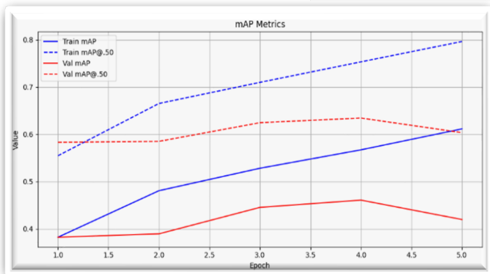
As shown in FIGURE 7, in (a) the training loss (blue) showed a steady decline, indicating successful learning, while the verification loss (red) initially decreased but later reached a plateau and increased slightly. In (c), the training accuracy and recall improved rapidly before stabilizing. The verification accuracy and recall followed a similar trend, though slight fluctuations after Epoch 3 indicated the possibility of overfitting in the future. In (b), the training mAP metrics (mAP and mAP@0.5)



a) Training and Validation Loss Curves

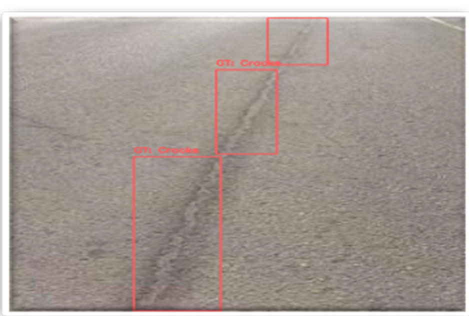


b) Precision and Recall Curves



c) Mean Average Precision (mAP) Metrics

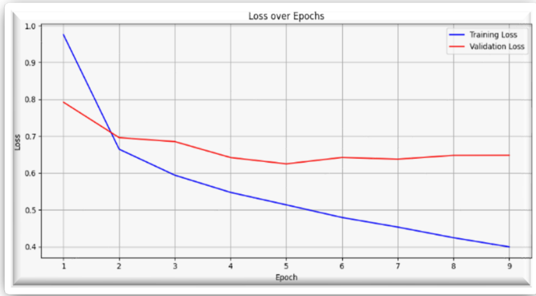
Figure 5: Faster R-CNN Training and Validation Metrics Curve



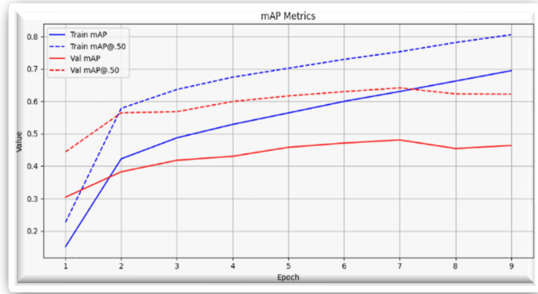
a) Detection of cracks

b) Detection of water collection

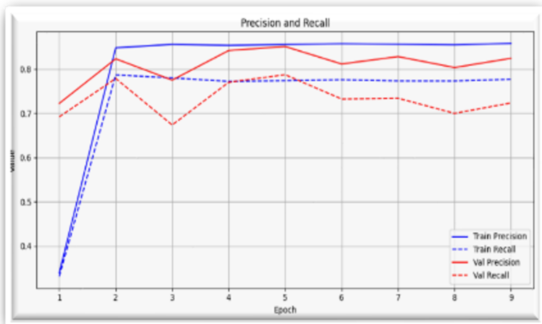
Figure 6: Faster RCNN-Examples of validation results



a) Training and Validation Loss Curves



b) Mean Average Precision (mAP)



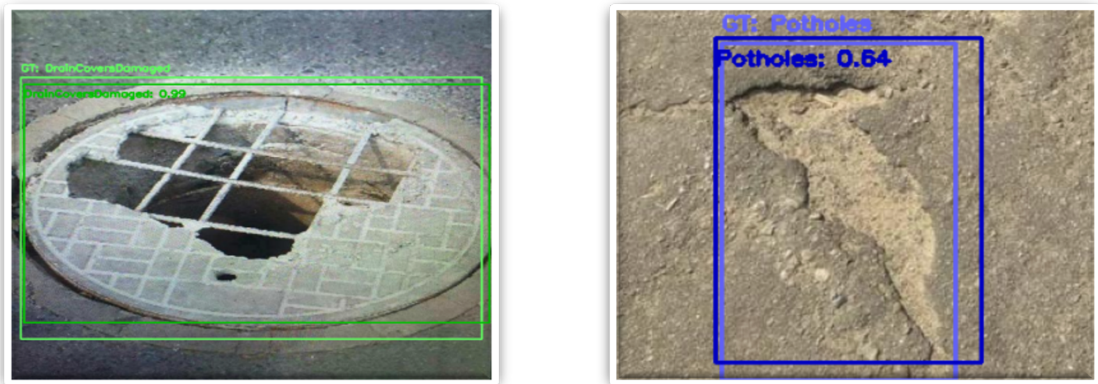
c) Precision and recall Curves

Figure 7: RetinaNet Training and Validation Metric curves

Table 8: RetinaNet per class results

Class	Precision	Recall	mAP@50
Potholes	0.93	0.90	0.70
Cracks	0.90	0.87	0.63
Drain Covers	0.87	0.83	0.56
Rainwater	0.82	0.76	0.51
Average	0.88	0.84	0.60

improved steadily; however, the verification mAP metrics reached a plateau or fluctuated after epoch 3, leaving a constant gap between the training and verification mAP. By Epoch 9, the metrics were as follows: precision of 0.8247, recall of 0.7236, and mAP@.50 of 0.6218. FIGURE 8 shows some examples of validation results with their confidence scores, and per class results are given in TABLE 8.



a) Detection of drain cover damage b) Detection of pothole

Figure 8: RetinaNet- examples of validation results

Table 9: Comparison of performance metrics across models

Model	Precision	Recall	mAP@50
YOLOv11X	82.1%	79.4%	80.4%
Faster RCNN	88.1%	84.5%	60.32%
RetinaNet	82.4%	72.3%	62.1%

8. DISCUSSION

Based on the results of the three models, distinct strengths and weaknesses for each model were noted; these can thus be used to aid the selection of the correct model for specific applications and needs. As shown in TABLE 9, YOLOv11X demonstrated a strong balance between precision (0.821), recall (0.794), and mAP@50 (0.804), making it ideal for consistent and adaptable object detection. Faster R-CNN exceled in terms of precision (0.8810) and recall (0.8452), minimizing false positives and negatives; however, its lower mAP@50 (0.6032) limit its adaptability, though its low validation loss (0.2608) ensures training stability. RetinaNet offered moderate performance, with lower precision (0.8247), recall (0.7236), and higher validation loss (0.6281), suggesting a need for optimization if this model were selected. Overall, YOLOv11X is the most balanced, Faster R-CNN suits high-precision tasks, and RetinaNet is best for simpler, resource-limited scenarios.

Moreover, despite the fact that this model reveals a satisfying precision result, it gives low mAP@50 results, especially for Faster RCNN and RetinaNet. This is usually due to the fact that the model makes confident predictions but fails to localize bounding boxes accurately. High precision suggests low false positives, but the reduced mAP reflects poor overlap Intersection over Union with ground truth annotations. Thus, more classified data may improve the performance and object detection for this model. Future work may look at expanding the data to include different types of cracks and potholes, and even variety of rainwater collection images to ensure different surface types, size and severity of issue, and weather conditions.

9. CONCLUSION AND SUGGESTIONS FOR FUTURE WORK

The study presented in this paper makes a significant contribution to the detection of on-street issues, thereby contributing to the development of safer, well-maintained civil infrastructure. This paper focused on developing a model capable of identifying common problems such as potholes, cracks, rainwater collection areas, and damaged drain covers in real time, which might thus be embedded in an edge-based system to deliver real-time reporting to authorized parties. The primary objective was to enhance road safety and maintenance efficiency, an aim achieved through the results of this study, which not only contribute to developing robust models for identifying multiple road problems simultaneously but also offer valuable insights into the effectiveness of such models in practical applications.

Future work related to this study might, however, usefully involve larger and more diverse datasets, including images taken under different lighting and weather conditions and at additional angles, in order to improve the models' generalization abilitie. Futhermore, the use of higher-resolution images and advanced methods (i.e., multi-scale feature extraction) may better equip the model to detect small and hard-to-detect objects like cracks. It is recommended that future researchers consider integrating the resulting detection models into mobile systems (i.e., drones or autonomous vehicles) to facilitate detection and reporting in real time.

10. ACKNOWLEDGMENTS

The researcher expresses their sincerest thanks to the following students undertaking a Bachelor's Degree in Data Science students for providing valuable contributions to this study: Razan Alsulami, Ghala Alotaibi, Juri Alghamdi, Majd Alshahrani, and Amal Mailbari. Their conscientious efforts to prepare, preprocess, and code data were critical in achieving the results of this study. The researcher reviewed and edited the output and assumes full responsibility for the content of this work.

References

[1] Shipu MK, Mamun FA, Razu SH, Nishat Sultana, M. TrafficNN: CNN-Based Road Traffic Conditions Classification. In: Ranganathan, G., Fernando, X., Shi, F., El Allioui, Y. (eds) Soft Computing for Security Applications. Advances in Intelligent Systems and Computing. Springer, Singapore. 2022;1397.

[2] Demirel Z, Nasraldeen ST, Pehlivan Ö, Shoman S, Albdairi M, et al. Comparative Evaluation of YOLO and Gemini AI Models for Road Damage Detection and Mapping. Future Transp. 2025;5:91.

[3] Megnidio-Tchoukouegno M, Adedeji JA. Machine Learning for Road Traffic Accident Improvement and Environ-Mental Resource Management in the Transportation Sector. Sustainability. 2023;15:2014.

[4] Zhou Y, Guo X, Hou F, Wu J. Review of Intelligent Road Defects Detection Technology. Sustainability. 2022;14:6306.

[5] Pandey AK, Iqbal R, Maniak T, Karyotis C, Akuma S, et al. Convolution Neural Networks for Pothole Detection of Critical Road Infrastructure. Comput Electr Eng. 2022;99:107725.

[6] Chun C, Ryu SK. Road Surface Damage Detection Using Fully Convolutional Neural Networks and Semi-Supervised Learning. Sensors. 2019;19:5501.

[7] Fan R, Bocus MJ, Zhu Y, Jiao J, Wang L, et al. Road Crack Detection Using Deep Convolutional Neural Network and Adaptive Thresholding. In2019 IEEE Intelligent Vehicles Symposium (IV). IEEE. 2019:474-479.

[8] Chitale PA, Kekre KY, Shenai HR, Karani R, Gala JP. Pothole Detection and Dimension Estimation System Using Deep Learning (YOLO) and Image Processing. In 2020 35th International Conference on Image and Vision Computing, New Zealand. IVCNZ. IEEE. 2020:1-6.

[9] Commandre B, En-Nejjary D, Pibre L, Chaumont M, Delenne C, et al. Manhole Cover Localiza-Tion in Aerial Images With a Deep Learning Approach. Int Arch Photogramm Remote Sens Spat Inf Sci. 2017:333-338.

[10] Yik YK, Alias NE, Yusof Y, Isaak S. A Real-Time Pothole Detection Based on Deep Learning Approach. J Phys: Conf Ser. 2021;1828:012001.

[11] Pang D, Guan Z, Luo T, Su W, Dou R. Real-Time Detection of Road Manhole Covers With a Deep Learning Model. Sci Rep. 2023;13:16479.

[12] Pham V, Nguyen D, Donan C. Road Damages Detection and Classification With YOLOv7. 2022. ArXiv Preprint: https://arxiv.org/pdf/2211.00091

[13] Sazara C, Cetin M, Iftekharuddin KM. Detecting Floodwater on Roadways From Image Data With Handcrafted Features and Deep Transfer Learning. 2019 IEEE Intelligent Transportation Systems Conference. ITSC. IEEE. 2019:804-809.

[14] https://www.kaggle.com/datasets/chitholian/annotated-potholes-dataset

[15] https://universe.roboflow.com/fumigation-robot/fumigation-robot

[16] https://universe.roboflow.com/manhole-pqaet/manhole-cover-detection-vpioj

[17] https://www.kaggle.com/datasets/rajdalsaniya/pothole-detection-dataset

[18] https://universe.roboflow.com/hanyang-university-bd2kb/puddle_night

[19] https://www.kaggle.com/datasets/rizwan123456789/road-crack-condition-images-pakistan

[20] https://universe.roboflow.com/jgate-90tes/texzt