DATA ANALYTICS AND APPLIED MATHEMATICS
E-ISSN: 2773-4854
VOLUME 6, ISSUE 2, 2025, 1 – 9
DOI: https://doi.org/10.15282/daam.v6i2.12679

RESEARCH ARTICLE

Potholes detection from dashcam using deep learning approach

Noryanti Muhammad[1,2], Wan Maisarah Wan Alias[1], Mohd Radhie Mohd Salleh[3] and Yuki Hayashida[4]

[1]Centre for Mathematical Sciences, Universiti Malaysia Pahang Al-Sultan Abdullah, Lebuh Persiaran Tun Khalil Yaakob, 26300 Kuantan, Pahang, Malaysia
[2]Centre for Artificial Intelligence & Data Science, Universiti Malaysia Pahang Al-Sultan Abdullah, Lebuh Persiaran Tun Khalil Yaakob, 26300 Kuantan, Pahang, Malaysia
[3]INet Spatial Sdn Bhd, 41A & 41B, Jalan Flora 1/9 Johor Bahru, 81300 Skudai, Johor, Malaysia
[4]Graduate School of Engineering, Division of Information Engineering, Mie University, Japan.

ABSTRACT - Deep learning has attracted considerable interest in the previous ten years and has established itself as a leading technology in the artificial intelligence sector. In object detection based on image processing, characteristics are extracted from images, and after that, data including category, position, and motion are collected and analysed. One of the practical applications of object detection is pothole detection, which plays an important role in improving road safety and maintenance. Due to the potential for crashes and injuries, potholes can be dangerous for cyclists, pedestrians, and moving automobiles. The discovery and repair of potholes may fundamentally change with the development of automated pothole detecting systems. Road repair is necessary to prevent such traffic accidents in the future. Therefore, to perform road maintenance, it is necessary to identify difficulties with things like potholes. The aims of this research is to identify number of potholes on asphalt or concrete roads from the images obtained for both primary and secondary data, to investigate the accuracy of the testing by using the deep learning approach which is Convolutional Neural Network (CNN) algorithm and to differentiate between normal road and potholes by evaluating them using confusion matrix which are “actual potholes”, “predicted potholes”, “actual normal” and “predicted normal”. Data collecting has been done around Universiti Teknologi Malaysia's road. Deep learning has been applied in this study is CNN, which was used as the model of this project. Results obtained are in confusion matrix which it concludes the actual number of potholes there. The actual potholes obtained were less than expected so this means more data needed if this project extended in the future. To improve and to achieve better performance of this project, data collecting can be done by adding other places so there will be more data and resolution of the images can be sharper so the results may be more accurate. To make our roadways reliable, efficient, and long-lasting, research on pothole detection is essential.

ARTICLE HISTORY
Received : 25th June 2025
Revised : 12th Sept 2025
Accepted : 17th Sept 2025
Published : 30th Sept 2025

KEYWORDS
Convolutional Neural Network (CNN)
Confusion matrix
Object detection
Potholes

1. INTRODUCTION

Potholes can be the big major causes in accident rates that happens today. As the World Health Organization (WHO) predicts that in 2030, fatal traffic accidents would rise to the fifth-highest cause of death [1]. According to Varona et al. [2], automatic monitoring systems such as potholes detection system or others are crucial to reducing accidents, enhancing traffic safety, and protecting vehicles from harm due to the expensive cost of maintaining an intact road surface. Even though is that, to reduce their role in unpleasant events, potholes should be repaired as soon as possible. If these potholes are filled in water during the rainy season, it might result in accidents that are also potentially lethal [3, 4]. In the past, certain methods for detecting potholes have been developed to deal with these problems, but they have a few drawbacks, including high implementation costs, a high number of hardware components needed, and poor speed and accuracy [5,6].

According to Ping et al. [7], a road surface structure breakdown known as a pothole, and this is too important to ignore because there is a possibility led to serious motor vehicle collision and affecting the efficiency of the road. A 2006 Asian Development Bank (ADB) study also revealed that approximately these paved roads are hardly half in good shape, which is a problem that almost all developed countries face. Numerous studies have been done in recent years to automatically locate potholes in the road. Support Vector Machine (SVM) had been started by Lin J et al. [8] to be used for pothole detection. The image's histogram was employed to extract the image region, then a basic kernel SVM was applied to identify the pothole. This technique made it easy to identify the pothole. In contrast, Maeda et al. [9] created a system to identify road damage in photos captured by mobile devices using Convolutional Neural Networks (CNN) techniques. To overcome the issue, they gathered a sizable dataset for pothole detection and used deep learning algorithms. The road damage detection system's performance in terms of accuracy and speed was positive.

***CORRESPONDING AUTHOR | N. Muhammad|** ✉ noryanti@umpsa.edu.my

Ping et al. [7] offer a solution designed to utilise artificial intelligence and machine learning techniques to build an accurate and effective pothole detection system. To determine which model generates the most accurate results, four deep learning models which are YOLO (You Only Look Once) algorithm [10], Single Shot Detector (SSD) algorithm, Histogram of Oriented Gradients (HOG) with Support Vector Machine (SVM), and Faster R-CNN are trained.

Yik et al. [11] stated that approximately 11.2% of the total fatal road crashes are related to a road defect, whereas 11.2% of all road accidents are related to the presence of potholes on the road, according to an analysis of Malaysian traffic deaths. This issue also includes the issue with paved roads in rural regions, which receive far less maintenance and consequently have a larger risk of developing potholes than roads in urban areas. As a result, 66% of the total road crash fatalities occurred in rural areas, compared to 34% of such fatalities in metropolitan areas. This situation illustrates how crucial road safety and health inspection are in preventing pothole problems. As we can see here, to avoid more road accidents in the future, there needs to be road maintenance. So, to do road maintenance, we need to detect the issues regarding road problems such as potholes. Detecting potholes and obtaining the coordinates of the road is the main purpose of this study.

2. LITERATURE REVIEW

Potholes are surface irregularities caused by the degradation of pavement materials and can become safety hazards, especially during the rainy season when they accumulate water [1]. Traditional detection methods, such as manual inspections or Ground Penetrating Radar (GPR), face limitations including high costs, limited access to rural roads, and dependence on user reports through apps like Waze or i-Tegur [10]. Given the vast number of road networks, continuous monitoring is impractical. Road surface distress generally falls into three categories: deformation (e.g., rutting, shoving), fractures (e.g., fatigue cracks), and disintegration (e.g., stripping, raveling) [11]. Additionally, the type of road surface such as asphalt, concrete, cobblestones, or dirt roads affects vehicle stability, making road-type recognition important in detection systems [2]. Therefore, a low cost, efficient, and automated system is necessary to monitor road conditions effectively.

Potholes typically form due to a combination of construction flaws, environmental conditions, and poor maintenance. Key causes include the use of low-quality materials, poor drainage design, and freeze thaw cycles, where water seeps into cracks, freezes, expands, and weakens the pavement [12,13]. Traffic load further accelerates deterioration, especially when existing cracks are present. Over time, general wear and tear, combined with environmental exposure such as sun, snow, and rainfall, can lead to the formation of potholes [14]. Improper construction practices like inadequate compaction and insufficient pavement thickness also contribute to early failure [15]. Addressing these causes requires proactive maintenance strategies including routine inspections, effective drainage systems, and proper crack sealing techniques.

Deep learning provides a scalable and accurate solution for automated pothole detection. Object detection methods are generally categorized into three groups: two-stage detectors, one-stage detectors, and advanced detectors [16,17,18,19]. Two-stage detectors such as R-FCN (Region-based Fully Convolutional Network) first propose candidate regions before classification, achieving high accuracy. One-stage detectors like RetinaNet perform detection in a single step, offering faster processing suitable for real-time applications. Advanced detectors, such as FoveaBox, enhance detection performance through refined feature extraction and optimization. These models allow efficient pothole detection from road imagery or video, reducing dependency on manual reporting and enabling faster maintenance response.

3. METHODOLOGY

This study focused on pothole detection whereas to differentiate between normal plain road and pothole. The training and testing dataset used is retrieved from Kaggle website (Pothole-Detection-Final-Project | Kaggle) from year 2021. There are two different categories of road which are normal road and potholes. Both categories are in images (.jpeg). As for the normal road, there are total of 352 images while potholes have 329 images. As for prediction, data was collected and provided by the company, I Net Spatial Sdn Bhd. Data collection was conducted around Universiti Teknologi Malaysia (UTM) Johor Bahru campus (Skudai, Johor) on 5th June 2023 by using a dashcam, so the data was in videos (.mp4). Then, the videos are exported to images by using VLC media player. The example of the model and train data from the UTM street is shown in Figure 1, where the videos has been split into many images. After doing the test accuracy of the model, evaluate them by using the confusion matrix. This study is applied CNN for the modelling part to identify the road issues which is a pothole.

Figure 1. Example of UTM street images

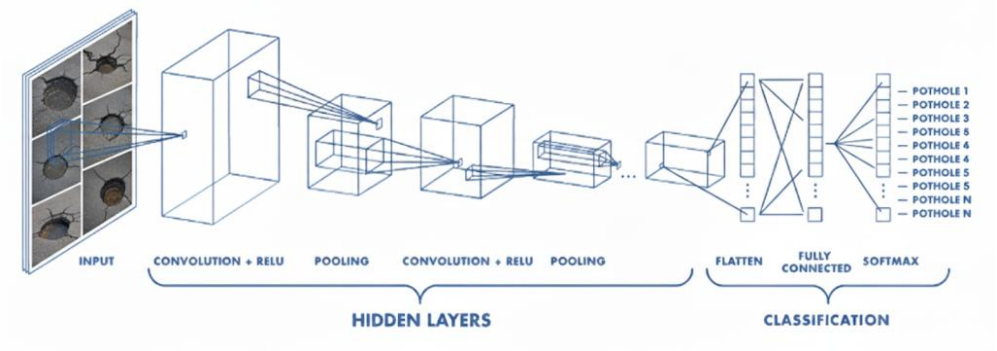

Figure 2. Architecture of CNN [21]

Figure 2 shows an architecture of CNN, which typically has three layers, a convolutional layer, a pooling layer, and a fully connected layer. The foundational component of the CNN is the convolution layer. It carries most of the computational load on the network. This layer conducts a dot product between two matrices, one of which is the kernel, a collection of learnable parameters and the other of which is the restricted portion of the receptive field. Compared to an image, the kernel is smaller in space but deeper. This indicates that the kernel height and width is spatially small if the image is made up of three (RGB) channels, but the depth is up to all three channels.

The kernel moves across the picture's height and width during the forward pass, creating an image representation of that receptive region. As a result, a two-dimensional representation of the image called an activation map is created, revealing the kernel's response at each location in the image. A stride is the name for the kernel's slidable size. By combining nearby pixels in a specific area of the image into a single representative value, the pooling layer minimizes the size of the image. Many other image processing systems have already been using the common technique known as pooling. It must be decided how to choose the pooling pixels from the image and how to set the representative value before we can carry out the activities in the pooling layer. The square matrix is typically used to choose the adjacent pixels, and the total number of pixels combined varies depending on the problem. The mean or maximum of the chosen pixels is typically used as the representative value. The pooling layer's operation is very straightforward. It is a two-dimensional system; a text explanation could make things even more unclear. In Figure 3, which represents the 4x4 pixel input image.

1	1	1	3
4	6	4	8
30	0	1	5
0	2	2	4

Figure 3. The four-by-four-pixel input image [21]

Figure 4 shows a 2x2 matrix has been created using the input image's pixels without overwriting any of the parts. The input image becomes the 2x2 pixel image after it has been processed by the pooling layer. The figure displays the resultant cases of pooling using mean pooling and maximum pooling.

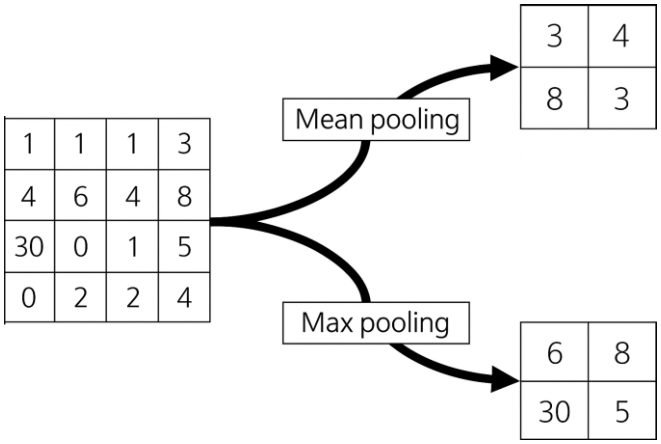

Figure 4. The resultant cases of pooling use two different methods [21]

As in Figure 4, the pooling procedure is a form of convolution operation in a mathematical sense. The convolution filter is fixed, and the convolution zones do not overlap, unlike the convolution layer. The pooling layer partially corrects for unsteady and tilted objects. The pooling layer, for instance, can enhance the recognition of a cat, even if it is off-center in the input image. Additionally, because the pooling procedure shrinks the size of the image, it is very helpful for reducing computing load and avoiding overfitting. Since convolution is a linear process while images are anything, but linear, non-linearity layers are frequently included right after the convolutional layer to add non-linearity to the activation map. Non-linear operations come in a variety of forms. The most common ones are Sigmoid, Tanh and ReLU. According to Mishra [21], the Equation (1) can be used to calculate padding layer where the size of the output volume has an activation map of size $W \times W \times D$, a pooling kernel of spatial size F, and stride S.

$$W_{out} = \frac{W - F}{S} + 1 \tag{1}$$

where W is weight and biases, F is the dimension of filter or filter size, and S is the number of strides. Other than that, the size of the output volume may be calculated convolution layer using the Equation (2), which an input of size $W \times W \times D$, whereby D is outnumber of kernels with a spatial dimension of F, stride S, and amount of padding P.

$$W_{out} = \frac{W - F + 2P}{S} + 1 \tag{2}$$

where W is weight and biases, F is the dimension of filter or filter size, S is the number of strides, and P is padding. Hence, an output volume of size $W_{out} \times W_{out} \times D_{out}$ has resulted from the equation.

In this study, both mean pooling and max pooling operations were illustrated (Figures 3 and 4) to demonstrate their effects on down sampling. However, for the actual CNN implementation, max pooling was selected as the pooling method. This choice was made because max pooling retains the most prominent features in an image, which is particularly useful for highlighting sharp edges and irregular patterns such as potholes. Mean pooling, while useful for smoothing, tends to dilute distinctive features, which may reduce detection accuracy in this context. Therefore, max pooling was applied consistently in all pooling layers of the proposed architecture.

According to Mishra [21], from the section "Motivation behind Convolution" stated that matrix multiplication is used in trivial neural network layers to describe the interaction between the input and output unit through a matrix of parameters. This implies that every input unit and output unit communicate with one another. Convolution neural networks (CNN), however, only interact sparsely. This is accomplished by making the kernel smaller than the input. For example, an image may include millions or thousands of pixels, but when it is processed using the kernel, we are able to identify significant information that is contained in tens or hundreds of pixels. This implies that we need to keep fewer parameters, which lowers the model's memory need and boosts the model's statistical effectiveness. CNN also has surpassed expectations and ascended to the throne as the most advanced computer vision method. CNN is without a doubt the most well-known of the several neural network types which include the others such as recurrent neural networks (RNN), long short-term memory (LSTM), artificial neural networks (ANN) and others. In the world of image data, these CNN models are widely used. On computer vision tasks like image classification, object identification, image recognition, and others, CNN perform remarkably well.

The evaluation method used in this paper is confusion matrix. There are four terms in the confusion matrix which are true positive (TP), true negative (TN), false positive (FP) and false negative (FN) which shown in Table 1. The scenarios consider in the confusion matrix between prediction and actual output are as follows, respectively.

i. True positives (TP): prediction YES, actual output YES.
ii. True negatives (TN): prediction NO, actual output NO.
iii. False positives (FP): prediction YES, actual output NO.
iv. False negatives (FN): prediction NO, actual output YES.

Table 1. Confusion matrix

		Prediction	
		Positives	Negatives
Actual	Positives	TP	FN
	Negatives	FP	TN

The confusion matrix is calculating the actual potholes and predicted potholes as the final output of this study. Equation (3) is accuracy formula used for the matrix.

$$\text{accuracy} = \frac{\text{TP+TN}}{\text{total sample}} \tag{3}$$

4. RESULTS AND DISCUSSION

In this section, the result of the paper is discussed. Figure 5 illustrates the Sequential CNN architecture implemented using the Keras deep learning framework [21, 23]. The model follows a conventional CNN structure comprising convolutional, pooling, dropout, and fully connected layers, adapted from established CNN design principles ([21],[24]).

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 124, 124, 32)      1432

 max_pooling2d (MaxPooling2D  (None, 68, 62, 62)       0
 )

 conv2d_1 (Conv2D)           (None, 60, 60, 64)        18496

 max_pooling2d_1 (MaxPooling  (None, 30, 26, 62)       0
 2D)

 conv2d_1 (Conv2D)           (None, 60, 28, 128)       78856

 max_pooling2d_1 (MaxPooling  (None, 14, 26, 128)      0
 2D)

 conv2d_1 (Conv2D)           (None, 60, 28, 128)       78856

 max_pooling2d_3 (MaxPooling  (None, 6, 6, 128)        0
 2D)

 conv2d_3 (Conv2D)           (None, 4, 4, 138)         147584

 max_pooling2d_4 (MaxPooling  (None, 2, 2, 128)        0
 2D)

 Flatten (Flatten)           (None, 512)               0

 dropout (Dropout)           (None, 512)               0

 dense (Dense)               (None, 119)               65664
 dense_3 (Dense)             (None, 2)                 0

=================================================================
Total params: 455,874
Trainable params: 455,874
Non-trainable params: 0
_________________________________________________________________
Noes
```

Figure 5. Linear stack of layers in Keras

The breakdown of the model architecture is as follows;

i. conv2d (Conv2D): This is the first convolutional layer with 32 filters and a kernel size of (3, 3). It takes input with shape (None, 124, 124, 3) and outputs a tensor with shape (None, 124, 124, 32). This layer has 2,432 parameters.

ii. max_pooling2d (MaxPooling2D): This is the first max pooling layer with a pool size of (2, 2). It takes the output of the previous layer and performs downsampling, resulting in a tensor with shape (None, 62, 62, 32).
iii. conv2d_1 (Conv2D): This is the second convolutional layer with 64 filters and a kernel size of (3, 3). It takes the output of the previous layer and outputs a tensor with shape (None, 60, 60, 64). This layer has 18,496 parameters.
iv. max_pooling2d_1 (MaxPooling2D): This is the second max pooling layer with a pool size of (2, 2). It performs downsampling on the output of the previous layer, resulting in a tensor with shape (None, 30, 30, 64).
v. conv2d_2 (Conv2D): This is the third convolutional layer with 128 filters and a kernel size of (3, 3). It takes the output of the previous layer and outputs a tensor with shape (None, 28, 28, 128). This layer has 73,856 parameters.
vi. max_pooling2d_2 (MaxPooling2D): This is the third max pooling layer with a pool size of (2, 2). It performs downsampling on the output of the previous layer, resulting in a tensor with shape (None, 14, 14, 128).
vii. conv2d_3 (Conv2D): This is the fourth convolutional layer with 128 filters and a kernel size of (3, 3). It takes the output of the previous layer and outputs a tensor with shape (None, 12, 12, 128). This layer has 147,584 parameters.
viii. max_pooling2d_3 (MaxPooling2D): This is the fourth max pooling layer with a pool size of (2, 2). It performs downsampling on the output of the previous layer, resulting in a tensor with shape (None, 6, 6, 128).
ix. conv2d_4 (Conv2D): This is the fifth convolutional layer with 128 filters and a kernel size of (3, 3). It takes the output of the previous layer and outputs a tensor with shape (None, 4, 4, 128). This layer has 147,584 parameters.
x. max_pooling2d_4 (MaxPooling2D): This is the fifth max pooling layer with a pool size of (2, 2). It performs downsampling on the output of the previous layer, resulting in a tensor with shape (None, 2, 2, 128).
xi. flatten (Flatten): This layer flattens the input tensor into a 1D tensor. It takes the output of the previous layer and reshapes it to (None, 512).
xii. dropout (Dropout): This layer applies dropout regularization to the input, randomly setting a fraction of input units to 0 at each update during training. It helps prevent overfitting. The output shape remains the same (None, 512).
xiii. dense (Dense): This is a fully connected layer with 128 units. It takes the output of the previous layer and produces a tensor with shape (None, 128). This layer has 65,664 parameters.
xiv. dense_1 (Dense): This is the final dense layer with 2 units, representing the output classes. It takes the output of the previous layer and produces a tensor with shape (None, 2). This layer has 258 parameters.

The total number of parameters in the model is 455,874, and all are trainable. Figure 6 shows the training and testing of the model. After the training and testing, the percentage of testing accuracy with (test_size) = 0.15 which equal to 15% and under epochs = 15.

```
In [11]: loss, accuracy = model.evaluate(x_test, y_test)

         print('Test accuracy: {:2.2f}%'.format(accuracy*100))

         4/4 [==============================] - 2s 277ms/step - loss: 0.3668 - accuracy: 0.8725
         Test accuracy: 87.25%
```

Figure 6. Testing accuracy

The performance of the model is evaluated using confusion matrix. This evaluation is used on testing data and prediction on data on site which is the data collected from UTM's street. Figure 7 shows a prediction on testing data set, where in total, the model correctly predicted 89 samples which are 44 samples of normal and 45 samples of potholes. There are 13 incorrect predictions which are 8 samples of potholes misclassified as normal and 5 samples of normal misclassified as potholes.

```
In [19]: #Create a Confusion Matrix for Evaluation
         # H = Horizontal
         # V = Vertical
         pd.DataFrame(confusion_matrix(y_test2, y_pred),
                      columns=["Predicted NORMAL", "Predicted POTHOLES"],
                      index=["Actual NORMAL", "Actual POTHOLES"])
```

Out[19]:

	Predicted NORMAL	Predicted POTHOLES
Actual NORMAL	44	8
Actual POTHOLES	5	45

Figure 7. Prediction on testing data set

While, in Figure 8 shows the model correctly predicted 203 samples which are 197 samples of normal and 6 samples of potholes. There are 220 incorrect predictions which are 218 samples of potholes misclassified as normal and 2 samples of normal misclassified as potholes.

```
In [27]: #Create a Confusion Matrix for Evaluation
         # H = Horizontal
         # V = Vertical
         pd.DataFrame(confusion_matrix(y2_test2, y2_pred),
                      columns=["Predicted NORMAL", "Predicted POTHOLES"],
                      index=["Actual NORMAL", "Actual POTHOLES"])
```

Out[27]:

	Predicted NORMAL	Predicted POTHOLES
Actual NORMAL	197	218
Actual POTHOLES	2	6

Figure 8. Prediction on testing data set (UTM's street)

The deep learning model developed using a sequential CNN architecture in Keras performed well during testing on a controlled dataset, achieving high accuracy with 89 correct predictions out of 102 samples. However, when evaluated on real-world data collected from UTM's streets, the model's performance declined, correctly identifying only 203 out of 423 samples. Notably, it misclassified 218 pothole images as normal, revealing a significant drop in accuracy. This highlights the model's sensitivity to differences between training and real-world data, indicating the need for further improvement in generalization and robustness.

To further evaluate the model's performance, additional metrics were calculated from the confusion matrix, including sensitivity, specificity, precision, and F1-score, which shown in Table 2. While the accuracy on the controlled dataset was relatively high, the field test on UTM's streets showed a considerable drop in performance, primarily due to the large number of false negatives (potholes classified as normal). This resulted in a low sensitivity or recall, indicating that many potholes were missed, while specificity remained relatively higher since most normal road samples were correctly identified. The precision and F1-score further reflected this imbalance, showing that although the model performed reasonably in detecting normal roads, it was less effective in identifying potholes in real world conditions. These findings suggest that the model generalizes poorly to unseen road data, emphasizing the need for a larger, more diverse dataset and additional fine-tuning to reduce false negatives and improve pothole detection reliability.

Table 2. Model performance

Dataset	Accuracy	Sensitivity (Recall)	Specificity	Precision	F1-Score
Controlled Dataset (Fig. 8)	0.873	0.849	0.898	0.900	0.874
UTM Streets (Fig. 9)	0.480	0.027	0.990	0.750	0.052

Based on Table 2, the additional performance metrics highlight a sharp contrast between controlled testing and real world data set. On the controlled dataset, the model achieved high sensitivity (0.849), specificity (0.898), precision (0.90), and F1-score (0.874), confirming balanced performance in detecting both normal roads and potholes. However, when applied to the UTM street data, sensitivity dropped drastically to 0.027, indicating that the model failed to detect the majority of potholes. Although specificity remained high at 0.990 and precision reached 0.750, the F1-score was only 0.052, reflecting severe imbalance in the classification outcomes. This demonstrates that while the model is reliable at identifying normal roads, it struggles to generalize to real-world pothole conditions, underscoring the need for a more diverse dataset, enhanced image quality, and domain-specific fine-tuning to reduce false negatives and improve robustness.

4. CONCLUSIONS

In conclusion, based on the results obtained, it seems that the data provided by the company is a bit unclear as the image of the pothole is not sharp. This has become a bit of low-quality data since the main object which is the potholes in the image not sharp. When it is low quality data, this may result not be the best one in terms of accuracy and performance. Other than that, the data is small size and cannot do more data analysis upon it. The evaluation of the modelling also seems a bit unsatisfied as the results too small. Hence, example can be taken from Figure 8 which there are small number of potholes in the UTM's data which is only 6 from hundreds of images of them. Hoping that there will be another location, and more locations can work on by deploying this project from another location too.

The recommendation that can be made throughout this project is high quality data is needed to acquire the best performance results. Other than that, use clearer videos so the images obtained can be clearer and sharper as well. By having clearer and sharper images, results that will be executed will achieve higher accuracy. In addition, if this project can be continued or be extended, hoping that there will be chances to do this project in another location so this project can be widely used also the coordinate of potholes been obtained so this idea can be proposed to government or any organizations that managed the road safety. The obtained coordinate then can be visualized into better visualization by using ArcMap and mark which location have potholes through the coordinates obtained.

ACKNOWLEDGEMENTS

The authors thank the Centre of Excellence for Artificial Intelligence & Data Science, Universiti Malaysia Pahang AL-Sultan Abdullah (UMPSA) for providing support and encouragement. The authors also would like to thank the reviewers for the valuable comments for improvements to this paper.

AUTHOR CONTRIBUTIONS

Noryanti Muhammad (Conceptualization; Supervision; Editing), Wan Maisarah Wan Alias (Data collection, Methodology; Formal analysis; Investigation; Visualization; Writing the original draft; Resources), Mohd Radhie Mohd Salleh (Data validation, Review and Editing), Yuki Hayashida (Review and Editing)

DECLARATION OF ORIGINALITY

The authors declare no conflict of interest to report regarding this study conducted.

GENERATIVE AI DECLARATIONS

The authors claim that artificially intelligent-assisted technologies in the form of generative AI were not used to generate content, ideas, or theories. We have just utilised AI to enhance readability and refine the language. This was used with extreme human control and oversight. The authors take full responsibility for reviewing and approving the content.

REFERENCES

[1] Holmes BD. Road traffic crashes in Federal Capital Territory, Nigeria: explanatory factors in statistical, geospatial, and qualitative analyses. The Medical College of Wisconsin. 2019.
[2] Varona J, Munoz-Bulnes J, Marcano A. Road surface type classification for smart vehicle navigation. Transportation Research Procedia. 2019;40:142–9.
[3] BH K, Kumari M, Kumari P. A review on pothole detection methods. International Journal of Research in Engineering, Science and Management. 2018;1(7):102–6.
[4] Senbagavalli M, Mahapatra S, Ruchita L. Revolutionizing road safety: an innovative approach to pothole detection and prevention using XAI and deep learning. In: 2025 4th International Conference on Sentiment Analysis and Deep Learning (ICSADL). IEEE; 2025. p. 878–84.
[5] Reddy ESTK, V R. Pothole detection using CNN and YOLO v7 algorithm. In: 2022 6th International Conference on Electronics, Communication and Aerospace Technology. IEEE; 2022.
[6] Reddy N, Varma V. Smart road maintenance: challenges and review of pothole detection methods. Journal of Civil Infrastructure. 2022;10(1):34–41.
[7] Ping P, Yang X, Gao Z. A deep learning approach for street pothole detection. In: International Conference on Big Data. IEEE; 2020.
[8] Lin TY, Dollár P, Girshick R, He K, Hariharan B, Belongie S. Feature pyramid networks for object detection. arXiv Preprint. 2016;arXiv:1612.03144.
[9] Maeda H, Sekimoto Y, Seto T, Kashiyama T, Omata H. Road damage detection and classification using deep neural networks with smartphone images. Computer-Aided Civil and Infrastructure Engineering. 2018;33(12):1127–41.
[10] Sawant S, Deshpande R, Karne K, Mehta K, Kasetwar A, Kasture A, Kasurde S. Pothole detection and road damage analysis using YOLOv9. In: International Joint Conference on Advances in Computational Intelligence. Singapore: Springer; 2025. p. 45–58.

[11] Yik W, Liew SY, Chong YW. A crowdsourced and automated approach for pothole monitoring in Malaysia. Malaysian Journal of Civil Engineering. 2021;33(1):59–72.
[12] Shaghouri M, Khodaii A, Yadollahi M. Pavement distress classification and its role in maintenance decision-making. Construction and Building Materials. 2021;276:122204.
[13] Ahmed KR. Smart pothole detection using deep learning based on dilated convolution. Sensors. 2021;21(24):8406.
[14] Ahmed A. Study on road pavement failures and pothole formation causes. International Journal of Pavement Research. 2021;6(2):45–52.
[15] Sharma R, Kanoungo R. Causes and prevention of potholes: a case study. International Journal of Advanced Research in Engineering and Management. 2015;1(4):21–5.
[16] Naveen N, Kumar V, Sinha R. Pothole detection using computer vision and machine learning. International Journal of Engineering and Technology. 2018;7(2.21):282–6.
[17] Mittal S, Bhatia A, Madan R. A survey on deep learning techniques for object detection. Archives of Computational Methods in Engineering. 2020;27(6):2089–115.
[18] Patawar A, Mehdi M, Kore B, Saval P. A pothole can be seen with two eyes: an ensemble approach to pothole detection. Machine Vision and Applications. 2025;36(3):1–20.
[19] Pawar S, Mahajan S, Pawar P, Khaire P, Dedgaonkar S, Shelke P. Road pothole detection and reporting system. In: 2025 International Conference on Emerging Smart Computing and Informatics (ESCI). IEEE; 2025. p. 1–6.
[20] Kandacharam S, Rajathilagam B, Vasudevan SK. Fusion framework for accident anticipation and incident detection in dashcam videos. SN Computer Science. 2025;6(5):1–17.
[21] Mishra M. Convolutional neural networks, explained. Medium. 2020 Sep 2.
[22] Ahmed MIB, Saraireh L, Rahman A, Al-Qarawi S, Mhran A, Al-Jalaoud J, Al-Mudaifer D, Al-Haidar F, AlKhulaifi D, Youldash M, et al. Personal protective equipment detection: A deep-learning-based sustainable approach. Sustainability. 2023;15(18):13990.
[23] Chollet F, Keras Team. Keras. GitHub. Retrieved from https://github.com/keras-team/keras, 2015.
[24] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521(7553):436–44.