Proceedings of the 3rd International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/6/20230948

A real-time automatic pothole detection system using convolution neural networks

Ricardo Bharat[1], Abiodun M Ikotun[1], Absalom E. Ezugwu[1,2,11], Laith Abualigah[3,4,5,6,7,8,11], Mohammad Shehab[9], and Raed Abu Zitar[10]

[1]School of Computer Science, University of KwaZulu-Natal, Pietermaritzburg Campus, Pietermaritzburg 3201, South Africa.
[2]Unit for Data Science and Computing, North-West University, 11 Hoffman Street, Potchefstroom, 2520, South Africa
[3]Computer Science Department, Prince Hussein Bin Abdullah Faculty for Information Technology, Al al-Bayt University, Mafraq 25113, Jordan.
[4]Hourani Center for Applied Scientific Research, Al-Ahliyya Amman University, Amman 19328, Jordan.
[5]Center for Engineering Application & Technology Solutions, Ho Chi Minh City Open University, Ho Chi Minh, Viet Nam
[6]Faculty of Information Technology, Middle East University, Amman 11831, Jordan.
[7]Applied science research center, Applied science private university, Amman 11931, Jordan.
[8]School of Computer Sciences, Universiti Sains Malaysia, Pulau Pinang 11800, Malaysia.
[9]Faculty of Computer Sciences and Informatics, Amman Arab University, Amman, 11953, Jordan
[10]Sorbonne Center of Artificial Intelligence, Sorbonne University-Abu Dhabi, Abu Dhabi, United Arab Emirates

[11]ezugwuabsalom79@gmail.com; Aligah.2020@gmail.com

Abstract. Detecting a pothole can help prevent damage to your vehicle and potentially prevent an accident. Different techniques, including machine learning, deep learning models, sensor methods, stereo vision, the internet of things (IoT), and black-box cameras, have already been applied to address the problem. However, studies have shown that machine learning and deep learning techniques successfully detect potholes. However, because most of these successful attempts are peculiar to the location of the study, we found no study which has addressed the peculiarity of potholes in South Africa using a tailored-trained deep learning model. In this study, we propose using a convolutional neural network (CNN), a type of deep learning model, to address this growing problem on South African roads. To achieve this, a CNN model was designed from scratch and trained with image samples obtained from the context of the study. The classifier was adapted to distinguish between a binary class which identifies the presence or absence of potholes. Results showed a significant performance enhancement at a classification accuracy of 92.72%. The outcome of this study showed that this machine learning approach holds great potential for addressing the challenge of potholes and road bumps in the region and abroad.

Keywords: Pothole, Machine Learning, Convolution Neural Network

1. Introduction

A pothole represents depression in a road surface built using asphalt, where broken pieces of the road have been removed by traffic. It is typically caused by underlying soil structure containing and traffic passing over the affected area depresses it. The presence of the first weakens the underlying soil it gets fatigued by the traffics, resulting in the breaking of the poorly supported asphalt surface in the affected area. The weakened asphalts in the affected area are ejected by the continuing traffic, creating a hole in the road [1].

Roads and highways are accessible to all residents in our country, and as a result, many vehicles, from as small as motorcycles to larger transportation trucks, regularly use them. This emphasizes that roads are essential for enabling transportation and communication. In South Africa, numerous potholes are formed due to the rainy weather, poor road infrastructure, and lack of maintenance culture. In a few areas, the road authorities respond swiftly and begin repairs, but in most cases, it takes many weeks and sometimes months to repair it. The standard or traditional road quality management methods involve looking for potholes manually s, recording their location, and returning to fix them once resources are available. These methods are time-consuming, expensive, and labour-intensive. It is not feasible in a country such as South Africa, where the pothole numbers are constantly increasing more than they are being fixed or repaired.

Other methods have been implemented for pothole detection. For example, Saurabh Pehere proposed a vision-based method [2]. Another method was proposed by Youngtae and Seungki [3], which uses laser scanning and vibration-based methods to detect potholes on the road. Other techniques for detecting potholes are presented in section 2 of this paper. All the previously researched methods have had limitations and still require improvements. These limitations include the methods requiring too much computation power to run on a mobile device and lower classification accuracies.

Machine learning(ML) techniques are regarded as the most successful in data-driven applications. Artificial intelligence is a branch premised on the idea that systems can learn from data, identify patterns after learning, and make decisions with limited human intervention. More impressive is the performance of CNN models in discovering and distinguishing patterns and features which support the classification process. Several benchmark CNN architectures like GoogleNet, AlexNet, LeNet, and ZFNet have been widely applied to feature extraction and classification challenges. Unfortunately, adapting these trained models to solve peculiar problems often requires fine-tuning the network for better adaptability.

In some cases, recent studies have proposed author-crafted CNN architectures to address the unique classification problems, and the results have confirmed that such models have performed well. This study proposes a new CNN architecture designed from scratch to tackle the difficult problem of classifying potholes in South Africa. Image samples were collected and pre-processed using resizing, data augmentation, and denoising methods. The emphasis is on using data augmentation techniques as a necessity to add up to the insufficient data and make up for the class imbalance in the original dataset. The dataset was user-sourced from several sources, including free stock photography websites, manually capturing images with a camera and pre-existing datasets.

The following are the technical contributions of the study:

a) Collate, curate, pre-process and build a relevant dataset of images of roads containing potholes and no potholes from different sources.

b) A CNN architecture was handcrafted from scratch and investigated to address the problem of pothole classification and detection.

c) The trained model was deployed on Android devices for improved usability and user satisfaction.

d) An evaluation of the performance of the proposed network was carried out, and a comparative analysis was made with similar methods.

This research paper is composed of 5 sections. Section 2 is the literature review justifying the importance of the study by reviewing and analyzing further related research. In section 3, the methods and techniques of how the problem was solved are discussed. In section 4, the results and the discussion describing the findings of the study are presented. Finally, in section 5, the conclusion and future work which describes how the work can be extended or improved, are discussed.

2. Literature review

Several pothole detection methods have already been researched and proposed; these methods' advantages and shortcomings are discussed in this section. In a study by Yu and Salari [4], the solution discussed is orientated around laser imaging techniques that collect road data. Then the data are analyzed to detect potholes using an artificial neural network (ANN), a machine learning model. This approach is not feasible for android devices because a substantial computation power is required to recognize images collected by the lasers. Lin and Liu's research [5] used a support vector machine algorithm to analyze road images for pothole detection. Although a high accuracy is credited to this approach, it requires extensive computational resources, making it unsuitable for android devices. In the research presented by [6], the authors employed a GIS system deployed on a smartphone that uses sensors to capture information about roads and then generate an automated report as output. Their method involved placing the smartphone inside a vehicle, and data was collected using the smartphone sensors. Using this data, the potholes are ranked based on their thickness and depth. The study by Maeda et al. [7] implements an approach for detecting road damage that uses computer vision-. The authors emphasized that no benchmark data set exists for road-damage detection; therefore, they constructed their dataset using a smartphone mounted on a dashboard. Object-detection algorithms are then used to predict the bounding boxes of the areas containing the road damageBounding boxes containing eight different categories of damages were predicted by the proposed model. In another paper by Danti, Kulkarni, and Hiremath [8], an image processing system is used as the primary classification /detection mechanism. The study aims to detect lanes, potholes and road signs. Potholes are recognized as distinctive black on the road, which is used as a characteristic for detecting them on the road. The algorithm did not produce desirable results, and as such, it requires a filtering method that is more effective to achieve better classifier accuracy. The work presented by Shaghouri, Alkhatib, and Berjaoui [9], proposed a method for pothole detection using deep learning algorithms. Using the YoloV4 architecture, the research was able to achieve the highest accuracy of 81%. In the research proposed by Pehere et al. [2], a vision-based technique was used for pothole detection. This method performed morphological operations on images to differentiate between normal pavement and pavement with potholes. This solution had accuracies ranging between 78% and 92%. In Hasan et al. [10], a method for the detection of speed bumps and potholes was proposed using a Region-based Convolutional Neural Network (RCNN). The researchers collected specific datasets to train the RCNN for the detection of unwanted obstacles on the road, reporting a training accuracy of 85%. Similarly, From the work of Aparna et al. [11], pothole detection was performed using a self-built CNN model and thermal images. An accuracy of 62.3% and 69.8% were obtained on the training and validation data, respectively, indicating poor detection accuracy. As the authors proposed in [12], their work revealed that datasets captured from a smartphone could be used to predict potholes. The identification of the potholes is achieved using k-Nearest Neighbour (KNN), Support Vector Machine (SVM), logistic regression, and Random Forest Tree models. The Random Forest tree and KNN recorded decent performance with an accuracy level of 89%. Furthermore, in the research proposed by Jo and Ryu [3], the vibration-based method with a black box camera over laser scanning was used for the detection of the pothole with an accuracy of 88%. It is important to note that most existing literature did not fully take advantage of machine learning. Most of the other techniques require too much computational power to be feasible to run on a mobile device. Another critical difference is that these models or techniques were not trained or evaluated on South African roads. Most of these studies concluded that more efficient methods would be needed for more accurate results.

DOI: 10.54254/2755-2721/6/20230948

3. Methodology

Android/Mobile smartphones have built-in sensors. Using these sensors, images and information will be captured and fed into the trained pothole classifier, which will then determine the state of the road by detecting any potholes observable at that current point in time and issue a warning of caution to the user of the application. The techniques used for accurate image classification involve data collection, pre-processing the data, defining a model, and training the defined model using the processed data.

3.1. Data collection

A dataset of 1248 images was compiled through the methods described below:

- Obtaining the relevant pothole/no pothole road images that are shared on Pixabay.com; This website contains free stock photography and royalty-free stock media, allowing the unrestricted use of the material. This was the main source of collecting images of roads without any potholes.
- Manually Capturing Images Using Mobile Phone Camera; These images were captured in Queensburgh, a town in Durban, South Africa.
- Using Images from 2 Pre-Existing Datasets [13, 14].

3.2. Data pre-processing

The dataset compiled contained hundreds of images that varied in size and, therefore, had to be resized to 3×250×250, which is also the expected input size of the CNN model to be trained. To have consistent-sized data and ease the image processing task, the photos were for training and testing of the model. Reducing the size of the images decreased both the processing time and the amount of memory needed for future processing. Due to the difficulty of obtaining a high-quality dataset, the next pre-processing step was implementing image augmentation techniques to the dataset [15]. These techniques include horizontally flipping and rotating images. These artificially created images from image augmentation helped increase accuracy, especially because the model training is done on a small dataset [16].

3.3. Proposed model

The chosen machine learning model (CNN) can recognize visual patterns in pixel images. Three layers were used to build the CNN architecture from scratch: a convolutional layer, a pooling layer, and a fully connected layer. The model is compiled using 11 of these layers, with the last layer having a sigmoid activation function that allows for classification between the two classes.

3.3.1. Model description. The first layer of the model consists of a convolution layer using a ReLu activation function. This layer is presented with a pre-processed image of the size 3×250×250. The convolutional filter size is 8x8, the number of filters is 16, the padding is 0, and the stride is 1. The ReLu activation is carried out on each feature map produced after the convolution operation. Layer 2 consists of a max pooling layer with an input of size 16@243x243. The pooling size is 2x2. On the completion of the max pooling operation, 16@121x121 size feature maps are obtained. Layer 3 forms the second convolutional layer that also uses the ReLu activation function. An output from the previous pooling layer serves this layer with an input of size 16@121x121. A total of 6 filters, with filter size 4*4, is used. This produces a feature map of size 16@118*118 after the convolution operation, and the ReLu activation is performed on it. Another max pooling layer with an input size of 16@118*118 forms the next layer as Layer 4 with a pooling size of 2*2, producing a feature map of 16@59*59 after the max pooling operation is obtained. Layer 5 forms the third convolutional layer that uses a ReLu activation function with an input size of 16@59*59 obtained from the preceding pooling layer. This layer has its number of filters as 32 with the filter size 8*8, producing feature maps of size 32@52*52 after the convolution operation, on which the ReLu activation is performed. The next layer is Layer 6, another max pooling layer with an input of size 32@52*52 and a pooling size of 2*2. Layer 6 produces a feature map of size 32@26*26 after the max pooling operation. Layer 7 forms the fourth convolutional layer with a ReLu activation function. It has 64 numbers of filters, a filter size of 4*4 and an input size of 32@26*26. It produces feature maps of size 64@23* after the convolution operation. Layer 8 is another max pooling layer with an input of 64@23*23 and a pooling size of 2*2. It produces feature maps of size 64@11*11 after the max pooling operation. Layer 9 is the fifth and final convolutional layer having an input of size 64@11*11. It has a pooling size of 2*2 and produces feature maps of size 32@26*26 after the max pooling operation. Its filter size is 4*4, and the number of filters used is 64 and produces feature maps of size 64@10*10 after the convolution operation. This layer, by being flattened, produces a one-dimensional vector of size 6400 and acts as a fully connected layer. Due to the reduction of the dimensions of the images, one more hidden layer (Layer 10) can now be added with 64 neurons. It can end the model architecture with an output layer (Layer 11) with two neurons for the two different classes. Layer 11 makes use of the SoftMax activation function. The CNN processes the image of the original road through the layers of the CNN to get to the final probability scores that determine which class it belongs to. Figure 1 summarizes and provides a visualization of the model architecture.

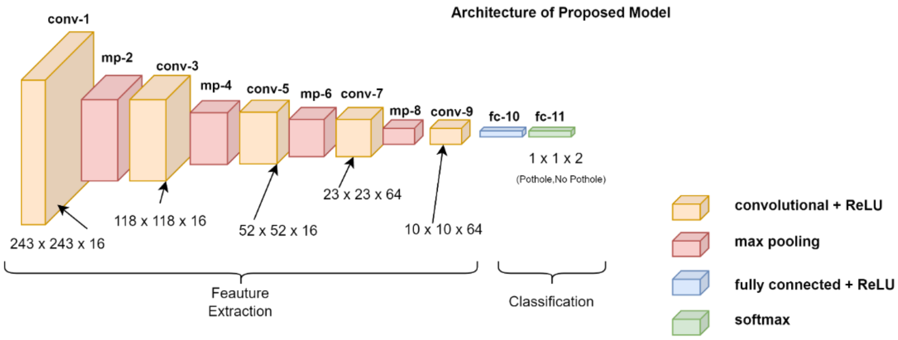

Figure 1. Summary of model architecture.

3.4. Compilation and training

Before continuing with the training of the CNN, the model had to be compiled. The Sparse Categorical Cross Entropy Loss Function to be used in training was first specified and including the Adam Optimizer to be used in training. The Adam optimizer is the optimization algorithm that updates the network weights for each iteration in the training data. Finally, the required metrics to evaluate the accuracy were used to track and monitor the training process and the constructed model is then trained using the training and testing data and their relevant labels. The number of epochs the model is to be trained for is also specified to be 25. After the model is trained successfully, it is saved and used in the android system's implementation. The accuracy results are discussed in Section 4.

3.5. Model implementation in android system

The system uses the OpenCV python library, which provides access to a live video feed from the android device's camera. Each frame from the live feed is captured, processed by resizing and normalization, and then fed into the classifier. The classification is displayed on the screen, and an audible warning is issued if a pothole is present.

3.6. Technical stack

There was a need for stronger computing power than provided by a standard CPU because image processing and training of the model was done on large-sized data. Thus, the image processing and training tasks were executed using a GPU instance provided by Google Collab. Google Collab allows a user to write and execute python code through a browser (Google Chrome was used). This platform conveniently provides the necessary modules for image processing and machine learning without additional installations. Kera's Python Deep Learning library (version 2.9.2) was used for building, training, and analyzing the performance of the models. This library allowed the sequential build of the model through stacking layers. PyCharm IDE was the environment used to develop the Android System, using the Open CV library to access a device's live camera video feed. This system was designed on a device with the following hardware specifications: an intel i5 processor and 4G of RAM.

DOI: 10.54254/2755-2721/6/20230948

4. Results and discussion

Figure 4 shows that the model's accuracy on the training data is 99.80%, and the validation accuracy is 92.72%. A 92.72 % test accuracy indicates a well-trained model sufficiently trained for classification for the pothole detection task. This accuracy can be improved dramatically by obtaining a larger dataset and using more relevant images of South African roads. However, we could not collect/use more images in the model training because of constraints of processing resources (Graphics Processing Unit (GPU) and RAM). Figure 2 shows the accuracy growth throughout the training. The val_accuracy line graph also plots the model's accuracy on the testing data throughout the training. The loss is also visualized in Figure 3. Knowing the loss allows us to see the variation between predicted and actual values. The satisfactory losses attained are 0.0007 for the training loss and 0.4450 for the validation loss. Figures 4 to 6 show that the model can correctly classify the class the image belongs to. The model also works well on images where the potholes are difficult to recognize, as shown by the classification made in Figure 6.

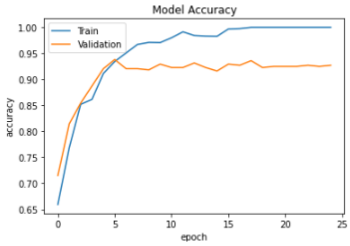

Figure 2. Accuracy vs epochs.

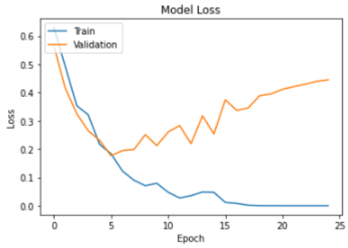

Figure 3. Loss vs epochs.

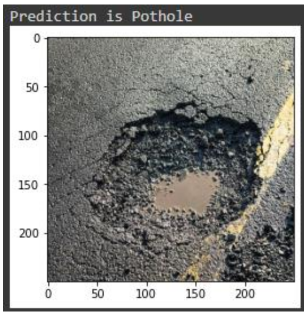

Figure 4. An example prediction.

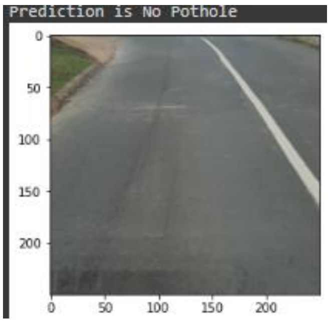

Figure 5. An example prediction.

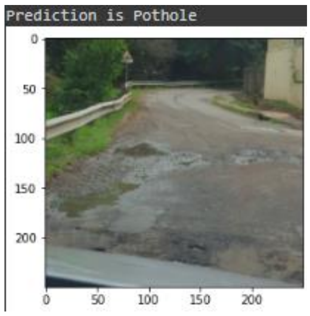

Figure 6. An example prediction.

A few other methods are listed along with their accuracies for comparison in Table 1. As can be seen from this comparison, our proposed deep learning method performed exceptionally well on pothole detection. It can also be seen that the machine learning methods produced much more excellent results.

Table 1. Comparison of results with other methods.

Authors	Method	Accuracy
Ricardo Bharat	**This Proposed Method (CNN)**	**92.72%**
Anas Al-Shaghouri [9]	Using YoloV4 architecture	81%
Saurabh Pehere [2]	Vision-based technique	78%
Zahid Hasan [10]	RNN	85%
Y.B Aparna [11]	Thermal Images and CNN	68%
Oche Alexander Egaji, Gareth Evans [11]	KNN and Random Forest tree	89%
JoYoungate [12]	Laser Scanning and Vibration-based methods	88%

5. Conclusion

This study proposed a real-time automatic pothole detection system using CNN. As seen from the model's results, the model performed exceptionally well in detecting potholes in a typical South African road. Interestingly, the proposed model achieved a 92.72% accuracy when validated on the testing data. The experimental results also revealed that the accuracy of the implemented model increases by increasing the number of convolution layers. Moreover, the accuracy and relevancy of the classifications also improved by increasing the quantity and quality of the data used in training and other image augmentation techniques. This study has proven to be a valuable tool in traversing the dangerous, pothole-plagued roads of South Africa. Being able to detect a pothole ahead of time will save South African road users from damage and other risks that may follow. A possible extension to this project will be to capture the location of the potholes detected using GPS coordinates and make it publicly available on the application. This will help road maintenance workers easily identify and eliminate the threats with ease and will also aid South African road users to choose a safer route. Alternate image augmentation techniques can also be explored in any future work. Investing more time and resources into this project can gradually eliminate the danger of driving on South African roads.

References

[1] Sathishkumar, V. E., Park, J., & Cho, Y. (2020). Using data mining techniques for bike sharing demand prediction in metropolitan city. Computer Communications, 153, 353-366.
[2] S. Pehere, P. Sanganwar, S. Pawar and A. Shinde. 2020. Detection of pothole by image processing using UAV. Journal of Science and Technology.
[3] J. Youngtae, R. Seungki. 2015. Pothole Detection System Using a Black-box Camera. Sensors. Basel, Switzerland, Nov. 2015, https://doi.org/10.3390/ s151129316
[4] X. Yu and E. Salari. 2011. Pavement pothole detection and severity measurement using laser imaging. IEEE International Conference on Electro/Information Technology. Mankato, Minn, USA, May 2011.
[5] J. Lin and Y. Liu. 2010. Potholes detection based on SVM in the pavement distress image. 9th International Symposium on Distributed Computing and Applications to Business, Engineering, and Science). pp. 544–547, Hong Kong, August 2010.
[6] D. Avellaneda, J. Lopez-Parra. 2016. Detection and localization of potholes in roadways using smartphones. DYNA 83.
[7] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, H. Omata. 2018. Road damage detection and classification using deep neural networks with smartphone images. Computer-Aided Civil and Infrastructure Engineering.
[8] A. Danti, J. Kulkarni and P. Hiremath. 2012. An Image Processing Approach to Detect Lanes, Potholes and Recognize Road Signs in Indian Roads. International Journal of Modeling and Optimization.
[9] A. A. Shaghouri, R. Alkhatib and S. Berjaoui. 2021. Real-time pothole detection using deep learning. arXiv preprint arXiv:2107.06356.
[10] Z. Hasan, S. N. Shampa, T. R. Shahidi, S. Siddique. 2020. Pothole and speed breaker detection using smartphone cameras and convolutional neural networks.2020 IEEE Region 10 Symposium (TENSYMP), pp 279–282. https://doi.org/10.1109/TEN SYMP50017.2020.9230587
[11] Y. B. Aparna, R. Rachna, G. Varun, A. Naveen, A. Aparna. 2019. Convolutional neural networks-based potholes detection using thermal imaging. J King Saud Univ Comput Inf Sci. https://doi.org/10.1016/j.jksuci.2019.02.004
[12] O. A. Egaji, G. Evans, M. G. Griffiths, G. Islas. 2021. Real-time machine learning-based approach for pothole detection. Expert Systems with Applications.
[13] Pothole Detection Dataset, A. Kumar. Retrieved on September 10, 2022, from: https://www.kaggle.com/datasets/atulyakumar98/pothole-detection-dataset

DOI: 10.54254/2755-2721/6/20230948

[14] Pothole 600, Google Sites. Retrieved on September,10, 2022 from: https://sites.google.com/view/pothole-600/dataset
[15] VE, S., Park, J., & Cho, Y. (2020). Seoul bike trip duration prediction using data mining techniques. IET Intelligent Transport Systems, 14(11), 1465-1474.
[16] VE, S., & Cho, Y. (2020). A rule-based model for Seoul Bike sharing demand prediction using weather data. European Journal of Remote Sensing, 53(sup1), 166-183.