ITM Web of Conferences **74**, 03001 (2025) https://doi.org/10.1051/itmconf/20257403001
ICCPCI-2024

Real-Time Road Surface Analysis for Pothole Detection

Ganjikunta Ganesh Kumar[1*], *Sibghatullah Inyatullah Khan*[1], *Mohammed Mahaboob Basha*[1], *Mallikarjun Mudda*[1], *Leela Prasad Yarrapathni*[2]

[1]Department of Electronics and Communication Engineering, Sreenidhi Institute of Science and Technology, Hyderabad, India
[2]Department of EEE, Shiv Nadar University, Greater Noida UP, India

Abstract. This paper presents a potentially low cost approach to detecting potholes using segmentation, by pairing a low cost computing platform (the Raspberry Pi) with a camera module and OpenCV image processing software. By getting real-time images processed with computational resources of Raspberry Pi and OpenCV image analysis functions, our system makes the search in real-time for potholes. The small form factor and flexibility of the Raspberry Pi makes it among the perfect platforms to interface with camera modules to capture high-resolution images of road surfaces. OpenCV provides an elaborate tool kit to enable image pre-processing, feature extraction, classification and precise crack detection. According to testing, this system has shown the ability to accurately detect potholes in a variety of environmental conditions and on different road types, making this solution cost-effective and scalable for transportation departments and cities looking to improve road safety and speed up pothole repairs.

1 Introduction

Road infrastructure is an integral component of transportation systems globally [1],[2]. It undergirds economic activity, and the social fabric of society. Yet roads deteriorating due to factors such as wear and tear, heavy traffic, and insufficient maintenance lead to massive issues. Potholes are among the top issues that impact road safety and efficiency. Conventional methods of pothole identification and fix are usually ineffective and sluggish.

This is an urgent problem that requires new, creative solutions. To fulfill this necessity, this study presents a novel solution that combines the Raspberry Pi, an inexpensive computing platform, with a camera module and the image processing capabilities of the OpenCV library. Our system can thus identify potholes from a vast amount of images, in real-time, as it combines the powerful computation capability of Raspberry Pi, with the

* Corresponding author: ganesh.g@sreenidhi.edu.in

capabilities of OpenCV. The compactness and versatility of the Raspberry Pi allows the camera module to interface with it seamlessly. This enables high-resolution images of road sur- faces to be captured in different environments. OpenCV is a comprehensive library of tools to prepare images, extract images features, and classify them. This helps in identifying potholes even when road conditions are dynamic.

The aim of this paper is to devise a novel approach for detecting potholes utilizing Raspberry Pi and OpenCV. The system, utilizing picture analysis, can identify potholes in real time with the aid of Raspberry Pi's computing power and OpenCV's features. This study is aimed at proving the effectiveness of this method in detecting potholes in different weather and road surface conditions.

Potholes provide a significant traversing problem to transportation companies and drivers all over the world; therefore numerous academic research papers have investigated innovative methods for identifying and monitoring potholes distinctly to enhance road safety while effectively maintaining the road infrastructure. One of the earliest researchers in employing accelerometric data for the detection of speed bumps is given by Celaya-Padilla et al. [3]. Although not directly related to pothole detection, this research illustrates how sensor-based techniques can be used in road surface analysis.

Numerous studies in pothole detection are available that involve low-cost computers, such as Raspberry Pis, and image processing. The authors in [4] used a system based on a Raspberry Pi for image processing capabilities to detect and report potholes in real time. This system uses image recognition techniques for detection purposes. This approach makes use of Raspberry Pi computing power to process road images and identify potholes at the site. Ajitha et al. [5] proposed the use of Internet of Things devices such as sensors to "smartly sense" potholes or any other potential road hazards in real time. Finally, this work demonstrates how inter-connected sensors can be used to monitor road conditions and detect hazards in real time. Although most of the researches focused on Raspberry Pi-based solutions for pothole detection purposes, researchers also explored smartphone solutions as a possible pothole detection mechanism. Jo and Ryu [6] proposed a video camera installed in an Android smartphone for the detection of potholes.

Automatic pothole detection systems pose many problems and hold a lot of possibilities [7]. More research is required to deal with the problems like scaling, stability, and flexibility towards different road conditions and environments. Current techniques hold good in terms of accuracy and time. Even more recent technologies such as machine learning and artificial intelligence could be incorporated into pothole detection systems to improve them further. This would be possible with more advanced research and with more proactive repair plans. The literature review shows the wide range of approaches and technologies that had been used in creating automatic pothole recognition systems. Researchers have looked into a lot of different ways.

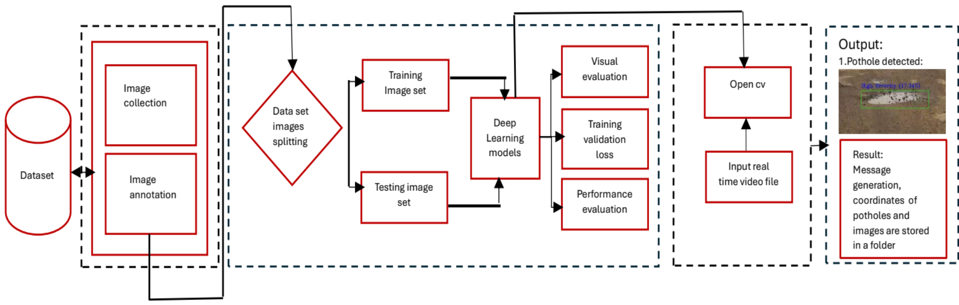

Fig. 1. Real-time pothole detection blocks

2 Proposed Methodology

The block diagram of the proposed methodology of real-time pothole detection system is shown in figure 1. In this system, OpenCV method has been used for image processing, and a very cheap computer tool known as Raspberry Pi. To make use of high-resolution images of the road surfaces, a Raspberry Pi and camera module are implemented in the system. Using the computing power of the Raspberry Pi and the lesion detection techniques available in OpenCV, image analysis is used to detect potholes in real time. You will use OpenCV that makes life easier for the preparation of the images and extraction of features and classification where you will be able to detect gaps with higher accuracy.

Raspberry Pi's small size and flexibility make it easy to connect to the camera module, which ensures that data is captured quickly. Experimental evaluation shows that the system works well in a range of weather and road situations. This method gives cities and transportation agencies an easy, scalable, and low-cost way to improve road safety and speed up the pothole fixing process. The suggested system helps smart transportation systems get better. It could be used in planning cities and keeping infrastructure in good shape, which are both big problems in today's transportation management.

2.1 Regulated Power Supply

The Raspberry Pi and Camera parts of the system get their power from the Regulated Power Supply, which stands for "power supply unit." It keeps these devices running smoothly by giving them the right amount of power and energy. The Regulated Power Supply is very important for keeping things working all the time, especially in remote or outdoor areas where regular power sources may not be easy to reach. In this project, the Regulated Power Supply makes sure that the Raspberry Pi and Camera always have power, so they can be used for real-time pothole spotting and tracking without having to be plugged into the wall.

2.2 Rasberry Pi

The raspberry pi is where all the computer work for the pothole sensing system is done. It runs the software routines that are needed and works with the data that the Camera module sends. The Raspberry Pi makes collaboration and operation in the field easier because it is cheap, small, and can do many things. In this project, the Raspberry Pi is in charge of handling the camera, using OpenCV to process images, and running the real-time methods for finding potholes.

2.3 Camera

The Camera gadget snaps high-resolution pictures of the road surfaces so that further analysis and processing take place. It connects to the Raspberry Pi and sends picture data so that it can be further processed. Camera is an important part of the project because it lets the system take pictures in real time of road conditions, such as potholes, so they can later be found.

2.4 OpenCV

Open Source Computer Vision Library or OpenCV is a great free computer vision and image processing software platform. It is awesome at setting up images, feature extraction, object finding, many more. OpenCV is used in this project to check pictures that camera clicks and detects the possible positions of potholes as per the defined features and patterns. With its powerful suite of tools and algorithms, it has become a necessity for monitoring and tracking potholes in real time. The block layout can be divided into four main components: Raspberry Pi, Camera, OpenCV, and Real-time Pothole Detection System (RPDS). Here, we use the Raspberry Pi to control all the functions as the main computer. It links to the Camera section and allows you to shoot high-definition images of the road. It is followed by OpenCV, which is a powerful library for computer vision and image processing tasks to analyze the captured images. With this, OpenCV does crucial functions such as picture preparation, feature extraction, and classification, enabling low latency in the detection of gaps.

The RPDS module takes the data to be received and applies algorithm on data to detect potholes based on parameters already set. Despite all this, if something is discovered, the system alerts or notifies the appropriate individuals so that they can respond immediately. Automatic pothole recognition and tracking utilizing the Raspberry Pi and Camera with OpenCV and RPDS software with its block diagram functions very smoothly to improve road safety and infrastructure upkeep.

2.5 Twilio API

The Twilio API provides an easy, yet powerful way to send SMS programmatically, which makes it the best choice for sending pothole detection alerts. In order to use the Twilio API for this, you have to register at Twilio and get your personal Twilio phone number, your Account SID and Auth Token. Using these credentials, you can now programmatically send an SMS containing the coordinates of the detected pothole to alert the appropriate authority. The integration allows alerts to automatically get sent out in real time improving communication around pothole repairs and maintenance.

3 Pothole Detection Working Process

3.1 Hardware Setup

Raspberry Pi is a cheap computer about the size of a credit card that has enough power to handle pictures. Camera Module: Connect a camera module that works with the Raspberry Pi board. This camera will take pictures of the surface of the road.

3.2 Software Setup

System of operation: Put on the Raspberry Pi a light Linux-based operating system, like Raspbian. OpenCV Installation: Put the open source computer recognition software OpenCV on the Raspberry Pi. This tool will be used to work with images.

3.3 Image Acquisition

As the car moves, the camera element keeps taking pictures of the road surface. The Raspberry Pi is given the images so that it can do more work on them.

3.4 Image Preprocessing

To make the next steps easier, change the taken photos to grayscale. Use noise reduction methods, like Gaussian blur, to improve the quality of the picture and lower the amount of distortion.

3.5 Feature Extraction

To find edges in the pictures, use edge detection methods (for example, Canny edge detector). Potholes usually show up as breaks in the smooth sides of the road. Get important details like size, shape, and changes in depth that show the presence of potholes.

3.6 Pothole Detection

Use a classification method (like a Support Vector Machine or a Convolutional Neural Network) to tell the difference between potholes and other problems with the road surface. Use a named collection of images with marked pothole sites to train the classification model.

3.7 Real-Time Detection

Keep processing receiving images in real time to find potholes as the car drives along the road. Show the found potholes on a graphical user interface (GUI) or send the data to a central computer so that it can be used in a different way [8].

3.8 Validation and Testing

Test the system thoroughly in a range of environments, such as those with different lighting, weather, and road surfaces, to see how reliable and accurate it is. You can measure how well the system works by looking at its sensitivity, precision, and false positive rate by comparing the found potholes to real-world data.

3.9 Optimization and Deployment

Take into account the Raspberry Pi's hardware limitations as you optimize the system for speed and efficiency. Put the software in a form that can be deployed and used in cars or on the side of the road.

3.10 Evaluation and Integration

Examine how well the method for finding potholes works in real life situations. Work with cities, transportation agencies, or private businesses to add the system to the way they already maintain infrastructure.

4. Experimentation and Results

4.1 Darknet Framework

We implemented the YOLO Tinyv4 algorithm [9] within our custom darknet framework, leveraging a Python-based open-source neural network library. This setup utilizes CUDA and C for accelerated processing, enabling efficient real-time object detection.

4.2 Dataset Annotation

Pothole detection models rely heavily on their training dataset. To create realistic pothole images, this study utilized one of the latest publicly available datasets [10] that capture real world situations like shadows, moving vehicles and changing illumination conditions that mimic real pothole environments as closely as possible. Some samples from the pothole image dataset are shown in Figure 2. Following the collection of the dataset [10], a critical phase involved the manual annotation of the gathered images. For this purpose, we employed LabelImg, a sophisticated graphical tool adept at generating annotations in the YOLOv4tiny darknet format.

To facilitate the training of the YOLOv4tiny model, it is imperative that annotations adhere to the specific YOLO format: <object><class><x><y><width><height>. In the context of our dataset, comprising a singular class, "pothole," the object class is designated as 0. Subsequent parameters delineate the precise coordinates, height, and width of the bounding box encompassing the annotated object. One example of annotating boundary boxes is shown in Figure 3. Each bounding box was precisely aligned to encapsulate the pothole, ensuring accurate and consistent annotations across the dataset [10]. This standardized annotation approach is essential for the effective training of the YOLOv4 Tiny model, enabling it to accurately detect and localize potholes in real-time.

4.3 Experimentation Protocol

The training of the YOLO and its variants is done on a system raspberry pi version: 3b+ with 32gb storage and 4gb ram. The dataset is split into 80% training of the model and 20% for the testing with labels of each image. The files needed for training are obj. names (class names), obj. data is the number of classes, then a path for train, test, and lastly a back up folder which saves weights.

Fig. 2. Example Images of Potholes from the Dataset

Fig. 3. Annotating Boundary Boxes with Labelimg

4.4 Generating Message

The system used the Twilio facilities to allow immediate notifications on the detection of a pothole. Thereafter, we configured the Twilio API account SID and identification token in the client module. This would generate messages automatically and send them to a particular number that would get instantaneous notifications of detected potholes.

In the system, the Twilio API client interaction interface included the Client module of Twilio, which enabled automated SMS notice generation and sending to provide information about determined potholes to a responsible recipient without any delay and with maximum effectiveness.

(a) Showing low level severity of pothole (b) Showing medium level severity of pothole

(c) Showing high level severity of pothole (d) Showing low level severity of pothole

Fig. 3. Severity levels of potholes

4.5 Performance Criteria

The model is able to balance high speed and accuracy which makes it perfect for real-time applications. We configure the DNN backend and target to CUDA, allowing us to utilize the GPU for faster inference by calling the setPreferableBackend and setPreferableTarget methods. The frame is then prepared as input to the model by resizing it to the shape of this (640x80) frame and normalizing the pixel values from ranges of 0-255 to 0-1 using setInputParams. The model then performs forward pass inference on each frame of the input video to detect potholes.

The detected objects bounding boxes, confidence scores, and class labels are extracted from the model's output. These are then filtered based on a confidence threshold (The Conf threshold is set to 0.5, determining the minimum confidence score required for an object to be considered a pothole detection.) and non-maximum suppression threshold (The NMS threshold is configured at 0.4, controlling the overlap threshold for non-maximum suppression to eliminate redundant detections.) to ensure accurate and reliable detection of potholes in the video frames. For each detected pothole, the system calculates the severity by encapsulating the box ratio on the width of the bounding box relative to the frame width. The severity is categorized as high (box ratio > 15), medium (10 <= box ratio <= 15), or low (box ratio < 10), and the coordinates of the pothole are saved for further processing and notification. Figure 4 shows the severity of the potholes. Finally, the coordinates of the detected potholes are sent as SMS notifications using the Twilio API, ensuring timely communication of the pothole locations to the designated recipient. Table 1 presents the parameters of confidence percentage, severity levels and co-ordinates of four pothole images.

5. Conclusion

This paper presents a novel approach for detecting potholes using Raspberry Pi and OpenCV, at an economical cost-point. This cost-cutting solution has the potential for scaling up for wider application while increasing road safety while speeding repair processes for potholes. Our system uses the computational capabilities of Raspberry Pi and the image processing tools of OpenCV to identify potholes in real time through image analysis. It has been tested over a multitude of environments and road types to successfully prove its reliability and versatility; insights realized from this study could significantly advance intelligent transportation systems while potentially offering benefits in infrastructure maintenance or urban planning applications. The small form factor of Raspberry Pi combined with the robust capabilities of image processing in OpenCV makes our system an attractive option for cities and transportation departments in pursuit of improving pothole detection efficiency and road maintenance efficiency.

Table 1. Pothole severity and Confidence percentage of a pothole.

Pothole Name	Confidence Percentage	Severity	Coordinates
Pothole 1	49	Low	[17.384, 78.4563]
Pothole 2	70	Medium	[17.384, 78.4562]
Pothole 3	90	High	17.384, 78.4564]
Pothole 4	49	Low	[17.384, 78.4564]

References

1. S. R. Kuthyar, S. Roopashree, V. Rasika, S. Manjesh, R. Girimaji, R. D. Arjun, and S. Priyadarshini, “An intelligent potholedetection and alerting system using mobile sensors and deep learning,” in 2021 IEEE 18th India Council International Conference (INDICON). IEEE, 2021, pp. 1–6.
2. R. Kavitha and S. Nivetha, “Pothole and object detection for an autonomous vehicle using yolo,” in 2021 5th international conference on intelligent computing and control systems (ICICCS). IEEE, 2021, pp. 1585–1589.
3. J. M. Celaya-Padilla, C. E. Galv´an-Tejada, F. E. L´opez Monteagudo, O. Alonso-Gonz´alez, A. Moreno-B´aez, A. Mart´ınezTorteya, J. I. Galv´an-Tejada, J. G. Arceo-Olague, H. Luna-Garc´ıa, and H. Gamboa-Rosales, “Speed bump detection using accelerometric features: A genetic algorithm approach,” Sensors, vol. 18, no. 2, p. 443, 2018.
4. M. M. Garcillanosa, J. M. L. Pacheco, R. E. Reyes, and J. J. P. San Juan, “Smart detection and reporting of potholes via image processing using raspberry-pi microcontroller,” in 2018 10th International Conference on Knowledge and Smart Technology (KST). IEEE, 2018, pp. 191–195.
5. U. Ajitha, “Smart sensing of potholes and obstacles in automotive applications using internet of things,” Int. J. Pure Appl. Math, vol. 119, no. 15, pp. 1479–1486, 2018.
6. Y. Jo and S. Ryu, “Pothole detection using android smartphone with a video camera,” 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:164209617

7. D. Desai, A. Soni, D. Panchal, and S. Gajjar, "Design, development and testing of automatic pothole detection and alert system," in 2019 IEEE 16th India council international conference (INDICON). IEEE, 2019, pp. 1–4.
8. M. H. Asad, S. Khaliq, M. H. Yousaf, M. O. Ullah, A. Ahmad, "Pothole detection using deep learning: A real-time and ai-on-the-edge perspective," Advances in Civil Engineering, vol.2022, 2022.
9. A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," arXiv preprint arXiv:2004.10934, 2020.
10. A. P. S. Rahman. Pothole image dataset. kaggle. [Online]. Available: https://www.kaggle.com/sachinpatel21/pothole-image-dataset/