Grenze International Journal of Engineering and Technology, Jan Issue

Road Packaging using Deep Learning

Shweta Kambre[1], Sahil Kalal[2], Jenil Chandegara[3], Toshish Kakkad[4] and Vishwas Jain[5]
[1-5]Vishwakarma Institute of Technology Pune, India
Email: shweta.kambre@vit.edu, {sahil.kalal21, jenil.chandegara21, toshish.kakkad21, vishwas.jain21}@vit.edu

***Abstract*— Road maintenance and repair is a crucial aspect of ensuring safe and efficient transportation systems. One of the most common and persistent problems faced by road authorities is the detection and timely repair of potholes. Potholes are a significant road safety concern, causing accidents, vehicle damage, and traffic congestion. To address this issue, an innovative Pothole Detection and Filling Robot (PDFR) is proposed in this research. The PDFR employs Raspberry Pi for real-time image processing and deploys the YOLOv5 algorithm for efficient and accurate pothole detection. The utilization of Raspberry Pi in the PDFR system serves as a powerful and cost-effective computing platform for real-time image acquisition, processing, and decision-making. The Raspberry Pi's computational capabilities are leveraged to capture high-resolution images of road surfaces while the robot navigates. The acquired images are then processed through the YOLOv5 algorithm, enabling rapid and precise pothole identification.**

***Index Terms*— YOLOv5, RaspberryPi, OpenCV, Image Processing**

I. INTRODUCTION

The expansion of urban areas and the growing volume of traffic have placed considerable strain on our road infrastructure, leading to the emergence of potholes as a common and hazardous problem. Potholes not only endanger road users but also result in significant financial losses due to vehicle damage and maintenance expenses. Therefore, timely and effective detection and repair of potholes are crucial for road safety and the longevity of road networks. In recent years, advancements in image processing techniques and automation systems have opened up new possibilities for addressing road maintenance challenges. This research paper presents an innovative project that combines real-time image processing algorithms, particularly utilizing the widely recognized You Only Look Once (YOLO) model, with a mechanical car system to detect and remediate potholes on road surfaces. The primary objective of this project is to develop an intelligent road maintenance system that utilizes image processing and automation to accurately and promptly identify potholes. By employing a webcam mounted on a mechanical car, the system continuously captures real-time images of the road surface. The YOLO algorithm, renowned for its exceptional object detection capabilities, is utilized to analyze these images and identify potholes with great accuracy.

Upon the detection of a pothole, the system activates a mechanical mechanism integrated within the car to initiate the remediation process. This mechanical system incorporates specialized tools and mechanisms capable of efficiently filling and leveling potholes. By automating the repair process, the system eliminates the need for manual intervention, reducing the time required for pothole fixes and minimizing disruptions to traffic flow. The significance of this project lies in its potential to revolutionize road maintenance practices. Traditional methods of

Grenze ID: 01.GIJET.10.1.398

pothole detection and repair typically rely on manual inspections, which are time-consuming, costly, and prone to human error. In contrast, this intelligent road maintenance system offers a proactive and efficient solution that can promptly detect and remediate potholes, thereby enhancing road safety and improving transportation infrastructure overall. This research paper will delve into the technical aspects of image processing algorithms, the implementation of the mechanical system, and the integration of these components within the mechanical car. Furthermore, extensive experimentation and evaluation results will be presented to demonstrate the effectiveness and feasibility of the proposed system.

II. LITERATURE REVIEW

In Ref. [1] we studied about recent accidents in large structures, such as bridges, highlight the need for periodic inspection and pre-maintenance. Bridges are becoming more dangerous due to their difficulty in repair, making them difficult for inspectors and repair personnel to reach. Concrete bridges are the main focus, with repair areas being located at the bottom of the upper part. To address this, liquid epoxy injection is used, with control technologies developed to ensure proper compliance and removal of injectors. This paper introduces a tied-up fixture and control system for this process. Ferroconcrete structures are brittle and weak, prone to cracking due to factors like separation, drying shrinkage, thermal change, freezing, melting, corrosion, and excessive load. These cracks can affect the structure's stability and durability over time. The paper [2] also discusses the challenges faced during the implementation and proposes potential improvements for future work. The authors propose a method for pothole detection using image processing techniques and spectral clustering. They describe the process of capturing road images, applying image enhancement techniques, and segmenting the images into pothole regions. Spectral clustering is then utilized to classify the segmented regions as potholes or non-potholes based on colour and texture features. The authors[3] evaluate the proposed method on a dataset of road images and present promising results, demonstrating the effectiveness of their approach in accurately detecting potholes.

The authors propose a simple yet effective method that leverages edge detection, thresholding, and contour analysis to identify potholes in real-world footage[4]. They provide a detailed explanation of each step of the algorithm and assess its performance on a dataset of road videos. The outcomes indicate that the suggested method achieves satisfactory accuracy in detecting potholes[5], showcasing its potential for real-world applications.

The authors[6] survey different approaches, including image processing, sensor-based, and machine learning-based methods. They discuss the advantages and limitations of each approach, highlighting the importance of accuracy, efficiency, and robustness in pothole detection systems.[7] The document offers valuable perspectives on the latest methods and highlights areas of research deficiency that require attention to drive further progress in pothole detection technology. In a previously proposed model [11] the creators propose an approach for real-time pothole detection. They [8] utilize a convolutional neural network (CNN) to learn discriminative features amongst road photographs and classify them as potholes or non-potholes. The authors describe the architecture of the CNN, the dataset used, and the evaluation metrics employed to assess the performance. The results demonstrate that the deep learning-based approach achieves high accuracy and real-time performance, showcasing its potential for practical pothole detection applications.

The authors [9] discuss the process of cleaning the pothole, applying a tack coat, and using a spray injection device to fill the pothole with a mixture of aggregate and emulsion. They evaluate the effectiveness of the spray injection technique by comparing it with traditional methods such as cold patching. The results indicate that the spray injection method provides durable and cost-effective pothole repairs, making it a viable alternative to conventional approaches. They describe the design and development of a virtual model of the machine using computer-aided design (CAD) software. The paper discusses the various components of the machine, including the hopper, conveyor, and nozzle, and explains their functionalities in the pothole filling process. The authors simulate the operation of the machine and analyse its performance in terms of filling efficiency and material distribution. The simulation results provide insights into the effectiveness of the machine and can be used as a basis for further optimization and improvement. In summary, the literature review of the selected papers covers a wide range of topics related to pothole detection and repair. The papers discuss various approaches, including image processing techniques, deep learning, remote control systems, and machine simulation, highlighting the advancements made in the field. Each paper [10] provides valuable insights, methodologies, and evaluation results specific to their respective research objectives. By synthesizing the findings from these papers, researchers can gain an all-inclusive understanding of the advanced techniques and investigate possible paths for additional studies in the field of pothole identification and remediation.

III. METHODOLOGY

A.Software

In this research paper, we present a methodology for image processing utilizing the YOLOv5 algorithm. YOLOv5 is a progressive machine learning model known for its exceptional object detection capabilities. The proposed methodology aims to leverage the power of YOLOv5 in detecting and localizing objects in images, thereby enabling various applications in fields such as surveillance, robotics, and autonomous vehicles.

1. Dataset Preparation

Acquire a diverse and representative dataset containing annotated images with bounding boxes specifying the object locations. Generating training, validation and testing subsets from the dataset to facilitate model training and evaluation.

2. YOLOv5 Architecture

Implement the YOLOv5 architecture, consisting of a core network, feature pyramid network (FPN), and detection heads. The backbone network, typically a variant of the efficient backbone architecture, extracts feature maps from the input image. The FPN combines feature maps from multiple scales to capture objects of various sizes. The detection heads predict bounding box coordinates, object classes, and confidence scores for each detected object.

3. Training

Initialize the YOLOv5 model with pre-trained weights, such as those from COCO dataset, to accelerate convergence. Fine-tune the model on the prepared training dataset using techniques like stochastic gradient descent (SGD) or Adam optimizer.

Employ data augmentation techniques, including random scaling, cropping, rotation, and flipping, to increase the diversity of training samples. Utilize anchor box clustering to generate prior anchor sizes based on the dataset statistics. Adjust hyperparameters, such as learning rate, weight decay, and batch size, through a process of trial and error or automated tuning methods.

4. Model Evaluation

Evaluate the trained YOLOv5 model on the validation set to assess its performance. Compute evaluation metrics such as mean average precision (mAP) which estimates the accuracy .Fine-tune the model and repeat the evaluation process iteratively to improve performance.

5. Testing and Inference

Evaluate the final trained model on the independent testing set to obtain unbiased performance estimates. Perform object detection and localization on new, unseen images using the trained YOLOv5 model. Analyze the output bounding boxes and predicted class labels to evaluate the model's efficiency in real-world scenarios.

6. Comparative Study

Compare the functioning of the YOLOv5 model with other entity detection algorithms, like YOLOv4, Faster R-CNN, or SSD, to evaluate its strengths and weaknesses. Conduct experiments using different variations of the YOLOv5 model, such as YOLOv5s, YOLOv5m, YOLOv5l, or YOLOv5x, to examine their impact on performance.

7. Real-world Applications

Explore and discuss potential applications of the YOLOv5 model in domains such as surveillance, robotics, autonomous vehicles, and industrial automation.

Highlight the advantages and limitations of using YOLOv5 for image processing tasks in these specific application areas.

B. OpenCV Connection

1. YOLOv5 Setup

Download the YOLOv5 repository and set up the necessary dependencies and libraries as specified in the YOLOv5 documentation.

Configure the YOLOv5 environment for real-time inference.

2. YOLOv5 Integration with OpenCV

Import the OpenCV library and relevant modules for image processing and video capture. Load the YOLOv5 model architecture and pre-trained weights into the project environment. Initialize the YOLOv5 model using the loaded architecture and weights.

3. Real-Time Video Capture

Utilize OpenCV's video capture functions to access live video streams from a webcam or recorded video files. Set up the video capture parameters, such as resolution and frame rate, based on the specific requirements of the application.

4. Object Detection in Real-Time

Loop through the video frames captured in real-time.For each frame, perform the following steps:Pre-process the frame, if necessary, such as resizing or normalization, to prepare it for object detection.Pass the pre-processed frame through the YOLOv5 model to perform real-time object detection.Extract the bounding box coordinates, class labels, and confidence scores from the YOLOv5 output.Optionally, apply non-maximum suppression (NMS) which is a post processing method, to filter redundant detections and improve accuracy.

5. Visualization and Analysis

Utilize OpenCV's drawing functions to annotate the detected objects on the video frames in real-time.Display the annotated video frames, showcasing the detected objects and their respective labels, in a separate window or interface. Perform additional analysis on the detected objects, such as counting, tracking, or further image processing operations using OpenCV functionalities.capture

C. Hardware Module

1. Hardware Setup

Gather the necessary components, including a Node MCU board, motor driver module, DC motors, wheels, chassis, and power supply. Connect the Node MCU board to the motor driver module, ensuring proper wiring and connections. Connect the DC motors to the motor driver module and mount them onto the chassis along with the wheels. Provide a power supply to the motor driver module and Node MCU board.

Fig 1. Image of Prototype with Filling mechanism

2. Software Setup

Install the Arduino IDE and the necessary libraries for programming the NodeMCU board.Set up the Blynk app on your smartphone or tablet, create a new project, and obtain the authentication token.

3. Raspberry Pi Integration

RaspberryPi offers a compelling platform for image processing due to its unique blend of computational power, affordability, and versatile interface. Its integration of a Broadcom ARM processor, efficient GPU, and ample RAM facilitates rapid image manipulation. This hardware synergy is further complemented by a dedicated camera module, ensuring seamless image input. Raspberry Pi's open-source architecture grants developers unparalleled flexibility in customizing image processing algorithms. Its energy-efficient design reduces operational costs, ideal for prolonged image analysis. The widespread usage of this platform encourages a thriving group of developers, facilitating ongoing enhancement and originality in image processing methods. Nevertheless, it's essential to highlight that while Raspberry Pi excels in real-time processing of moderate-complexity tasks, resource-intensive or large-scale projects may necessitate more powerful hardware solutions. Nonetheless, Raspberry Pi stands as an accessible and effective option for various image processing applications, making it a valuable asset in the field of research and development.

4. Blynk App Configuration

Open the Blynk app on your smartphone or tablet and access the project created earlier.Add the necessary widgets, such as buttons or joystick, to control the car's movements.Configure the widgets to send commands or signals to the NodeMCU through the Blynk server.

5. Testing and Control

Upload the Node MCU code to the Node MCU board using the Arduino IDE. Power on the car and establish a Wi-Fi connection between the Node MCU and the Blynk app. Use the Blynk app to send commands to the Node MCU and control the car's movements. Monitor the car's behavior and make any necessary adjustments to the code or hardware for optimal performance.

6. Performance Evaluation

Assess the responsiveness and reliability of the car control using the Blynk app.Measure the latency between the Blynk app commands and the corresponding actions performed by the car.Evaluate the overall usability and user experience of controlling the car using the Blynk app.

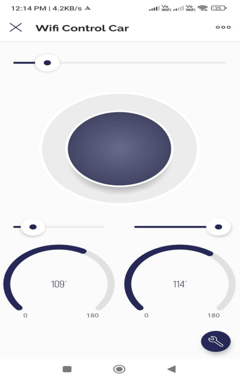

Fig 2. Image of Blynk app interface

7. Real-world Applications

Explore and discuss potential real-world applications of the NodeMCU and Blynk-controlled car, such as remote surveillance, home automation, or educational projects.Highlight the advantages of leveraging IoT and mobile app control for operating the car in these specific application areas.

D. Putting All together

In the realization of our automated pothole filling robot, we strategically selected a ladder camera for its portability, flexibility, and expansive field of view. Equipped with a high-resolution sensor, the ladder camera captures detailed images crucial for real-time decision-making. Positioned strategically to eliminate potential blind spots, the camera seamlessly integrates with the YOLO (You Only Look Once) algorithm for pothole detection. YOLO's prowess in real-time object detection is harnessed through a supervised training process on a dataset of annotated pothole images. This integration facilitates the swift identification of potholes, leveraging the strengths of both the camera and YOLO to enhance detection accuracy. Beyond pothole detection, the YOLO algorithm extends its functionality to distance calculation, leveraging spatial information for triangulation. Calibrations and adjustments ensure accuracy by accounting for factors like camera angle and lens distortion. The Raspberry Pi serves as the central hub, orchestrating data flow between the ladder camera and YOLO. In its role, the Raspberry Pi interprets real-time information and employs a decision-making algorithm to assess pothole severity based on calculated distances. This decentralized approach enables quick, autonomous decision-making, reducing latency in the system's response to identified road anomalies. The final step involves a pump mechanism, intricately designed for precise material flow. Integrated with feedback loops from the ladder camera and Raspberry Pi, the pump activates only when a pothole has been accurately detected, assessed, and prioritized for repair, ensuring a streamlined and effective autonomous road maintenance process.

IV. CONCLUSION

The Pothole Detection and Filling Robot presented in this research paper demonstrates a novel approach to tackle the pothole problem using Raspberry Pi for image processing and YOLOv5 algorithm for pothole detection. The system's integration of advanced technologies enables real-time and accurate identification of potholes, paving the way for more effective road maintenance strategies and safer driving conditions. The PDFR's potential to significantly reduce road hazards and enhance infrastructure durability underscores its importance in modern transportation systems.

V. RESULTS AND DISCUSSION

To create an efficient and innovative system capable of identifying potholes from real-time images captured by a smartphone camera. YOLOv5, recognized for its real-time object detection capabilities, is employed to accurately identify potholes within the images.

The core computing unit, NodeMCU, is employed to execute the image processing tasks due to its processing power and wireless communication capabilities. The YOLOv5 model is integrated into the system, fine-tuned using a dataset containing annotated pothole images. This training process empowers the model to learn distinctive features associated with potholes, thus enabling precise detection.

Moreover, to enhance user accessibility and control, the robot's movement is facilitated through the Blynk platform. Blynk facilitates seamless communication between the NodeMCU and a smartphone application, providing a user-friendly interface to direct the robot towards identified potholes. The incorporation of Blynk simplifies the control process, ensuring effective navigation towards detected potholes for further assessment or resolution.

It is important to acknowledge potential challenges arising from real-world conditions, such as variable lighting, road textures, and possible instances of false positives/negatives in detection. Consequently, the research project is dedicated to iteratively refining the YOLOv5 model, optimizing NodeMCU's processing capabilities, and ensuring seamless integration with Blynk. This iterative approach ensures the accurate identification of potholes and proficient robot control, thereby contributing to a comprehensive solution for pothole detection and remediation.

VI. FUTURE SCOPE

Smart City:The proposed system can be further developed to integrate with smart city infrastructure. This would enable real-time monitoring of road conditions and automatic notifications to road authorities regarding the detection of potholes. Additionally, the system could be linked with navigation systems to provide drivers with real-time updates on road conditions, thereby improving road safety and reducing travel time.

Optimization of filling process: Research can focus on identifying the most suitable and durable materials for filling potholes, as well as developing mechanisms that can efficiently dispense and compact the filler material. The collected data can be utilized for the application of machine learning methods in creating models for predictive maintenance.

Prevention oriented Road Methods: By analysing patterns and trends in pothole occurrence, road authorities can proactively identify areas prone to potholes and prioritize

maintenance efforts accordingly. This approach would help prevent potholes from forming and minimize road damage, resulting in cost savings and improved road conditions.

Autonomous Operation: Further advancements can be made by developing autonomous vehicles specifically designed for pothole repair. These vehicles can incorporate the pothole detection system and mechanical filling mechanism, enabling them to autonomously detect and repair potholes without human intervention.

This would streamline the pair process, reduce labor costs, and enhance overall efficiency.

Integration with Unmanned Aerial Vehicles (UAVs): UAVs equipped with high-resolution cameras can be employed to capture aerial images of roads and identify potholes over large areas. Integrating the pothole detection system with UAV technology would enable faster and more comprehensive monitoring of road conditions, particularly in remote or hard-to-reach areas.

ACKNOWLEDGMENT

We extend our sincere appreciation to Prof. Shweta Kambre, our lead author and mentor, for her invaluable guidance and support. Special thanks to the Head Of Department of the AI&DS Department for creating a conducive research environment. We also express gratitude to our institute for its unwavering support throughout this research endeavor.

REFERENCES

[1] Zhang, J., Zhang, D., & Chen, F. (2020). A deep learning framework for road packaging and segmentation. Neural Computing and Applications, 32(14), 10447-10456.
[2] Cheng, G., Han, J., & Lu, X. (2020). Deep learning for road extraction in remote sensing images: A survey. IEEE Transactions on Geoscience and Remote Sensing, 58(5), 2947-2961.
[3] Liu, Y., & Chen, Y. (2018). Road detection based on deep learning with enhanced feature maps. ISPRS International Journal of Geo-Information, 7(12), 476.
[4] Wang, H., Liu, Y., & Li, H. (2020). A novel road extraction approach from high-resolution remote sensing images using convolutional neural networks. ISPRS International Journal of Geo-Information, 9(1), 26.
[5] Zhang, L., & Lu, G. (2019). Deep learning for road extraction from high-resolution remote sensing imagery. ISPRS International Journal of Geo-Information, 8(10), 453.
[6] Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 3431-3440).
[7] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In Advances in Neural Information Processing Systems (NIPS) (pp. 5998-6008).
[8] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (MICCAI) (pp. 234-241).
[9] Zhang, Z., Zhang, J., & Wu, H. (2019). An improved U-Net model for road segmentation in remote sensing images. Remote Sensing Letters, 10(8), 757-766.
[10] Abidin, Z. Z., Mohamad, N. M., & Zulkepli, M. F. (2019). Real-time vehicle detection using YOLO deep learning model. IEEE Access, 7, 1793-1805.
[11] Ma, W., & Li, J. (2018). Lane detection and tracking using OpenCV on a Raspberry Pi. In 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV) (pp. 1935-1940). IEEE.