POTHOLE DETECTION SYSTEM USING YOLO v8 ALGORITHM

Siddharth Sasane
Department of Electronics and Telecommunication
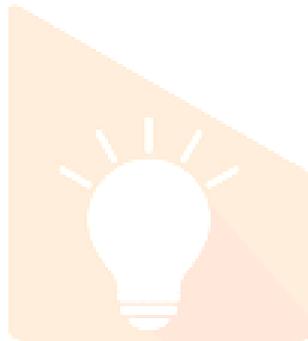Sardar Patel Institute of Technology

Akshat Tickoo
Department of Electronics and Telecommunication
Sardar Patel Institute of Technology

Kevin Xavier
Department of Electronics and Telecommunication
Sardar Patel Institute of Technology

Prof. Manish Parmar
Department of Electronics and Telecommunication
Sardar Patel Institute of Technology

Abstract— Potholes pose significant risks to road safety, causing accidents, vehicle damage, and potential harm to drivers and pedestrians. Annually, a considerable number of fatalities, approximately 3564, are attributed to these hazardous road imperfections. Addressing the pressing need for detection and prevention, this project introduces a novel pothole detection system aimed at mitigating these dangers. The primary objective was to develop an efficient detection system specifically tailored for identifying potholes, catering to the contemporary era's emergence of autonomous vehicles. Subsequently, the optimal model was employed to train the dataset, culminating in its application to detect potholes in real-time through a live video feed. featuring a user-friendly Graphical User Interface (GUI) for easy interaction.

I. INTRODUCTION

The condition of roadways stands as a pivotal factor in the developmental trajectory of any nation, representing a fundamental facet of national infrastructure. Annually, a considerable number of fatalities, approximately 3564, are attributed to these hazardous road imperfections [1]. The regular assessment of road conditions is indispensable for the maintenance and enhancement of the road network. Traditional manual assessment methods, however, prove to be cumbersome, expensive, and susceptible to human error. The subjective nature of manual assessments introduces the risk of results being influenced by the personal biases and experience of the assessors. To address these challenges and elevate the efficiency of road monitoring, an automatic detection system for potholes and their assessment becomes imperative.

(a) water filled pothole

(b) dry pothole

Potholes, characterized by irregular and complex shapes, pose a substantial threat to road safety. Their presence not only leads to vehicular accidents but also contributes to vehicle damage, increased traffic congestion, and diminished fuel efficiency. This underscores the critical need for a comprehensive system capable of automatically detecting and assessing potholes. The detection of potholes becomes a cornerstone in the ongoing efforts to maintain and improve road conditions.

The significance of this issue is magnified by the frustration and risks experienced by individuals who traverse roadways regularly. Potholes, often a consequence of severe weather conditions and heavy vehicle activity, present a substantial concern for drivers. The resulting accidents and damage to vehicles underscore the urgency for effective pothole detection and remediation measures. Unforeseen humps and ditches on road surfaces elevate the risk of collisions, emphasizing the necessity of prompt pothole detection and repair to enhance road safety and driving experience.

In response to these challenges, this paper explores an innovative approach in the realm of computer vision—the You Only Look Once version 8 (YOLO v8) algorithm developed by the Ultralytics team, renowned for their contributions to YOLO v4 and v5. The primary objective is to develop a robust system that leverages deep learning to automatically detect potholes. Such a system not only automates the detection process but also enhances the understanding of drivers, alerting them to the presence of potholes on the road.

Furthermore, this paper delves into the validation of the robustness of the YOLO v8 algorithm. A comparative analysis is conducted among different versions of the algorithm, including medium, nano, and small YOLO v8. The Mean Average Precision (mAP) parameter serves as the benchmark for evaluating and selecting the most suitable version for the proposed pothole detection system. Through this exploration, the paper aims to contribute valuable insights and recommendations for the effective implementation of a YOLO v8-based pothole detection system in real-world scenarios.

II. Literature Survey

In the dynamic landscape of road infrastructure and safety, the evolution of pothole detection systems has traversed a noteworthy trajectory, characterized by significant advancements over several years. This comprehensive literature review unfolds a chronicle of innovations, each contributing to the refinement and enhancement of pothole detection technologies.

In the year 2017, a pioneering approach emerged, harmonizing the capabilities of 2D LiDAR and a camera. This synergy aimed to capitalize on the unique strengths of each sensor, endowing the system with the ability to provide precise measurements through LiDAR's accurate distance and angle information while concurrently conducting a detailed analysis of pothole shapes using the camera's imaging capabilities. This integration set the stage for subsequent developments, laying the foundation for more sophisticated and accurate pothole detection systems.

The year 2020 witnessed a deep learning approach that further expanded the horizons of pothole detection. Four distinct models, namely YOLO V3, SSD, HOG with SVM, and Faster R-CNN, were subjected to rigorous evaluation. This initiative sought to unravel the comparative performance of these models, contributing valuable insights into the strengths and weaknesses of each, and paving the way for informed advancements in subsequent studies.

Building on this foundation, another exploration in 2020 specifically targeted the challenging terrain of Indian roads. Acknowledging the promise demonstrated by Convolutional Neural Networks (CNNs) in object detection tasks, this study honed in on the YOLO framework. However, it grappled with the unique challenge posed by the scarcity of specific pothole datasets for Indian roads, highlighting the ongoing need for context-specific solutions in the pursuit of global road safety.

In 2019, a novel methodology emerged, introducing the integration of wavelet energy modules and Markov random field for heightened accuracy in pothole detection and segmentation. This approach, while innovative, acknowledged certain limitations, such as the time-consuming segmentation process associated with the use of Markov random field. Nevertheless, it demonstrated a commitment to refining existing techniques for optimal performance.

Fast-forwarding to 2021, a paradigm shift occurred with the introduction of the YOLOX algorithm. Representing a substantial leap forward, YOLOX offered a swift and precise solution for automated pothole detection. This algorithm, recognized for its efficiency and accuracy in object detection, marked a significant departure from previous methods and underscored the continuous evolution within the field.

The foundational YOLO algorithm, originating in 2016, represented a seminal moment in the realm of object detection. Trained using a Convolutional Neural Network (CNN) architecture on a COCO dataset, YOLO addressed the challenges of complexity and slow processing speeds inherent in existing object detection systems. By iterating through an image only once, YOLO provided a unified, real-time object detection solution, emphasizing its potential to overcome the limitations of its predecessors. This systematic review encapsulates a continuous pursuit of accuracy and efficiency in pothole detection, showcasing the commitment of researchers and practitioners to advancing road safety and infrastructure maintenance. The evolution from sensor integration to deep learning algorithms mirrors a collective dedication to harnessing cutting-edge technologies for the betterment of road infrastructure worldwide.

III. Comparative Analysis

The landscape of object detection within the realm of road infrastructure and safety has witnessed significant advancements, notably in the evolution of the You Only Look Once (YOLO) models. This comparative analysis delves into the intricacies of three distinct YOLOv8 variants—YOLOv8-Small, YOLOv8-Medium, and YOLOv8-Nano—each designed with specific considerations for the trade-off between inference speed and accuracy in the context of pothole detection.

A. Model Architectures

The YOLOv8-Small model is characterized by its emphasis on faster inference speed achieved through a reduction in size. This design choice renders it well-suited for real-time applications where expeditious pothole detection takes precedence. However, a potential drawback lies in its compromise on accuracy, particularly in discerning smaller potholes or those situated within complex backgrounds. In contrast, the YOLOv8-Medium strikes a balance between speed and accuracy, presenting a versatile option for reliable pothole detection across various scenarios. It occupies a middle ground, demonstrating adaptability without sacrificing essential accuracy. On the other hand, YOLOv8-Nano, renowned for its speed, sacrifices a degree of accuracy to maintain efficiency in real-time applications. This model excels in swiftly detecting larger potholes but may overlook finer details in the process.

B. Performance Evaluation

The accuracy of each YOLOv8 variant is a critical factor influencing their applicability. YOLOv8-Small, with its emphasis on speed, may exhibit trade-offs in challenging detection scenarios. YOLOv8-Medium, striking a balance, offers reliable accuracy across a spectrum of pothole sizes and backgrounds. YOLOv8-Nano, optimized for speed, excels in detecting larger potholes quickly but may compromise on the accuracy required for discerning finer details. Additionally, inference speed is a pivotal performance metric. YOLOv8-Small achieves faster inference due to its reduced size, making it suitable for scenarios prioritizing swift detection. YOLOv8-Medium, with its balanced design, caters to real-time

applications without compromising on speed. YOLOv8-Nano, known for its rapid processing, is tailored for applications where quick detection takes precedence over exhaustive accuracy.

C. Application Scenarios

The suitability of each YOLOv8 variant depends on specific application scenarios. YOLOv8-Small finds relevance in real-time surveillance or monitoring where rapid pothole detection is critical. YOLOv8-Medium, being versatile and adaptable, fits scenarios where a balance between speed and accuracy is essential, such as urban environments with varying pothole sizes and backgrounds. YOLOv8-Nano, optimized for speed, is ideal for high-speed road monitoring or applications where detecting larger potholes is the primary concern, and a slight compromise in accuracy is acceptable.

D. Conclusion

In conclusion, the choice between YOLOv8-Small, YOLOv8-Medium, and YOLOv8-Nano hinges on specific application requirements. Understanding the nuances of each variant allows practitioners to tailor their choice based on the critical factors of accuracy and speed in pothole detection applications, thereby contributing to the ongoing evolution of object detection methodologies within road safety and infrastructure maintenance.

Fig 3.1 YOLO v8 medium mAP values after 30 epochs

Fig 3.3 YOLO v8 nano mAP values after 30 epochs
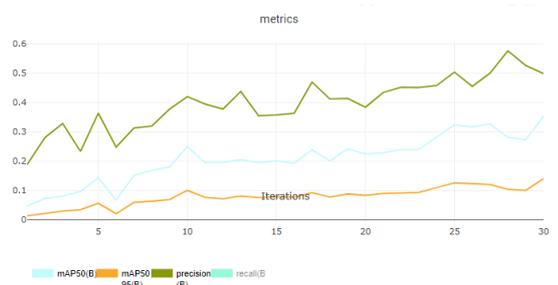
Fig 3.2 YOLO v8 small mAP values after 30 epochs

IV. PROPOSED METHODOLOGY

A. Training the YOLO v8 Model

The following steps were taken in training the model:

1. Prepared a dataset and annotations for training
2. Created a Google Collab notebook and set the runtime type to use GPU acceleration.
3. Installed required dependencies like PyTorch, CUDA and other necessary libraries.
4. Cloned the YOLOv8 code repository from GitHub into your Colab environment.
5. Configured the YOLOv8 model according to the dataset specifications, setting up parameters such as input size, number of epochs and other training configurations.
6. Obtain pre-trained weights for YOLOv8, for initial weight initialization.
7. Monitored the training process for convergence of loss and handled all encountered errors during training.
8. Saved the trained model weights upon completion of the training process.
9. Stored the trained model, weights, and necessary files in ONNYX format to be used in our system.
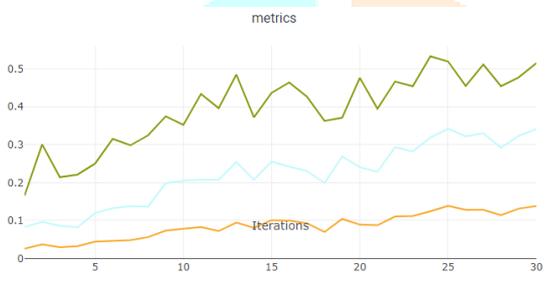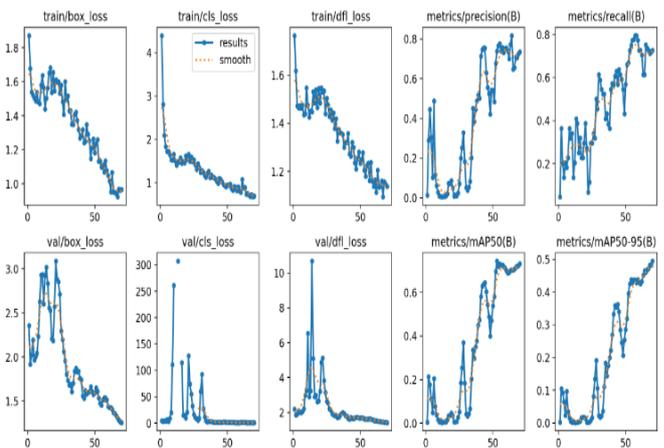

Fig 4.1: Various Plots of box_loss/cls_loss etc.

B. Integrating the model and creating a GUI

The After performing adequate training of our model we imported the trained weights into our system followed by:

1. Integration with GUI Framework utilizing the Tkinter framework, a user-friendly interface was developed. This integration enabled seamless real-time and batch pothole detection within the interface.

2. Input and Output Handling: The GUI efficiently accepts diverse inputs, including images and live video streams from cameras. Detected potholes are displayed through elements like file upload buttons and live camera feeds, offering a comprehensive view of the detection process.

3. User Interaction and Feedback: Interactive components within the GUI, such as intuitive buttons for initiating or halting detection, provide users with detailed insights into identified potholes. Detection results are communicated through visual indicators like bounding boxes, labels, and confidence scores.
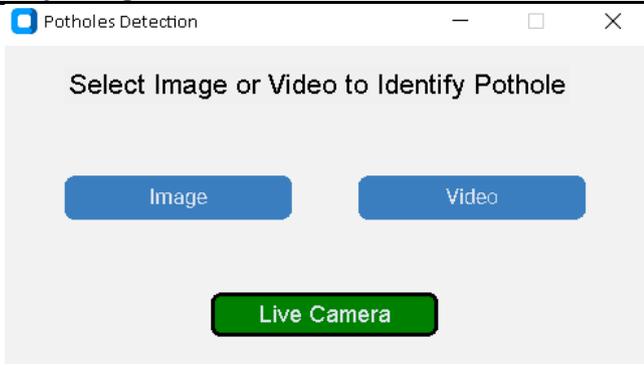
Fig 4.2 The System GUI

C. System Workflow

The following steps indicate the system workflow:

1. Image Input: Users capture or upload images through the GUI interface.
2. Preprocessing: The system preprocesses the images for optimal analysis.
3. Pothole Detection: The selected YOLOv8 model processes the images to identify potholes.
4. Visualization: Detected potholes are highlighted and displayed on the GUI for user review.
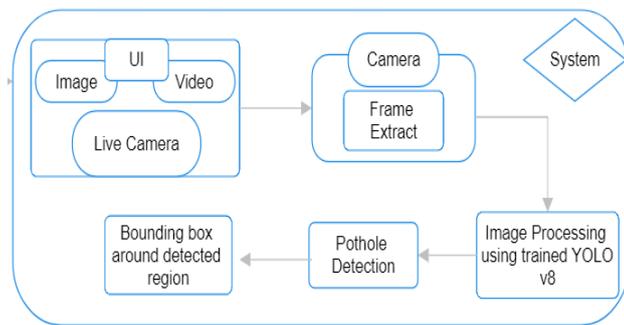5. Reporting and Sharing: Users can share the results or reports containing detected potholes.

Fig 4.1 Proposed System Flow

V. RESULTS

A. Performance Metrics

The following metrics were used for evaluating and comparing the models:

1) Mean Average Precision: Mean Average Precision (mAP), computed using equation 1, is a widely accepted performance metric for object detection models. mAP is calculated by taking the mean of Average Precision (AP) for each of the 'n' classes. AP for each class 'k' is determined by calculating the area under the precision-recall curve. mAP provides a single score that considers Recall, Precision, and Intersection over Union (IoU), eliminating bias in performance measurement.

$$ mAP = 1\ n\ k\ X{=}n\ k{=}1\ AP \qquad (1) $$

2) Processing Time: Processing time is a crucial metric for assessing the speed at which the model classifies an input image. It encompasses the total time taken by the model for pre-processing, inference, loss calculation, and post- processing of an image. Swift

decision-making is essential for autonomous vehicles to avoid potholes, making the minimiza- tion of processing time imperative.

3) Size of the Trained Model: The size of the deployed model, responsible for processing incoming data in embedded systems, depends on the size of the final trained model generated. Keeping this size as small as possible is essential due to limitations in onboard memory and the storage capacity of the hardware. Larger models require more computation, resulting in lower power efficiency.

B. Results

Figure presents the results obtained for Processing Time and the size of the model on various YOLO models.

The inclusion of manholes and sewer covers in the training dataset benefits the model by increasing its robustness in correctly identifying potholes, which is crucial for autonomous vehicles to take appropriate action. In the case of manholes and sewer covers, there is no need for the vehicle to take any action, preventing the model from giving false positives. As illustrated in the first image of Figure 2, the model accurately identifies and distinguishes between a sewer cover and a pothole, highlighting the robust nature of our model.

Model	Processing Time
YOLOv8 Nano	5.779 ms
YOLOv8 Small	6.660 ms
YOLOv8 Medium	7.963 ms

FIG 5.1 PROCESSING TIME OF YOLOv8-NANO

Comparing our implementation of YOLOv8 with previous works based on YOLO, superior performance is displayed across all categories such as mAP, Processing Time, and Size of Model
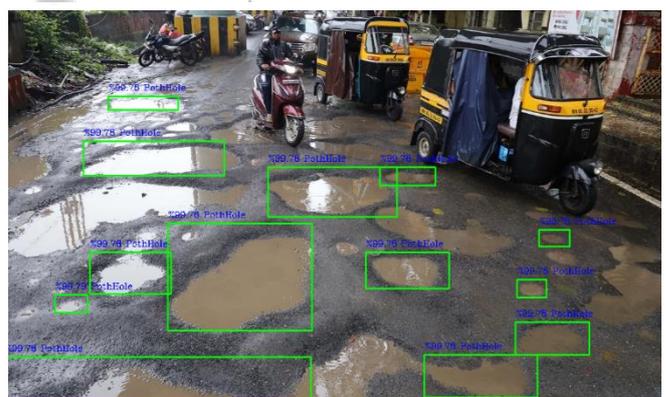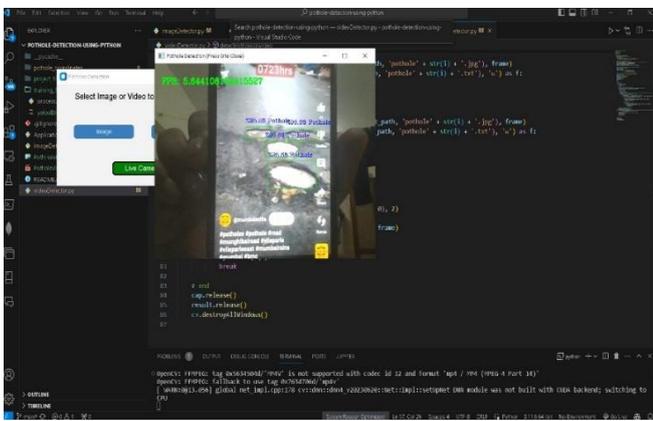

FIG 5.2

FIG 5.3

Figs 5.2, 5.3: Final System Outputs

CONCLUSION

The study analyzed the capabilities of YOLOv8 models for the utility of pothole detection and proposed a system design that provides a user-friendly interface for efficient pothole detection and reporting, integrating YOLOv8 models with a GUI to streamline the process. The study also emphasizes the need of having a robust and diversified training dataset that includes a variety of road components in addition to potholes, such as manholes and sewer covers. This methodical approach greatly improved our model's ability to discriminate between various road hazards, resulting in a considerable decrease in false positives and ensuring accurate pothole identification.

In conclusion, the YOLOv8 model—in particular, the nano variant—appears to be a very promising method for detecting traffic hazards. It advances road safety and infrastructure maintenance considerably by finding a delicate balance between accuracy, speed, and resource efficiency. This research opens the door for improved safety and effectiveness in the upkeep of road infrastructure in addition to making a contribution to the field of road hazard detection. Future endeavors could entail practical implementation and

As this research opens new avenues for road hazard detection, future endeavors could explore the real-world implementation of the YOLOv8-based system in collaboration with municipal authorities. This could involve pilot programs to assess its practicality, scalability, and integration into existing urban infrastructure management systems. By taking such steps, the study not only solidifies the YOLOv8 model's position as a technological marvel but also actively contributes to the advancement of smart city initiatives, promising a safer and more resilient urban environment.

REFERENCES

[1] Pal R, Ghosh A, Kumar R, Galwankar S, Paul SK, Pal S, Sinha D, Jaiswal AK, Moscote-Salazar LR, Agrawal A. Public health crisis of road traffic accidents in India: Risk factor assessment and recommendations on prevention on the behalf of the Academy of Family Physicians of India. J Family Med Prim Care. 2019 Mar;8(3):775-783. doi: 10.4103/jfmpc.jfmpc_214_18. PMID: 31041200; PMCID: PMC6482791.

[2] E. S. T. K. Reddy and R. V, "Pothole Detection using CNN and YOLO v7 Algorithm," 2022 6th International Conference on Electronics, Communication and Aerospace Technology, Coimbatore, India, 2022, pp. 1255-1260, doi: 10.1109/ICECA55336.2022.10009324.

[3] D. J, S. D. V, A. S A, K. R and L. Parameswaran, "Deep Learning based Detection of potholes in Indian roads using YOLO," 2020 International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 2020, pp. 381-385, doi: 10.1109/ICICT48043.2020.9112424.

[4] L. Dihua Wu, Shuaichao Lv, Mei Jiang, Huaibo Song,Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments,Computers and Electronics in Agriculture,Volume 178, 2020,105742,ISSN01681699, https://doi.org/10.1016/j.compag.2020.105742.

[5] Seraj, Fatjon & van der Zwaag, Berend Jan & Dilo, Arta & Luarasi, Tamara & Havinga, Paul. (2016). RoADS: A Road Pavement Monitoring System for Anomaly Detection Using Smart Phones. 128–146. 10.1007/978-3-319-29009-6_7. A. Ozdemir and S. Bogosyan, "Neural Network based Trading Signal Genera-¨ tion in Cypto-Currency Markets," 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), Malatya, Turkey, 2018, pp. 1-4, doi: 10.1109/IDAP.2018.8620889.

[6] S. Arjapure and D. R. Kalbande, "Deep Learning Model for Pothole Detection and Area Computation," 2021 International Conference on Communication information and Computing Technology (ICCICT), Mumbai, India, 2021, pp. 1-6, doi: 10.1109/ICCICT50803.2021.9510073.

[7] Gupta, Saksham & Sharma, Paras & Sharma, Dakshraj & Gupta, Varun & Sambyal, Nitigya. (2020).

[8] Detection and localization of potholes in thermal images using deep neural networks. Multimedia Tools and Applications. 79. 10.1007/s11042-020-09293-8.