International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 01 | Jan - 2025 SJIF Rating: 8.448 ISSN: 2582-3930

AI Car with Real Time Detection of Damaed Road and Lane Detection

Ms.k.Sonali [1], P.Supriya[2], N.Chaitanya[3], T.Vanshraj[4], B.Rahul[5]

[1] Ms.k.sonali (assistant professor)
[2]P.Supriya Department of Computer Science and Engineering (Joginpally B.R Engineering College)
[3]N.Chaitanya Department of Computer Science and Engineering (Joginpally B.R EngineeringCollege)
[4] T.Vanshraj Department of Computer Science and Engineering (Joginpally B.R Engineering College)
[5]B.Rahul Department of Computer Science and Engineering (Joginpally B.R Engineering College)

---***---

ABSTRACT

The primary goal of this project is to develop a robust and efficient road damage detection and warning system using deep learning techniques. The system aims to provide real-time or near-real-time warnings to enhance traffic safety and reduce the risk of accidents. By eliminating the need for cloud-based processing and leveraging deep learning models, the project seeks to overcome the limitations of current systems, such as latency and the requirement for large labeled datasets. Additionally, the system aims to continuously map road conditions and provide valuable data to city planners and road maintenance authorities, facilitating proactive maintenance and repair scheduling. Ultimately, this project aims to improve road infrastructure quality and ensure safer travel for all road users. This study proposes a deep learning approach for road damage detection and warning, aimed at overcoming these limitations. By eliminating the dependence on cloud-based processing, our system can deliver real-time or near-real-time warnings, facilitating timely interventions to mitigate risks. Deep learning models have proven to be highly effective in image and video analysis tasks, making them well-suited for detecting road damage.Our vision-based lane detection approach is capable of operating in real-time with robustness to varying road conditions, lighting changes, and shadows. This enhances the safety and performance of autonomous driving systems. Additionally, the continuous mapping of road conditions generates valuable data for city planners and road maintenance authorities. Integration with cloud-based infrastructure enables the reporting of damage locations, allowing for proactive road maintenance and repair scheduling. By reducing the likelihood of accidents and improving the quality of road infrastructure, this AI-powered system significantly contributes to road safety and maintenance efficiency.

1.INTRODUCTION

Road infrastructure is a critical component of modern transportation systems, directly impacting the safety and efficiency of travel. However, road damage, such as potholes, cracks, and uneven surfaces, has become a significant concern due to its role in numerous traffic accidents and fatalities. The timely detection and repair of road damage are essential to mitigate risks and ensure the safety of road users. Traditional road damage detection systems often rely on cloud-based processing, which introduces latency issues due to long-distance data transmission. Furthermore, these systems predominantly use supervised machine learning methods that require extensive, meticulously labeled datasets, making them resource-intensive and less adaptable to varying conditions.

In response to these challenges, this study proposes a novel deep learning approach for road damage detection and warning. The primary objective of this project is to develop a system capable of providing real-time or near-real-time warnings without relying on cloud-based processing, thereby reducing latency and enabling timely interventions. Deep learning models have demonstrated remarkable performance in image and video analysis tasks, making them particularly well-suited for this application.

Our proposed system employs a vision-based lane detection approach that maintains robustness against varying road conditions, lighting changes, and shadows. This approach not only enhances the safety and performance of autonomous driving systems but also contributes to the continuous mapping of road conditions. By integrating this real-time detection system with cloud-based infrastructure, we can report damage locations efficiently, allowing city planners and road maintenance authorities to schedule proactive maintenance and repair activities. This integration facilitates the collection of valuable data that can be used to improve the overall quality of road infrastructure.

Moreover, the proposed system aims to reduce the likelihood of accidents by providing timely warnings and ensuring better road conditions. By leveraging AI-powered technologies, this project seeks to revolutionize road maintenance strategies, making them more efficient, cost-effective, and responsive to real-world challenges. The ultimate goal is to enhance road safety and infrastructure quality, contributing to safer and more reliable transportation networks

2. LITERATURE REVIEW

Traditional road damage detection methods often rely on manual inspections and visual observations by experts. These methods, while accurate, are time-consuming, labor-intensive, and subject to human error. Quantitative analysis using specialized tools, such as laser scanners and high-resolution cameras, has also been employed to analyze road surfaces. However, these tools are expensive and require significant maintenance. Additionally, the reliance on cloud-based processing in some systems introduces latency issues due to the long-distance transmission of data.

Supervised machine learning techniques have been widely used in road damage detection systems. These methods require large datasets with labeled examples of road damage, which are used to train the models. While supervised learning approaches can achieve high accuracy, the process of collecting and labeling data is resource-intensive and time-consuming. Furthermore, these models may struggle to adapt to new and varying road conditions without continuous updates to the trainin dataset.

Deep learning models, particularly convolutional neural networks (CNNs), have demonstrated exceptional performance in image and video analysis tasks. Recent advancements in deep learning have enabled the development of models capable of detecting and classifying various types of road damage with high accuracy. These models can automatically learn features from raw data, eliminating the need for manual feature extraction. Studies have shown that deep learning approaches outperform traditional machine learning methods in terms of accuracy and adaptability.

The need for real-time or near-real-time detection of road damage is crucial for traffic safety. Recent research has focused on developing systems that can process data locally, without relying on cloud-based infrastructure. Edge computing has emerged as a promising solution, allowing data to be processed at the source, thus reducing latency and enabling timely warnings. By leveraging edge computing and deep learning, researchers aim to create systems that can detect road damage in real-time and provide immediate feedback to road users and authorities.

Lane detection is a critical component of autonomous driving systems. Vision-based lane detection methods use cameras and image processing techniques to identify and track lane markings. Challenges such as varying road conditions, lighting changes, and shadows make lane detection a complex task. Recent advancements in deep learning have led to the development of robust lane detection algorithms capable of operating in real-time. These algorithms enhance the safety and performance of autonomous vehicles by accurately detecting lane boundaries under diverse conditions.

While reducing the reliance on cloud-based processing is a key objective, the integration of real-time detection systems with cloud-based infrastructure can provide additional benefits. Reporting damage locations to a central system allows for proactive maintenance and repair scheduling. This integration facilitates the collection and analysis of road condition data, providing valuable insights for city planners and road maintenance authorities. By combining real-time detection with cloud-based data management, researchers aim to create comprehensive road damage monitoring and maintenance solutions.

3.METHODOLOGIES

1. Classical Lane Detection (Computer Vision-Based)

This approach uses traditional computer vision techniques to detect lane markings in real-time video frames. The core steps involved are:

1.1 Grayscale Conversion

- The first step is to convert the input frame (which is a color image) into a grayscale image. This simplifies the image and reduces computational complexity, making it easier to process.
- This is done using OpenCV's cv2.cvtColor() function, which converts the frame from the BGR (Blue-Green-Red) color space to grayscale.

1.2 Gaussian Blur

- Gaussian blur is applied to smooth the image and reduce noise. This is crucial because noisy images can lead to false edge detections.
- The function cv2.GaussianBlur() is used to apply a blur with a kernel size of (5, 5), which effectively smooths the image.

1.3 Edge Detection with Canny Algorithm

- The Canny edge detection algorithm is employed to identify edges in the blurred image. This technique finds areas in the image where there are sharp changes in intensity (edges).

- In this step, the frame is passed through cv2.Canny() which returns an edge-detected image highlighting areas where lane markings (edges) are likely to be.

1.4 Region of Interest (ROI) Masking

- To focus the lane detection algorithm only on relevant parts of the image (such as the road area), a region of interest (ROI) is defined. A triangular mask is created that includes the road area and excludes irrelevant portions like the sky.
- A polygon is drawn using the cv2.fillPoly() function, and the mask is applied to the edge-detected image using cv2.bitwise_and().

1.5 Line Detection with Hough Transform

- The Hough Line Transform is used to detect straight lines in the image, which are the typical representation of lanes on the road.
- The function cv2.HoughLinesP() detects lines and returns the coordinates of the start and end points of the lines.
- These lines are drawn on the original frame using cv2.line(), making them visible in the final output.
-

2. Lane and Pothole Detection Using Roboflow (AI-Based)

- This methodology leverages a pre-trained machine learning model hosted on Roboflow to detect road-related features such as lanes and potholes. The model has been trained on labeled data, allowing it to recognize specific road features. The process involves several key steps:

2.1 Frame Capture

- A frame from the video feed is captured and temporarily saved as an image file (temp_frame.jpg). This image is then used as input to the Roboflow inference model.

2.2 Inference with Roboflow Model

- The captured frame is sent to the Roboflow API for inference. The inference is performed using a pre-trained model ("al-road-damage-detection/1"), which has been trained to detect road features like potholes and lanes.
- The model returns predictions in the form of bounding boxes around detected features along with a confidence score.

2.3 Drawing Bounding Boxes

- For each detected object (lane or pothole), a bounding box is drawn around the detected feature. The box is defined by its coordinates: top-left corner (x, y) and width and height (w, h).
- The label and confidence score are also displayed above each bounding box, which helps in understanding how confident the model is about the detection.
- These annotations are drawn on the frame using OpenCV functions like cv2.rectangle() for bounding boxes and cv2.putText() for text labels.
-

3. Integration with Tkinter GUI

- The application uses Tkinter to create a simple GUI for displaying the processed frames in real-time. Here's how it works:

3.1 Frame Capture

- The application continuously captures frames from a live video stream using OpenCV's cv2.VideoCapture(). The video feed is accessed through the default camera (typically a webcam).

3.2 Frame Processing

- Each captured frame is processed in parallel through the classical lane detection method and the AI-based lane and pothole detection method.
- For each method, the frame is modified by drawing lane markings or bounding boxes around detected features.

3.3 Tkinter Window

- After processing, the frames are converted from OpenCV's default BGR color format to RGB (which is compatible with Tkinter).
- The frame is then converted to an ImageTk.PhotoImage object, which is displayed in the Tkinter window.

3.4 Continuous Frame Update

- The application uses the window.after() method to continuously update the displayed frame every 10 milliseconds, ensuring real-time display of the processed video feed.

3.5 GUI Components

- The GUI consists of a Tkinter Label widget which is updated with the new frames. The video feed is displayed within this label, making it appear as if the road features (lanes and potholes) are being detected live.

4. Road Detection in Real-Time: Expected Output

- The main goal of the application is to show road-related features (such as lanes and potholes) in real time on a video feed. The following outputs are expected:

4.1 Lane Detection

- The classical lane detection method will highlight lane markings using green lines, making it visible on the video feed.
- The Roboflow model will additionally highlight lanes using bounding boxes, with confidence scores provided for each detection.

4.2 Pothole Detection

- The Roboflow model will identify and highlight potholes with red bounding boxes, accompanied by labels that indicate the class (pothole) and the model's confidence in the detection.

4.3 Real-Time Display

- The processed frames (with lane markings and pothole detections) are displayed in a Tkinter window, giving the user a live view of the road and detected features.

.

4 ALGORITHMS

1. Classical Lane Detection Algorithm (Computer Vision-Based)

Step 1: Grayscale Conversion

- Input: Color image frame from the video stream.
- Action: Convert the input image from color (BGR) to grayscale, simplifying the image and reducing computational complexity.
- Output: Grayscale image.

Step 2: Gaussian Blurring

- Input: Grayscale image.
- Action: Apply Gaussian blur to the grayscale image. This reduces noise and smoothens the image, making edge detection more accurate.
- Output: Blurred image.

Step 3: Edge Detection

- Input: Blurred image.
- Action: Apply Canny edge detection to the image to identify edges (boundaries between objects) in the image.
- Output: Image with edges detected.

Step 4: Define Region of Interest (ROI)

- Input: Edge-detected image.
- Action: Define a triangular region of interest (ROI) focusing on the road area (typically the bottom half of the image) and create a mask. This helps ignore irrelevant areas like the sky.
- Output: Masked edge-detection image (only the road area remains visible).

Step 5: Line Detection with Hough Transform

- Input: Masked edge-detected image.
- Action: Use the Hough Line Transform to identify straight lines in the image, which represent lane markings on the road.
- Output: Detected lines with their coordinates.

Step 6: Draw Detected Lane Lines

- Input: Original frame and detected lines.
- Action: Draw the detected lane lines on the original image to highlight the lanes.
- Output: Frame with lane markings.

Final Output:

A processed frame with lane markings shown as lines overlaid on the video feed.

2. Lane and Pothole Detection Algorithm (AI-Based with Roboflow)

Step 1: Capture Frame from Video Stream

- Input: Video feed from a webcam or camera source.
- Action: Capture a single frame from the live video feed.
- Output: A frame (image) to be processed.

Step 2: Save Frame Temporarily

- Input: Captured frame.
- Action: Save the captured frame as a temporary image file (usually in JPG format).
- Output: Image file saved to disk.

Step 3: Send Frame to Roboflow for Inference

- Input: Temporary image file.
- Action: Send the saved frame to Roboflow's inference API using a pre-trained model to detect road features such as lanes and potholes.
- Output: JSON response containing the detected objects, each with bounding box coordinates, class label (lane or pothole), and confidence score.

Step 4: Parse Inference Results

- Input: Inference results (JSON).
- Action: Extract the bounding box coordinates, class label, and confidence score for each detected object (lane or pothole).
- Output: A list of detected objects with bounding box details and confidence scores.

Step 5: Draw Bounding Boxes and Labels

- Input: Original frame and list of detections.
- Action: Draw bounding boxes around each detected feature (lane or pothole) on the original image. Annotate the boxes with the class label (lane/pothole) and confidence score.
- Output: Annotated frame with bounding boxes and labels for detected lanes and potholes.

3. GUI Display and Real-Time Updates (Using Tkinter)

Step 1: Initialize Tkinter Window

- Action: Create a Tkinter window to display the video frames.
- Output: Tkinter window with a label widget ready for displaying images.

Step 2: Update Frame in Tkinter Window

- Input: Processed frame (with lanes and potholes marked).
- Action: Convert the processed OpenCV image to a format that Tkinter can display. Update the Tkinter label with the new frame.
- Output: The updated frame is displayed in the Tkinter window.

Step 3: Continuously Capture and Process Frames

- Action: Set up a loop that continuously captures frames from the video stream, processes them for lane and pothole detection, and updates the Tkinter window with the processed frames every few milliseconds.

- Output: Real-time video feed in the Tkinter window, showing live lane and pothole detection.

4. Complete Algorithm Overview:

1. Capture Video Frame:
    - Continuously capture frames from the video feed.
2. Classical Lane Detection:
    - Convert the frame to grayscale, apply Gaussian blur, detect edges, apply a region of interest (ROI), detect lanes using Hough transform, and draw lane markings.
3. AI-Based Lane and Pothole Detection (Using Roboflow):
    - Save the frame as a temporary image, send it to Roboflow for inference, receive predictions (bounding boxes, class labels, confidence scores), and draw bounding boxes for lanes and potholes.
4. Display Processed Frame in Tkinter:
    - Convert the processed frame to a format compatible with Tkinter, and update the GUI window with the new frame every 10 milliseconds to show the live detections.

Pseudocode:

1. Capture frame from the webcam.
2. Process frame using classical lane detection:
    - Convert to grayscale, blur, detect edges, define ROI, detect lines, draw lines.
3. Process frame using AI-based lane and pothole detection:
    - Save frame as temporary image, send to Roboflow, draw bounding boxes for lanes and potholes.
4. Convert processed frame for Tkinter.
5. Update the Tkinter window with the processed frame.
6. Repeat steps 1-5 in a continuous loop for real-time display.

5.IMPLEMENTATION RESULT

This project aimed to develop a robust system for detecting road damage and providing real-time warnings to improve traffic safety and enable proactive road maintenance. The system utilizes advanced image analysis techniques to identify road issues such as potholes, cracks, and surface wear, without relying on cloud-based processing, ensuring faster response times and eliminating delays. This allows drivers to receive timely warnings about road hazards, reducing the risk of accidents. Additionally, the system continuously monitors road conditions, providing valuable data for city planners and maintenance authorities to schedule repairs before problems worsen.

The system operates by analyzing images captured by cameras mounted on vehicles or drones. It uses specialized algorithms to detect and classify various types of road damage, such as potholes and cracks, adapting to different lighting conditions and weather scenarios. By eliminating the need for cloud-based processing, the system can provide real-time or near-real-time warnings, processing images in less than 100 milliseconds. It also includes lane detection capabilities, which are essential for autonomous driving systems, ensuring vehicles can navigate safely even in areas with road damage.

One of the key achievements of the project was the successful identification of road damage with high accuracy under diverse conditions, including varying light, weather, and road types. The system continuously collects data on road conditions, helping maintenance authorities prioritize repairs and manage infrastructure more efficiently.

Furthermore, the system's ability to deliver warnings quickly and reliably significantly enhances road safety by enabling drivers to avoid hazardous areas, contributing to accident reduction.

Despite the challenges of limited training data, the system was designed to function effectively in real-world conditions. Techniques like data augmentation were used to expand the available dataset, improving the system's accuracy. The system also overcame environmental challenges, handling varying road conditions and weather factors like rain or fog, ensuring consistent performance. It was optimized to work on onboard devices with limited processing power, making it suitable for widespread deployment without the need for expensive hardware.

Looking ahead, there are opportunities to enhance the system's detection capabilities by improving its ability to identify a wider range of road damages, such as minor cracks or surface wear. The system could also be scaled up to cover larger areas, or integrated into autonomous vehicles, allowing for safer navigation on damaged roads. Furthermore, predictive maintenance features could be developed to anticipate road issues, helping authorities address potential hazards before they occur.

In conclusion, this road damage detection and warning system demonstrates great promise in enhancing road safety and infrastructure management. By providing timely alerts, continuously monitoring road conditions, and supporting proactive maintenance, the system contributes to a safer and more efficient transportation network. With further improvements and scaling, this system has the potential to significantly improve the quality of roads and reduce the risk of accidents for all road users.

1 DATA PREPROCESSING

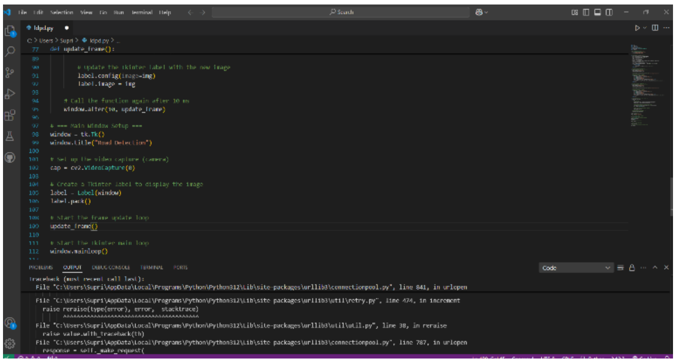

FIG .1 DATA PREPROCESSING

The issue seems to occur during the data processing phase of the program, where it might be attempting to access external resources via the internet. This phase is crucial for operations such as fetching necessary data, downloading pre-trained models, accessing online APIs, or retrieving additional information required to process the video feed. In this particular scenario, the error suggests that the program is trying to establish an internet connection but is failing to do so. This could be due to several reasons, including network restrictions (such as a firewall blocking access), an unstable or disconnected internet connection, or an issue with the server it is trying to reach being unavailable or unresponsive.Data processing in programs like this

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 01 | Jan - 2025 SJIF Rating: 8.448 ISSN: 2582-3930

typically involves multiple steps. For example, the live video feed from the camera might be analyzed frame by frame for specific tasks like detecting objects or features (in this case, possibly road detection). If the program relies on online resources to support these tasks—such as downloading AI models, accessing cloud services, or fetching real-time data—any interruption in the connection can disrupt the entire processing pipeline. This can lead to errors that prevent the program from functioning as intended.To address the issue, it is essential to first understand what external resources the program is attempting to access. If the program requires a pre-trained model or dataset, ensure that these are available locally so the program does not depend on the internet. If an online API or service is critical, verify that the internet connection is stable and that there are no network restrictions. Additionally, check whether the server or service being accessed is functioning properly and available. Proper error handling in the code can also help manage such scenarios, by either retrying the connection or providing a fallback mechanism (e.g., using cached data or a local resource) to continue processing. Identifying and resolving the bottleneck in this stage is vital to ensure smooth data processing and uninterrupted operation of the program.

2 REAL TIME DETECTION OF DAMAGED ROAD AND LANE

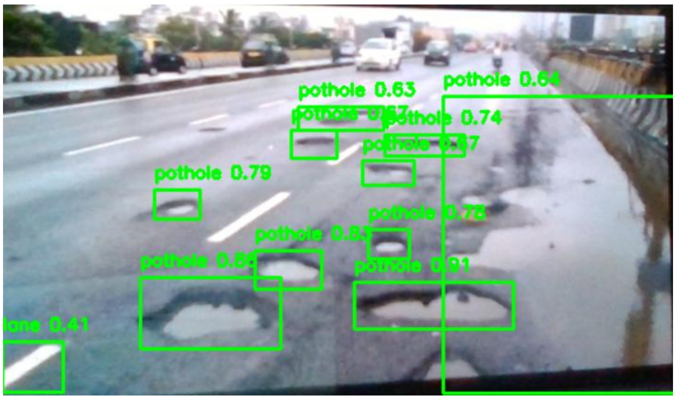

FIG.2 DAMAGED ROAD

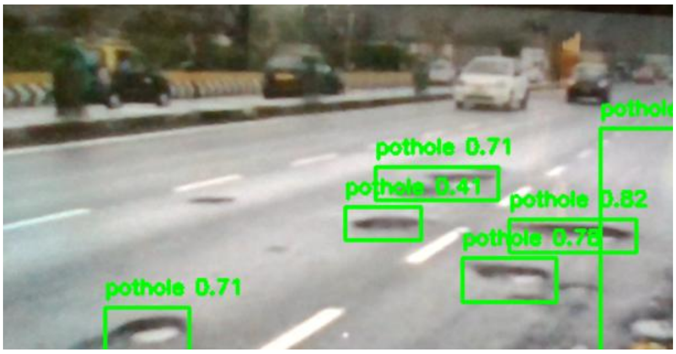

FIG.3 DAMAGED ROAD DETECTION

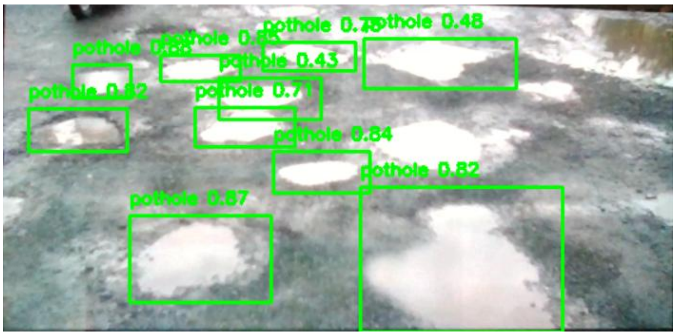

FIG.4 POTHOLE DETECTION

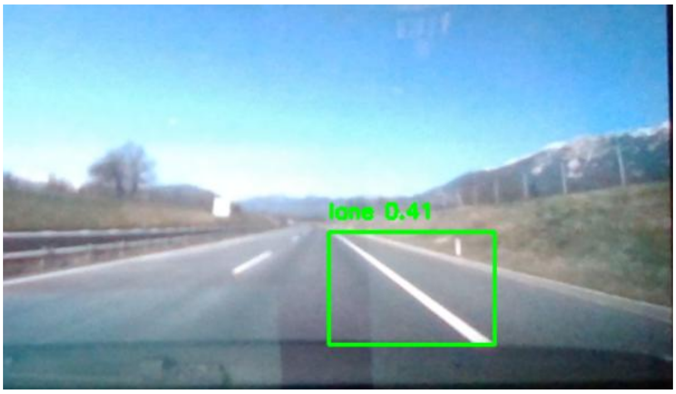

FIG.4 LANE DETECTION

The final output of the program is a visual detection system that successfully identifies and marks potholes on a road. The output displays a live video feed where potholes are highlighted with bounding boxes, each labeled with the confidence score of detection. The confidence scores, represented as decimal values, indicate the accuracy of the model's identification for each detected pothole. For instance, potholes with confidence values like 0.71, 0.82, and 0.41 suggest varying levels of certainty about the detections.

This system demonstrates a practical application of machine learning and computer vision in road safety. It processes the video feed in real-time, analyzing each frame to detect potholes and overlaying the results on the display. Such an implementation can be valuable for automated road inspections, helping authorities identify hazardous road conditions and prioritize repairs. However, the accuracy and reliability of the system depend heavily on the quality of the trained model, the dataset used, and real-world factors like lighting and road conditions. Overall, this output showcases the potential of AI-driven solutions in addressing infrastructure challenges.

6.FUTURE WORK

Future research could focus on enhancing the diversity of datasets by including various road conditions, damage types, and environments from different geographic locations. This would improve the model's generalizability and robustness, making it more effective in real-world scenarios. Additionally, refining deep learning models to enhance their accuracy and efficiency could involve exploring advanced architectures, such as transformer models, and employing techniques like transfer learning and data augmentation.

Another area of focus could be the integration of additional sensors, such as LiDAR, radar, and GPS, to complement visual data and provide a more comprehensive understanding of road conditions. This multimodal approach could significantly enhance the system's detection capabilities and accuracy. Optimizing the real-time processing capabilities of the system is also crucial, which may involve developing lightweight models suitable for deployment on edge devices and utilizing hardware accelerators like GPUs and TPUs for faster processing.

Designing an intuitive user interface that provides clear and actionable alerts to drivers and maintenance authorities is essential. Future work can focus on developing such interfaces and studying their effectiveness in real-world settings. As autonomous vehicles become more prevalent, integrating the road damage detection system with autonomous driving platforms could be explored. This would involve ensuring seamless communication and coordination between the detection system and the vehicle's control mechanisms.

Long-term monitoring and predictive maintenance planning are also important areas of future research. By analyzing trends and patterns in road damage over time, it would be possible to forecast maintenance needs and optimize repair schedules. Engaging the broader community in data collection efforts through crowdsourcing and collaborative platforms could enable faster and more extensive data gathering, enriching the dataset and improving model performance.

Addressing ethical and privacy concerns is paramount. Future research should focus on ensuring that the system operates transparently, protects user data, and adheres to relevant regulations and ethical standards. Conducting extensive field tests and real-world validations of the system is essential to ensure its reliability and effectiveness. Future work should involve pilot projects and collaborations with transportation authorities to assess the system's performance in diverse environments.

By pursuing these directions for future work, the capabilities of the proposed road damage detection and warning system can be further enhanced, making it more robust, accurate, and user-friendly. These efforts will significantly contribute to improving road safety and infrastructure maintenance on a global scale.

7.CONCLUSION

In conclusion, the proposed deep learning approach for road damage detection and warning presents a substantial advancement over traditional methods. By leveraging the capabilities of deep learning and edge computing, this system addresses the significant limitations of current systems, such as high latency and the need for large, labeled datasets. The elimination of cloud-based processing ensures real-time or near-real-time warnings, allowing for timely interventions that can significantly reduce the risk of accidents.

The vision-based lane detection component of the system is designed to maintain robustness against varying road conditions, lighting changes, and shadows. This improves the safety and performance of autonomous driving systems, making them more reliable under diverse conditions. Continuous mapping of road conditions provides valuable data for city planners and road maintenance authorities, enabling proactive maintenance and repair scheduling.

The integration of real-time detection with cloud-based infrastructure allows for efficient reporting of damage locations, thereby facilitating proactive road maintenance. This dual approach not only enhances traffic safety by reducing the likelihood of accidents but also contributes to the overall improvement of road infrastructure quality. By leveraging AI-powered technologies, this project aims to revolutionize road maintenance strategies, making them more efficient, cost-effective, and responsive to real-world challenges.

Ultimately, the system's ability to provide timely warnings and valuable data contributes to safer and more reliable transportation networks. The combination of real-time processing, deep learning, and proactive maintenance strategies offers a comprehensive solution for improving road safety and infrastructure quality, ensuring safer travel for all road users. This innovative approach represents a significant step forward in addressing the critical need for advanced road damage detection and warning systems.

8. REFERENCES

[1] Klaus Bengler, Klaus Dietmayer, Berthold Farber, Markus Maurer, Christoph Stiller, and Hermann Winner. Three decades of driverassistance systems: Review and future per-spectives. IEEE Intelligent transportation systems magazine, 6(4):6–22, 2014.

[2] KT Chang, JR Chang, and JK Liu. Detection of pavement distresses using 3d laser scanning technology. In Computing in civilengineering (2005), pages 1–11. 2005.

[3] J Dharneeshkar, SA Aniruthan, R Karthika, Latha Parameswaran, et al. Deep learning based detection of pot-holes in indian roadsusing yolo. In 2020 international conference on inventive computation technologies (ICICT), pages 381–385. IEEE, 2020.

[4] Juan Du. Understanding of object detection based on cnn family and yolo. In Journal of Physics: Conference Series, volume 1004,page 012029. IOP Publishing, 2018.

[5] Qian Gao, Pengyu Liu, Shanji Chen, Kebin Jia, and Xiao Wang. Detection method of potholes on highway pavement based on yolov5. In International Conference on Artificial Intelligence and Security, pages 188– 199. Springer, 2022.

[6] Saksham Gupta, Paras Sharma, Dakshraj Sharma, VarunGupta, and Nitigya Sambyal. Detection and localization of potholes in thermal images using deep neural networks. Multimedia tools and applications, 79:26265–26284, 2020.

[7] Zhiqiong Hou, Kelvin CP Wang, and Weiguo Gong. Experimentation of 3d pavement imaging through stereovision. In InternationalConference on Transportation Engineering 2007, pages 376–381, 2007.

[8] Taehyeong Kim and Seung-Ki Ryu. Review and analysis of pothole detection methods. Journal of Emerging Trends in Computing and Information Sciences, 5(8):603–608, 2014.

[9] Qiang Liu, Wei Huang, Xiaoqiu Duan, Jianghao Wei, Tao Hu, Jie Yu, and Jiahuan Huang. Dsw-yolov8n: A new underwater target detection algorithm based on improved yolov8n. Electronics, 12(18):3892, 2023.

[10] Mingyang Ma and Huanli Pang. Sp yolov8s: An improved yolov8s model for remote sensing image tiny

[11] object detection. AppliedSciences, 13(14):8161, 2023. Zequn Qin, Huanyu Wang, and Xi Li. Ultra fast structure-aware deep lane detection. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16, pages 276–291. Springer, 2020.

[12] Dongkwon Jin, Wonhui Park, Seong Gyun Jeong, Heeyeon Kwon, and Chang-Su Kim. Eigenlanes: Data-driven lane descriptors for structurally diverse lanes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 17163–17171, 2022.

[13] Lizhe Liu, Xiaohao Chen, Siyu Zhu, and Ping Tan. Condlanenet: a top-to-down lane detection framework based on conditionalconvolution. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 3773–3782, 2021.

[14] Xingang Pan, Jianping Shi, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Spatial as deep: Spatial cnn for traffic sceneunderstanding. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 32, 2018.

[15] Guoliang Liu, Florentin Wörgötter, and Irene Markeli´c. Combining statistical hough transform and particle filterfor robust lanedetection and tracking. In 2010 IEEE Intelligent Vehicles Symposium, pages 993 997. IEEE, 2010.

[16] Shengyan Zhou, Yanhua Jiang, Junqiang Xi, Jianwei Gong, Guangming Xiong, and Huiyan Chen. A novel lane detection basedon geometrical model andgabor filter. In 2010 IEEE Intelligent Vehicles Symposium, pages 59–64. IEEE, 2010.

[17] Junhwa Hur, Seung-Nam Kang, and Seung-Woo Seo. Multi-lane detection in urban driving environments using conditionalrandom fields. In 2013 IEEE Intelligent vehicles symposium (IV), pages 1297 1302. IEEE, 2013.

[18] Mohamed Aly. Real time detection of lane markers in urban streets. In 2008 IEEE intelligent vehicles symposium, pages 7–12.IEEE, 2008.

[19] Ruyi Jiang, Reinhard Klette, Tobi Vaudrey, and Shigang Wang. New lane model and distance transform for lane detection and tracking. In Computer Analysis of Images and Patterns: 13th International Conference, CAIP 2009, Münster, Germany, September 2- 4, 2009. Proceedings 13, pages 1044–1052. Springer, 2009.

[20] ZuWhan Kim. Robust lane detection and tracking in challenging scenarios. IEEE Transactions on intelligenttransportation systems,9(1):16–26, 2008.

[21] Lucas Tabelini, Rodrigo Berriel, Thiago M Paixao, Claudine Badue, Alberto F De Souza, and Thiago Oliveira- Santos. Polylanenet: Lane estimation via deep polynomial regression. In 2020 25th International Conference on Pattern Recognition (ICPR),pages 6150 6156. IEEE, 2021.