Smart Roadway Monitoring: Pothole Detection and Mapping via Google Street View

Shazab Ali, Meng Xu, and Daehan Kwak(✉)

Department of Computer Science and Technology, Kean University, Union, USA
{alisha,mengxu,dkwak}@kean.edu

Abstract. Potholes pose significant financial and safety hazards to motorists worldwide, emphasizing the demand for innovative solutions for detection and repair. Conventional methods, reliant on manual inspection and patching, prove to be inefficient and unsustainable, prompting the need for automated detection systems. However, merely expediting the patching process does not address the underlying issues that cause the potholes in the first place. This paper introduces a pothole detection and mapping system over Google Street View, utilizing highly effective learning models and Google Map's APIs. Our system extracts images along specified routes from the Google Street View API, processes them using a detection model, and plots the results on an interactive map. Additionally, it compiles these findings into a video that simulates a drive along the route. By leveraging deep learning techniques, we provide users with valuable insights into road conditions, facilitating proactive maintenance strategies. The evaluation demonstrates high classification accuracy and sensitivity in pothole detection. Additionally, the system's capacity to analyze data over time enables municipalities to identify and pinpoint persistent pothole-prone areas, paving the way for targeted interventions to prevent future hazards. Future work includes expanding the dataset and developing a user-friendly interface to enhance the system's capabilities and usability. Our system offers a promising solution for long-term pothole repair and maintenance, contributing to safer and more sustainable transportation infrastructure for communities around the world.

Keywords: Deep learning · Google Street View · Pothole detection · Pothole mapping · Roadway monitoring system

1 Introduction

Potholes are holes in road surfaces that result from gradual damage caused by traffic and weather. Asphalt, commonly used in road surfaces for its durability, flexibility, and low cost, paradoxically becomes a breeding ground for potholes. Potholes arise in four main steps [6]. Moisture infiltrates the asphalt which allows

H. R. Arabnia et al. (Eds.): CSCE 2024, CCIS 2260, pp. 151–163, 2025.
https://doi.org/10.1007/978-3-031-85923-6_12

water to drain into it allowing for the moisture to become trapped underneath. During freezing and thawing, ice expands and causes cracks within the asphalt. Voids form as hollow spaces accumulate moisture and rapid freeze-thaw cycles. With the impact of traffic, the pavement begins to collapse, causing potholes to become larger and larger overtime.

The damage caused by potholes may seem trivial, but the extent of the damage they cause is significant, both financially and physically. A study investigating the impact of potholes on motorists across 11 distinct regions in the UK between 2019 and 2020 revealed that the cumulative damage inflicted on motorists annually exceeded 1.25 billion dollars [6]. Furthermore, the study [6] indicated that 32% of drivers who encountered potholes in the past year reported sustaining damage to their vehicles. These damages encompassed a range of issues, including but not limited to tire damage, wheel misalignment, and suspension repairs.

Potholes not only pose a financial burden, but also create significant safety hazards as well, with the potential for serious and fatal consequences. Road accidents attributed to potholes claimed 5,626 lives between 2018 and 2020 [21]. These accidents highlight the critical need for improved road maintenance and proactive measures. The intersection between financial burden and loss of life underscores the urgent need for effective pothole detection and mapping systems. By prioritizing these systems, we can enhance road safety, facilitate timely repairs, and reduce the economic and safety impact.

Traditional methods of pothole detection and repair rely on manual inspection and patching, both of which are inefficient. Patching, a common practice, involves filling potholes with temporary asphalt, however, this "quick fix" does not last long and constantly requires refilling especially in areas that receive bad weather or traffic frequently. Alternatively, methods such as reconstruction and overlays [9] provide long-term solutions by replacing entire sections of asphalt. However, the cost associated with these approaches limits their capability of widespread implementation.

In this paper, we introduce a pothole detection and mapping system leveraging well-established deep learning models and Google Maps APIs. Our system operates along a route of interest and extracts images from the Google Street View API, which are then analyzed using a pothole detection model. The detected potholes are plotted on a map before being converted into video. This facilitates both visual and statistical assessment of road conditions in a given area over time. This approach allows us to pinpoint the areas where there is recurring road damage and implement targeted reinforcement to prevent potholes from forming. By eliminating the need for constant road surveying and patching, our system aims to mitigate both injury and financial burdens caused by pothole-related incidents.

The paper is organized as follows. Section 2 introduces the related work to our approach and potential alternatives, and Sect. 3 discusses and explains the methodology of our proposed approach. Section 4 delves into the implementation and application, and Sect. 5 summarizes our results. Section 6 explains the challenges faced and the resulting output, and Sect. 7 summarizes our approach and discusses future work.

2 Related Work

Advancements in computer vision [20,22,23,27] and intelligent transportation systems [7,8,15] have led to more sophisticated methods for monitoring roadway conditions and supporting autonomous decision-making. Numerous studies have concentrated on refining models for enhanced pothole detection using image-based approaches [2,3,17,24,28]. At the same time, crowdsourcing has emerged as an invaluable approach for the real-time collection and dissemination of information across various domains [11–14,16]. Research has increasingly focused on combining these two methods to tackle real-world pothole detection. For example, several approaches leverage image-based machine learning algorithms alongside sensor technologies for pothole detection, while utilizing crowdsourcing mechanisms to disseminate pothole locations [10,19,26]. This dual approach not only aids in the precise detection of potholes but also allows real-time reporting and subsequent mitigation, therefore enhancing the practical implementation.

Studies in [10] and [19] have explored the option of using vehicle-mounted cameras in order to automate the process and improve detection accuracy. In [19], the You Only Look Once version 5 (YOLOv5) deep learning algorithm was used to detect potholes from dash camera images. These images served as input to CNN models within the YOLOv5 framework. During training, the CNN models were fine-tuned using transfer learning techniques. This involved leveraging pre-trained weights from models trained on larger datasets like COCO to expedite the learning process and improve performance. Next, three different architectures of YOLOv5 (small, medium, large) were evaluated during training of 500 epochs. The effectiveness of the YOLOv5 approach is demonstrated through its ability to detect potholes accurately and in real time. By analyzing the trade-offs between detection accuracy and speed across the different model sizes, the proposed solution offers flexibility in optimizing its performance based on the task presented.

YOLOv5 is not the only machine learning approach that has been used for this task. The study in [18] highlights a range of deep learning techniques suitable for pothole detection, composed of both object detection and semantic segmentation algorithms. The object detection methods range from single-shot multi-box detectors (SSD) to region-based convolutional neural networks (R-CNN) similar to what we employed in our experiment. In addition, semantic segmentation-based methods have also gained traction, utilizing networks such as U-Net and DeepLabv3+ to segment road images at the pixel level, providing us with a detailed understanding of road anomalies. These techniques leverage attention mechanisms and data fusion strategies to refine features and improve the accuracy of the segmentation.

Additionally, researchers have explored using crowd-sourcing for pothole detection and accelerometers to collect data on road conditions [18,26]. In [26], two groups of experiments were conducted utilizing smartphones in one group and high-precision devices in the other. In the smartphone group, three phones were positioned in different locations of the vehicle, with the high-precision device on the floor. Supervised learning models were then employed to identify the potholes, while the acceleration data was labeled using the ground truth from the windshield camera. The labeled data were used in the training of these models to recognize patterns in the acceleration data that indicated the appearance of road anomalies. The algorithms were fine-tuned using the collected data from both the experimental and verification groups to ensure their effectiveness in different driving conditions. This information would be uploaded by the users to further train the model around different road conditions.

It is important to note that although these approaches are good for detecting potholes, they lack any real-world application beyond monitoring and detecting. In addition, crowdsourcing from users is challenging to achieve and creates privacy and data authenticity concerns. To address these challenges, our system utilizes deep learning techniques and leverages Google Street View and Google Maps APIs to obtain real-world street-level road data. Additionally, we implement a mapping system to make use of the detections as a comprehensive data tool rather than a plain detection tool.

3 Methodology

3.1 Proposed System

Our proposed pothole detection system aims to provide users with a seamless and efficient way to not only detect potholes but monitor them as well. By combining powerful machine learning techniques and Google Street View imagery, the system offers users valuable insights into road conditions in their area and long-term data to prevent future hazards before they arise.

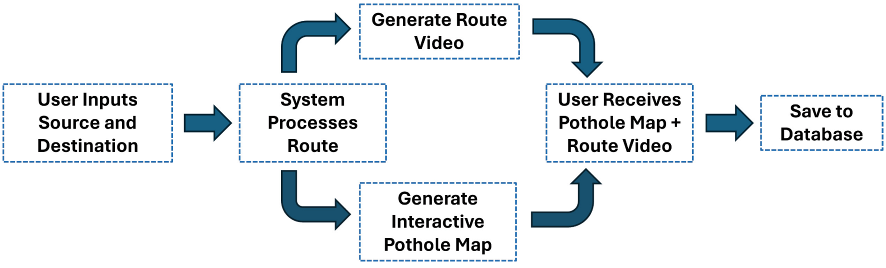


Fig. 1. Workflow of the pothole detection and mapping system.

3.2 System Implementation

The user flow of the proposed system is depicted in Fig. 1. The program starts with user input of the route of interest, where users can specify the source and destination. The system then takes this information and begins processing it using the Google Maps Directions and Street View APIs followed by a deep learning model, as seen in Fig. 1. Once the information is processed, an interactive map is generated that displays the location of the potholes at the exact coordinates where they were detected. In addition to the interactive map, the system compiles the detected potholes into a video simulation of the route for users to view. Lastly, the user receives the map and video and can store them in a database to monitor the road conditions over time in their town or area. Google Street View updates its imagery at varying frequencies, with urban areas typically updated once every year, while less populated areas may take three or more years [4]. This is beneficial for our system since we are not trying to simply gather road data or detect potholes for the current day but to gather data over time to prevent the potholes from forming in the first place. By leveraging these images and maps, within a few years, municipalities can discern patterns of road degradation and prioritize areas for appropriate intervention. For instance, areas exhibiting frequent and recurring pothole formation may warrant more extensive measures such as complete reconstruction and recurrence rather than just patching. As described earlier, reconstruction and reinforcement are expensive procedures and cannot be done everywhere, so utilizing our system, enables implementation in places where it is needed. Conversely, areas experiencing sporadic or isolated potholes will suffice with simpler solutions such as patching.

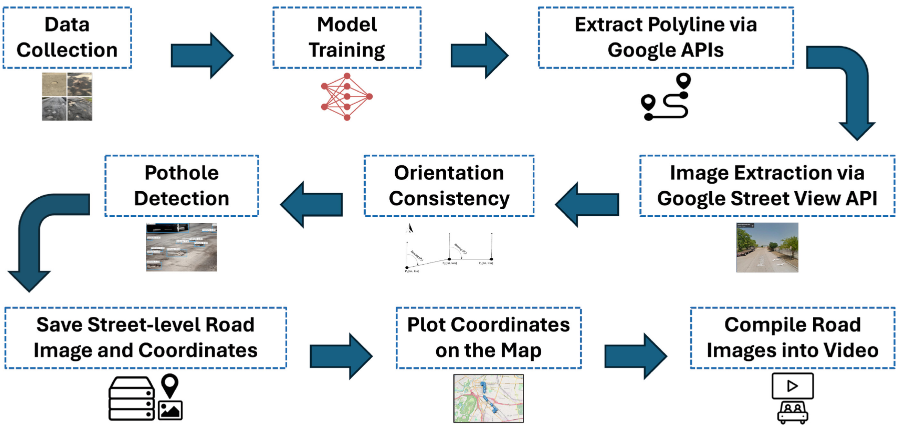


Fig. 2. System architecture of the pothole detection and mapping system.

3.3 System Architecture

Figure 2 illustrates the proposed systems architecture, while Fig. 1 depicts the user implementation. This highlights the various steps the data must go through to get the results. The first step in building the system's architecture was to collect a large dataset of potholes. A dataset used by a related paper that used deep learning to detect potholes which consisted of pothole images from various locations all around the world [5] was utilized. This dataset consisted of thousands of images; however, only the United States data was used as it was the closest to the type of images desired for detection. Additionally, the dataset was augmented with our images to enhance its size and diversity. The augmented images were from routes that were set from different parts of the region such as Wayne, New Jersey, to Newark, New Jersey, then the images were extracted and annotated manually using Roboflow [1].

Following data collection, we trained a Detectron2 model using the RCNN R50 FPN architecture from the model zoo [25]. This model was chosen due to its efficiency and relatively high accuracy compared to other available methods and models. The Feature Pyramid Network (FPN) incorporated into the architecture enables effective feature extraction allowing the model to capture both fine-grained details and broader contextual information in the training images. The ResNet-50 backbone provides a balance between the model efficiency and complexity making it suitable for this application where speed is an important factor. Overall, the choice of the RCNN R50 FPN architecture was because of our emphasis on achieving both high accuracy and efficiency in pothole detecting tasks.

Next, utilizing the Google Maps Directions API and the polyline library in Python, a polyline representation of the route of interest entered by the user was extracted. This ensured all the data points along the route were captured. We then employed the Google Maps Street View API to extract images at each data point along the route. Additionally, the bearing between each current and previous point was calculated to ensure the consistent orientation of the captured images. This step was crucial because without it, all the images would be facing different directions, making the last step of video compilation impossible but most importantly, preventing the detection of potholes if the orientation were not facing the direction of the road in progress. For example, without consistent bearings, the orientation of the street view image might face a home or the side of the road instead of the road itself, leading to inaccurate and unusable data.

With the images extracted, they were run through the model as they arrived. This process allowed for the detection of potholes as they were encountered and saved the coordinates in a CSV file. Running the images through the model in this order prevented duplicate images of detected and undetected potholes and eliminated the need to run the program twice to obtain the coordinates.

Upon saving of all the images and coordinates, the CSV file that stored the pothole coordinates was used to create the map. We used the folium library to create an interactive map, as seen in Fig. 6, to visualize the potholes along the route. Lastly, all the saved images were compiled into a video using OpenCV, with the addition of interpolation between the frames to give the illusion of a seamless driving experience as it progressed through the route.

4 Experiments

4.1 Dataset

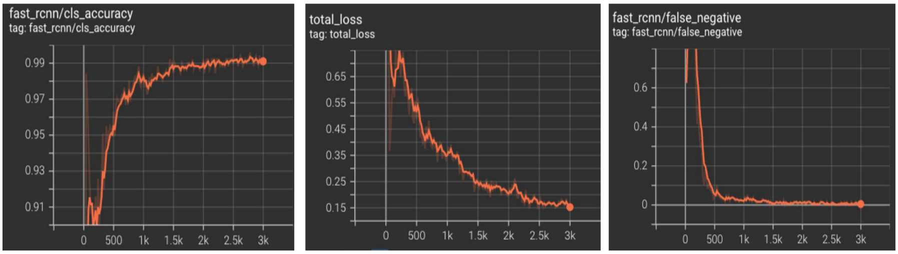
The dataset used in our experiment comprises of street-level images obtained from the Google Street View Static API [5]. These images were captured under sunny conditions but at different times of the day and with varying sunlight intensities. The dataset consists of 1,977 images in the training set and 50 images in the test set, totaling 2,027 images. No separate validation set was used in this study, as it was not the scope of this research. Image annotations were provided in COCO format, allowing compatibility with our deep learning framework. Each image was annotated with bounding boxes delineating the location of the potholes. Prior to model training, images were preprocessed to ensure uniformity and compatibility with the model. Images were resized to a resolution of 640×640 pixels. A single class label, "pothole", was assigned to all annotated images, indicating the presence of a pothole within a bounding box region. In this study, data augmentations such as exposure and saturation adjustments were not applied to the dataset. We hope to test this in the future to enhance the model's performance.

4.2 Implementation Details

The training process was conducted using the Detectron2 library, leveraging the Faster R-CNN architecture with a ResNet-50 backbone pre-trained on the COCO dataset. The implementation details are as follows: The learning rate was set to 0.0025; the batch size was 256; the model was trained for 3,000 iterations; no preprocessing was done as this was completed while annotating the images; lastly, the optimizer that was used during training was stochastic gradient descent (SGD). The training was conducted in a GPU-enabled environment in Google Colab to accelerate computation. After training, the model's performance was evaluated on the test set using a detection threshold of 0.8.

5 Results

Figure 3 highlights the performance of the model in terms of classification accuracy, total loss, and false negatives, as expressed through the TensorBoard library. The classification accuracy peaked at 91% by the end of the training process. This demonstrates the model's high capability of predicting the classification of potholes correctly. In Fig. 3c, we can see the total loss of the model starting at 0.65 and bottoming out at around 0.15 by the end. This difference between the beginning and the end indicates that the model was learning throughout the training process and no errors occurred during training. Lastly, the false negative rate was initially high but neared zero at the end. This decrease reflects



(a) Accuracy (b) Total Loss (c) False Negative

Fig. 3. Model performance in terms of accuracy, loss, and false negatives.

the model's enhanced sensitivity to detecting potholes, which is important for our task. Overall, these graphs collectively demonstrate the good health of the model and its effectiveness for the task.

(a) Labeled ground truth (b) Detected image

Fig. 4. Comparison of ground truth and system detection for pothole identification.

We selected a challenging route for a case study to test the effectiveness of the system. The chosen route was from 42 Maryland Ave, Paterson, New Jersey to 26 19th Ave, Paterson, New Jersey. This route was specifically selected because it is known to have a dense population of potholes, which is an ideal environment for evaluating the performance of the system. To establish the ground truth of this area, we extracted images from the polyline using the Google Maps Street

View API and Directions API and manually labeled the route. This was done to ensure the accuracy of our evaluation.

Figure 4 represents a side-by-side comparison between a snapshot from the ground truth video Fig. 4a and the snapshot obtained from our system's detection Fig. 4b. Despite the large population of potholes in the area, the system only missed two predictions and detected one false positive compared to the ground truth.

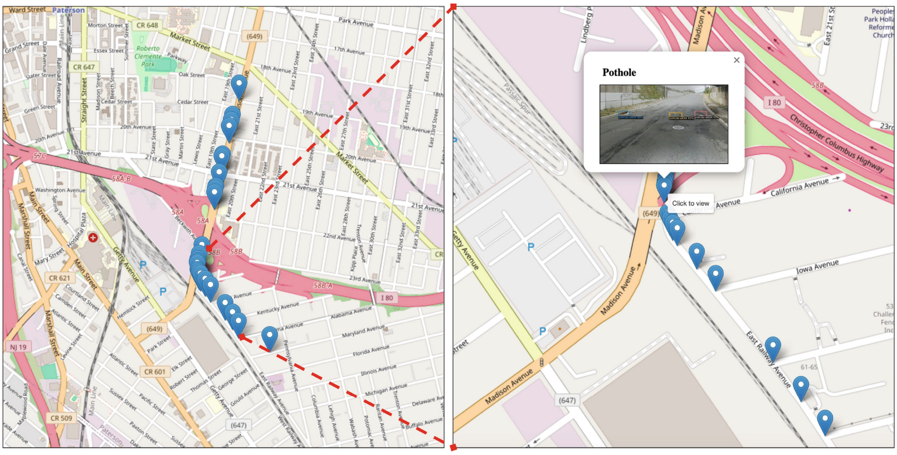


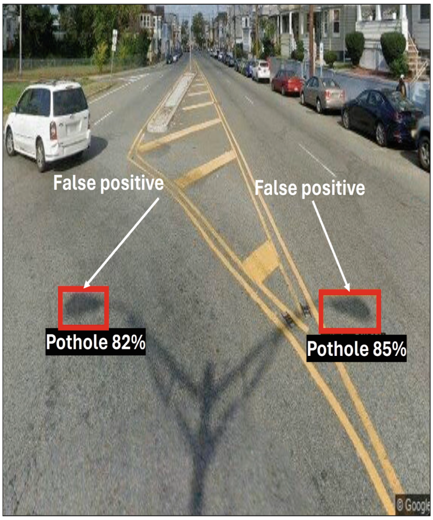
Fig. 5. Pothole detection mapping system. This figure provides a visual representation of pothole distribution along the route of interest, offering valuable insight into the road's condition.

Our proposed system's mapping layout is illustrated in Fig. 5, which provides a visual representation of pothole distribution along the route of interest, offering valuable insight into the road's condition. When a request is sent to the Directions API to create a polyline of the route, a JSON file containing the metadata of the latitude and longitude coordinates is generated. The Google Street View Static API reads these coordinates and returns street-level road images. When the image is run through the model to detect a pothole, if a pothole is detected, the coordinates are saved in a CSV file. However, if nothing is detected, the metadata is discarded even though the image is still saved. The coordinates in the CSV file indicate where the potholes are to be marked as pinpoints on the map, as seen in Fig. 5. Once this process is complete, the CSV file is saved and loaded into the Folium Python library, which marks the pothole positions on the map using the saved coordinates and creates an interactive map for the user to swipe through and analyze. The interactive map allows users to swipe through, zoom in and out, and click on any pinpoint location to view the actual potholes to conduct a more thorough investigation of the area of interest.

The map illustrates scattered pothole pinpoints at the beginning and end of the route, with a higher density concentration observed in the middle. This higher density indicates that the middle segment of the route receives a higher frequency of road deterioration, which could be due to several factors such as heavy traffic or bad weather. By identifying these segments, authorities can prepare these areas for additional measures such as reinforced construction or enhanced road surface materials. This addresses safety concerns for motorists and the financial burdens associated with frequent patching.

6 Challenges

While the overall accuracy along the case study route was high, there were a few instances where false positives were detected. For example, in Fig. 6a and 6b we see two instances where this occurred. In the left image, the system incorrectly identified the shadow of a light post as a pothole. Similarly, the image on the right shows a misclassification of a sewer lid as a pothole. This was likely due to us not augmenting the dataset to account for different exposures, which would help the model learn the features of potholes at different times of the day. Since the Google Maps Street View API provides images taken at various times of day and under different weather conditions, this variability influences the appearance of potholes, likely leading to these errors.



(a) Lamp shadow detected as pothole

(b) Sewer detected as pothole

Fig. 6. Examples of false positives in pothole detection.

7 Conclusion and Future Work

In conclusion, our system presents a promising solution for pothole detection and mapping using deep learning techniques and real-world street-level road data from Google Street View API. The system's ability to analyze data over time will allow municipalities to identify persistent pothole-prone areas, enabling proactive intervention before the potholes even occur. This will not only reduce repair frequency and cost but will also increase safety. Through continued innovation, we hope to create a safer and more sustainable transportation infrastructure for all communities.

Moving forward, we aim to enhance the system's capabilities by expanding and augmenting the pothole dataset to include a wider range of road conditions and lighting scenarios. We plan to be able to adjust the images to account for the discrepancies caused by the different times of day when Google Street View car collected its images. Additionally, we plan to develop a user-friendly GUI to streamline route and input data retrieval so users of all technological backgrounds can use it. Furthermore, enabling users to upload their data to a cloud server will allow municipalities to efficiently input and track their data over time.

Acknowledgment. This work was supported in part by NSF grant DUE-2247157, CNS-2318696, and the Office of Research and Sponsored Programs, Kean University.

References

1. Roboflow: Computer vision tools for developers and enterprises. https://roboflow.com/. Accessed 16 May 2024
2. Akagic, A., Buza, E., Omanovic, S.: Pothole detection: an efficient vision based method using RGB color space image segmentation. In: 2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). IEEE (2017). https://doi.org/10.23919/mipro.2017.7973589
3. Ansari, S.: Building a realtime pothole detection system using machine learning and computer vision. https://towardsdatascience.com/building-a-realtime-pothole-detection-system-using-machine-learning-and-computer-vision-2e5fb2e5e746. Accessed 16 May 2024
4. Antonelli, W.: It can take years for google maps to update certain features - here's how they get the data to update street view, traffic, and more (2021). https://www.businessinsider.com/guides/tech/how-often-does-google-maps-update. Accessed 06 Sept 2024
5. Arya, D., Maeda, H., Ghosh, S.K., Toshniwal, D., Sekimoto, Y.: RDD2022: a multi-national image dataset for automatic road damage detection (2022). https://doi.org/10.48550/ARXIV.2209.08538
6. Bučko, B., Lieskovská, E., Zábovská, K., Zábovský, M.: Computer vision based pothole detection under challenging conditions. Sensors **22**(22), 8878 (2022). https://doi.org/10.3390/s22228878

7. Dacayan, T., Ponte, E., Huang, K., Kwak, D.: Utilizing a spatial grid for automated parking space vacancy detection. In: 2023 International Conference on Computational Science and Computational Intelligence (CSCI). IEEE (2023). https://doi.org/10.1109/csci62032.2023.00169
8. Devarakonda, S., Chittaranjan, S., Kwak, D., Nath, B.: In: MobiQuitous 2020 - 17th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, MobiQuitous 2020, pp. 206–214. ACM (2020)
9. Hafezzadeh, R., Autelitano, F., Giuliani, F.: Asphalt-based cold patches for repairing road potholes - an overview. Constr. Build. Mater. **306**, 124870 (2021). https://doi.org/10.1016/j.conbuildmat.2021.124870
10. Hoseini, M., Puliti, S., Hoffmann, S., Astrup, R.: Pothole detection in the woods: a deep learning approach for forest road surface monitoring with dashcams. Int. J. For. Eng. **35**(2), 303–312 (2023). https://doi.org/10.1080/14942119.2023.2290795
11. Kwak, D., Kim, D., Liu, R., Iftode, L., Nath, B.: Tweeting traffic image reports on the road. In: Proceedings of the 6th International Conference on Mobile Computing, Applications and Services. MobiCASE (2014). https://doi.org/10.4108/icst.mobicase.2014.257815
12. Kwak, D., Kim, D., Liu, R., Nath, B., Iftode, L.: DoppelDriver: counterfactual actual travel times for alternative routes. In: 2015 IEEE International Conference on Pervasive Computing and Communications (PerCom), pp. 178–185. IEEE (2015). https://doi.org/10.1109/PERCOM.2015.7146525
13. Kwak, D., Liu, R., Kim, D., Nath, B., Iftode, L.: Seeing is believing: sharing real-time visual traffic information via vehicular clouds. IEEE Access **4**, 3617–3631 (2016). https://doi.org/10.1109/ACCESS.2016.2569585
14. Liu, R., et al.: Themis: a participatory navigation system for balanced traffic routing. In: 2014 IEEE Vehicular Networking Conference (VNC), pp. 159–166. IEEE (2014). https://doi.org/10.1109/VNC.2014.7013335
15. Liu, R., et al.: Balanced traffic routing: design, implementation, and evaluation. Ad Hoc Netw. **37**, 14–28 (2016). https://doi.org/10.1016/j.adhoc.2015.09.001
16. Liu, R., Yang, Y., Kwak, D., Zhang, D., Iftode, L., Nath, B.: Your search path tells others where to park: towards fine-grained parking availability crowdsourcing using parking decision models. Proc. ACM Interact. Mob. Wearable Ubiquit. Technol. **1**(3), 1–27 (2017). https://doi.org/10.1145/3130942
17. Liu, Z., Gu, X., Chen, J., Wang, D., Chen, Y., Wang, L.: Automatic recognition of pavement cracks from combined GPR B-scan and C-scan images using multiscale feature fusion deep neural networks. Autom. Constr. **146**, 104698 (2023). https://doi.org/10.1016/j.autcon.2022.104698
18. Mednis, A., Strazdins, G., Zviedris, R., Kanonirs, G., Selavo, L.: Real time pothole detection using android smartphones with accelerometers. In: 2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS), pp. 1–6 (2011). https://doi.org/10.1109/DCOSS.2011.5982206
19. Patel, R., Huang, L., Vejarano, G.: Pothole detection from dash camera images using yolov5. In: 26th International Conference on Image Processing, Computer Vision, & Pattern Recognition (IPCV 2022), Las Vegas, NV, USA (2022)
20. Ponte, E., Amparo, X., Huang, K., Kwak, D.: Automatic pill identification system based on deep learning and image preprocessing. In: 2023 Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE). IEEE (2023). https://doi.org/10.1109/csce60160.2023.00324
21. PTI: Accidents caused by potholes killed 5626 people during 2018–2020. https://www.tribuneindia.com/news/nation/accidents-caused-by-potholes-killed-5626-people-during-2018-2020-424252. Accessed 15 May 2024

22. Serrano, G., Kwak, D.: Real-time sign language recognition using computer vision and AI. In: 2023 International Conference on Computational Science and Computational Intelligence (CSCI). IEEE (2023). https://doi.org/10.1109/csci62032.2023.00198
23. Shahzad, M., Ali, F., Shirazi, S.H., Rasheed, A., Ahmad, A., Shah, B., Kwak, D.: Blood cell image segmentation and classification: a systematic review. PeerJ Comput. Sci. **10**, e1813 (2024). https://doi.org/10.7717/peerj-cs.1813
24. Wang, D., Liu, Z., Gu, X., Wu, W., Chen, Y., Wang, L.: Automatic detection of pothole distress in asphalt pavement using improved convolutional neural networks. Remote Sens. **14**(16), 3892 (2022). https://doi.org/10.3390/rs14163892
25. Wu, Y., Kirillov, A., Massa, F., Lo, W.Y., Girshick, R.: Detectron2 (2019). https://github.com/facebookresearch/detectron2
26. Xin, H., et al.: Sustainable road pothole detection: a crowdsourcing based multi-sensors fusion approach. Sustainability **15**(8), 6610 (2023). https://doi.org/10.3390/su15086610
27. Xu, M., Huang, J., Huang, K., Liu, F.: Incorporating tumor edge information for fine-grained BI-RADS classification of breast ultrasound images. IEEE Access **12**, 38732–38744 (2024). https://doi.org/10.1109/access.2024.3374380
28. Zhang, Y., et al.: Road damage detection using UAV images based on multi-level attention mechanism. Autom. Constr. **144**, 104613 (2022). https://doi.org/10.1016/j.autcon.2022.104613