Automated Road Damage Detection and Interactive Mapping Using YOLOv11, YOLOv12, and DeepSORT

Thomas Dermengea
Queen's University
22gq34@queensu.ca

Ethan Solnik
Queen's University
22kg37@queensu.ca

Anthony Grecu
Queen's University
22rpv4@queensu.ca

Edan Kroi
Queen's University
22qds@queensu.ca

Scott Doggett
Queen's University
20sjd7@queensu.ca

Kiarash Soleimaniroozbahani
Queen's University
21ksr5@queensu.ca

***Abstract*—Road damage significantly impacts infrastructure efficiency, road safety, and maintenance budgets. This paper introduces an automated road damage detection and visualization system employing advanced deep learning models, specifically YOLOv11 and YOLOv12 [1], [2], with ongoing testing of DeepSORT [3] for improved detection tracking across video frames. Utilizing a Raspberry Pi [6] equipped with a camera and GPS module, synchronized video and GPS data are captured. Data is uploaded to a Node.js [8] and Next.js [9] web platform for processing, resulting in an interactive, color-coded map allowing detailed damage analysis and route navigation based on road damage data. Our model achieves a mean Average Precision (mAP) of 54%, indicating significant practical applicability.**

I. Introduction

Automated detection of road damage, including potholes and surface cracks, is essential to maintaining road safety, reducing maintenance costs, and enhancing transportation efficiency. Traditional manual inspections are costly, slow, and error-prone. This research presents a comprehensive automated approach leveraging state-of-the-art deep learning models, YOLOv11 and YOLOv12 [1], [2], combined with the DeepSORT tracking algorithm [3], currently under evaluation.

II. Methodology

A. Datasets and Model Development

The models were trained and evaluated on datasets from the Canadian Road Damage Detection Challenge (CRDDC 2022) [4] and the IEEE Big Data Cup 2022 [5]. YOLOv11 and YOLOv12 models were selected for their accuracy and efficiency in real-time detection scenarios. Training utilized Google Colab Pro+ [7] with NVIDIA A100 GPUs, significantly accelerating training times and enabling extensive hyperparameter tuning. Data augmentation techniques, including random cropping, rotations, brightness adjustments, and scaling, were employed to enhance model robustness across varying road conditions and lighting scenarios.

B. Hardware Setup for Data Collection

A Raspberry Pi [6] equipped with a high-resolution camera and GPS module was mounted on a vehicle for field data collection. Synchronized video footage in MP4 format and timestamped GPS coordinates were captured and logged, enabling accurate geospatial tagging of detected road damage.

C. Processing Workflow

The system's operational workflow includes:

1) **Upload Interface:** Users upload recorded MP4 videos and corresponding GPS text logs via a custom-built web interface.
2) **Damage Detection and Tracking:** Uploaded videos undergo inference using YOLOv11 and YOLOv12 models [1], [2]. DeepSORT tracking is actively tested to improve tracking accuracy across video frames [3].
3) **Output Integration:** Detection results are combined with GPS data into structured CSV files, containing detailed timestamps, locations, and damage classifications.
4) **Visualization and Navigation:** CSV data generates an interactive, geospatial map displaying road damage markers color-coded by severity. Users can interact with the map to obtain detailed information and utilize navigation features, allowing route optimization based on preferences to either avoid or intersect damaged roads.

D. Web Platform and Storage

The web application, developed with Node.js [8] and Next.js [9] frameworks, offers an intuitive user interface, efficient processing pipeline, and robust data management facilitated by Azure Blob Storage [10]. This cloud backend ensures secure, scalable storage of video files, GPS data, processed outputs, and visualization results.

III. Applications and Use Cases

The developed system addresses multiple practical scenarios:

- **Municipal Road Inspection:** Automating routine road condition assessments to maintain safety.
- **Road Maintenance Planning:** Optimizing resource allocation by mapping damaged road sections.
- **Autonomous Vehicle Navigation:** Enhancing vehicle route safety by providing detailed road condition data.

- **Insurance Assessment:** Providing visual evidence of road conditions to support claim processing.
- **Public Safety Enhancement:** Identifying and repairing hazardous sections to reduce accidents.
- **Logistics Optimization:** Rerouting shipments to avoid damaged roads, minimizing transportation costs.
- **Infrastructure Development:** Assessing road quality pre- and post-construction for improved durability.
- **Environmental Impact Analysis:** Studying the influence of weather and disasters on road conditions.
- **Disaster Recovery:** Prioritizing road repairs after natural disasters to quickly restore essential transportation routes.
- **Low-Clearance Vehicles:** Allowing low-clearance vehicles to select routes that avoid severe road damage and speed bumps.

IV. Experimental Results

Our system achieved a mean Average Precision (mAP) of 54%, demonstrating effective real-world applicability in diverse environmental conditions. The ongoing integration of DeepSORT is expected to further improve detection accuracy by enhancing anomaly tracking across consecutive video frames.

A. Model Evaluation

Figures 1 and 2 illustrate confusion matrices for YOLOv12-l and YOLOv12-s models, respectively, detailing true positives, false positives, and class misclassifications. Figure 3 displays training and validation loss curves along with precision and recall metrics, highlighting training progress and model performance.

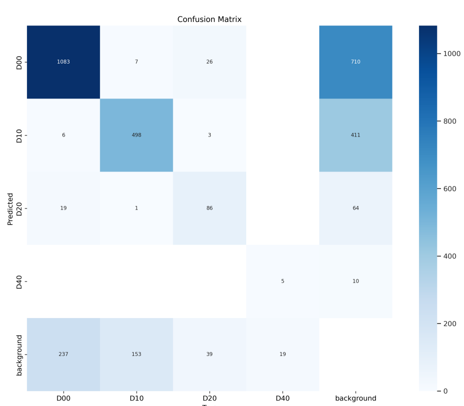

Fig. 1. Confusion Matrix for YOLOv12-l Model

V. Conclusion and Future Work

This study successfully integrates advanced object detection methods, effective tracking algorithms, and practical visualization and navigation tools into a cohesive system for road damage detection and management. Future enhancements include refining the DeepSORT integration, improving detection accuracy in challenging environmental scenarios, and further optimizing model efficiency for edge device deployment.

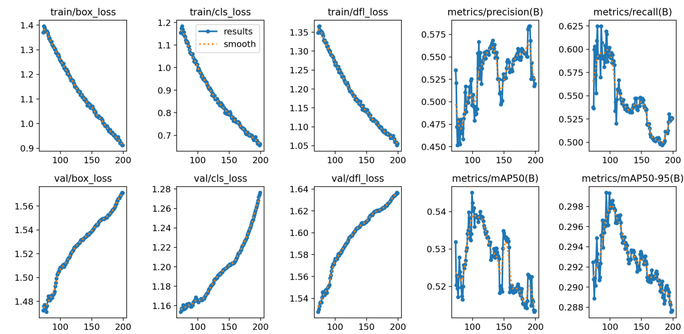

Fig. 2. Training and Validation Metrics

References

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[2] A. Bochkovskiy, C. Wang, and H. Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv preprint arXiv:2004.10934*, 2020.

[3] W. Wojke, N. Bewley, and D. Paulus, "Simple Online and Realtime Tracking with a Deep Association Metric," in *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, 2017, pp. 3645–3649.

[4] "Canadian Road Damage Detection Challenge (CRDDC) 2022," [Online]. Available: https://example.com/crddc2022.

[5] "IEEE Big Data Cup 2022," [Online]. Available: https://example.com/ieeebigdatacup2022.

[6] "Raspberry Pi Documentation," [Online]. Available: https://www.raspberrypi.org/documentation/.

[7] "Google Colaboratory," [Online]. Available: https://colab.research.google.com/.

[8] "Node.js," [Online]. Available: https://nodejs.org/.

[9] "Next.js," [Online]. Available: https://nextjs.org/.

[10] "Microsoft Azure Blob Storage," [Online]. Available: https://azure.microsoft.com/en-us/services/storage/blobs/.