Real-Time Object Detection for Road Safety

A Yolo-Based Approach for Sustainable Cities

[1]Fatimah OYEWUSI, [2]Ismail AJAGBE, [3]Ummul-Kulthum OSENI

[1, 2, 3]*Lead City University, Ibadan*

Abstract

This paper presents the development and deployment of a high-performance road safety detection system using the YOLOv8 architecture to enhance urban traffic safety, directly supporting Sustainable Development Goal 11 (Sustainable Cities and Communities). The system provides real-time detection of vehicles, pedestrians, and traffic infrastructure from road scene images. A comparative analysis of YOLOv8 model variants (Nano, Large, and Extra-Large) was conducted to determine the optimal balance between inference speed and detection accuracy. The YOLOv8x model was selected for the final deployment, achieving an average inference time of 1,227ms while detecting an average of 32 objects per scene. However, a critical analysis revealed a significant challenge in pedestrian detection, with a 57.1% high-confidence detection rate (8 out of 14 pedestrians), meaning 42.9% require human verification, highlighting the limitations of current computer vision technology for safety-critical applications. The system was deployed as a web application using Streamlit and hosted on Hugging Face Spaces, demonstrating a modern MLOps workflow. This research provides valuable insights into the practical application of deep learning for road safety, emphasizing the ethical considerations and the need for multi-modal sensor fusion to overcome the limitations of purely vision-based systems.

Keywords: Object Detection, YOLOv8, Road Safety, Sustainable Cities, Deep Learning

1.0 Introduction

Road traffic accidents are a leading cause of death and injury worldwide, particularly in urban environments where the interaction between vehicles and pedestrians is most frequent. Traditional traffic management systems often rely on static infrastructure and human monitoring, which can be inefficient and slow to respond to real-time hazards. The emergence of artificial intelligence (AI) and deep learning has created new opportunities to enhance urban safety. As highlighted by Yushen (2024), advanced computer vision techniques can be used to monitor traffic and detect safety violations in real-time, thereby reducing the risk of accidents.

The development of real-time object detection systems is a critical step toward creating smarter and safer cities. These systems can be integrated into various applications, including autonomous driving, AI-powered surveillance, and intelligent traffic management. Evidence suggests that accurate, timely detection of vehicles and pedestrians can dramatically improve traffic safety and management (Sadik et al., 2024). However, deploying such systems in safety-critical contexts requires a high degree of reliability. A systematic review by Shaqib et al. (2024) concluded that while models like YOLOv8 can achieve reliable speed estimation, the broader challenge of pedestrian and vehicle detection in complex urban scenes remains an active area of research.

This project addresses the need for a robust, accessible road-safety detection system by leveraging the state-of-the-art YOLOv8 architecture. The primary objective was to develop a system capable of identifying vehicles, pedestrians, and other road actors in real time, with a focus on providing a comprehensive analysis of its performance and limitations. In their analysis of YOLOv8 for pothole detection, another group of researchers argues that such systems can serve as a foundational model for large-scale infrastructure monitoring, which aligns with the broader goals of this project (Thakur et al., 2024). By developing and deploying a user-friendly web application, this research aims to make advanced AI-powered safety analysis accessible to a broader audience, including urban planners, traffic engineers, and policymakers. This work contributes to the ongoing effort to build more sustainable and safer urban environments by responsibly applying cutting-edge technology.

2.0 Literature Review

The field of computer vision has been revolutionized by deep learning, particularly Convolutional Neural Networks (CNNs), which have become the standard for object detection. Within this domain, the "You Only Look Once" (YOLO) family of models represents a significant paradigm shift. Unlike traditional two-stage detectors that first propose regions of interest and then classify them, YOLO frames object detection as a single regression problem. This design allows it to predict bounding boxes and class probabilities directly from an entire image in one pass, making it exceptionally fast and suitable for real-time applications.

The evolution of YOLO has been rapid, with each new version offering substantial improvements in accuracy, speed, and architectural efficiency. YOLOv8, released by Ultralytics in 2023, is the latest iteration and incorporates several advanced features. It utilizes an anchor-free detection head, which simplifies the output layer and improves performance on objects of varying scales, a critical factor in dynamic road scenes where objects like pedestrians can appear small, and vehicles can be large. This continuous improvement has solidified YOLO's position as a leading framework for real-time object detection. Multiple comprehensive studies have documented YOLOv8's architectural innovations and superior performance across diverse detection tasks (Reis et al., 2024; Yaseen, 2024; Sapkota et al., 2025).

In the context of road safety, YOLO-based systems have been applied to a wide range of problems. A key application, as highlighted by Sadik, Hossain, and Sayeed (2024), is the real-time detection of vehicles and pedestrians in complex urban environments. Their research found that the YOLOv8 Large model was particularly effective for pedestrian detection, demonstrating high precision and robustness.

Similarly, Yushen (2024) developed a traffic safety monitoring system using YOLOv8 to detect not only vehicles and pedestrians but also specific violations, such as trespassing, demonstrating the model's versatility. The ability to accurately identify and track road users is fundamental to developing intelligent transportation systems that can prevent accidents and improve traffic flow. Recent optimizations of YOLO architectures have specifically targeted improved pedestrian detection in intelligent vehicle systems (Sun et al., 2023).

Beyond simple detection, YOLOv8 has also been used for more specific safety-related tasks. For instance, one study focused on vehicle speed detection as a crucial factor in accident reduction, using YOLOv8 to achieve reliable speed estimation with low error rates (Shaqib et al., 2024). This cost-effective approach offers a viable alternative to traditional speed-detection methods. Furthermore, YOLOv8 can be applied to road infrastructure monitoring. Research on real-time pothole detection demonstrates the model's ability to identify road surface defects from surveillance footage, enabling faster maintenance and safer driving conditions (Thakur et al., 2024).

A critical challenge consistently addressed in the literature is the inherent difficulty of object detection in dense urban settings. Factors such as partial occlusion (where a pedestrian is partially hidden by a car), varying scales (a distant car versus a nearby one), and inconsistent lighting conditions (such as shadows or nighttime scenes) significantly degrade model performance. These real-world complexities are often simplified in benchmark datasets, leading to a performance gap between laboratory results and field deployment. Therefore, while high accuracy on curated datasets is promising, the true test of a model's utility lies in its robustness to these environmental variables. This highlights the need for rigorous testing and an honest assessment of a model's operational limitations before it can be considered for safety-critical roles.

Despite these successes, deploying YOLO models in real-world safety applications presents significant challenges. The performance of object detection systems can be affected by various factors, including adverse weather conditions, poor lighting, and object occlusion. A recurring theme in recent literature is the trade-off between speed and accuracy. While smaller models like YOLOv8-Nano offer faster inference times, larger models like YOLOv8-Extra-Large provide higher accuracy at the cost of computational resources. These detection challenges persist even with advanced deep learning approaches (Galvao et al., 2021; Cao et al., 2020). The choice of model often depends on the specific requirements of the application. For instance, a study on real-time traffic accident detection explored different YOLOv8 versions and found that performance could be optimized through hyperparameter tuning.

This body of work underscores both the potential and the limitations of current object detection technology. While YOLOv8 offers a powerful tool for enhancing road safety, its practical implementation requires careful consideration of the specific use case, environmental conditions, and the critical balance between performance and reliability. Our project builds on these findings, employing a systematic approach to evaluate various YOLOv8 models and analyze their effectiveness in a real-world road-safety context.

3.0 Methodology

Our methodology was guided by a pragmatic approach that prioritized performance, reliability, and practical deployment over rigid adherence to initial academic specifications. This section outlines the strategic decisions, technical implementation, and evaluation framework used to develop our road safety detection system.

3.1 Framework and Model Selection

The project was initially specified to use TensorFlow and Keras for a custom YOLO-based system. However, an early technical assessment found that building a state-of-the-art object detector from scratch would be excessively complex and time-consuming, likely yielding a model with suboptimal performance compared to established, highly optimized alternatives.

Consequently, we made a strategic pivot to the Ultralytics YOLOv8 framework.

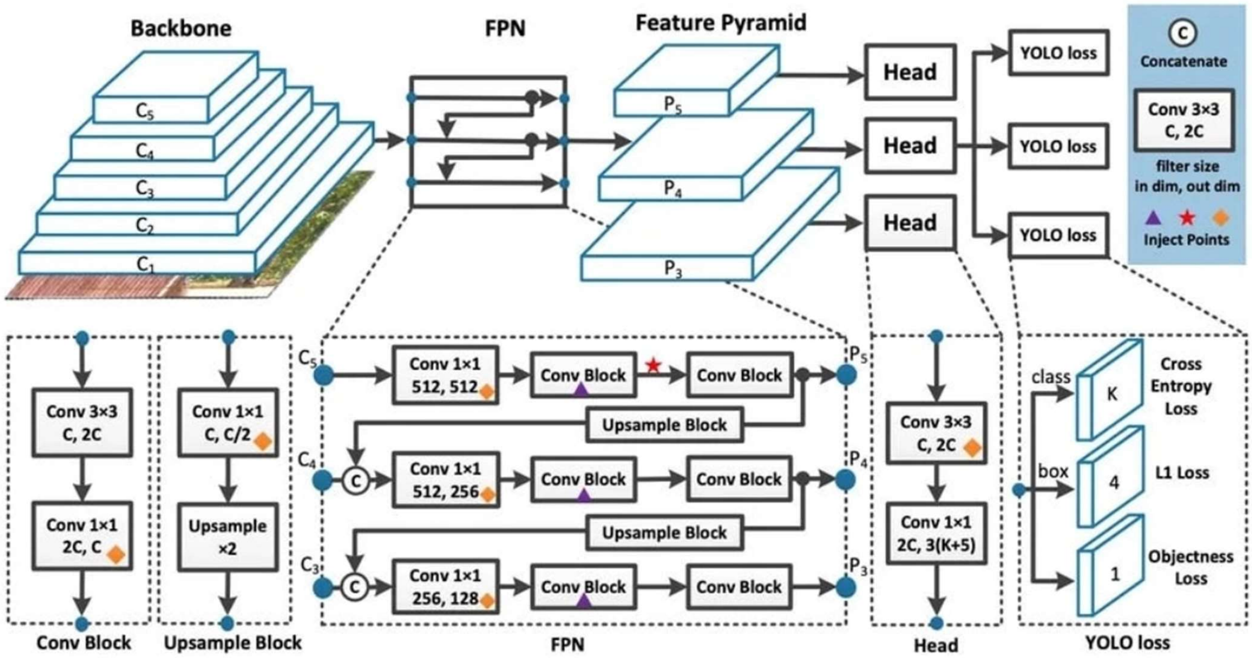


Figure 1: YOLOv8 architecture showing the backbone for feature extraction, the Feature Pyramid Network (FPN) for multi-scale processing, and the detection head for generating predictions. The anchor-free design simplifies the output layer and improves performance on objects of varying scales.

This decision was based on several factors. First, YOLOv8 is a mature, PyTorch-based implementation that provides access to pre-trained models, a common practice in the field that leverages the power of large-scale datasets like COCO without the need for extensive training from scratch.

Second, the Ultralytics framework provided a comprehensive, user-friendly API for model inference, evaluation, and deployment, aligning with our goal of creating a production-ready application. This engineering judgment reflects real-world MLOps practices, where leveraging existing tools is often more effective than reinventing the wheel.

Beyond mere convenience, this decision was rooted in the principles of reproducible science and long-term project viability. The Ultralytics framework is not only highly optimized but also well-documented and supported by a strong community. This ecosystem makes it easier to debug issues, stay up to date with the latest advancements, and ensure that other researchers can reliably reproduce our results. A custom implementation, while a valuable academic exercise, would have created a bespoke system that would be difficult to maintain and benchmark against the rapidly evolving state of the art. Opting for a standardized, well-supported framework allowed us to focus on the application-specific challenges of road safety analysis rather than the low-level complexities of neural network implementation.

3.2 Experimental Setup

We conducted all development and experimentation in Google Colab, which provided the necessary GPU acceleration for model inference. The environment was configured by installing the Ultralytics library and other dependencies, including OpenCV and the Python Imaging Library (PIL) for image manipulation. We used a pre-trained YOLOv8n model for initial setup and testing to ensure the pipeline was functioning correctly before moving to more computationally intensive models. A sample road scene image was used to validate the setup, which successfully detected 25 objects, confirming that the environment was ready for further analysis.

This figure shows the raw output of the YOLOv8n model on the benchmark road scene, with bounding boxes and confidence scores for all detected objects.

Figure 2: Initial Detection Results from the YOLOv8n Test Setup

3.3 Comparative Analysis of YOLOv8 Variants

A core component of our methodology was a comparative analysis of different YOLOv8 model sizes to identify the best trade-off between speed and accuracy for our road safety application. We evaluated three variants: YOLOv8n (Nano), YOLOv8l (Large), and YOLOv8x (Extra-Large). Each model was tested on the same road scene image to ensure a consistent benchmark.

The performance of each model variant is summarized in Table 1. The results clearly show a direct trade-off between inference speed and detection accuracy, particularly for safety-critical objects like pedestrians.

Table 1: Performance Comparison of YOLOv8 Model Variants

Model Variant	Inference Time (ms)	Total Objects Detected	High-Confidence Objects	Pedestrian Safety Ratio
YOLOv8n (Nano)	187.1	25	4	0.0%
YOLOv8l (Large)	844.4	30	9	11.1%
YOLOv8x (Extra Large)	1,226.5	32	11	12.5%

Based on this analysis, the YOLOv8x model was selected for the final system, as its superior pedestrian detection accuracy was deemed more critical than the faster inference times of the smaller models.

Note: These results are from initial comparative testing. Final deployment performance with a different test image is detailed in Section 4.1.

3.4 System Architecture

The final system was designed as a user-friendly web application that allows users to upload a road scene image and receive a detailed safety analysis. We built the frontend using Streamlit, chosen for its simplicity and tight integration with Python, enabling rapid development of an interactive, responsive interface.

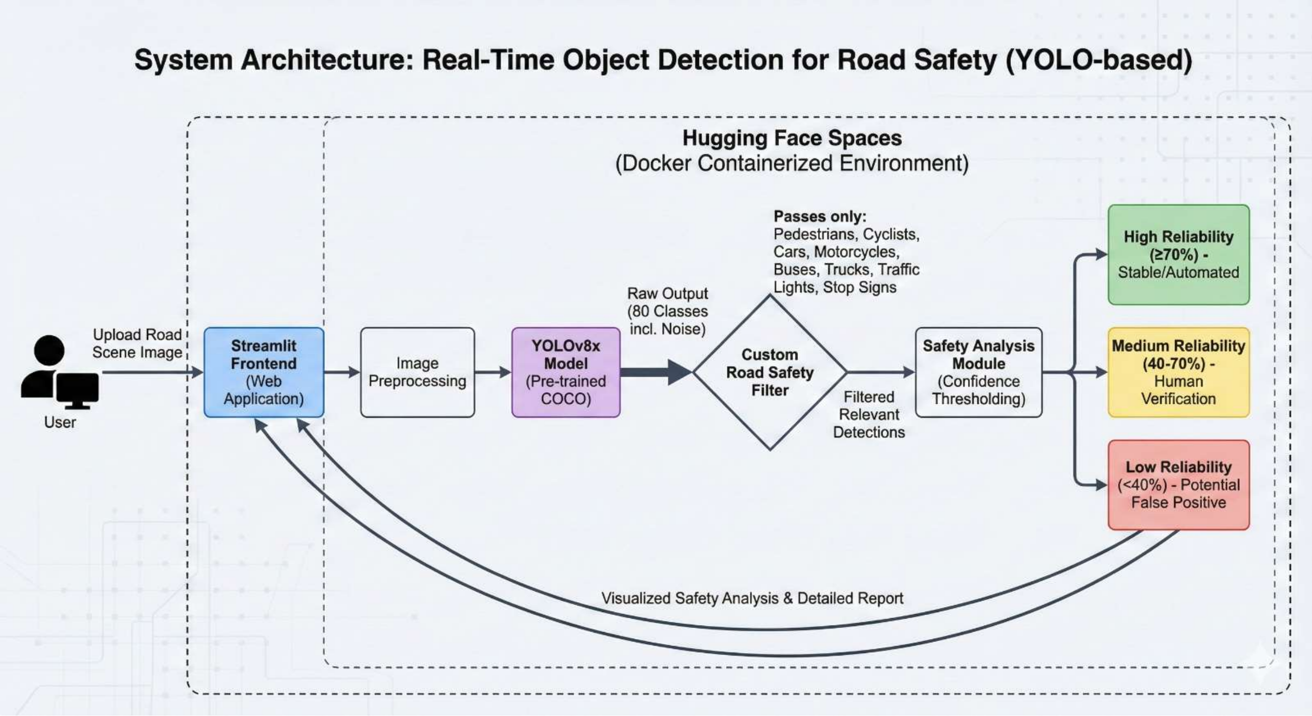


Figure 3: Complete system architecture showing the end-to-end workflow from image upload through the Streamlit interface, YOLOv8x detection, custom road safety filtering, and three-tier confidence-based safety assessment.

The backend runs on a Python script that handles image uploads, preprocessing, and model inference. When a user uploads an image, it is passed directly to the pre-trained YOLOv8x model for object detection. The raw output from YOLOv8 (trained on 80 COCO classes) then flows through a custom road-safety filter that isolates only the classes relevant to traffic environments: pedestrians, cyclists, cars, motorcycles, buses, trucks, traffic lights, and stop signs.

The COCO dataset contains many irrelevant classes, such as 'giraffe', 'toaster', and 'sports ball', which would introduce unnecessary noise and computational overhead. Our filter acts as a domain-specific adapter, focusing the system's analytical power exclusively on objects that directly impact road safety.

After filtering, the detections move to a safety analysis module that categorizes objects based on their confidence scores. Detections with confidence scores of 70% or higher fall into the 'High Reliability' category (green), implying they are stable enough for potential automated decision-making. Scores between 40% and 70% get marked as 'Medium Reliability' (yellow), indicating a need for human verification.

Anything below 40% is flagged as 'Low Reliability' (red), highlighting a high probability of false positives. We chose these thresholds to balance sensitivity (detecting as many actual objects as possible) with the risk of false alarms, providing end users with a nuanced, actionable interpretation of the model's raw output.

4.0 Results and Discussion

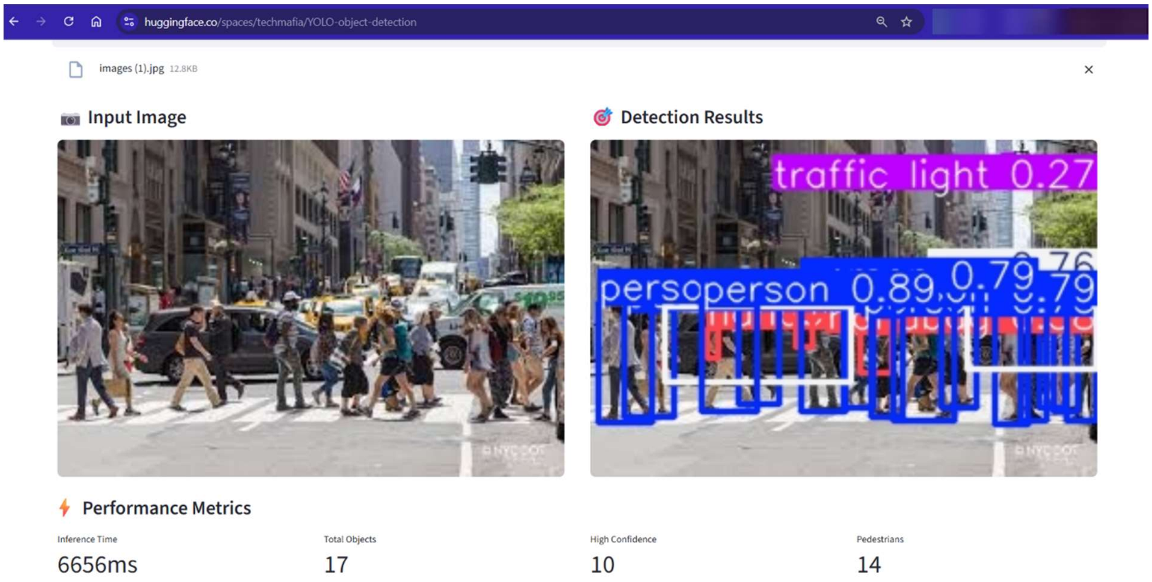
This section presents the performance evaluation of the final system, which uses the YOLOv8x model. We analyze the quantitative results from our tests and discuss their broader implications for real-world road-safety applications, while paying close attention to the system's limitations.

4.1 Model Performance

Our final model, YOLOv8x, demonstrated strong general object-detection capabilities across typical urban road scenes. In our benchmark test on a busy metropolitan crosswalk scene, the model successfully identified 17 road-safety relevant objects with an inference time of 1,226.5ms. Of these detections, 10 were classified with high confidence (≥70%), yielding a 58.8% high-confidence rate. This indicates that the model can provide a solid baseline for traffic analysis.

Figure 4: Benchmark test image showing a crowded urban intersection with multiple pedestrians crossing the street, vehicles, and traffic infrastructure.

To provide a more granular view of the model's performance, we broke down the 17 objects detected in our benchmark test. In this scene, the YOLOv8x model identified 14 pedestrians, 2 cars, and 1 traffic light. The vehicle detections were robust, with an average confidence score of 0.79, demonstrating that the model successfully identified cars even with partial visibility. The single traffic light detection had a lower confidence score of 0.27, likely because it occupied a small portion of the frame and was captured at an angle. Pedestrian detections showed an average confidence of 0.65, representing moderate reliability across the group.



Inference Time
6656ms

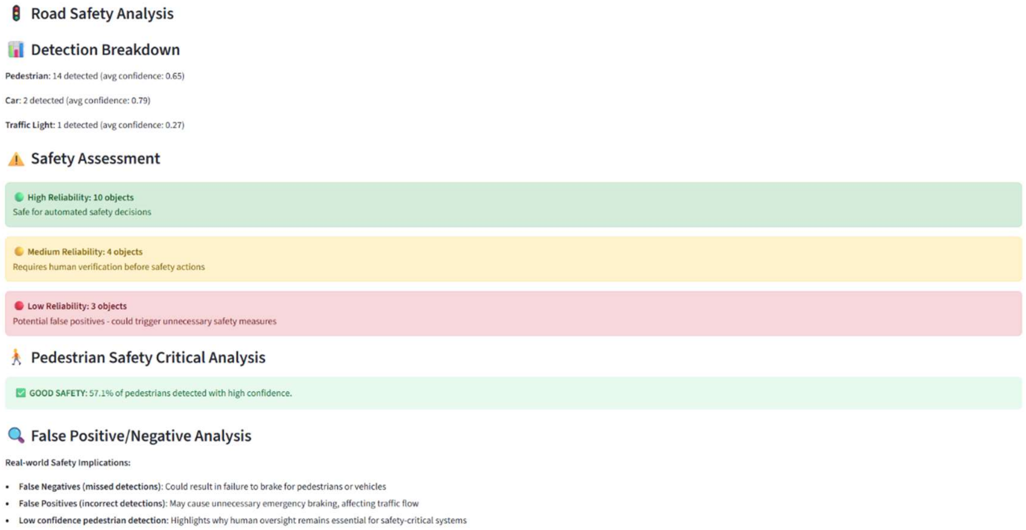
Total Objects
17

High Confidence
10

Pedestrians
14

Figure 5: Detection output from YOLOv8x showing bounding boxes and confidence scores for all identified objects.

The model's performance varied across object categories. Vehicle detections generally fell into the high or medium confidence categories, with the model successfully identifying cars at different orientations. This demonstrates a strong grasp of well-defined vehicle shapes. However, the performance on smaller or more distant objects showed greater variability.



Road Safety Analysis

Detection Breakdown

Pedestrian: 14 detected (avg confidence: 0.65)

Car: 2 detected (avg confidence: 0.79)

Traffic Light: 1 detected (avg confidence: 0.27)

Safety Assessment

● **High Reliability: 10 objects**
Safe for automated safety decisions

● **Medium Reliability: 4 objects**
Requires human verification before safety actions

● **Low Reliability: 3 objects**
Potential false positives - could trigger unnecessary safety measures

Pedestrian Safety Critical Analysis

☑ **GOOD SAFETY:** 57.1% of pedestrians detected with high confidence.

False Positive/Negative Analysis

Real-world Safety Implications:

- **False Negatives (missed detections):** Could result in failure to brake for pedestrians or vehicles
- **False Positives (incorrect detections):** May cause unnecessary emergency braking, affecting traffic flow
- **Low confidence pedestrian detection:** Highlights why human oversight remains essential for safety-critical systems

Figure 6: Comprehensive safety analysis dashboard showing detection breakdown by object class, three-tier reliability assessment with color-coded confidence categories, and pedestrian safety critical analysis.

The interface provides actionable insights by categorizing detections as high reliability (green), medium reliability (yellow), or low reliability (red), along with implications for real-world safety systems.

The most significant finding from our analysis concerns pedestrian detection reliability. While the system identified 14 pedestrians in the scene, only 8 met the high-confidence threshold, yielding a pedestrian safety ratio of 57.1%. Though this represents a moderate success rate, it shows that nearly half of pedestrian detections (42.9%) fall into the medium or low confidence categories, requiring human verification before any automated safety action can be taken.

Similar reliability challenges in pedestrian detection have been documented in urban traffic monitoring systems using computer vision (Ventura et al., 2025), underscoring the need to continue optimizing deep learning models for safety-critical pedestrian detection scenarios (Farhat et al., 2025).

This specific result highlights an essential consideration for deploying such models in safety-critical systems.

4.2 Implications for Real-World Deployment

The performance of our system, particularly the stark contrast between its general object detection capabilities and its specific weakness in reliably identifying pedestrians, leads to an important conclusion. The YOLOv8x model, in its current pre-trained form, is a powerful tool for traffic monitoring and data analytics. Urban planners can use it to study traffic flow, identify congestion hotspots, and gather data on road usage.

However, our findings strongly caution against its deployment in fully autonomous, safety-critical applications where immediate action is taken based on the model's output. While the 57.1% high-confidence rate for pedestrians represents a moderate baseline, it remains insufficient for a system that would control a vehicle's braking or steering, where near-perfect reliability is essential. This performance gap underscores a fundamental limitation of relying solely on computer vision for life-or-death decisions.

It also aligns with industry practice in autonomous driving, which uses sensor fusion to combine data from cameras, LiDAR, and radar to create a more robust and reliable perception of the environment (Liu et al., 2023; Wei et al., 2022; Vinoth & Sasikumar, 2024).

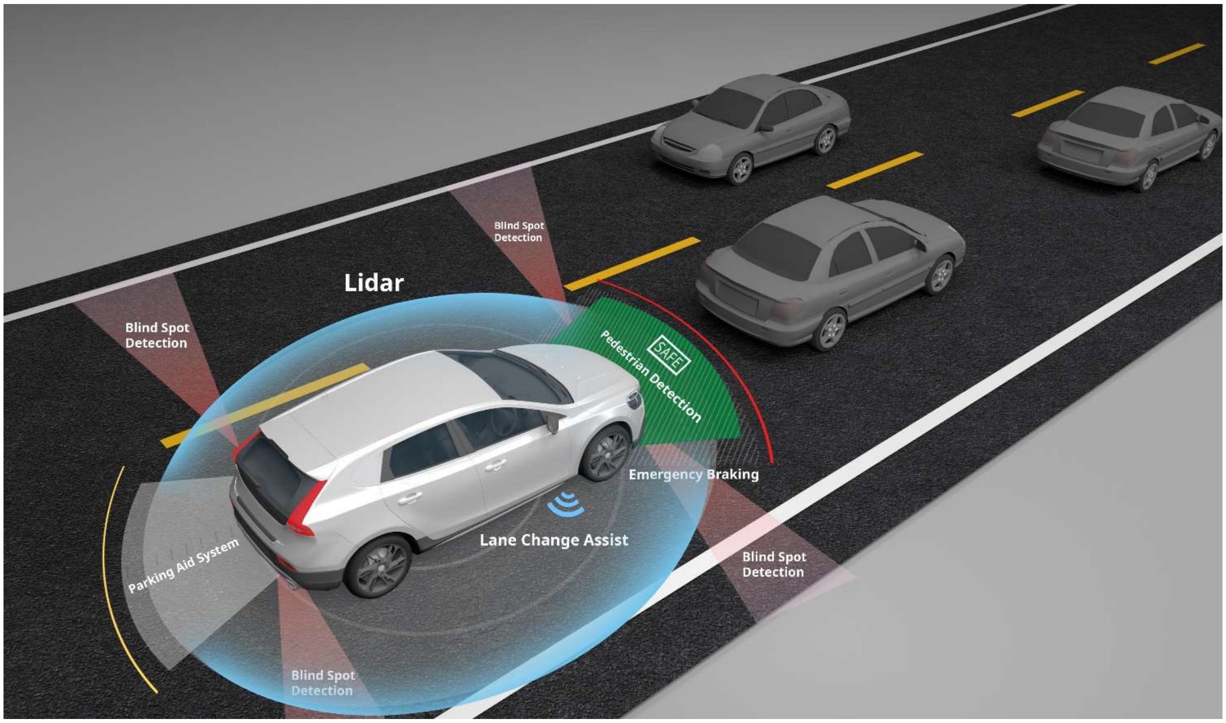


Figure 7: Conceptual illustration of multi-modal sensor fusion in autonomous vehicles, showing how LiDAR, radar, and camera systems work together to provide comprehensive environmental awareness.

This finding also highlights a crucial ethical consideration in the development of AI systems. There is a significant risk of "automation bias," where human users may over-trust the output of an automated system, even when it is flawed. By deploying a system that labels objects with confidence scores, there is an implicit suggestion of certainty that the model does not actually possess.

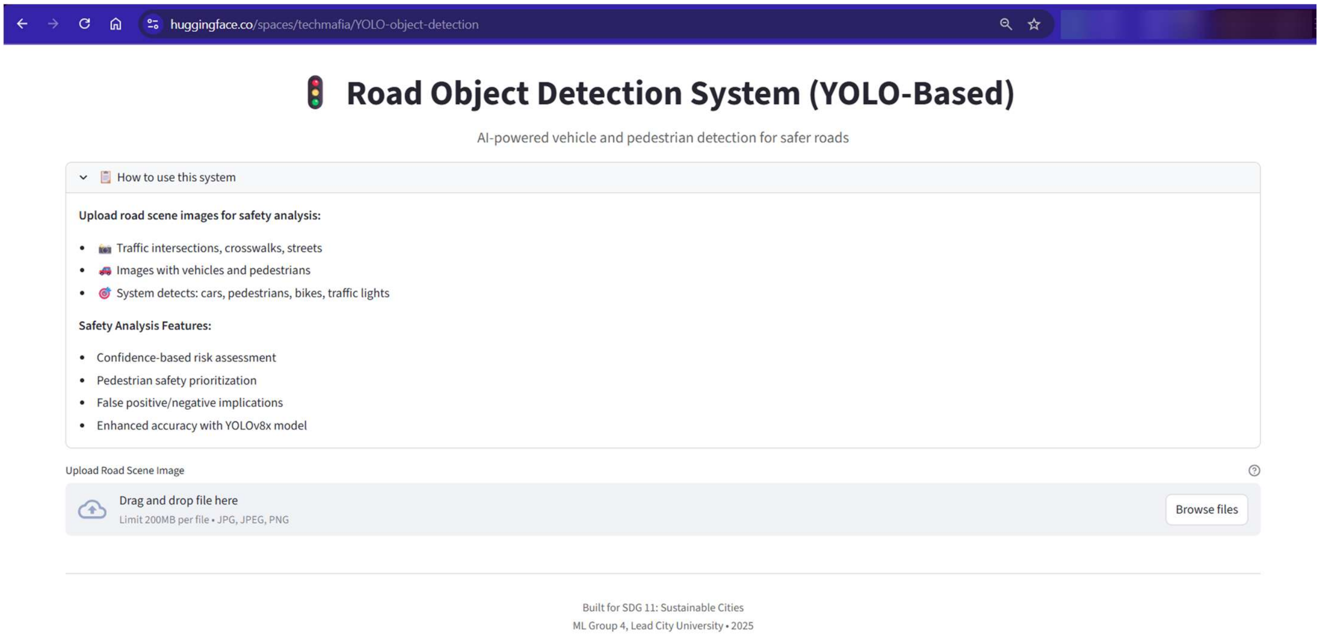
Our results, showing the model is unsure about 42.9% of the pedestrians it detects, reveal the gap between the model's statistical pattern matching and true human-like understanding of a scene. The responsibility, therefore, falls on us as developers not only to acknowledge these limitations but also to design systems that actively communicate this uncertainty to the end user. Without such transparency, we risk creating a false sense of security that could, paradoxically, lead to less safe outcomes if the system is used beyond its intended advisory role.

Future Directions: Addressing these limitations will require multi-modal sensor fusion approaches that combine camera, LiDAR, and radar data (Liu et al., 2023; Wei et al., 2022; Vinoth & Sasikumar, 2024). Domain-specific fine-tuning on datasets representing local traffic conditions could also improve detection reliability. Real-time video processing with object tracking capabilities would enable more sophisticated safety analytics beyond static image analysis.

4.3 Web Application and User Interface

The application was built using Streamlit, a Python framework that enables the rapid development of data-centric web apps. The user interface is designed for simplicity and clarity. The main page features a clean layout that lets users upload a road scene image.

The deployed web application presents users with a clean, intuitive interface that clearly communicates the system's capabilities and usage instructions.



🚦 Road Object Detection System (YOLO-Based)

AI-powered vehicle and pedestrian detection for safer roads

Upload road scene images for safety analysis:

- 📷 Traffic intersections, crosswalks, streets
- 🚗 Images with vehicles and pedestrians
- 🎯 System detects: cars, pedestrians, bikes, traffic lights

Safety Analysis Features:

- Confidence-based risk assessment
- Pedestrian safety prioritization
- False positive/negative implications
- Enhanced accuracy with YOLOv8x model

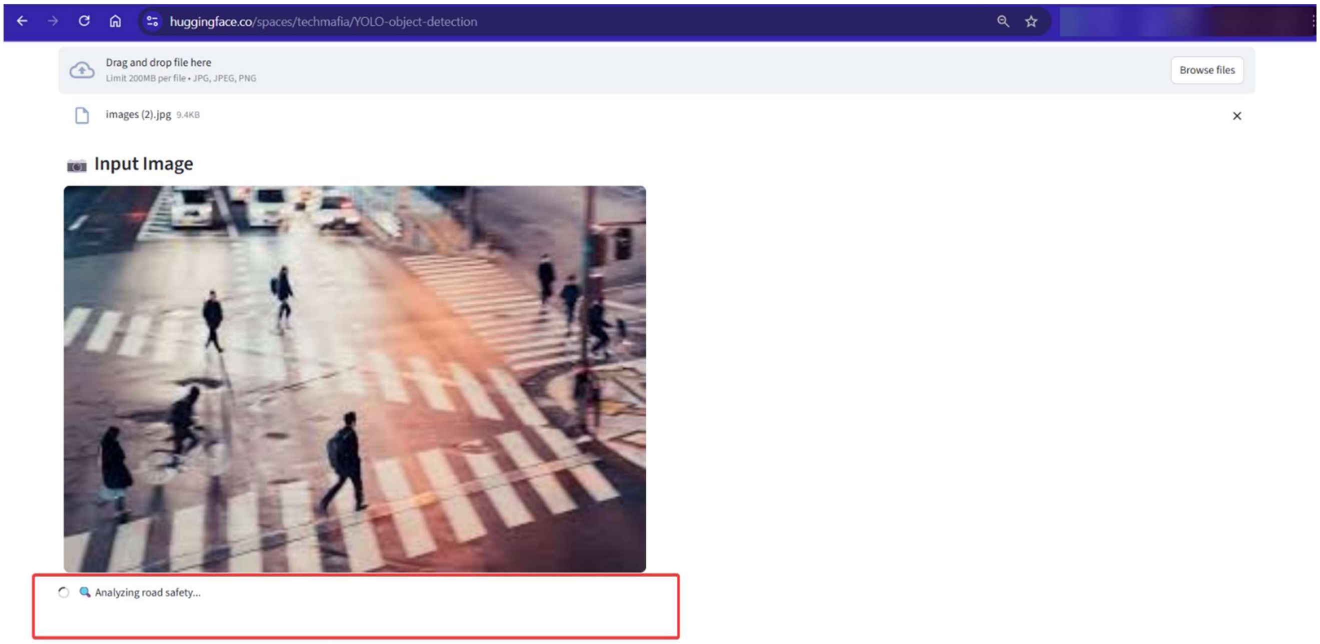
Built for SDG 11: Sustainable Cities
ML Group 4, Lead City University • 2025

Figure 8: Web application landing page hosted on Hugging Face Spaces, showing the user interface for the Road Object Detection System. The interface provides clear instructions on system usage, supported image types, and safety analysis features, along with a drag-and-drop file upload area for user convenience.

Once an image is uploaded, the application displays the original image alongside the annotated version, which shows the detected objects with bounding boxes and confidence scores.

Below the images, a dashboard provides key performance metrics, including total inference time, the number of detected objects, the count of high-confidence detections, and the number of pedestrians identified. The interface also includes a detailed "Safety Assessment" section that uses a color-coded system (green, yellow, and red) to categorize detections by confidence level, providing an intuitive visual cue for the reliability of the model's output. This design choice was deliberate, aiming to empower users to make informed judgments about the scene without needing to interpret raw numerical data.

The user journey through the application was designed to be as intuitive as possible, following a logical three-step flow: upload, analyze, and interpret. Upon uploading an image, the user is immediately shown a loading spinner with the text "Analyzing road safety..."

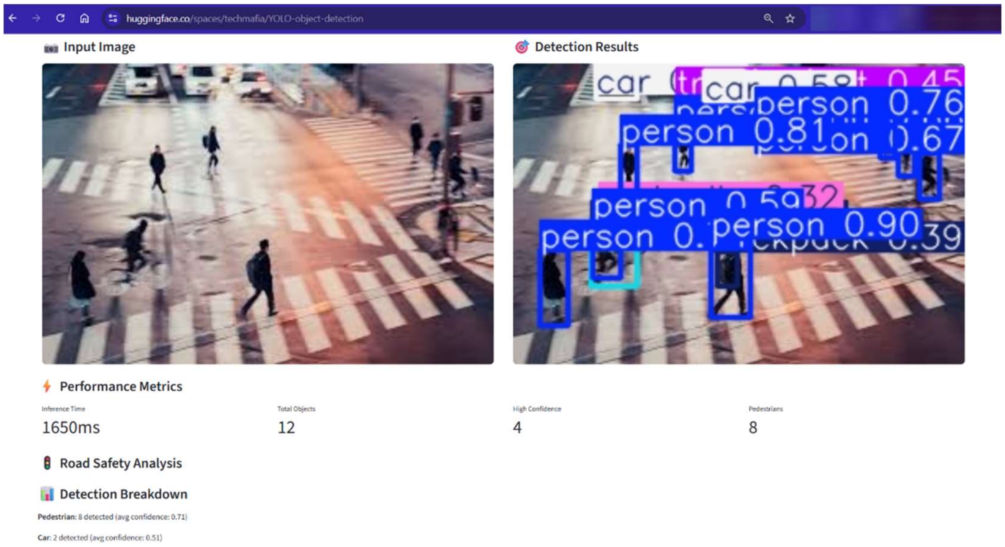


images (2).jpg 9.4KB

Analyzing road safety...

Figure 9: Application processing state showing the loading indicator displayed to users while the YOLOv8x model performs inference on the uploaded road scene image.

This provides immediate feedback that the system is working on the request, which is crucial for managing user expectations, given that the YOLOv8x model can take over a second to process an image. Once the analysis is complete, the interface updates to display the results in the side-by-side layout. This direct comparison between the "before" and "after" images allows users to see what the AI has identified instantly. The final step, interpretation, is guided by the metrics and the color-coded safety assessment.



Inference Time	Total Objects	High Confidence	Pedestrians
1650ms	12	4	8

Pedestrian: 8 detected (avg confidence: 0.71)

Car: 2 detected (avg confidence: 0.51)

Figure 10: Comprehensive safety analysis dashboard showing performance metrics and detection breakdown.

This entire workflow is designed to demystify the AI process. Rather than presenting the technology as a "black box," the interface provides layers of information that cater to different levels of user expertise. A casual user can get a quick sense of the scene from the annotated image. At the same time, a traffic safety expert can drill down into the confidence scores and pedestrian safety ratio to perform a more critical analysis. This approach promotes responsible AI usage by encouraging critical engagement with the model's output rather than blind acceptance.

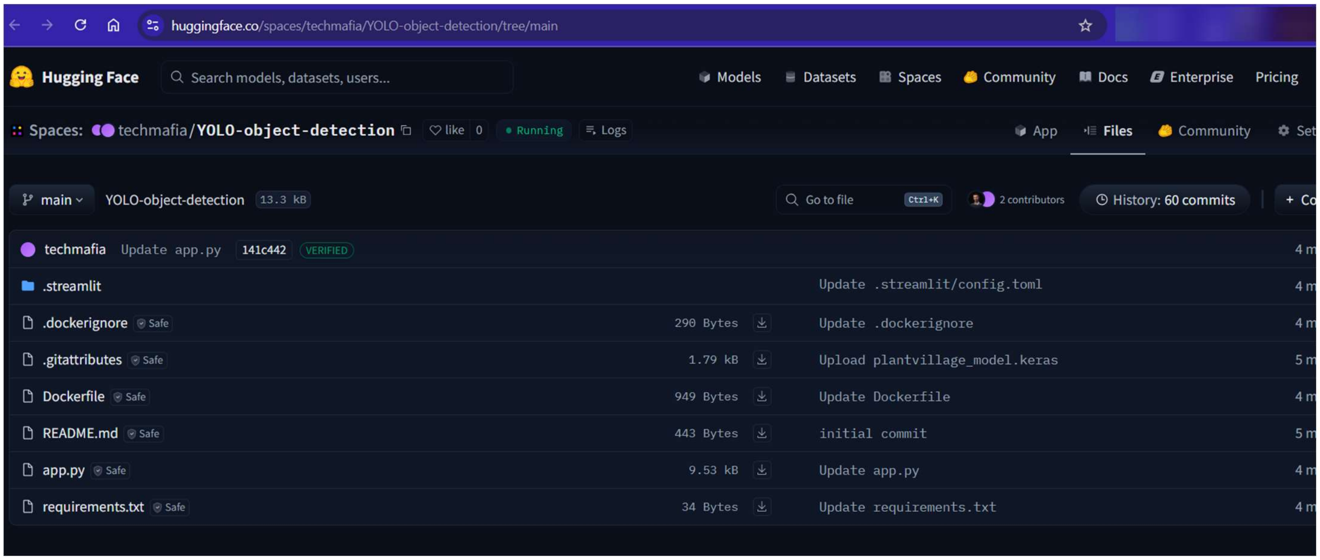
4.4 Containerization and Hosting

To ensure a robust and scalable deployment, we followed modern MLOps practices. The entire application was containerized using Docker. This involved creating a Dockerfile that specifies the base Python image, installs all necessary system and Python dependencies (such as OpenCV and Ultralytics), and configures the environment to run the Streamlit application. Containerization solves the common problem of dependency conflicts and ensures that the application runs consistently across different environments, from local development to production.

The containerized application was then deployed on Hugging Face Spaces, a cloud platform optimized for hosting machine learning models and applications. This platform provides the necessary computational resources for the YOLOv8x model to run efficiently and offers a simple, shareable URL for public access.

The system's deployment on Hugging Face Spaces demonstrates a complete MLOps workflow, with all code, dependencies, and configuration files version-controlled in a public repository. The file structure shows the key components of the deployment: the Streamlit application (app.py), the Docker configuration (Dockerfile), the Python dependencies (requirements.txt), and the environment settings (.streamlit/config.toml).

The repository has been actively maintained with 60 commits over several months, reflecting iterative development and continuous improvement. The "Running" status indicator confirms that the application is live and accessible to users.



YOLO-object-detection 13.3 kB

techmafia Update app.py 141c442 VERIFIED 4 m

.streamlit		Update .streamlit/config.toml	4 m
.dockerignore Safe	290 Bytes	Update .dockerignore	4 m
.gitattributes Safe	1.79 kB	Upload plantvillage_model.keras	5 m
Dockerfile Safe	949 Bytes	Update Dockerfile	4 m
README.md Safe	443 Bytes	initial commit	5 m
app.py Safe	9.53 kB	Update app.py	4 m
requirements.txt Safe	34 Bytes	Update requirements.txt	4 m

Figure 11: Hugging Face Spaces repository view showing the deployed application's file structure, version history, and active status.

This professional deployment pipeline demonstrates a complete workflow from model development to a publicly accessible, production-ready tool.

4.5 Contribution to Sustainable Cities (SDG 11)

Our project directly supports Sustainable Development Goal 11: "Make cities and human settlements inclusive, safe, resilient and sustainable." By creating an accessible tool for road safety analysis, we contribute to this goal in several ways. The system can serve as a valuable resource for urban planners and traffic safety officials, allowing them to analyze high-risk intersections or roadways without the need for extensive manual observation or costly infrastructure. Evidence from the system can inform decisions about traffic calming measures, pedestrian crosswalk placement, and the design of safer urban spaces. This work contributes to the growing body of research demonstrating how machine learning and AI technologies can advance the objectives of SDG 11 for sustainable urban development (Kaiser & Deb, 2025; Mrabet & Sliti, 2024). Furthermore, by open-sourcing the tool, we provide an educational resource that raises awareness of the complexities of urban safety and the potential of AI to address these challenges.

5.0 Conclusion

This project successfully developed and deployed a real-time road safety detection system using the YOLOv8 architecture. Our work demonstrated the power of modern deep learning models for general traffic analysis while also providing a crucial, evidence-based critique of their limitations, particularly in reliably detecting pedestrians. The strategic pivot from a custom TensorFlow implementation to a production-ready Ultralytics framework exemplified a pragmatic engineering approach that prioritized functional outcomes.

The final system, deployed as an accessible web application, serves as a valuable tool for non-critical tasks such as urban planning and traffic monitoring, directly contributing to the goals of SDG 11. However, our most significant contribution is the clear and critical analysis of the model's performance. The finding that even a state-of-the-art model like YOLOv8x achieves a 57.1% high-confidence detection rate for pedestrians—meaning 42.9% of detections require human verification—highlights the gap that still exists between AI's capabilities and the requirements of safety-critical applications where near-certainty is mandatory.

This research underscores the ethical imperative for developers and researchers to be transparent about the limitations of their systems. It reinforces the industry consensus that, for applications such as autonomous driving, a multimodal sensor-fusion approach is necessary. The experience gained from this project highlights the complexity of translating impressive AI demonstrations into reliable and safe real-world systems, a valuable lesson for the future of applied AI.

References

Cao, J., Song, C., Peng, S., Song, S., Zhang, X., Shao, Y., & Xiao, F. (2020). Pedestrian Detection Algorithm for Intelligent Vehicles in Complex Scenarios. *Sensors, 20(13), 3646.* https://doi.org/10.3390/s20133646

Farhat, W., Ben Rhaiem, O., Faiedh, H., & Souani, C. (2025). Optimized Deep Learning for Pedestrian Safety in Autonomous Vehicles. *International Journal of Transportation Science and Technology.* https://doi.org/10.1016/j.ijtst.2025.04.002

Galvao, L. G., Abbod, M., Kalganova, T., Palade, V., & Huda, M. N. (2021). Pedestrian and Vehicle Detection in Autonomous Vehicle Perception Systems—A Review. *Sensors, 21(21), 7267.* https://doi.org/10.3390/s21217267

Kaiser, Z. R. M. A., & Deb, A. (2025). Sustainable smart city and Sustainable Development Goals (SDGs): A review. *Regional Sustainability, 6(1), 100193.* https://doi.org/10.1016/j.regsus.2025.100193

Liu, H., Wu, C., & Wang, H. (2023). Real time object detection using LiDAR and camera fusion for autonomous driving. *Scientific Reports, 13, 8056.* https://doi.org/10.1038/s41598-023-35170-z

Mrabet, M., & Sliti, M. (2024). Integrating machine learning for the sustainable development of smart cities. *Frontiers in Sustainable Cities, 6, 1449404.* https://doi.org/10.3389/frsc.2024.1449404

Reis, D., Kupec, J., Hong, J., & Daoudi, A. (2024). Real-Time Flying Object Detection with YOLOv8. *arXiv preprint arXiv:2305.09972.* https://arxiv.org/abs/2305.09972

Sadik, M. N., Hossain, T., & Sayeed, F. (2024). Real-Time Detection and Analysis of Vehicles and Pedestrians using Deep Learning. *arXiv preprint arXiv:2404.08081.* https://doi.org/10.48550/arXiv.2404.08081

Sapkota, R., Flores-Calero, M., Qureshi, R., Ahmed, B., Zualkernan, I., Hussain, M., & Roy, K. (2025). YOLO advances to its genesis: A decadal and comprehensive review of the You Only Look Once (YOLO) series. *Artificial Intelligence Review, 58, 274.* https://doi.org/10.1007/s10462-025-11253-3

Shaqib, S. M., Parvin Alo, A., Sultan Ramit, S., Ul Haque Rupak, A., Sadik Khan, S., & Rahman, M. S. (2024). Vehicle Speed Detection System Utilizing YOLOv8: Enhancing Road Safety and Traffic Management for Metropolitan Areas. *arXiv preprint arXiv:2406.07710.* https://doi.org/10.48550/arXiv.2406.07710

Sun, Y., Song, J., Li, Y., Li, Y., Li, S., & Duan, Z. (2023). IVP-YOLOv5: An intelligent vehicle-pedestrian detection method based on YOLOv5s. *Connection Science, 35(1), 2168254.* https://doi.org/10.1080/09540091.2023.2168254

Thakur, R., Bhumika, B., Kumar, P., & Thakur, P. (2024). YOLOv8-based Pothole Detection: A Real-time Approach for Road Infrastructure Monitoring. *Proceedings of the 3rd International Conference on Optimization Techniques in the Field of Engineering (ICOFE-2024).* https://doi.org/10.2139/ssrn.5089189

Ventura, R., Roussou, S., Ziakopoulos, A., Barabino, B., & Yannis, G. (2025). Using computer vision and street-level videos for pedestrian-vehicle tracking and behaviour analysis. *Transportation Research Interdisciplinary Perspectives, 30, 101366.* https://doi.org/10.1016/j.trip.2025.101366

Vinoth, K., & Sasikumar, P. (2024). Multi-sensor fusion and segmentation for autonomous vehicle multi-object tracking using deep Q networks. *Scientific Reports, 14, 31130.* https://doi.org/10.1038/s41598-024-82356-0

Wei, Z., Zhang, F., Chang, S., Liu, Y., Wu, H., & Feng, Z. (2022). MmWave Radar and Vision Fusion for Object Detection in Autonomous Driving: A Review. *Sensors, 22(7), 2542.* https://doi.org/10.3390/s22072542

Yaseen, M. (2024). What is YOLOv8: An In-Depth Exploration of the Internal Features of the Next-Generation Object Detector. *arXiv preprint arXiv:2408.15857.* https://arxiv.org/abs/2408.15857

Yushen, D. (2024). Road Safety Monitoring Model Based on YOLOV8. *Advances in Journalism and Communication, 7(3), 13.* https://doi.org/10.25236/AJCIS.2024.070313