SMART VEHICLE INFOTAINMENT SYSTEM USING COMPUTER VISION AND AR

Course Code and Name: BECE498J - Project - II / Internship

Team Members:

Sudharson Aswin N K (21BEC0025)

Tushar Sharma (21BEC0490)

Guide Name: Dr. Rohit Mathur

AGENDA

- INTRODUCTION
- LITERATURE REVIEW
- PROBLEM FORMULATION
- RELEVANCE OF THE PROBLEM STATEMENT W.R.T SDG
- PROPOSED SYSTEM/ARCHITECTURE/DESIGN
- ANALYTICAL AND THEORETICAL DESCRIPTION
- HARDWARE/SOFTWARE TOOLS AND SIMULATION/DESIGN PARAMETERS
- RESULTS ANALYSIS
- CONCLUSION
- CONTRIBUTION OF INDIVIDUAL TEAM MEMBERS
- IMPACT OF THE PROJECT ON SOCIETY AND ENVIRONMENT
- PROJECT OUTCOME(PUBLICATION/PATENT DETAILS)
- REFERENCES.

RELEVANCE TO SUSTAINABLE DEVELOPMENT GOALS (SDGS)

SDG 9: Industry, Innovation, and Infrastructure

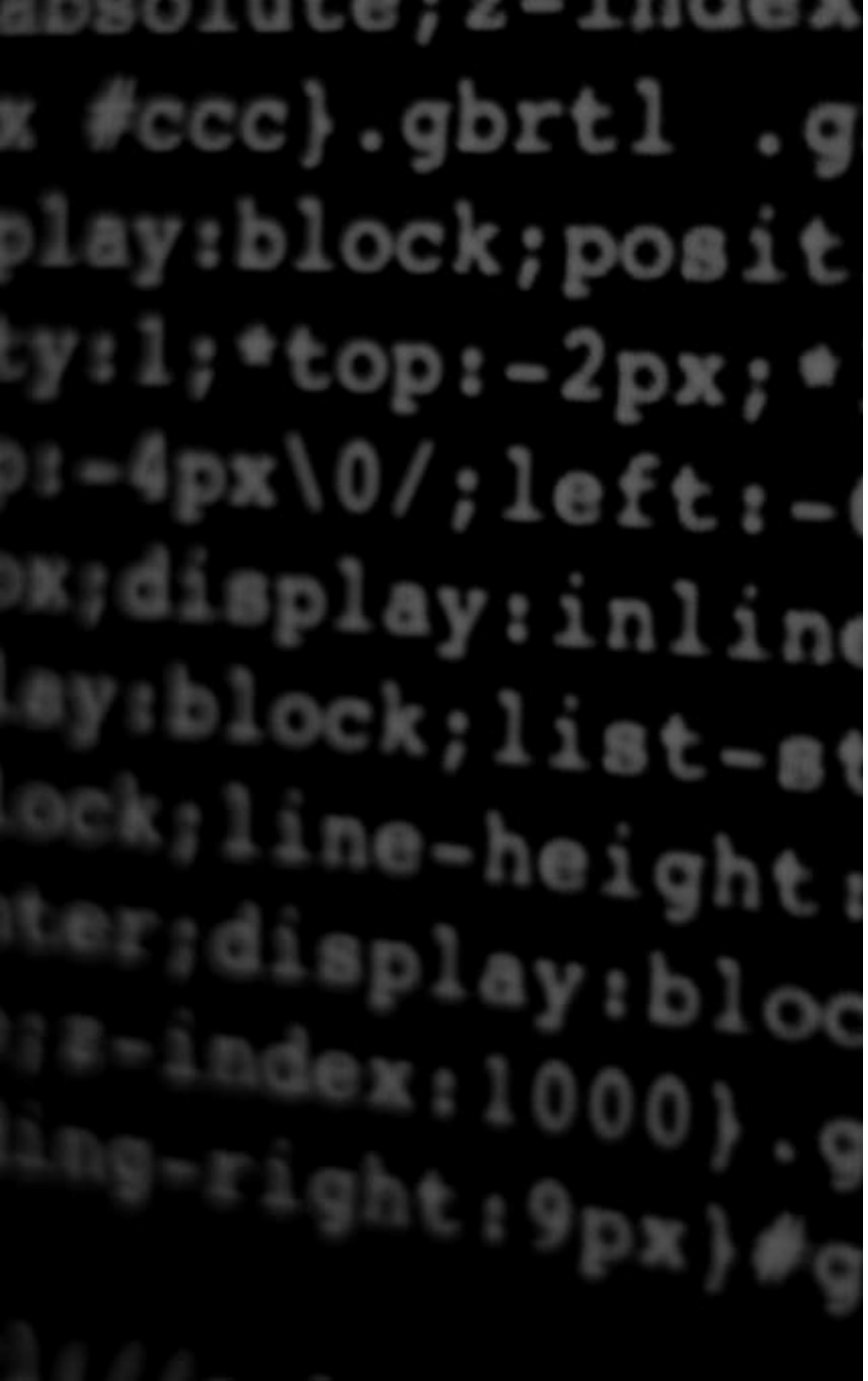
Industry, Innovation, and Infrastructure

Supports innovation in intelligent transport systems through AI and AR, enhancing digital infrastructure in mobility

SDG 11: Sustainable Cities and Communities

Sustainable Cities and Communities

Promotes safer and smarter mobility, contributing to more efficient and sustainable urban transportation.



INTRODUCTION

- Modern vehicles are increasingly integrating smart infotainment systems to enhance user experience, safety, and navigation. With 33% of new vehicles featuring ADAS-equipped cameras in 2021 (projected to reach 50% by 2030), the role of computer vision and AR in driving assistance is expanding.

- This project proposes a Smart Vehicle Infotainment System that uses a Raspberry Pi 5 and camera module to:

- Detect QR codes & traffic signs for real-time road data (speed limits, intersections, school zones).

- Overlay AR-based 3D Alerts and Information.

- Using AI models like Haar cascade classifiers and Yolo to detect and alert users about the objects in front of it

- By leveraging libraries and framework like Multithreading and multiprocessing, OpenCV, Flask, Tensor Flow and other, the system aims to improve road safety, navigation, and commercial engagement in a cost-effective manner

LITERATURE REVIEW

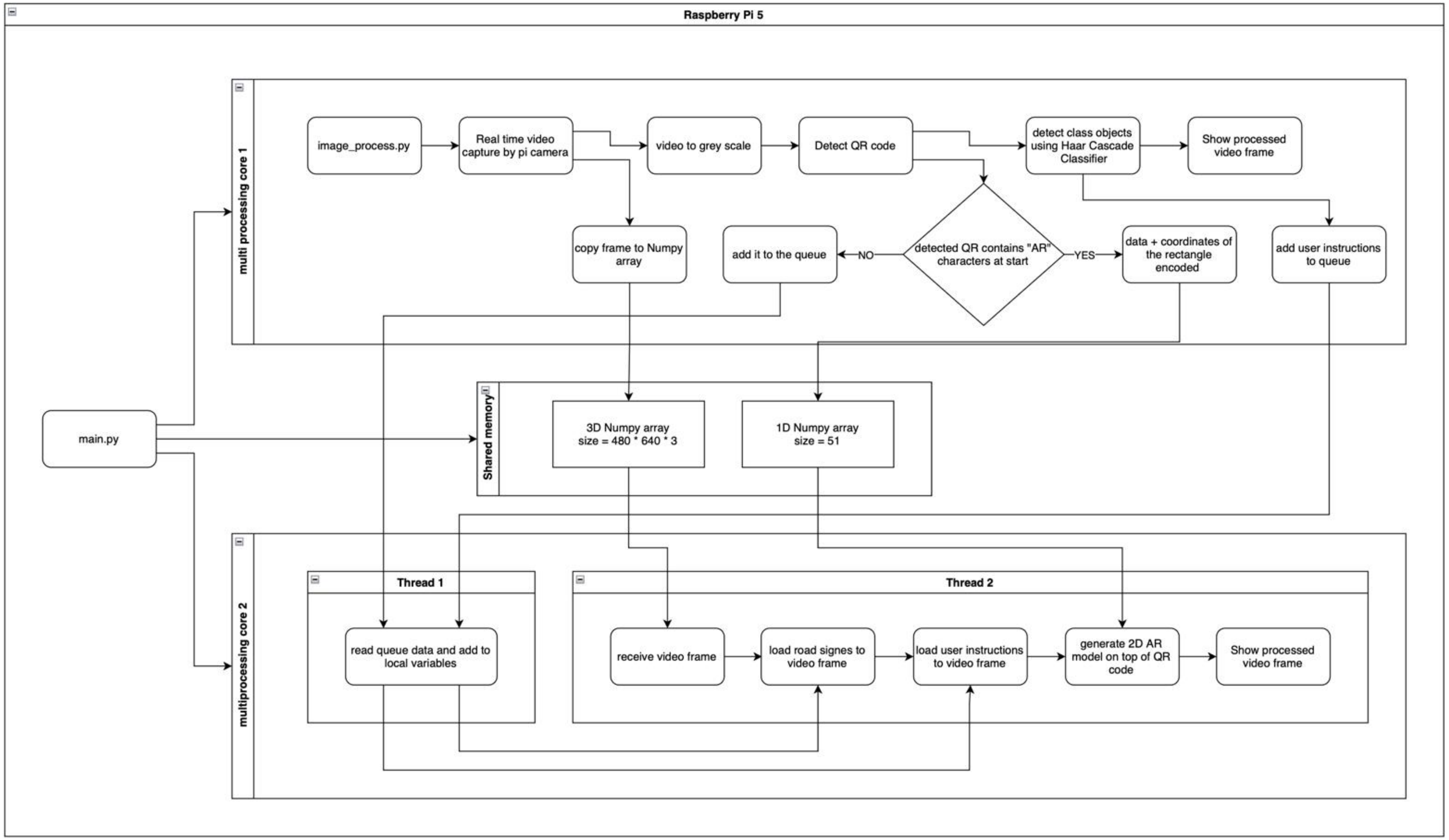
Title	Authors	Key Relevance to Project
The Role of AI in ACL Injuries	Andriollo et al. (2024)	Highlights the integration of AI into domain-specific decision systems. Demonstrates lightweight AI for diagnostics, supporting our rationale for embedded inference on Raspberry Pi.
Deep Fusion: A CNN-LSTM Image Caption Generator	Bhatt et al. (2023)	Combines CNN for visual feature extraction and LSTM for interpretation. Reinforces the use of deep learning in visual scene understanding, relevant for AR overlay contexts.
Visual Image Caption Generator Using Deep Learning	Sharma et al. (2021)	Presents use of convolutional models for extracting semantic features from visual input — a parallel to our object detection phase using YOLOv8s.

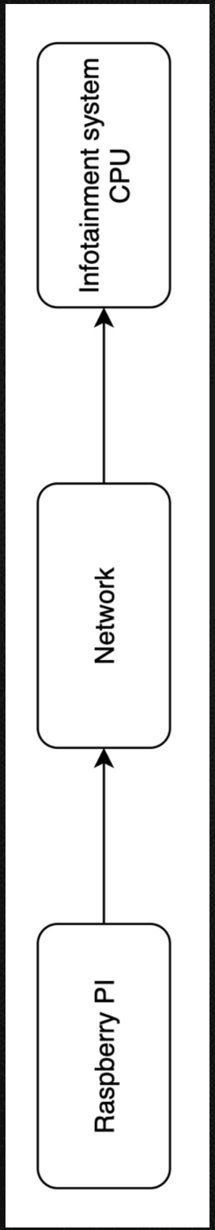
Title	Authors	Key Relevance to Project
Image Caption Using CNN & LSTM (Unpublished)	Mohamed A. A. (2020)	Emphasizes computational trade-offs in embedded AI models. Supports our use of Haar Cascade for low-power prototypes and YOLOv8s in high-performance configurations.
Image Caption Generation with Visual Attention and Spatial Context	Sasibhooshan et al. (2023)	Explores attention mechanisms for spatial localization — aligns with our use of pose estimation and marker-based AR model placement via QR codes.
Artificial Intelligence in Management of ACL Injuries	Corban et al. (2021)	Shows practical implementation of ML in healthcare devices, highlighting modular design and real-world constraints — a methodology similar to our system's embedded + distributed model.

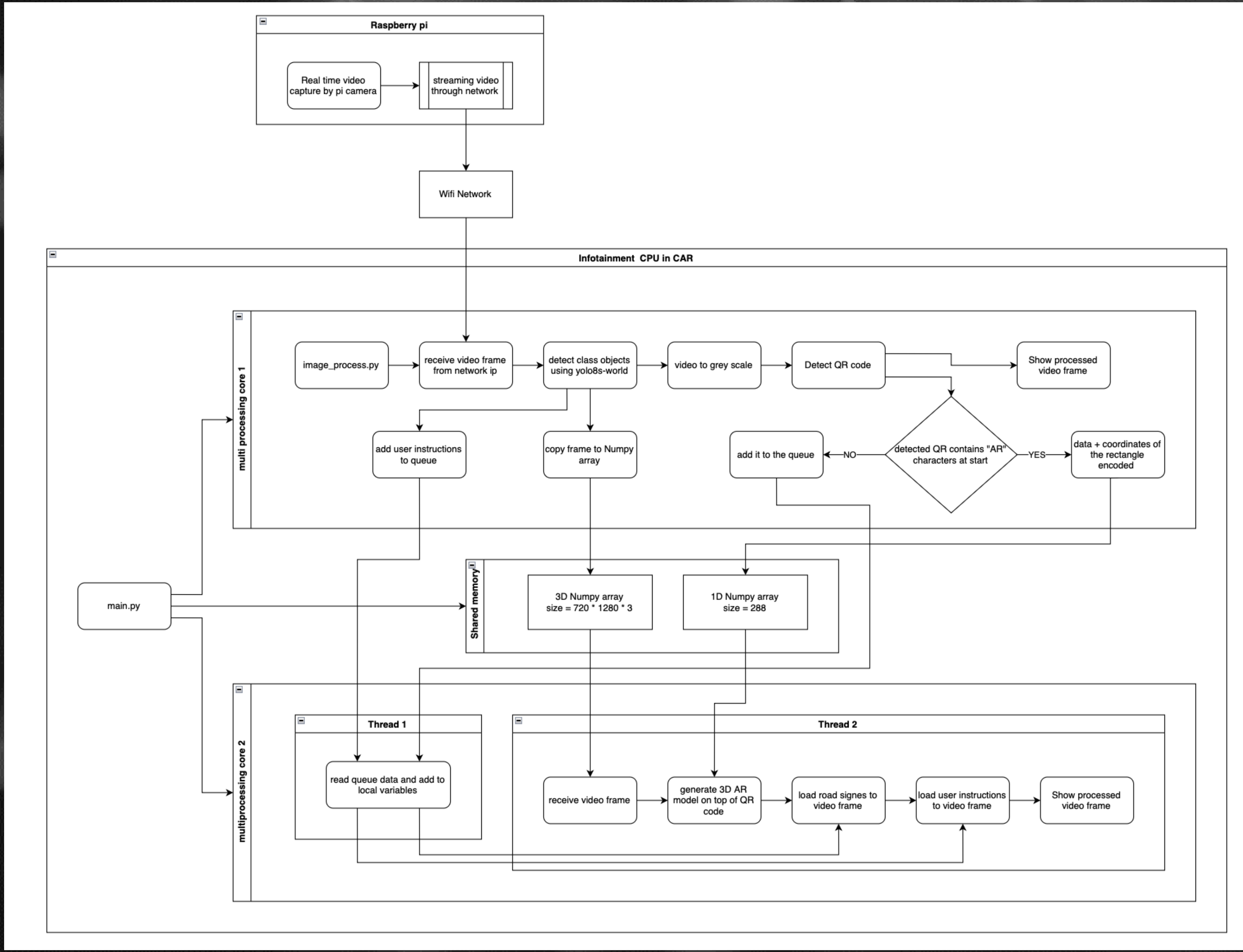
PROBLEM FORMULATION

- Existing road communication systems rely heavily on static signage and physical infrastructure, which may fail to deliver dynamic, real-time guidance in complex or adverse environments.
- Advanced driver-assistance systems (ADAS) integrated in premium vehicles are often costly and not scalable to mass-market or embedded platforms.
- Rendering AR content in real time requires significant computational power, posing a challenge for lightweight devices like Raspberry Pi without onboard acceleration.
- Object detection models (e.g., YOLOv8s) and 3D AR overlays introduce latency and resource conflicts if not parallelized effectively.
- The challenge lies in designing a modular system that combines computer vision, pre-trained AI models, QR code-based cues, and efficient AR rendering — all while maintaining real-time performance on constrained embedded hardware.

PROGRAM ARCHITECTURE




Raspberry PI
Network
Infotainment system CPU


Raspberry pi
Real time video capture by pi camera
streaming video through network
Wifi Network
Infotainment CPU in CAR
multi processing core 1
image_process.py
receive video frame from network ip
detect class objects using yolo8s-world
video to grey scale
Detect QR code
Show processed video frame
add user instructions to queue
copy frame to Numpy array
add it to the queue
NO
detected QR contains "AR" characters at start
YES
data + coordinates of the rectangle encoded
main.py
Shared memory
3D Numpy array size = 720 * 1280 * 3
1D Numpy array size = 288
multiprocessing core 2
Thread 1
read queue data and add to local variables
Thread 2
receive video frame
generate 3D AR model on top of QR code
load road signes to video frame
load user instructions to video frame
Show processed video frame

OVERVIEW OF ANALYTICAL MODEL

- Objective: Real-time detection of road elements using computer vision and AI models on embedded hardware.
- Input: Live video stream from Pi Camera Rev 1.3 (OV5647, 720p @ 30fps).
- Output: AR overlays (2D/3D) aligned with object detections and QR codes.
- Two configuration modes:
  - System 1: Embedded-only inference (Haar Cascade)
  - System 2: Distributed inference (YOLOv8s + Flask streaming)

System 1:

- Uses Haar Cascade Classifiers for lightweight detection (Stop signs, traffic cues).
- Runs locally on Raspberry Pi using OpenCV with ~22–30 FPS at 480p.

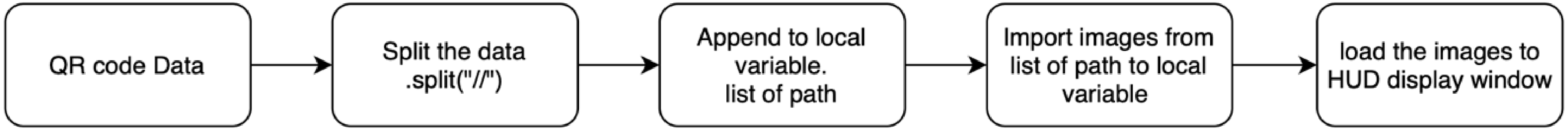
System 2:

- YOLOv8s-world pretrained model from Ultralytics GitHub.
- Detected classes include person, car, truck, traffic light, etc.
- Achieves ~15–30 FPS on remote compute system (Intel i5 / 8 GB).

THEORETICAL MODEL FOR QR & AR RENDERING

QR Code Detection:

- QR decoding handled via OpenCV and PyZBar integration.
- Position-based rendering: AR overlays triggered via decoded metadata.

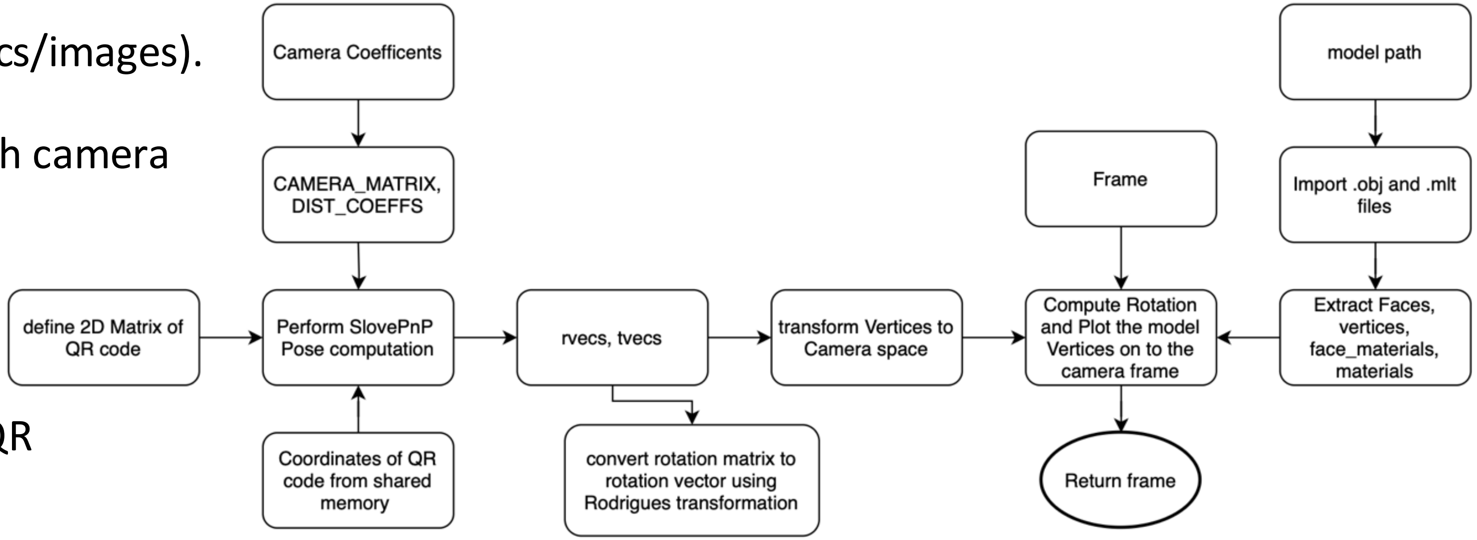


AR RENDERING:

- System 1: 2D overlay (HUD graphics/images).
- System 2: 3D model rendering with camera pose estimation.

Pose Estimation:

- Uses cv2.solvePnP() with known QR coordinates and image points.
- Projects AR object spatially accurate within frame.



PARALLEL EXECUTION ARCHITECTURE

- Multithreading:

Manages live frame acquisition, QR scanning, and AR rendering concurrently.

- Multiprocessing:

YOLOv8s detection offloaded to a separate core to maintain UI responsiveness.

- Shared Memory:

Enables communication between processes (e.g., frame buffer, detection queue).

- Result:

Smooth overlay pipeline (~30 FPS), responsive alerts, minimal lag in both configurations.

HARDWARE/SOFTWARE TOOLS & DESIGN PARAMETERS

Hardware :

1. Raspberry Pi 5	(8Gb) (2.4Ghz quAd core)
2. Pi Camera Module	rev1.3 (5mp) (720p 30fps)
3. Network Connection	Wifi / Mobile hotspot
4. Power supply	5V/5A or 5V/3A(minimum)
5. Development System	(Minimum) intel i5, 8gb, 256GB

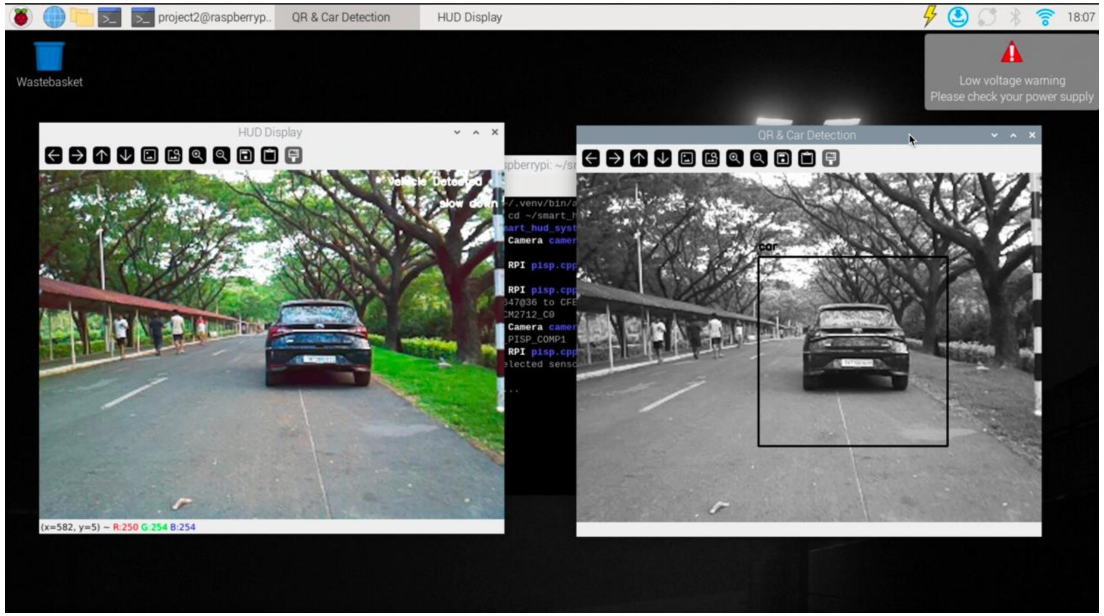
Software libraries/frameworks:

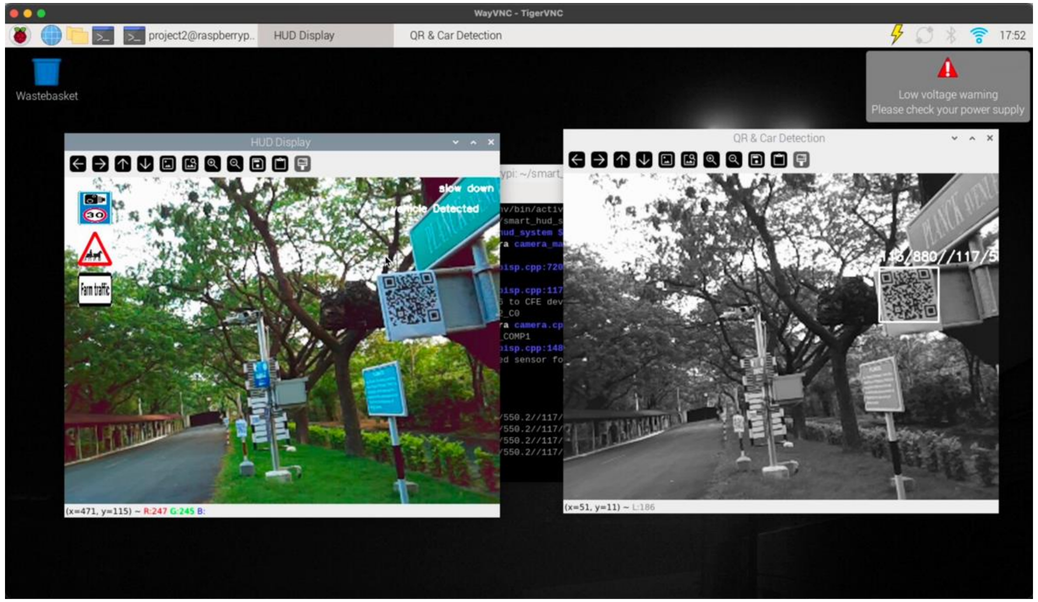
• Python 3.8 - 3.12	• Opencv
• Numpy	• Haar Cascade Classifier
• Ultralytics - YOLO8s-world	• Multiprocessing • Threading
• Opencv	• Tensorflow Or Pytorch
• VS Code • Blender	• Putty • Tiger VNC

RESULTS ANALYSIS

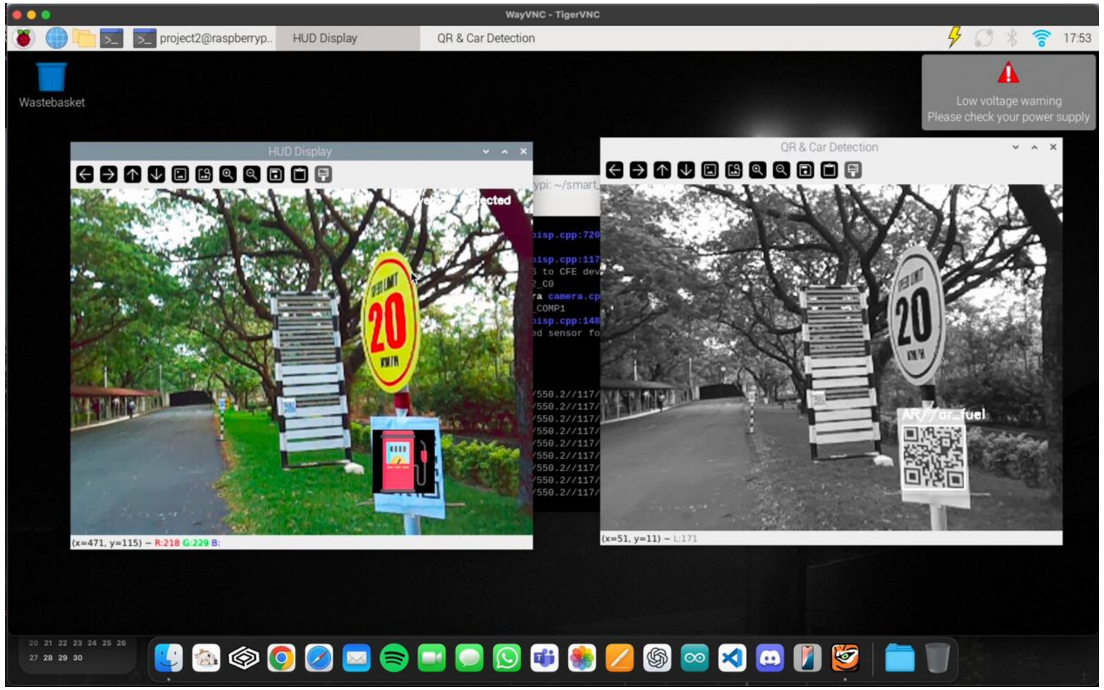
SYSTEM RUNNING ON RASPBERRY PI

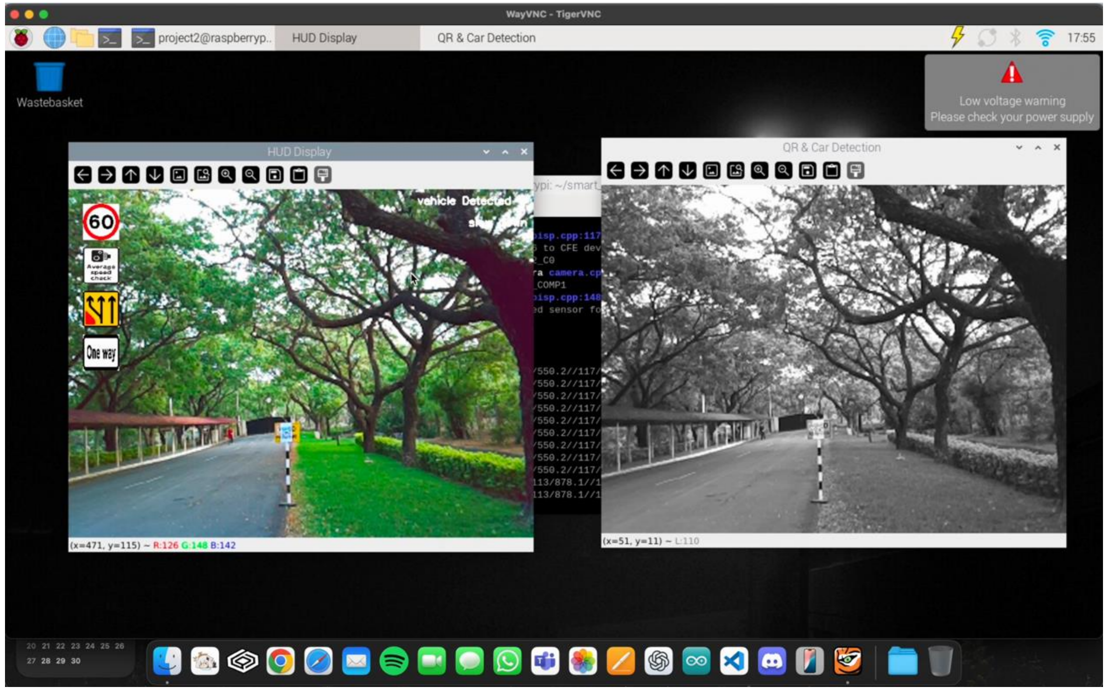
VEHICLE DETECTION WITH HAAR CASCADE CLASSIFIER AND QR CODE DETECTION AND ROAD SIGN UPDATES ON HUD DISPLAY





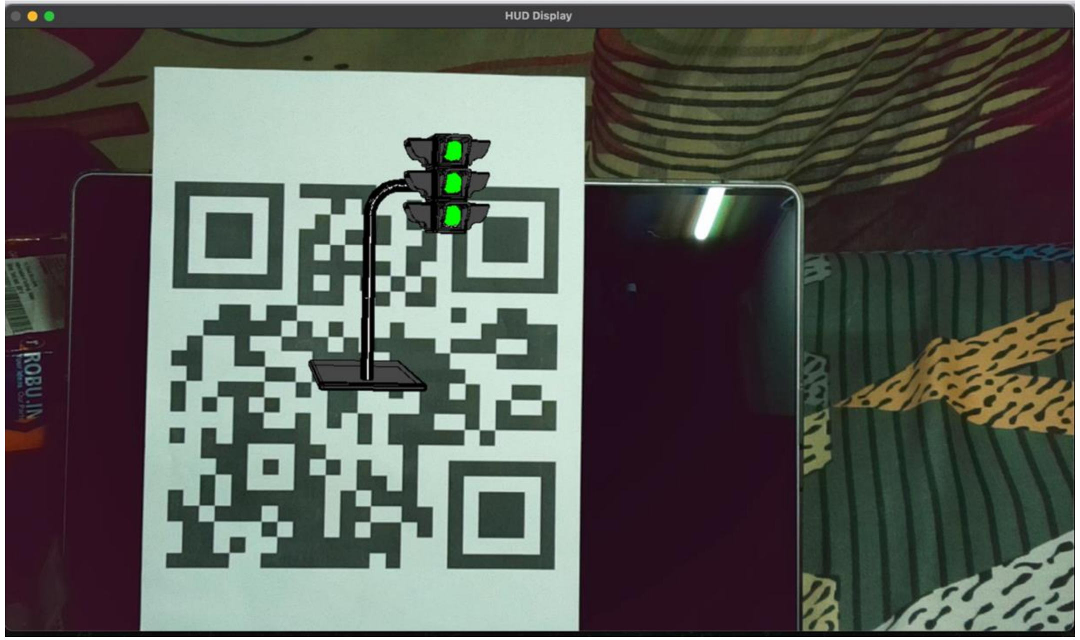
2D AR OVERLAY ON QR CODE



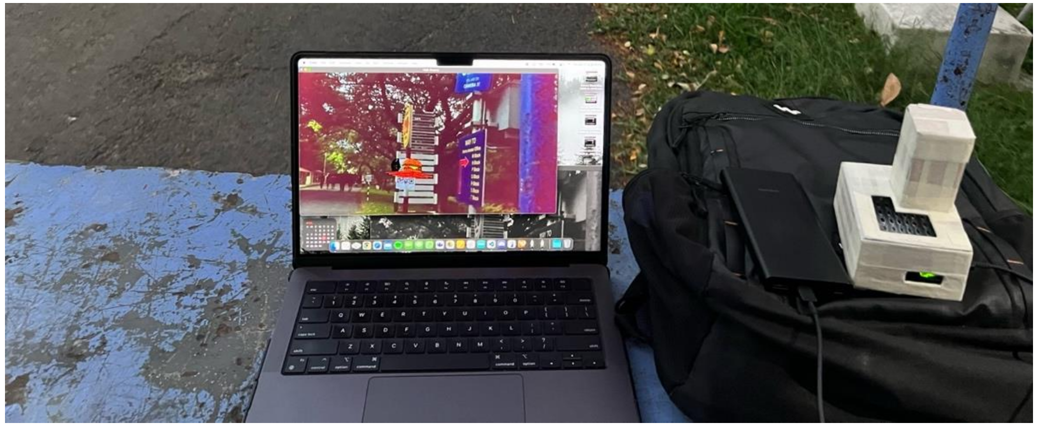


DISTRIBUTED COMPUTATION SYSTEM

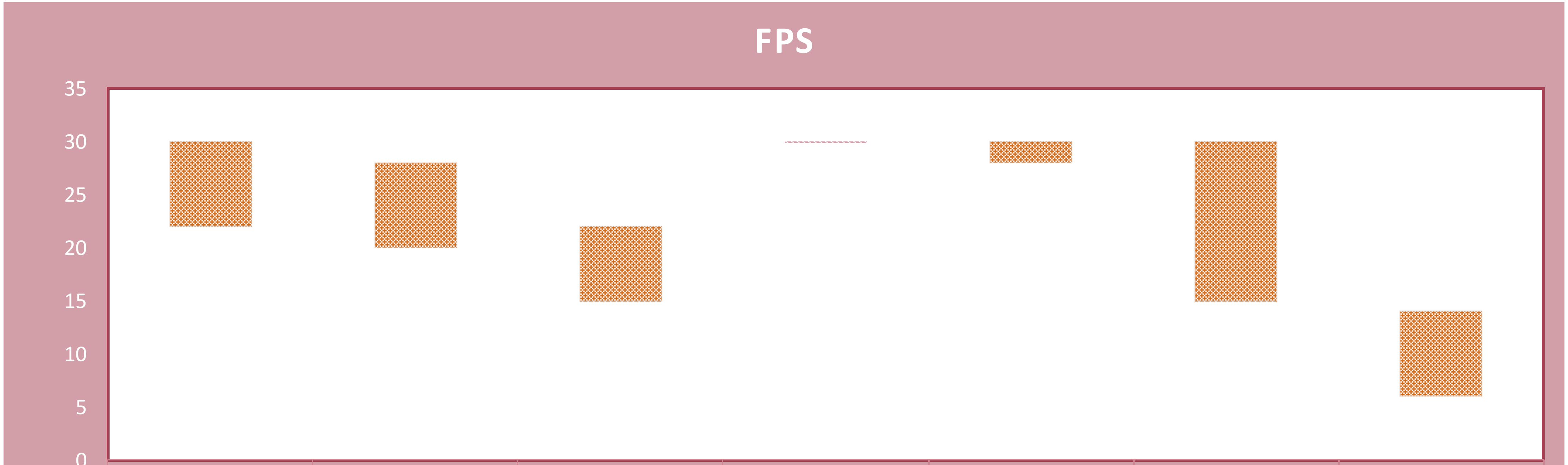
CLASS OBJECT DETECTION WITH YOLO8S-WORLD AND QR CODE DETECTION AND AR GENERATION DEMO



AR GENERATION DEMO AND SETUP



SYSTEM PERFORMANCE ANALYSIS



	System 1 - Idle	System 1 - QR Code Detection	System 1 - Haar Cascade Classifier	System 2 - Idle	System 2 - QR Code Detection	System 2 - YOLOv8s-world	System 2 - 3D model
High	22	20	15	30	28	15	6
Low	30	28	22	30	30	30	14

CONCLUSION

- ✓ Successfully developed a modular Smart Vehicle Infotainment System with AR-HUD functionality using embedded vision.
- ✓ Implemented two working configurations:
  - ➢ System 1: Lightweight, embedded solution using Haar cascade at 480p.
  - ➢ System 2: High-performance mode with YOLOv8s and 3D AR rendering at 720p.
- ✓ Achieved stable real-time detection and AR overlay under constrained hardware using multithreading and multiprocessing.
- ✓ Demonstrated integration of computer vision, AI models, and AR rendering on Raspberry Pi 5.
- ✓ System validated under various conditions, proving feasibility, scalability, and potential for real-world automotive deployment

CONTRIBUTION OF TEAM MEMBERS

Sudharson Aswin N K:

- Implemented core functionalities using Python with integrated AI and CV modules.
- Developed object detection, HUD overlay logic, and real-time QR code processing.
- Tuned system for optimal performance on limited hardware.
- Managed system resource allocation and algorithmic efficiency.

Tushar Sharma:

- Handled hardware procurement and Raspberry Pi remote setup (SSH/VNC).
- Built test environments and validated core features.
- Documented methodology, progress, and final outcomes for future use.

IMPACT OF THE PROJECT ON SOCIETY AND ENVIRONMENT

Social Impact :

- Enhances driver situational awareness through AR-based HUD overlays.
- Reduces reliance on physical infrastructure using computer vision + QR-based cues.
- Applicable in smart cities, emergency zones, and infrastructure-deficient areas.
- Contributes to safer navigation by delivering real-time object and marker detection.

Environmental Impact :

- Minimizes energy consumption via edge processing on embedded hardware (Raspberry Pi 5).
- Reduces need for large-scale roadside installations through deployable QR-code systems.
- Encourages smoother, informed driving, potentially lowering fuel usage and emissions.

PROJECT OUTCOME

This Project is intended for publication in leading journals such as IEEE, as well as for patent consideration.

REFERENCES

[1] Zhang, Y., Chen, L., & Wu, X. (2023, May). A human-vehicle game stability control strategy considering drivers' steering characteristics. IEEE Transactions on Intelligent Transportation Systems, 24(5), 4123–4135. https://doi.org/10.1109/TITS.2023.3245678

[2] Lin, M., Zhao, J., & Wang, H. (2022, November). A real-time three-dimensional tracking and registration method in the AR-HUD system. Sensors, 22(21), 8429. https://doi.org/10.3390/s22218429

[3] Park, J., Nguyen, T., & Kim, H. (2023, February). A robust QR and computer vision-based sensorless steering angle control, localization, and motion planning of self-driving vehicles. IEEE Access, 11, 15432–15447. https://doi.org/10.1109/ACCESS.2023.3241234

[4] Chatterjee, R., & Saxena, A. (2023, June). A unified interactive model evaluation for classification, object detection, and instance segmentation in computer vision. Procedia Computer Science, 224, 950–957. https://doi.org/10.1016/j.procs.2023.06.104

[5] Lee, H., Xu, Z., & Wang, S. (2022, October). An AR-based meta vehicle road cooperation testing system: Framework, components modeling, and an implementation example. IEEE Transactions on Vehicular Technology, 71(10), 10328–10340. https://doi.org/10.1109/TVT.2022.3198743

[6] Torres, M., & He, Y. (2023, March). Comparing state-of-the-art and emerging augmented reality interfaces for autonomous vehicle-to-pedestrian communication. ACM Transactions on Human-Robot Interaction, 12(2), 1–20. https://doi.org/10.1145/3579781

[7] Singh, A., Gupta, M., & Prasad, R. (2024, January). Efficient vehicle detection and classification using YOLO v8 for real-time applications. In 2024 International Conference on Smart Systems and Technologies (SST) (pp. 210–215). IEEE. https://doi.org/10.1109/SST58999.2024.00123

[8] Sharma, K., & Patil, V. (2022, December). Infotainment enabled smart cars: A joint communication, caching, and computation approach. *IEEE Internet of Things Journal, 9*(24), 25981–25990. https://doi.org/10.1109/JIOT.2022.3215678

[9] Mehta, R., & Chauhan, S. (2023, April). Intelligence in parking: QR based earmarking vehicle stability in smart-city. *International Journal of Intelligent Transportation Systems Research, 21*(2), 145–153. https://doi.org/10.1007/s13177-023-00289-z

[10] Zhao, Y., Liu, J., & Gao, R. (2021, July). Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis. *IEEE Transactions on Intelligent Vehicles, 6*(3), 402–420. https://doi.org/10.1109/TIV.2021.3086789

[11] Batra, D., & Kumar, A. (2022, September). Looking-in and looking-out of a vehicle: Computer-vision-based enhanced vehicle safety. *In 2022 5th International Conference on Computer Vision and Image Processing (CVIP)* (pp. 314–321). IEEE. https://doi.org/10.1109/CVIP55445.2022.00059

[12] Al-Qaness, M. A. A., & Fan, H. (2023, February). Metaverse for intelligent transportation systems (ITS): A comprehensive review of technologies, applications, implications, challenges and future directions. *IEEE Access, 11*, 15023–15045. https://doi.org/10.1109/ACCESS.2023.3243456

[13] Tiwari, S., & Singh, P. (2022, June). Motion-based vehicle surround analysis using an omni-directional camera. *Procedia Computer Science, 200*, 1236–1243. https://doi.org/10.1016/j.procs.2022.01.298

[14] Verma, M., Sharma, T., & Sinha, R. (2023, August). Performance evaluation of YOLO models in varying conditions: A study on object detection and tracking. *Journal of Big Data, 10*(1), 57. https://doi.org/10.1186/s40537-023-00763-w

[15] Roy, A., Dutta, M., & Saha, B. (2021, November). QR-code traffic signs recognition for ADAS smart driving system. *In 2021 International Conference on Robotics, Intelligent Automation and Control Technologies (RIACT)* (pp. 146–150). IEEE. https://doi.org/10.1109/RIACT54277.2021.9669037

[16] Chen, X., & Wu, Y. (2020, December). The role of machine vision for intelligent vehicles. *IEEE Transactions on Intelligent Transportation Systems, 21*(12), 5231–5243. https://doi.org/10.1109/TITS.2020.2998711

[17] Kim, T., & Park, D. (2022, July). Vehicle detection by means of stereo vision-based obstacles features extraction and monocular pattern analysis. *Pattern Recognition Letters, 156*, 25–32. https://doi.org/10.1016/j.patrec.2022.01.006

[18] Ramesh, R., & Patel, S. (2023, May). Vision-based vehicle guidance. *Journal of Intelligent & Robotic Systems, 107*, 25–38. https://doi.org/10.1007/s10846-023-01796-1

THANK YOU