ML-based Traffic Detection and Autonomous Signal Operation

Vaibhav Wagh[1], Nikhil Terkar[2], Rohit Terkar[3], Khushi Sharma[4], Dr. Girisha Bombale[5]

Department of Computer Engineering, SND COE & RC, Yeola, Maharashtra, India.

Abstract: This research paper presents the development and real-time implementation of an AI-based traffic detection and autonomous signal operation system using the YOLOv5 object detection algorithm. The system aims to dynamically manage traffic flow based on vehicle density, thereby improving road efficiency and reducing congestion. Unlike prior approaches that relied on Raspberry Pi and external cameras, the proposed setup utilizes a standard laptop for processing live video input and an Arduino microcontroller to operate physical traffic signals via LEDs. The YOLOv5 model is employed for accurate and fast detection of vehicles such as cars, buses, trucks, and motorcycles using a webcam. Based on the detected vehicle count, green signal duration is adjusted dynamically in real-time. The Arduino board receives control commands (G, Y, R, and O) via serial communication to reflect the traffic signal states using LEDs. This low-cost, modular prototype offers a practical and scalable solution for smart traffic management in urban as well as rural environments. The paper also presents system architecture, hardware/software integration, and experimental results validating the effectiveness of the proposed solution.

Keywords: AI-based Traffic Control, YOLOv5, Real-Time Object Detection, Arduino, Serial Communication, Traffic Signal Automation, Vehicle Detection, Smart Traffic System.

I. INTRODUCTION

Traffic congestion is a growing issue in urban areas around the world, leading to increased travel times, pollution, and accidents. Traditional traffic management systems, which rely on fixed-time or sensor-based signal control, often struggle to adapt to real-time traffic conditions, exacerbating these challenges. As cities continue to expand, the need for intelligent, efficient traffic management solutions has become increasingly urgent.

Artificial Intelligence (AI) and machine learning offer innovative solutions to optimize traffic control by enabling systems to respond dynamically to real-time traffic data. This paper proposes an AI-based traffic detection and autonomous signal operation system using the YOLO (You Only Look Once) algorithm for vehicle detection and Raspberry Pi as the core processing unit. By leveraging live traffic footage captured by cameras, the system analyzes vehicle density and adjusts traffic signals accordingly, thereby reducing congestion and improving traffic flow.

This research explores the integration of image processing, machine learning, and real-time traffic control, presenting a scalable and adaptable solution for both high-traffic urban environments and less congested areas. The use of the YOLO algorithm allows for accurate and efficient vehicle detection, while the Raspberry Pi ensures cost-effective implementation, making the system suitable for deployment in a wide range of traffic scenarios.

II. RELATED WORK

Image-Based Traffic Detection Systems

In more recent research, image-based traffic detection has gained traction due to the proliferation of cameras in urban environments. **R. Cucchiara et al. (2000)**, in *Image analysis and rule-based reasoning for a traffic monitoring system*, pioneered the use of video surveillance for vehicle detection and traffic monitoring, employing feature extraction techniques like background subtraction and optical flow. However, these methods had limitations in low-light conditions and weather changes, which led to the development of more advanced image processing techniques.

Machine Learning and Deep Learning for Traffic Detection

Machine learning techniques have greatly improved traffic detection capabilities. **Cai et al. (2016)**, in *Vehicle detection and classification using convolutional neural networks*, demonstrated how convolutional neural networks (CNNs) could be applied to traffic footage for vehicle detection and classification. The research showed high detection accuracy, but the computational complexity of CNNs required expensive hardware for real-time deployment, which limited their widespread adoption.

Building on the use of CNNs, **Joseph Redmon et al. (2016)** introduced the **YOLO (You Only Look Once)** algorithm in their paper *You Only Look Once: Unified, Real-Time Object Detection*. YOLO is a breakthrough in object detection that processes the entire image in one pass, enabling real-time detection of multiple objects, including vehicles, with high accuracy. Unlike earlier methods that required scanning an image multiple times (e.g., region-based CNNs), YOLO's single-stage detection framework offers superior speed and efficiency, making it ideal for real-time traffic detection applications.

AI-Driven Traffic Signal Control

Traffic signal control using AI has been an area of active research. **Ceyhun Genc and D. van der Schaar (2010)**, in *Dynamic traffic signal control with reinforcement learning*, proposed an adaptive traffic control system based on reinforcement learning, which could dynamically adjust signal timings based on real-time traffic conditions. The system outperformed traditional fixed-time control methods by reducing vehicle waiting times at intersections, but the complexity and computational cost limited its practicality for real-world deployment in resource-constrained environments.

Similarly, **K. Fu et al. (2018)**, in *Multi-agent deep reinforcement learning for traffic signal control*, explored the use of multi-agent reinforcement learning to optimize traffic signals across a network of intersections. This approach demonstrated significant improvements in traffic flow but required extensive computation power and real-time data availability, limiting its feasibility in developing regions or smaller municipalities.

Raspberry Pi and Low-Cost IoT Solutions

Recent works have aimed to develop cost-effective traffic management systems using microcontrollers. **M. Abdoos et al. (2014)**, in *Traffic control in a smart city using IoT-based devices*, proposed an Internet of Things (IoT) architecture utilizing low-cost microcontrollers like the Raspberry Pi for real-time data collection and signal control. Although their study focused on sensor-based traffic management, it demonstrated the feasibility of using Raspberry Pi for traffic applications due to its affordability and processing power.

*Building on these advancements, **P. Bhadani and D. Jagtap (2020)**, in Intelligent traffic management system using Raspberry Pi and OpenCV, integrated computer vision with a Raspberry Pi to detect traffic patterns and manage signals in real-time. Their system demonstrated the potential for low-cost, scalable implementations in urban and rural areas, though vehicle detection accuracy was still a challenge, particularly in adverse weather conditions.*

III. SYSTEM ARCHITECTURE AND PROPOSED SYSTEM

1. Proposed System

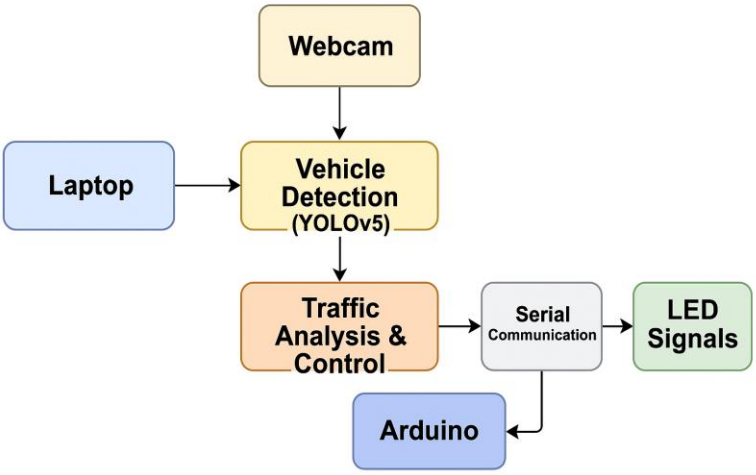


Fig. Architecture Block Diagram

The proposed system is a real-time, intelligent traffic detection and control solution designed using accessible and cost-effective components. A standard laptop serves as the central processing unit, running a Python-based application that captures live video feed through a connected webcam. Each frame is analyzed using a pre-trained YOLOv5n object detection model to detect vehicles such as cars, buses, trucks, and motorcycles.

The system dynamically calculates traffic density based on the number of detected vehicles and adjusts the duration of the green traffic signal accordingly. Traffic signal control is achieved using an Arduino UNO, which receives commands from the laptop via serial communication. These commands correspond to signal states—green, yellow, or red—and control three LEDs connected to the Arduino to simulate traffic light behavior.

The overall architecture comprises three main modules:

- **Vehicle Detection Module**: Captures and processes video frames using OpenCV and YOLOv5 for real-time object detection.
- **Traffic Analysis and Control Module**: Computes traffic density and determines signal timings based on detection output.

- **Signal Output Module**: Transmits control signals via serial to the Arduino, which activates the appropriate LED to represent signal status.

This setup offers an adaptive, low-cost alternative to conventional traffic control systems, with potential for scalability, integration with IoT infrastructure, and real-world deployment in smart city environments.

2. Hardware Components

2.1 Processor:

A standard laptop serves as the core processing unit of the system. It handles live image capture, vehicle detection using a pre-trained YOLOv5 model, and traffic density analysis. The laptop runs Python scripts to process input from the camera and sends control signals to the microcontroller for operating the traffic lights. Its higher processing power compared to embedded boards allows for faster real-time detection and decision-making.

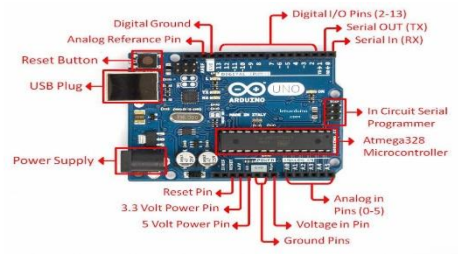
2.2 USB Camera or Webcam:

A USB camera or built-in laptop webcam is used to capture real-time traffic images. The captured images are processed using machine learning algorithms to detect vehicles and estimate traffic density accurately.

2.3 Traffic Light LED's:

Red, yellow, and green LEDs represent the traffic lights in the hardware prototype. These are connected to the Arduino UNO and are controlled based on the vehicle density detected by the system.

2.4 Arduino UNO (Traffic Signal Controller):



The Arduino UNO acts as the interface between the laptop and the physical traffic lights. It receives serial commands from the laptop and controls the red, yellow, and green LEDs accordingly. It ensures a smooth and timely transition of traffic signals based on the analysis.

2.5 Power Supply:

The system is powered by a stable power source connected to the Raspberry Pi, ensuring continuous operation even during peak traffic hours.

3. Software Components

- **YOLOv5 (You Only Look Once) Algorithm:**
  The YOLOv5 object detection algorithm is utilized for detecting vehicles in real-time from the live camera input. It identifies and classifies objects such as cars, buses, motorcycles, and trucks with high accuracy and speed, enabling dynamic traffic analysis based on vehicle density.
- **OpenCV (Open Source Computer Vision Library):**
  OpenCV is employed for handling all image and video processing operations. It captures frames from the live feed, preprocesses the input for the YOLOv5 model, overlays vehicle detection information, and displays the processed output to the user in real-time.
- **Python Scripts:**
  Python serves as the core programming language for the system. Custom scripts manage the entire workflow including video capture, vehicle detection using YOLOv5, calculation of traffic density, and communication with the Arduino UNO via serial interface to control the traffic lights accordingly.
- **PyTorch:**
  PyTorch, a deep learning framework, is used to load and run the pre-trained YOLOv5 model. It enables efficient and scalable inference on the laptop.
- **Serial Communication Library (PySerial):**
  The PySerial library is used to establish and manage communication between the laptop and Arduino UNO over a serial port, allowing real-time transmission of control commands for the traffic signal operation.

4. Traffic Density Estimation

Once the vehicles are detected using the YOLO model, the system counts the number of vehicles in each frame. Traffic density is estimated based on the number of vehicles present in each frame. The system classifies the traffic into categories such as "Low," "Medium," or "High" traffic density. This information is used to dynamically adjust the traffic signal timings.

5. Signal Control Logic

Based on the number of vehicles detected in real-time, the system dynamically adjusts the traffic light durations to improve traffic flow efficiency:

1) **Green Signal Duration:** When the system detects high vehicle density, it increases the green signal duration to allow more vehicles to pass. For lower traffic conditions, the green time is reduced to avoid unnecessary delays and to provide timely access to other lanes.
2) **Red Signal Duration:** The red light duration for the opposite or intersecting directions is automatically adjusted based on the green time of the current lane, ensuring fairness and continuous flow across the junction.
3) **Dynamic Real-Time Adaptation:** The system recalculates the vehicle count every few seconds using the live camera feed. It then updates the traffic light timings in real-time to respond promptly to changing traffic conditions, enabling adaptive traffic control.

6. Communication Flow

1) **Video Input:** A webcam or USB camera connected to the laptop continuously captures the live traffic feed.
2) **Image Processing:** The captured frames are processed using OpenCV. Each frame is resized, preprocessed, and then passed to the YOLOv5 object detection model.
3) **Vehicle Detection and Counting:** The YOLOv5 model detects and classifies objects such as cars, trucks, buses, and motorcycles. The number of detected vehicles is counted to estimate traffic density.
4) **Decision Making:** Based on the number of vehicles detected, the laptop runs a decision-making algorithm in Python to determine the appropriate green signal duration. The logic ensures shorter waits during light traffic and longer green signals during congestion.
5) **Signal Output via Arduino:** The laptop sends control characters ('G', 'Y', 'R', 'O') over a serial connection to the Arduino UNO. The Arduino receives these commands and turns on the corresponding LED (green, yellow, or red) to simulate real-world traffic light behavior.

7. Data Flow Diagram

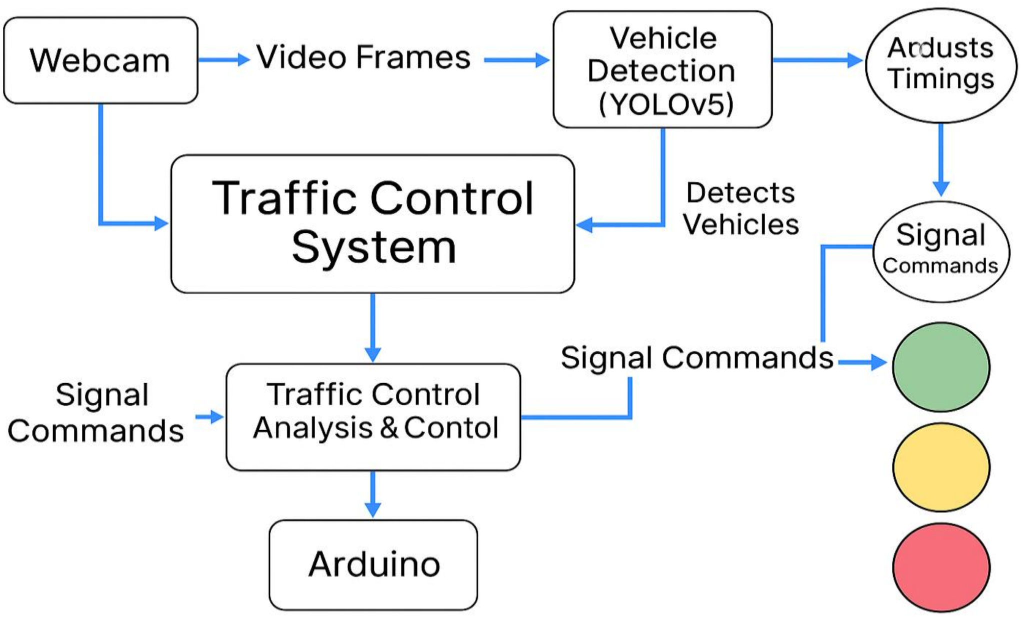
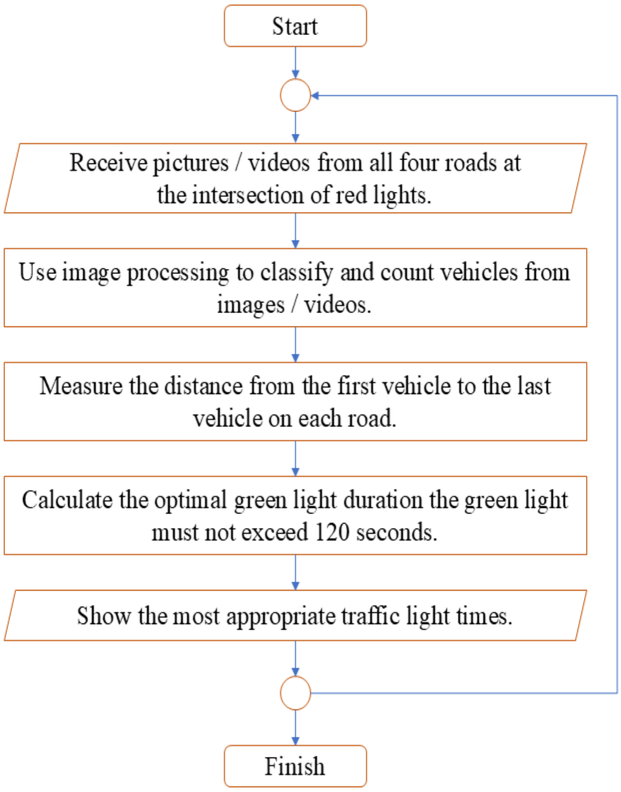


Fig. Data Flow Diagram

8. Algorithm



IV. EXPERIMENTAL RESULT

To validate the functionality and efficiency of the proposed system, a prototype was developed using a laptop for real-time object detection and an Arduino UNO for traffic signal simulation using LEDs. The system was tested under various traffic scenarios to measure its responsiveness, accuracy, and signal timing behavior.

A. Test Setup

- **Hardware:**
  - Laptop with Intel CPU (no GPU)
  - Arduino UNO connected via USB
  - Three LEDs (Red, Yellow, Green) with 220Ω resistors
  - USB webcam or laptop's built-in webcam
- **Software:**
  - Python 3.10
  - OpenCV for live video processing
  - PyTorch for loading YOLOv5n
  - PySerial for communication between laptop and Arduino
- **Detection Classes Used:** Car, Bus, Truck, Motorcycle
- **Confidence Threshold:** 0.4

The YOLOv5 model was configured to detect specific vehicle types from the video stream. The detected vehicle count was used to dynamically adjust the green signal duration, as shown in Table below.

B. Signal Timing Strategy

Vehicle Count Detected	Green Signal Time (sec)
0	5
1-3	10
4-6	15
>6	20

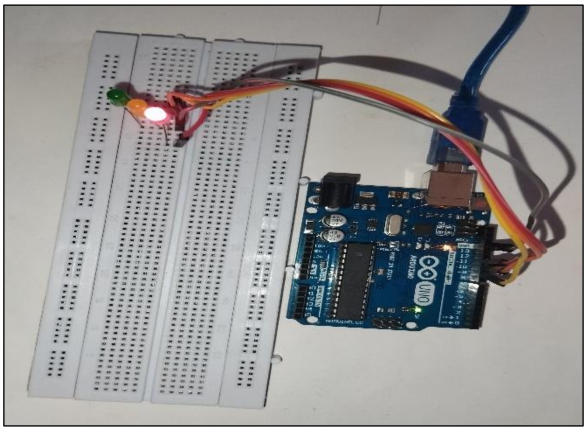
Yellow and red signals were fixed at 3 and 5 seconds respectively.

C. Live Testing Observations

The system was tested in different indoor scenarios with simulated vehicles (e.g., toy cars, printed photos on paper, or images on a screen). The following observations were recorded:

- **Detection Accuracy:** The YOLOv5n model was able to accurately detect multiple vehicles in the frame with minimal false positives.
- **Responsiveness:** Green signal time changed correctly based on the number of vehicles in each frame.
- **Smooth Video Feed:** Frame skipping and multithreading helped ensure that video feed remained smooth without lag, even on CPU-only systems.
- **Hardware Output:** The Arduino LEDs reflected the traffic light logic in real-time. Green, yellow, and red LEDs lit up according to detected traffic conditions.

E. Screenshots of the Prototype



F. Performance Summary

Test Condition	Result
Detection Delay	< 1 second
Frame Resolution	480 × 360 (optimized for CPU)
Serial Communication Lag	Negligible
Max Vehicle Load Tested	8 (on a printed image)

V. ADVANTAGES AND DISADVANTAGES

1. Advantages

- Enables real-time traffic signal control reducing congestion and emissions.
- Enhances road safety and encourages disciplined driving.
- Supports efficient traffic management in rapidly urbanizing cities.
- Facilitates smart, data-driven planning and decisions.
- Saves time and increases operational efficiency at intersections.
- Reduces commuter wait times and improves user satisfaction.
- Contributes to better air quality and public health.
- Minimizes road accidents by optimizing signal timing.
- Operates autonomously without human intervention.
- Uses YOLOv5 for fast, accurate multi-vehicle detection.
- Cost-effective using a laptop and Arduino instead of embedded boards.
- Aligns with smart city goals for intelligent transportation systems.

2. Disadvantages

- Requires upfront investment in hardware components.
- Detection accuracy can drop in low-light or adverse weather.
- No fallback mechanism in case of hardware or communication failure.
- YOLO model may require retraining for different environments.
- Some human supervision still needed during initial deployment.

3. Applications

Artificial intelligence-based systems can be a smart investment for governments looking to get the most out of their restricted funds for streets and roads. The ability to manage more traffic and a growing population while lowering the capital expenses of creating new or rebuilt roadways is made feasible by them. Only 15% of the world's population was residing in metropolitan regions a century ago. More than fifty-five percent of people live in cities today, and sixty-eight percent are projected to do so in 2050. Hence urbanization needs smart traffic system implementation. With all the development comes an equal rise in traffic, necessitating the use of technology to help control the movement of people and commodities. The AI based system can also help the traffic agencies in penalizing the offenders and act as deterrent. The cameras installed can detect and identify the vehicles which disobey traffic rules and fine them electronically. The AI based traffic system can cut-down the waiting time at traffic signals by almost half. Hence there is a scope and need for smart traffic implementation in real world to ease clogged roads and cope up with growing number of cars.

VI. CONCLUSION

The integration of AI-based traffic detection and autonomous signal operation using a laptop, real-time image processing, and an Arduino-controlled LED system presents a practical and cost-effective advancement in smart traffic management. By utilizing a standard webcam and the YOLOv5 object detection model, the system dynamically adjusts traffic signal timings based on real-time vehicle density without the need for expensive embedded platforms. The use of Arduino for traffic light simulation via serial communication enables accurate and responsive physical signal control, making the prototype highly accessible for educational and research purposes. This implementation demonstrates that complex AI models can be effectively deployed on conventional computing hardware while maintaining real-time performance. The system not only reduces traffic congestion by adapting to current conditions but also lays a foundation for scalable, modular, and locally deployable smart traffic solutions.

VII. FUTURE WORK

In the future, the AI-based traffic detection and autonomous signal control system can be enhanced by integrating cloud platforms for real-time data collection, analytics, and storage. This would allow the system to learn from long-term traffic patterns, improving its decision-making capabilities through predictive modeling. Expanding the solution to coordinate multiple intersections could lead to city-wide traffic optimization, helping reduce congestion on a broader scale. Additionally, incorporating vehicle classification would allow the system to identify and prioritize specific types of vehicles, such as emergency services and public buses, thereby improving emergency response times and public transport efficiency.

Further enhancements could include integrating the system with broader smart city infrastructure by connecting it to Internet of Things (IoT) networks like public transportation systems, environmental sensors, and weather monitoring tools. Such integration would enable multi-dimensional decision-making and more responsive traffic control. Moreover, deploying more advanced AI models on edge computing platforms, such as updated Raspberry Pi or NVIDIA Jetson devices, would reduce reliance on centralized systems, minimize latency, and support real-time operations more effectively. These advancements would improve the scalability, efficiency, and adaptability of the system for diverse urban and semi-urban environments.

REFERENCES

[1] Gong, Suning & Huan, Zhiying & JI, Mingmei & Chen, Xinxin & Bao, Yuanqiu. (2021). “ITLCS Based on OpenCV Image Processing Technology”. Journal of Physics: Conference Series. 2143. 012031. 10.1088/1742-6596/2143/1/.

[2] Sharma Moolchand and Bansal, Ananya and Kashyap, Vaibhav & Goyal, Pramod & Sheakh, Dr.Tariq (2020). “Intelligent Traffic Light Control System Based on Traffic Environment Using Deep Learning”.

[3] N. Seenouvong, U. Watchareeruetai, C. Nuthong, K. Khongsomboon and N. Ohnishi, "A computer vision based vehicle detection and counting system," 2016 8th International Conference on Knowledge and Smart Technology (KST), 2016, pp. 224-227, doi: 10.1109/KST.2016.7440510.

[4] Tanvi Sable, Nehal Parate, Dharini Nadkar, Swapnil Shinde. “Density and Time-based Traffic Control System using Video Processing”. ITM Web of Conferences 32, 03028 (2020).

[5] S. Zhao and F. You, "Vehicle Detection Based on Improved Yolov3 Algorithm," 2020 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), 2020, pp. 76-79, doi: 10.1109/ICITBS49701.2020.00024.

[6] O.I. Olayode, L.K. Tartibu, M.O. Okwu, Application of Artificial Intelligence in Traffic Control System of Non-autonomous Vehicles at Signalized Road Intersection., Procedia CIRP, Volume 91, 202Pages 194- 200, ISSN 2212-8271

[7] V. Zinchenko, G. Kondratenko, I. Sidenko and Y. Kondratenko, "Computer Vision in Control and Optimization of Road Traffic," 2020 IEEE Third International Conference on Data Stream Mining & Processing (DSMP), 2020, pp. 249-254, doi: 10.1109/DSMP47368.2020.9204329.

[8] B. Ghazal, K. ElKhatib, K. Chahine, and M. Kherfan, “Smart traffic light control system,” in Proc. 3rd Int. Conf. (EECEA), Apr. 2016, pp. 140-145.

[9] C. K. Toh, J. A. Sanguesa, J. C. Cano, and F. J. Martinez, “Advances in smart roads for future smart cities,” Roy. Soc. Publ. J., vol. 476, no. 20190439, pp. 1-24, Dec. 2019.

[10] S. Singh, “Critical reasons for crashes investigated in the national motor vehicle crash causation survey,” Tech. Rep., 2015.

[11] R. Lanctot et al., “Accelerating the future: The economic impact of the emerging passenger economy,” Strategy Analytics, vol. 5, 2017.