Information System and Smart City 2025, 5(1), 2253.
https://doi.org/10.59400/issc2253

Article

Energy-optimizing machine learning-driven smart traffic control system for urban mobility and the implications for insurance and risk management

Chizoba P. Chinedu[1], Queensley C. Chukwudum[2,*], Eberechukwu Q. Chinedu[3]

[1] Department of Mechatronics, Nile University, Abuja 900001, Nigeria
[2] Department of Insurance and Risk Management, University of Uyo, Uyo 520003, Nigeria
[3] Godfrey Okoye University, Enugu 400001, Nigeria
*** Corresponding author:** Queensley C. Chukwudum, queensleyv@yahoo.com

CITATION

Chinedu CP, Chukwudum QC, Chinedu EQ. Energy-optimizing machine learning-driven smart traffic control system for urban mobility and the implications for insurance and risk management. Information System and Smart City. 2025; 5(1): 2253. https://doi.org/10.59400/issc2253

ARTICLE INFO

Received: 10 December 2024
Accepted: 17 February 2025
Available online: 27 February 2025

Abstract: Heavy traffic during peak hours, such as early mornings and late evenings, is a significant cause of delays for commuters. To address this issue, the prototype of a dual smart traffic light control system is constructed, capable of dynamically adjusting traffic signal duration based on real-time vehicle density at intersections, as well as the brightness of the streetlights. The system uses a pre-trained Haar Cascade machine learning classifier model to detect and count vehicles through a live video feed. Detected cars are highlighted with red squares, and their count is extracted. The vehicle data is then transmitted to an Arduino microcontroller via serial communication, facilitated by the pySerial library. The Arduino processes this information and adjusts the timing of the traffic lights accordingly, optimizing traffic flow based on current road conditions. A novel approach involves optimizing energy usage through real-time data integration with the power grid. Street lighting is then dynamically adjusted at night times—brightening during high-traffic periods and dimming during low-traffic times. The brightness levels are set at 30%, 50%, 75%, and 100% based on the number of cars detected, with above 50% indicating the presence of cars. This adaptive control enhances energy efficiency by reducing energy consumption while maintaining road safety. The simulated and experimental results are provided. The former demonstrated a lower accuracy compared to the latter, particularly during the transition to the green light, across all traffic density levels. Additionally, the simulation was only capable of representing discrete lamp brightness levels of 0%, 50%, and 100%, in contrast to the experimental results, which showed a clear differentiation between 50%, 75%, and 100% brightness levels. Details of the model limitations are outlined with proposed solutions. The implications of the optimized system for auto insurance, liability coverage, and risk management are explored. These are areas that are rarely addressed in current research.

Keywords: Arduino microcontroller; traffic density estimation; liability coverage; image sensor

1. Introduction

Despite the importance of road transportation, the story is quite sad and different within Africa, particularly sub-Saharan Africa. Its road safety performance is the worst globally [1], and as a result, road fatalities have become one of the leading causes of death on the continent [2,3], inflicting lifelong injuries and disabilities, as well as being responsible for thousands of deaths and economic losses. To worsen the situation, mobility structures and services are not safe or sustainable. These have a significant potential of stunting the economic growth of African countries.

Vehicular networks, a key component of intelligent transportation systems, enable communication between vehicles, infrastructure, and other network entities to enhance road safety, traffic management, and driving efficiency. These networks leverage wireless technologies such as Dedicated Short-Range Communications (DSRC) and cellular-based Vehicle-to-Everything (C-V2X) for real-time data exchange [4]. Research has explored routing protocols to improve data transmission reliability in dynamic vehicular environments, with position-based and cluster-based protocols being prominent solutions [5]. Security and privacy remain significant challenges, as vehicular networks are susceptible to cyber threats like eavesdropping and spoofing [6]. Integrating vehicular networks with edge and cloud computing has further enhanced their capabilities by enabling low-latency data processing and intelligent decision-making [7]. Machine learning techniques are increasingly being applied to optimize network performance, traffic flow prediction, and anomaly detection [8]. These advancements have paved the way for the development of smart traffic systems, which rely on vehicular networks to improve traffic efficiency, reduce congestion, and enhance urban mobility.

Particularly in metropolitan regions, traffic congestion during peak hours has grown to be a serious problem, causing delays, higher fuel usage, and more pollution. Often, depending on set signal timings, traditional traffic control systems lack the flexibility to manage changing traffic patterns. In order to promote sustainable urban mobility, several studies have investigated creative ideas to maximize traffic flow, including microcontroller-based and machine learning-driven devices.

Several researchers have developed smart traffic control systems using microcontrollers, such as the Atmega 32 and the Arduino Mega 2560. The ability of these microcontrollers to effectively control traffic as made evident in the simulation, shows their appropriateness for road traffic control [9]. Some of these microcontrollers can further be customized to detect traffic violators [10]. Additionally, Fadoro [11] highlighted that microcontrollers can serve as a "training kit in learning traffic light control system design and operation" and a "teaching aid in schools for various road users".

Some studies have incorporated machine learning (ML) technology in their designs. For example, Bisen et al. [12] implemented ML in assisting ambulances to navigate through clogged traffic, while Balasubramanian et al. [13] introduced an accident alert sound system into their ML-based IoT Adaptive Traffic Management (ATM) architecture. Tiwari [14] provided a data-based ML solution drawn from transportation data and accounting for seasonality. The You Only Look Once (YOLO) model integrated with a single image processing via a neural network was developed by Khan et al. [15]. The type of vehicle (two-wheelers, four-wheelers, etc.), road width, and junction crossing time were some of the real-time parameters accounted for in their model. Other interesting constructions include the use of radio frequency identification, which was presented by Lanke and Koul [16] as a less expensive form of traffic control. The use of images and video-based information models focusing on a computer vision-based traffic control system was proposed by Abbas [17].

These solutions have greatly increased traffic efficiency and lowered energy usage, yet still there are gaps. Current designs ignore the possibility of maximizing streetlight brightness in concert with traffic control at night times, when proper illumination is absolutely essential for guaranteeing visibility and safety, particularly during intense traffic congestion. Moreover, the effects of improved traffic systems for sectors such as risk management, liability coverage, and motor insurance have not gotten the needed attention in current studies.

This work attempts to close these gaps by providing a two-fold contribution to the literature. First, we develop a prototype of a dual smart traffic light control system that combines dynamic streetlight brightness optimization with real-time traffic signal modification. Aside from enhancing the visibility for road users, the optimized streetlight brightness allows for pedestrians and cyclists to be more visible, reducing the risk of collisions in mixed-use traffic areas common in African cities. Second, we explore the implications of smart traffic systems for auto insurance, liability coverage, and risk management, providing a complete modern traffic management solution.

The remaining part of the paper is as follows. Section 2 provides details of the material and methods used, while section 3 presents the diagrammatic flow. In section 4, real-world tests and simulations are conducted. Section 5 presents the broader impacts for insurance and risk management, and we conclude in section 6.

2. Material and methods

This research work is based on the design and implementation of a smart traffic light circuit with a detection system made from the following materials: Arduino Uno, light-emitting diodes (LED), and a 74HC595 shift register.

2.1. Arduino uno microcontroller

The Arduino Uno microcontroller (**Figure 1**) receives data relating to the number of cars on a road from a pre-trained Haar Cascade classifier via the PySerial library in Python. Based on this information, it sends digital signals to control the traffic light LEDs and the shift register (74HC595), which drives the street lamps. The Arduino operates using a pre-programmed code written by Sobral [18] in the Arduino IDE.

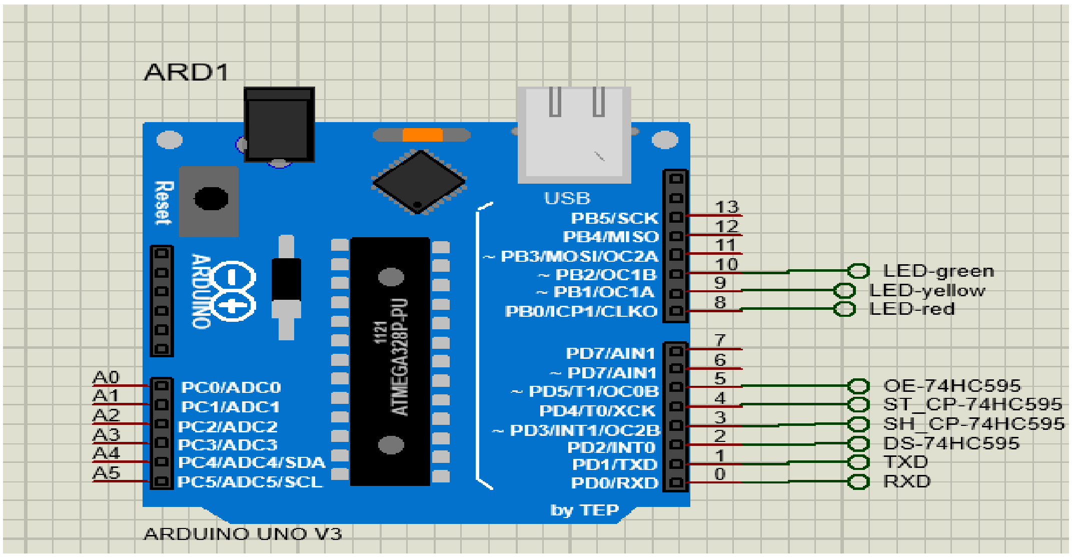


Figure 1. Arduino Uno microcontroller serving as the central processing unit.

2.2. Configuration of the COMPIN

Virtual COM port pairs (COM1 and COM2) were created using the virtual serial port tool application. The COMPIM component in Proteus was configured with the following settings: a baud rate of 9600, 8 data bits for both physical and virtual communication, and COM2 as the physical port. This setup in **Figure 2** allows the COMPIM component to receive car count data transmitted via the PySerial library in Python. For physical implementation, a USB 2.0 Type A to Type B cable was used to connect the system to the Arduino Uno. The COMPIM's pins 2 (RXD) and 3 (TXD) were connected to pins 0 (RX) and 1 (TX) on the Arduino Uno, respectively, to facilitate data transmission.

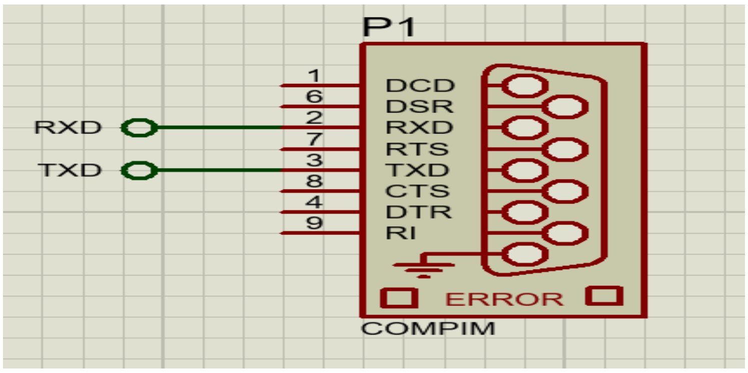


Figure 2. Virtual COMPIM port for simulation and hardware communication.

2.3. Configuration of the 74HC595 shift register

An 8-bit shift register is used to control the brightness of eight light bulbs, representing streetlights. A byte of data is sent to the Data Serial (DS) pin (Pin 14) of the shift register (**Figure 3**). The Serial Clock (SH_CP) pin (Pin 11) shifts in each bit of the byte sequentially. Once the entire byte is loaded, the Register Clock (ST_CP) pin (Pin 12) transfers the stored data to the 8-bit parallel output, illuminating the corresponding light bulbs.

In **Figure 3**, we achieve a fade effect by utilizing the Output Enable (OE) pin (Pin 13). When OE is connected to 5V, it disables the outputs. By gradually reducing the voltage on the OE pin to zero, the brightness of the bulbs slowly increases, creating a fade-in effect.

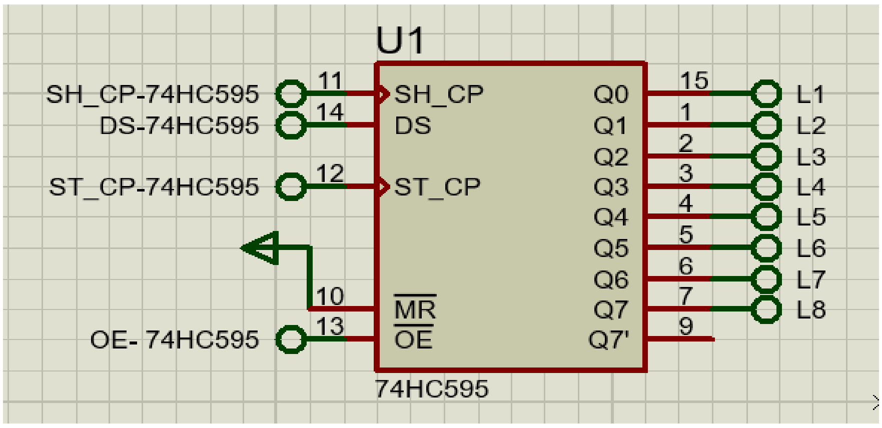


Figure 3. 74HC595 shift register for output expansion.

2.4. Configuration of the traffic lights

The red, yellow, and green traffic lights (LEDs) are connected to the digital output pins of the Arduino, where they receive digital signals to control their timing for traffic light simulation. The LEDs are based on Kirchoff's Voltage Law (KVL), which quantifies how voltage varies around a loop in a circuit. We apply the loop laws to determine the values of their limiting resistors. From the data sheet on the Green, Red, and Yellow LEDs on Proteus, the voltage drop across the diode is 2.2 V, and the full drive current is 10 mA. Taking KVL around the loop, we have:

$$10 \times 10^{-3} R_1 + 2.2\ \mathrm{V} = 5\ \mathrm{V}$$

$$R_1 = \frac{5 - 2.2}{10 \times 10^{-3}} = 0.28\ \mathrm{k\Omega} = 280\ \Omega$$

Although the calculated resistor value is 280 Ω, a 220 Ω resistor was selected due to its availability in the market. This value is close enough to ensure proper operation while slightly increasing the LED brightness within safe limits. **Figure 4** presents the diagrammatic representation.

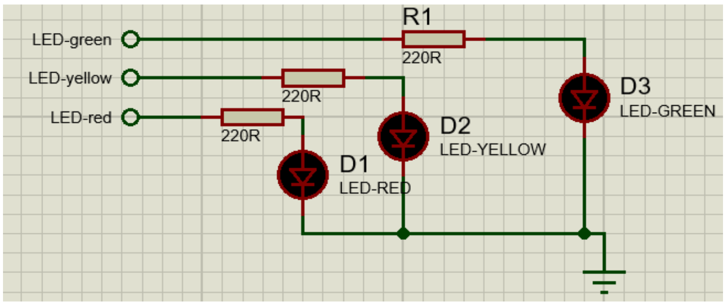


Figure 4. LED traffic light for smart traffic control circuit.

The complete circuit diagram for the smart traffic light circuit is presented in **Figure 5**.

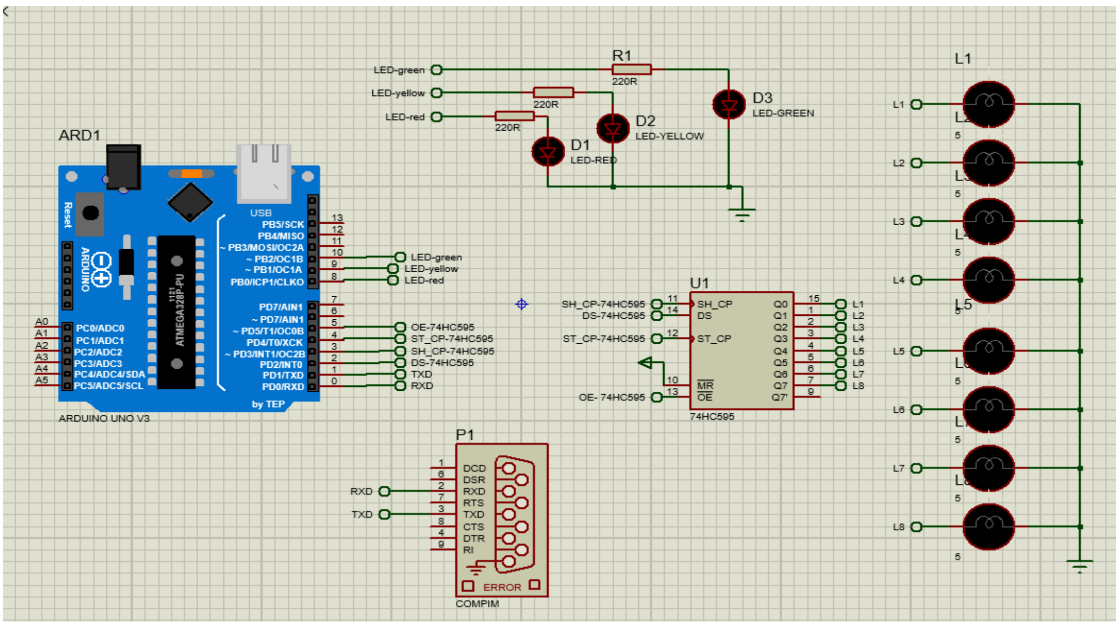


Figure 5 Circuit diagram of the smart traffic system.

2.5. Performance evaluation metrics

The four performance metrics used include (1) detection accuracy to assess the precision of vehicle, detection by the system's sensors and algorithms; (2) streetlight energy efficiency to evaluate energy savings achieved through dynamic brightness optimization based on the real-time traffic data; (3) adaptability to different traffic conditions, which evaluates the system's performance across various scenarios, such as peak hours and off-peak hours; and (4) system uptime to monitor the percentage of time the system operates without delay, failure, or downtime.

3. Diagrammatic representations of the system

3.1. Block diagram of the system

The system block diagram is illustrated in **Figure 6**. The Haar cascade classifier model processes the video feed to determine the vehicle count. This data is then transmitted to the Arduino via the PySerial library in Python. The Arduino, in turn, controls the traffic light LEDs and adjusts the brightness of the street lamps using the 74HC595 shift register, ensuring efficient traffic management and energy optimization.

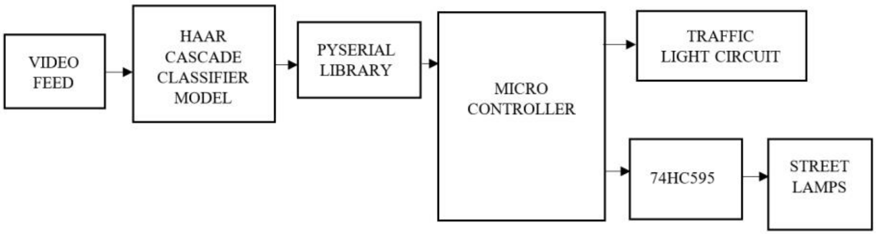


Figure 6. Block diagram showing system components and flow.

3.2. Input/output flowchart

The flowchart (**Figure 7**) illustrates the decision-making process of the smart traffic light system. It begins by initializing the Haar cascade model to process the video feed and detect car counts. The detected car count is sent to the Arduino via serial communication. Based on the car count, the Arduino determines the traffic light timing (red, yellow, green) and adjusts the street lamp brightness accordingly. The system uses the following brightness levels: 30% brightness when no car is detected, 50% brightness when less than 10 cars are detected, 75% brightness when the number of cars falls between 10 and 20, and 100% brightness when there are more than 20 cars detected. The logic ensures optimal traffic flow and energy efficiency, with varying brightness levels (mid-low, mid-high, bright) corresponding to the number of cars detected.

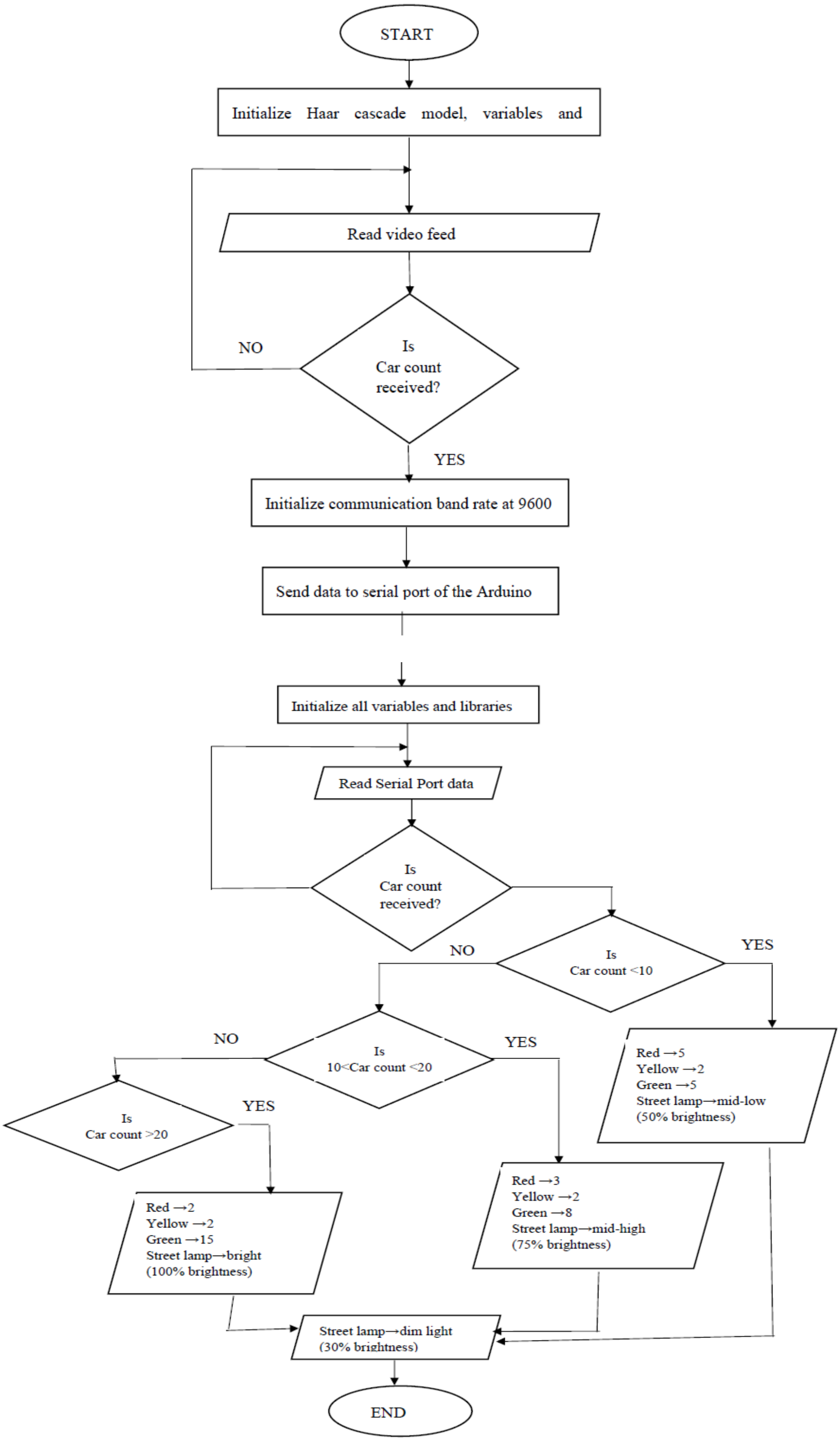
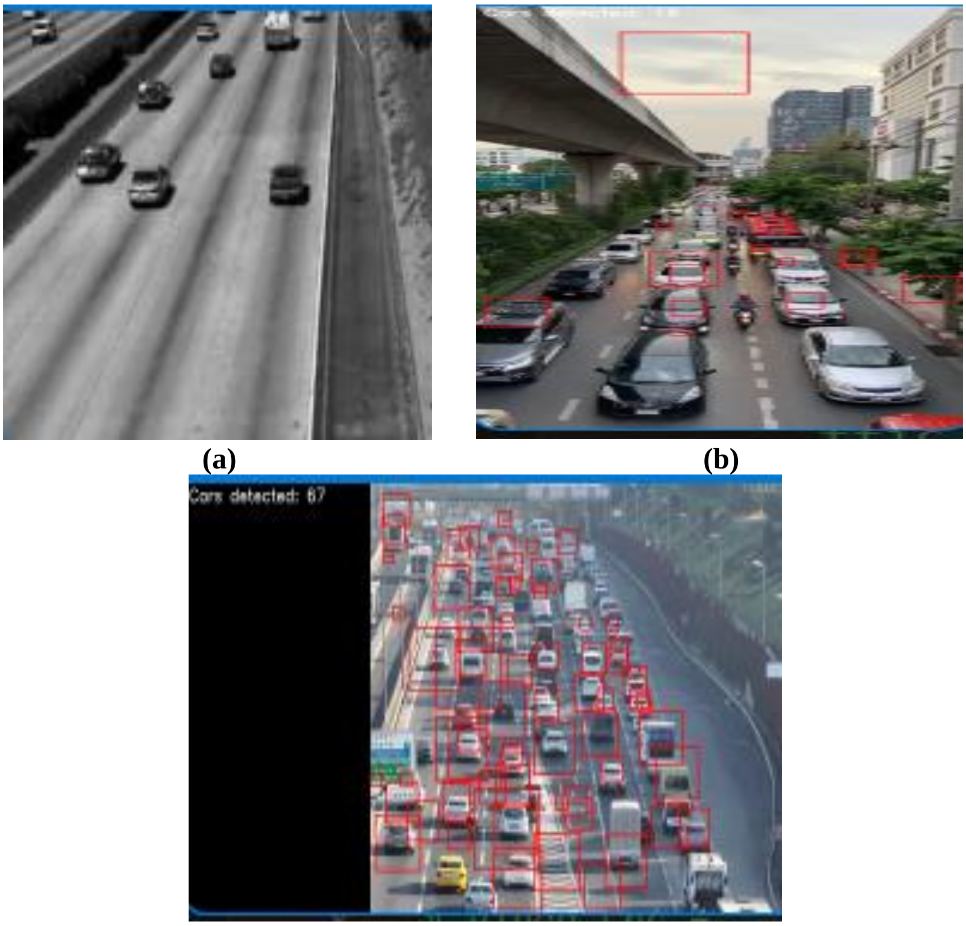


Figure 7. Flowchart of the dual smart traffic control process.

4. Real-world data testing

The system was tested on three pre-recorded traffic videos (www.pexels.com) with varying levels of traffic density (light, moderate, and heavy). The video feed was processed using a computer vision model designed for vehicle detection and counting. The model analyzed the pre-recorded video frames, identified vehicles, and quantified their presence in each frame. The extracted vehicle count data was transmitted to a microcontroller for further processing and analysis.

The light traffic case (**Figure 8a**) had a high detection accuracy with minimal false positives. There was good detection accuracy for the moderate traffic scenario (**Figure 8b**), with some false positives in overlapping/shadowed regions. For the heavy traffic (**Figure 8c**), reduced accuracy was observed due to overlapping vehicles, occlusions, and reduced visibility.



(a) (b)

(c)

Figure 8. (a) Video feed for low vehicle density; **(b)** moderate vehicle density; **(c)** and high vehicle density.

4.1. A comprehensive overview of the procedure

The procedural steps and computational considerations taken during the implementation of the test project are as follows.

4.1.1. Video analysis using the Haar Cascade Classifier

The video analysis was performed using the Haar Cascade Classifier model to detect vehicles in pre-recorded videos. The video stream for high-density traffic is shown in **Figure 8c**. In terms of detection performance, the model could identify cars successfully but could not detect motorcycles within the video frames. The detection accuracy faced challenges in regions where vehicles overlapped or where shadows were present, leading to a few false positives. The video was processed at an average rate of 5.65 frames per second (fps). This frame rate was chosen to balance the computational load while maintaining feasibility for real-time processing. Each video frame was converted to grayscale before detection, as shown in **Figure 9**. This step reduces the image data size and enhances edge and feature detection capabilities.

Figure 9. Video feed of high-density traffic converted to grayscale.

4.1.2. Data extraction and communication

The data extraction and communication process involved two key components: vehicle count and serial communication. For vehicle counting, the Haar cascade classifier model was responsible for detecting and counting the number of cars in each frame. To ensure accuracy, minimal filtering was applied to the data to address overlapping detections. The vehicle count data was then transmitted to an Arduino Uno via serial communication, using the PySerial library. The configuration for this setup was a transmission path from COM1 (host) to COM2 (Arduino Uno). The communication operated at a baud rate of 9600 bits per second, with 8 bits per byte, no parity, and a single stop bit. Before transmission, the vehicle count data was converted to string format, as serial communication protocols typically use ASCII-based data transmission.

4.1.3. Arduino-base control logic

For the prototype project, the Arduino Uno was physically connected to the computer via a type A/B USB cable on COM4, receiving serial data from the computer for processing. For virtual simulation, the compiled hex file from the Arduino IDE was uploaded to the blank hex file property field of the virtual Arduino Uno in Proteus. The COMPIM component linked the communication pin (COM2) to the TX and RX pins of the virtual Arduino. This setup allowed bidirectional data transfer between the virtual Arduino and the host system, enabling the execution of the uploaded code on Proteus 8 simulation software. The Arduino processed the received vehicle count data as integers and used conditional statements to determine the operational parameters for the street lighting system and the time required for the green light. Based on the detected number of cars, the system adjusted its behavior, as shown in **Table 1**.

Table 1. Operational parameters for the street lighting system based on vehicle density.

Vehicle Count	Lamp Brightness	Time to Green light (s)
$0 \leq$ Count < 10	50%	7
$10 \leq$ Count ≤ 20	75%	5
Count > 20	100%	4

To control the brightness of the street lamps, the Arduino utilized Pulse Width Modulation (PWM) signals. These varying signals were sent to the OE pin of the shift register, enabling a change in lamp brightness based on the detected vehicle count.

4.2. Results and discussions

Simulated and experimental tests were conducted. The simulation was done on proteus simulation software while the experimental test was conducted using the Arduino uno microcontroller. The results in both cases are presented.

4.2.1. Simulated results

After acquiring the vehicle count data using the COMPIM module in Proteus 8 simulation software and transmitting it to an Arduino Uno microcontroller, the traffic signals and street lamp brightness were then managed via the 74HC595 shift register. The circuit diagrams in **Figures 10–12** reflect the dynamics of the smart traffic system under test. The **l**ow traffic density (**Figure 10**) shows the lamps operating at 50% brightness (section L1 of **Figure 10**). This is represented in the simulation by dimly radiating yellow lines around the lamp icons. Thus, this reduced illumination effectively simulated energy savings during periods of low traffic.

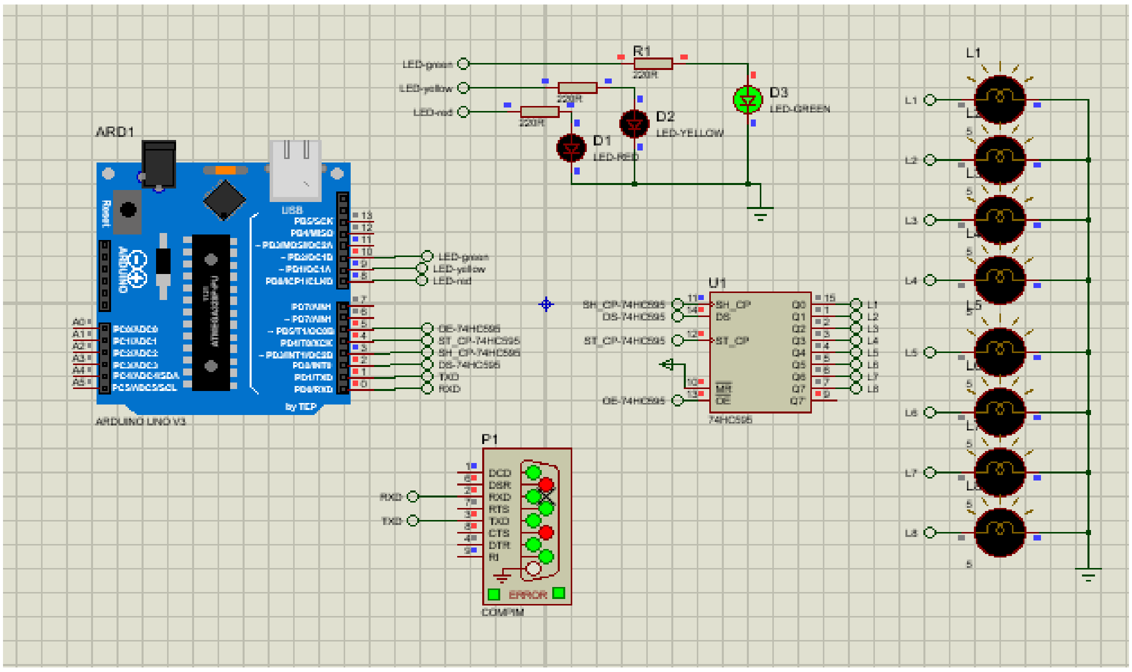


Figure 10. Low density traffic simulation on proteus simulation software.

For the moderate traffic density (**Figure 11**), the street lamps functioned at 75% brightness. In the simulation, this was depicted as moderately bright yellow lines radiating from the lamp icons, visually distinguishing the condition from the low-density scenario. A 100% street lamp brightness was observed for the high traffic density (**Figure 12**). This is indicated by intense yellow radiating lines in the simulation. This served as a clear representation of maximum illumination during high traffic volumes. It was, however, observed that the Proteus 8 simulation did not provide a clear distinction between the moderate and intense street light brightness levels.

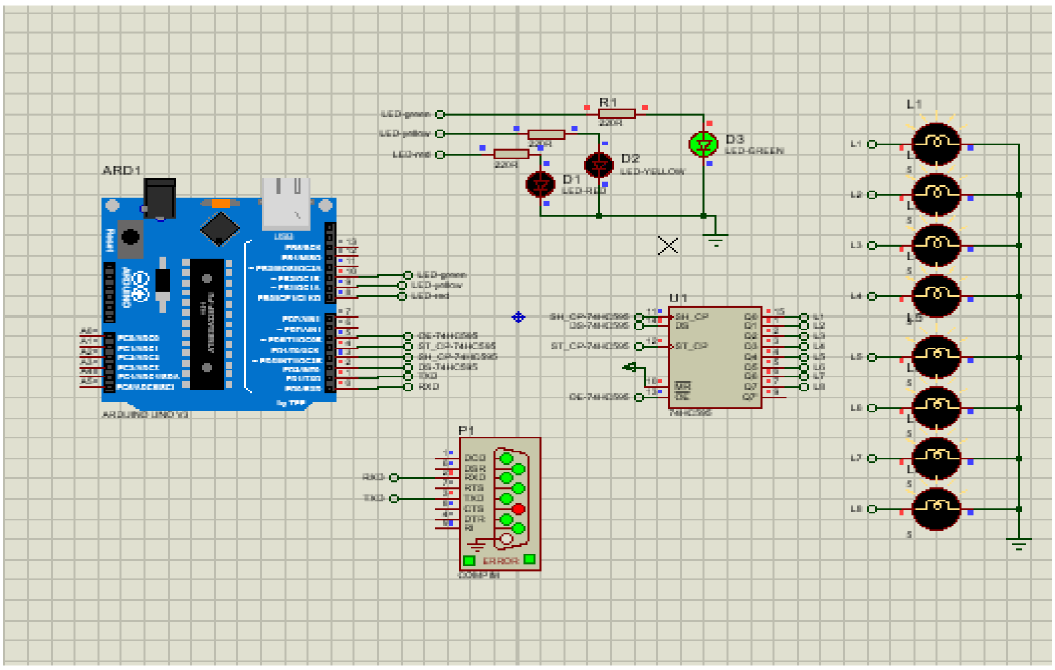


Figure 11. Moderate density traffic simulation on proteus simulation software.

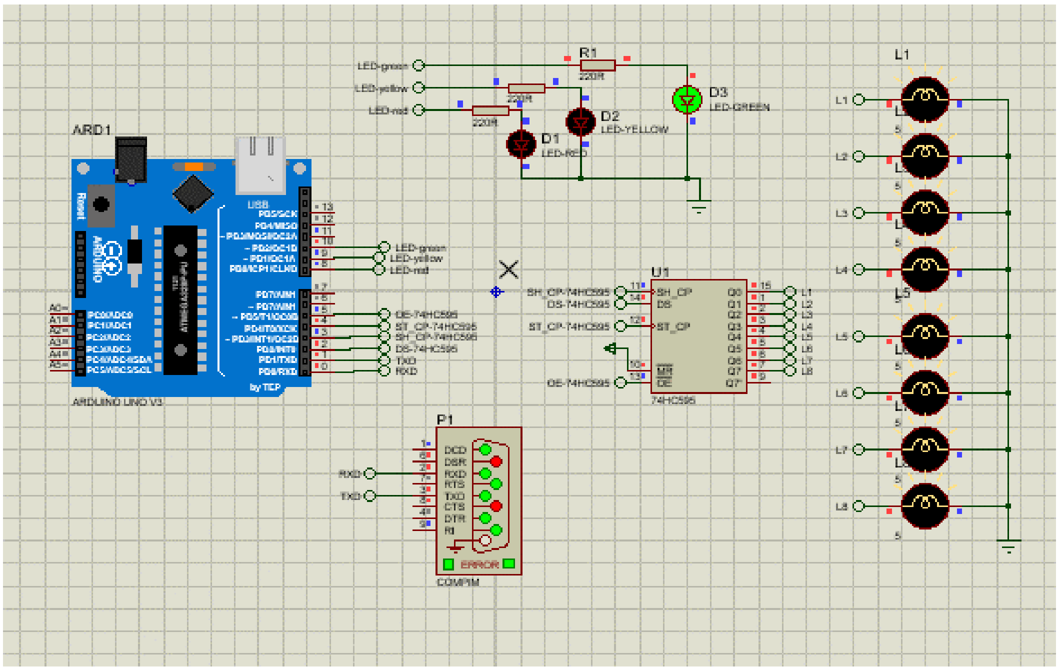


Figure 12. High density traffic simulation on proteus simulation software.

The adaptive response of the system successfully demonstrated the ability to adjust both traffic signal timings and street lamp brightness based on real-time vehicle density data. However, the host computer's limited CPU resources caused delays in traffic light transitions. Specifically, the observed duration of the green 'go' light exceeded the programmed value, highlighting a discrepancy between simulation and theoretical design. These delays did not affect the system's core functionality but emphasized the need for a more powerful host system for seamless performance. The simulation validated the feasibility of the proposed adaptive traffic control design, showcasing its potential for real-world applications in traffic and energy management.

4.2.2. Experimental results

The experiment featured two sets of LEDs—eight LEDs on the left-hand side represented the streetlights, while three LEDs on the right-hand side (arranged as red, yellow, and green) represented traffic lights (**Figure 13**). The streetlights, controlled by the 74HC595 shift register, varied in brightness based on traffic density. A 50% brightness for low traffic density is shown in **Figure 13**. **Figure 14** displays the 75% brightness for the moderate traffic density, and **Figure 15** shows the 100% brightness for the high traffic density.

Figure 13. Demo of LED brightness with less than 10 cars detected.

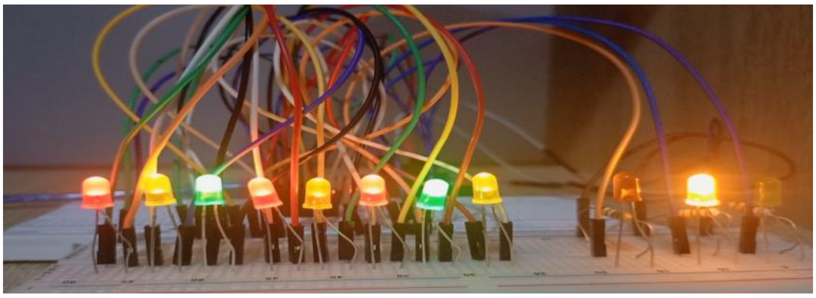


Figure 14. Demo of LED brightness with less than 20 cars detected.

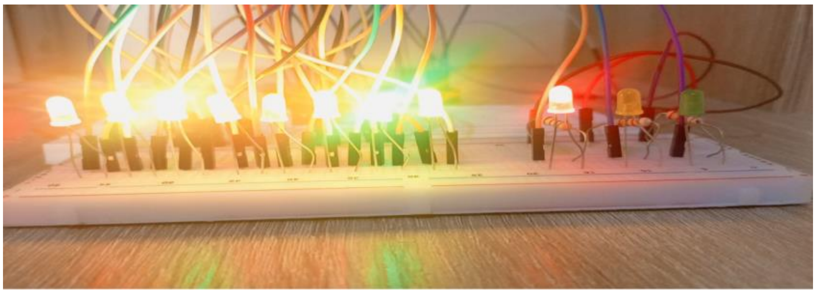


Figure 15. Demo of LED brightness with over 20 cars detected.

Comparing the experimental results with that of the simulations performed on Proteus 8 simulation software, it was observed that the experimental test project demonstrated higher accuracy compared to the simulation, particularly during the transition to the green light. In the experimental setup, the green light programmed delay for low traffic density was 7 s, and the Arduino accurately executed this timing due to its chip's dedicated processing capability. While, in the Proteus simulation, the same transition took approximately 20 s. This discrepancy was attributed to the computational limitations of the host computer running the simulation, which caused delays in processing. This pattern was observed across all traffic density levels, where the Arduino consistently executed transitions with precise timing, while the simulation introduced notable delays. Additionally, although the street lamps in the simulation were configured for analog control, thereby allowing for varying analog values between 0 and 255, the simulation was only capable of representing discrete brightness levels of 0%, 50%, and 100%. In contrast, the experimental results demonstrated a clear differentiation between 50%, 75%, and 100% brightness levels, depicted by the LEDs and illustrated in **Figures 13–15**. The superior performance of the experimental setup highlights the efficiency of using a dedicated microcontroller for real-time traffic management systems.

4.3. Limitations of the smart traffic light system

The smart traffic light system designed in this project shows the potential for integrating computer vision and microcontroller-based automation. However, several limitations impact its performance and scalability.

4.3.1. Limited general purpose input/output (GPIO) pins, single-core processing and universal asynchronous receiver-transmitter (UART) communication

The microcontroller limitations include limited GPIO pins, and single-core processing, and UART communication. More specifically, the Arduino Uno provides only 14 digital I/O pins and 6 analog input pins, which can become insufficient for handling the complex input and output requirements of the system, especially if additional sensors or actuators need to be integrated. Microcontrollers such as the ESP32, Raspberry Pi, or STM32, with significantly more GPIO pins, higher clock speeds, and advanced processing capabilities, could improve the scalability and efficiency of the system. Additionally, the Arduino Uno uses a single UART interface to receive data from the computer vision model. However, it cannot simultaneously read serial data while performing the time-critical tasks of controlling the traffic lights and adjusting the brightness of street lamps. The Arduino Uno can be replaced with an ESP32 microcontroller. The ESP32's dual-core processor allows one core to handle serial communication with the computer vision model while the other core manages traffic light control and street lamp brightness.

4.3.2. Limited computer vision model

The computer vision model also poses some limitations. It is important to note that the Haar Cascade classifier, while effective for simple detection tasks, has several limitations. Environmental conditions significantly impact the performance of its vehicle detection capability. These include adverse weather conditions such as rain, fog, or snow, which degrade visibility and disrupt lighting consistency, leading to reduced detection accuracy. Similarly, nighttime detection is challenging in low-light environments and requires additional preprocessing techniques, such as histogram equalization, to enhance visibility and improve performance. Furthermore, in crowded scenes, overlapping cars often lead to misdetections or missed detections. The detection speed is also impacted as the Haar Cascade's detection process, while relatively fast, struggles with real-time performance when scaled to larger resolutions or high-density scenes. Details of the vehicular detection challenges and proposed solutions are laid out.

Occlusion due to vehicle alignment

When vehicles are aligned successively behind one another, the computer vision model experiences occlusion, where some vehicles are partially or completely obscured. This results in undercounting, leading to an inaccurate estimation of traffic density as seen in **Figure 16**. To mitigate occlusion, the camera should be mounted at an elevated position, such as on top of the traffic light pole or a high vantage point, ensuring a clear field of view. Additionally, integrating depth sensors (LiDAR or stereo cameras) can enhance vehicle detection in occluded scenarios by providing three-dimensional spatial awareness.

Figure 16. Occlusion due to vehicle alignment.

Lane misclassification

The current model is trained solely for vehicle detection and does not incorporate lane segmentation. As a result, it fails to distinguish between vehicles in the traffic lane it controls and those in adjacent or returning lanes. This misclassification leads to an overestimation of vehicle count. For instance, in high-traffic density scenarios, **Figure 17**, the model detects 48 vehicles instead of the actual 24 due to miscounting vehicles from opposing lanes. Implementing a lane segmentation module using deep learning-based semantic segmentation (U-Net, DeepLabV3, etc.) or classical computer vision techniques (for instance, Hough Transform for lane line detection) will enable the model to accurately associate vehicles with their respective lanes. Additionally, integrating Region of Interest (ROI) filtering can ensure that only vehicles within the designated traffic lane are considered.

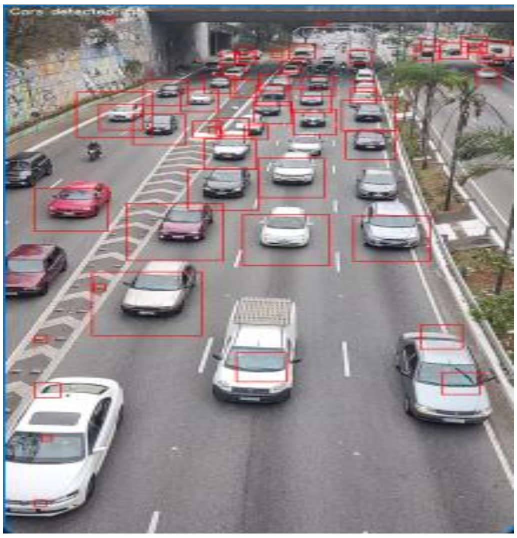
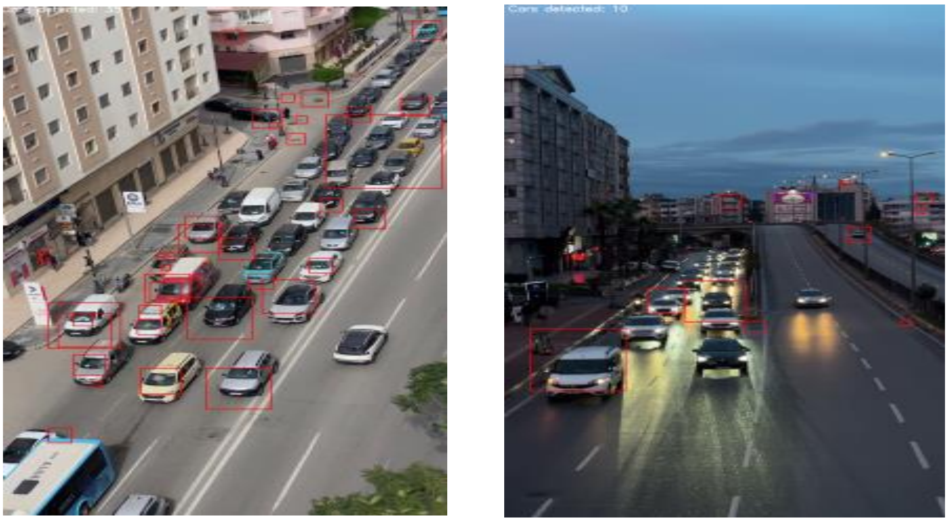


Figure 17. Lane misclassification.

Reduced accuracy in extreme lighting conditions

Under bright sunlight, the model's accuracy deteriorates due to overexposure and the presence of false positives, where trees, shadows, and reflective surfaces (e.g., car windows) are mistakenly detected as vehicles. In high-traffic density conditions, **Figure 18**, this leads to significant overcounting—detecting 47 cars instead of the actual 30, representing a 56% reduction in model accuracy. In addition to challenges posed by bright sunlight, computer vision models often experience decreased accuracy during evening hours due to low-light conditions. This reduction in illumination can lead to increased noise and decreased contrast in images, making it difficult for models to accurately detect and classify objects.

To enhance robustness against varying lighting conditions, a set of techniques can be employed. Adaptive exposure control involves adjusting camera exposure settings to balance brightness levels across different lighting conditions. Polarization, infrared filtering, and thermal imaging focus on using polarized lenses or infrared and thermal cameras to reduce glare and reflections. Data augmentation focuses on training the model with diverse lighting conditions using techniques such as brightness normalization, histogram equalization, and shadow-aware augmentation to improve generalization. Adaptive histogram equalization enhances image quality in low-light environments, while background subtraction methods effectively address detection issues caused by adverse weather effects. Post-processing enhancements can also be deployed with optical flow or tracking algorithms. This involves implementing object tracking (e.g., Kalman filters or SORT, DeepSORT) to reduce transient false detection caused by momentary reflections and shadows and improve detection consistency in consecutive frames.



(a) (b)

Figure 18. **(a)** Reduced accuracy in sunny conditions; **(b)** and in low lighting conditions.

Other potential improvements in computer vision model limitations include using advanced machine learning models. Convolutional Neural Networks (CNNs), including models like YOLO [15] and Faster R-CNN, provide superior performance by effectively identifying objects in diverse scenarios. Pre-trained models, such as MobileNet or SSD, can also be employed for faster inference, particularly on hardware with limited computational capabilities [19,20]. Enhancements to preprocessing techniques can further improve detection under challenging conditions.

5. Implications for auto insurance, risk management and liability support

In this section, suggestions are provided on how smart traffic systems can significantly improve the insurance industry. A few concrete examples are highlighted.

5.1. Auto insurance

5.1.1. Risk-based premium adjustment

Smart traffic systems have the potential to transform the insurance industry, particularly in how premiums are calculated and risk is managed. A key area of application is in dynamic risk profiling, which leverages data collected from these systems—such as peak traffic times, accident hotspots, and congestion patterns—to provide insurers with valuable insights into risk-prone areas. With this information, insurers can adjust premiums based on the likelihood of accidents in specific areas or at particular times. This approach not only enhances the accuracy of risk assessment but also benefits drivers who frequent safer, less congested routes by offering them more favorable premium rates.

Another significant innovation lies in the personalization of premiums. By utilizing real-time traffic data, insurers can account for individual driving habits, including driving frequency and preferred routes. For instance, a driver who regularly navigates through high-risk intersections may see a corresponding adjustment in their premium, reflecting the increased exposure to potential accidents. Conversely, drivers who consistently avoid high-risk areas, particularly during peak traffic hours, could benefit from lower premiums. Insurers might even incentivize safer driving practices by offering discounts to those who follow optimized routes, encouraging more responsible behavior on the roads.

This is becoming a desired need with the rise in electric vehicles and self-driving cars. Such insights have been described by Balasubramanian et al. [21], who highlighted the future potential impact of artificial intelligence on the insurer. They describe a digital personal assistant who plans a route and shares it with the mobility insurer, which suggests an alternate route with reduced accident risk and lower chances of vehicle damage, along with the revised monthly premium.

5.1.2. Improved claims processing

The integration of smart traffic systems into insurance processes offers transformative opportunities for improving claims management and reducing fraud. One of the most impactful applications is in data-driven accident analysis. By leveraging traffic flow and video data from intersections, insurers can reconstruct accident scenarios with greater accuracy. Precise accident reconstruction is essential not only for understanding the circumstances surrounding such events but also for enhancing advanced driver assistance systems and improving various vehicle features where applicable [22]. Studies like Chen et al. [23]and Pádua et al. [24] have studied ways in which road traffic accidents can be reconstructed based on multiple scenarios. By incorporating data from smart traffic systems, more specific real-life scenarios can be investigated. This capability will significantly streamline claims investigations, enabling faster and more precise determination of fault and assessment of damages. As a result, insurers can reduce the time and resources traditionally required for such investigations, improving efficiency and customer satisfaction.

Besides enhancing accident analysis, smart traffic systems can play a critical role in fraud detection. The availability of real-time traffic data and vehicle density information allows insurers to verify the circumstances of a claimed accident. For example, claims of staged accidents or fabricated incidents can be cross-referenced against actual traffic patterns and video evidence from the time and location in question. This robust validation process helps mitigate instances of fraudulent claims, protecting insurers from financial losses while fostering trust and accountability within the system.

5.1.3. Reduced loss ratios

Smart traffic systems can play a significant role in reducing loss ratios for insurers by addressing some of the root causes of traffic accidents. By managing congestion more effectively, these systems help decrease the frequency of accidents, particularly in high-risk areas. Many accidents are caused by sudden braking, aggressive lane changes, or violations such as running red lights—all of which are more likely to occur in congested conditions. By optimizing traffic flow and reducing bottlenecks, smart traffic systems can mitigate these behaviors, leading to fewer accidents and, consequently, fewer insurance claims. This reduction in claims directly translates to lower loss ratios for insurers, improving their financial performance and enabling them to allocate resources more efficiently.

Insurers can actively promote the adoption of smart traffic technology by offering incentives for safer driving areas. For instance, they might collaborate with local governments to implement these systems in accident-prone zones by providing community discounts or risk-based incentives. Such partnerships not only encourage the use of smart traffic solutions but also create a safer driving environment for all road users. The widespread adoption of these technologies benefits insurers by reducing claims and benefits drivers by fostering safer and more efficient road networks.

5.1.4. Enhanced telematics programs

The system could provide real-time data to usage-based insurance models, which typically rely on telematics to assess driver behavior [25,26]. Combined with telematics devices in cars, smart traffic systems could offer insurers even more granular data on driving conditions, helping to create more accurate risk profiles.

5.2. Risk management

Smart traffic systems provide transformative capabilities in the realm of risk management, offering a proactive approach to monitoring, planning, and response. One of the key applications lies in traffic and infrastructure monitoring. By collecting and analyzing data on traffic flow and density, these systems can predict and manage risks associated with infrastructure wear and tear. This insight allows for the development of more effective maintenance schedules, reducing the likelihood of sudden failures such as road collapses or signal malfunctions. This can complement other road safety protocols, such as the Traffic Accidents Reduction Strategy for

Vehicular Ad Hoc Networks developed by Aldegheishem et al. [27]. Proactive infrastructure management not only enhances safety but also minimizes disruptions to transportation networks.

The data generated by these systems also support data-driven policy and insurance adjustments. Traffic patterns can reveal areas with high accident probabilities, enabling targeted interventions and informing risk-based insurance premiums. Such insights allow insurers and policymakers to optimize strategies for managing risks across transportation networks, fostering safer and more reliable mobility.

In the context of resilience planning and recovery, smart traffic systems prove invaluable during emergencies, such as hurricanes, floods, earthquakes, or large public gatherings. By adapting to real-time traffic patterns, these systems can dynamically reroute vehicles and manage flow to support emergency plans. This reduces risks related to overcrowding or infrastructure strain in critical areas, enhancing the overall safety and efficiency of response efforts. Bhavana and Praveen [28] proposed an Internet of Things (IoT)-based intelligent traffic navigation to address emergency situations. Their technological design can also be connected to receive a data feed from a smart traffic system. Insurers, too, benefit from this data, as it allows them to assess risks in impacted areas and develop coverage options tailored to catastrophic events, further strengthening risk management frameworks.

Finally, smart traffic systems play a pivotal role in post-event claim efficiency. After major traffic incidents or disasters, the system's detailed records of traffic patterns and densities provide insurers with a reliable source of information to handle a surge in claims. This speeds up claim resolutions, improves customer satisfaction, and reduces the administrative burden on insurers.

5.3. Liability and litigation support

Smart traffic systems have significant potential to improve liability and litigation support by offering accurate data and reducing the risk of traffic-related incidents. One major benefit is enhanced liability determination. By providing precise records of vehicle density, signal timing, and traffic flow, these systems enable more accurate assessments of liability in traffic accidents occurring at monitored intersections. MAPFRE [22] noted that insurance companies significantly depend on accurate accident assessment information to determine liability and provide fair compensations. This detailed information serves as reliable evidence for litigation or settlements, offering clarity on the conditions during an incident. As a result, disputes can be resolved more efficiently, reducing the time and costs associated with legal proceedings.

Another advantage is the reduction of liability for municipalities. Poor traffic management and unsafe intersections are often cited as contributing factors in claims against local governments. By implementing smart traffic control systems, municipalities can address these issues by reducing accident frequency and improving road safety. This proactive approach minimizes the likelihood of liability claims, ultimately benefiting both municipalities and their insurers. Over time, the reduction in claims could lead to lower municipal insurance premiums, allowing resources to be redirected toward further improving urban infrastructure.

6. Conclusion

This work presents a complete solution to solve traffic congestion and energy efficiency issues in metropolitan regions through a twin smart traffic light control system. The system dynamically changes traffic light lengths depending on real-time vehicle density by using machine learning under a pre-trained Haar Cascade classifier and merging it with Arduino-operated adaptive traffic signals. Dynamic streetlight brightness management, depending on traffic patterns at night, offers a fresh method of energy optimization while preserving road safety.

This method illustrates the possibility for multidisciplinary uses in addition to improving traffic flow and lowering gasoline use. Examining the effects on motor insurance, liability coverage, and risk management expands the scope of smart traffic systems and prepares the ground for more study on their social and economic effects.

Future research could investigate how to increase vehicle identification accuracy by integrating innovative deep learning models such as convolutional neural networks or YOLO. Furthermore, improving the system's adaptability might include weather conditions, pedestrian activity, and other environmental elements. This study shows the viability and possibilities of using technology and creativity to provide smart traffic control systems for contemporary cities.

Author contributions: Conceptualization, CPC and QCC; methodology, CPC and QCC; software, CPC; validation, CPC and QCC; formal analysis, CPC; investigation, CPC; resources, CPC and QCC; data curation, CPC; writing—original draft preparation, CPC, QCC and EQC; writing—review and editing, CPC, QCC and EQC; visualization, CPC and QCC; supervision, QCC; project administration, CPC and QCC. All authors have read and agreed to the published version of the manuscript.

Conflict of interest: The authors declare no conflict of interest.

References

1. Thomas P, Welsh R, Mavromatis S, et al. Survey results: Road safety data, data collection systems and definitions. Safer Africa Project; 2017.
2. Adeloye D, Thompson JY, Akanbi MA, et al. The burden of road traffic crashes, injuries and deaths in Africa: A systematic review and meta-analysis. Bulletin of the World Health Organization. 2016; 94(7): 510–521A. doi: 10.2471/BLT.15.163121
3. Kazeem Y. Death rates from traffic accidents are higher in Africa than anywhere else. Available online: https://www.weforum.org/agenda/2019/02/death-rates-from-traffic-accidents-are-higher-in-africa-than-anywhere-else/ (accessed on 5 December 2024).
4. Hartenstein H, Laberteaux LP. A tutorial survey on vehicular ad hoc networks. IEEE Communications Magazine. 2008; 46(6): 164–171.
5. Yan G, Rawat DB. Vehicle-to-Vehicle Connectivity Analysis for Vehicular Ad-Hoc Networks. Ad Hoc Networks, 2016; 58 (C), 25 - 35. doi: 10.1016/j.adhoc.2016.11.017.
6. Raya M, Hubaux JP. Securing vehicular adhoc networks. Journal of Computer Security. 2007; 15: 39–68.
7. Taleb T, Samdanis K, Mada B, et al. On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration. IEEE Communications Surveys & Tutorials, 2017; 19(3), 1657-1681. doi: 10.1109/COMST.2017.2705720.

8. Veres M, Moussa M. Deep Learning for Intelligent Transportation Systems: A Survey of Emerging Trends. IEEE Transactions on Intelligent Transportation Systems, 2020; 21(8), 3152-3168. doi: 10.1109/TITS.2019.2929020
9. Arifin S, Razi SA, Haque A, et al. A Microcontroller Based Intelligent Traffic Control System. American Journal of Embedded Systems and Applications. 2019; 7(1): 21–25. doi: 10.11648/j.ajesa.20190701.13
10. Nyiekaa EA, Irokwe JN, Shaapera TM. Design and Implementation of a 4-Way Traffic Control and Detection System. International Journal of Innovations in Engineering and Science. 2022; 10(4): 16–27.
11. Fadoro JO. Design and Implementation of Electronic Traffic Light. Villanova Journal of Science, Technology and Management. 2019; 1(1).
12. Bisen K, Shahare R, Wasnik K, et al. Machine Learning-Based Intelligent Traffic System. International Research Journal of Modernization in Engineering Technology and Science. 2023; 5(4). doi: 10.56726/irjmets35340
13. Balasubramanian SB, Balaji P, Munshi A, et al. Machine learning based IoT system for secure traffic management and accident detection in smart cities. PeerJ Computer Science. 2023; 9: e1259–e1259. doi: 0.7717/peerj-cs.1259
14. Tiwari P. The machine learning framework for traffic management in smart cities. Management of Environmental Quality. 2024; 35(2): 445–462. doi: 10.1108/MEQ-08-2022-0242
15. Khan H, Kushwah KK, Maurya MR, et al. Machine learning driven intelligent and self adaptive system for traffic management in smart cities. Computing. 2022; 104(5): 1203–1217. doi: 10.1007/s00607-021-01038-1
16. Lanke N, Koul S. Smart Traffic Management System. International Journal of Computer Applications. 2013; 75(7): 19–22. doi: 10.5120/13123-0473
17. Al-Abaid SAF. A Smart Traffic Control System Using Image Processing: A Review. Journal of Southwest Jiaotong University. 2020; 55(1). doi: 10.35741/issn.0258-2724.55.1.31
18. Sobral AC. Vehicle Detection with Haar Cascades. Available online: https://github.com/andrewssobral/vehicle_detection_haarcascades?tab=readme-ov-file (accessed on 5 December 2024)
19. Aulia U, Hasanuddin I, Dirhamsyah M, et al. A new CNN-BASED object detection system for autonomous mobile robots based on real-world vehicle datasets. Heliyon. 2024; 10(15). doi: 10.1016/j.heliyon.2024.e35247
20. Khan D, Waqas M, Tahir M, et al. Revolutionizing Real-Time Object Detection: YOLO and MobileNet SSD Integration. Journal of Computing & Biomedical Informatics. 2023; 6(1). doi: 10.56979/601/2023
21. Balasubramanian R, Libarikian A, McElhaney D. Insurance 2030—The impact of AI on the future of insurance. Available online: https://www.mckinsey.com/industries/financial-services/our-insights/insurance-2030-the-impact-of-ai-on-the-future-of-insurance (accessed on 5 December 2024).
22. MAPFRE. The growing potential of technology in traffic accident reconstruction. Available online: https://www.mapfre.com/en/insights/innovation/technology-traffic-accident-reconstruction/ (accessed on 5 December 2024).
23. Chen Y, Zhang Q, Yu F. Transforming traffic accident investigations: A virtual-real-fusion framework for intelligent 3D traffic accident reconstruction. Complex Intelligent Systems. 2025; 11. doi: 10.1007/s40747-024-01693-9
24. Pádua L, Sousa J, Vanko J, et al. Digital Reconstitution of Road Traffic Accidents: A Flexible Methodology Relying on UAV Surveying and Complementary Strategies to Support Multiple Scenarios. International Journal of Environmental Research and Public Health. 2020; 17(6): 1868. doi: 10.3390/ijerph17061868
25. Li HJ, Luo XG, Zhang ZL, et al. Driving risk prevention in usage-based insurance services based on interpretable machine learning and telematics data. Decision Support Systems. 2023; 172: 113985. doi: 10.1016/j.dss.2023.113985
26. Ziakopoulos A, Petraki V, Kontaxi A, et al. The transformation of the insurance industry and road safety by driver safety behaviour telematics. Case Studies on Transport Policy. 2022; 10(4): 2271–2279. doi: 10.1016/j.cstp.2022.10.011
27. Aldegheishem A, Yasmeen H, Maryam H, et al. Smart Road Traffic Accidents Reduction Strategy Based on Intelligent Transportation Systems (TARS). Sensors. 2018; 18(7): 1983. doi: 10.3390/s18071983
28. Bhavana S, Praveen P. IoT-Enabled Intelligent Traffic Navigation with Accident Management for Critical Emergency Response in Heavy Congestion Using Machine Learning. Available online: https://advance.sagepub.com/users/810526/articles/1212008-iot-enabled-intelligent-traffic-navigationwith-accident-managementfor-critical-emergency-response-in-heavy-congestion-using-machine-learning (accessed on 5 December 2024).