

Article

Not peer-reviewed version

---

# An Android-Based Smart RSU Framework for Enhanced Urban Traffic Management

---

Mohammed Fadhil\* and [Qutaiba Ibrahim Ali](#)

Posted Date: 24 March 2025

doi: 10.20944/preprints202503.1759.v1

Keywords: Roadside Units (RSUs); traffic monitoring; YOLOv8; SAHI algorithm; android-based RSUs; real-time object detection; urban traffic management



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Article

# An Android-Based Smart RSU Framework for Enhanced Urban Traffic Management

Mohamed Fadhil \* and Qutaiba I. Ali

University of Mosul/IRAQ

\* Correspondence: eng.mohammed.fadhil.r@gmail.com

**Abstract:** This paper addresses critical challenges in the deployment and effectiveness of traditional Roadside Units (RSUs) in traffic monitoring systems and proposes a novel, cost-effective approach using Android-based smart RSUs. Leveraging mobile phone architecture, YOLOv8, the SAHI algorithm and chat gpt-4o, the system provides real-time traffic data collection, vehicle detection, and congestion analysis. This paper evaluates the performance of different cost tiers of mobile devices, discusses traditional traffic monitoring challenges, and identifies key gaps in current RSU technologies. The proposed system offers enhanced scalability, flexibility, and reduced cost, making it an ideal solution for urban traffic management.

**Keywords:** Roadside Units (RSUs); traffic monitoring; YOLOv8; SAHI algorithm; android-based RSUs; real-time object detection; urban traffic management

---

## I. Introduction

As urban areas continue to grow due to population increases and urban expansion, traffic congestion has emerged as a significant challenge and need effective solutions, Effective traffic control is essential for mitigating the adverse effects of congestion on the economy, environment, and quality of life.

The rapid increase in urban populations, projected to rise from 54% in 2014 to 66% by 2050[1,2], has led to a corresponding surge in traffic volume. This growth results in more vehicles on the road, exacerbating congestion and straining existing infrastructure. Traditional methods of expanding road networks are often unsustainable due to limited land resources and high costs on other hand, improving traffic control systems can provide ,a more viable solution to manage the increased demand for transportation services effectively[2,3].

Traffic congestion has far-reaching impacts on various aspects of urban life:

**Economic Impact:** Congestion results in increased travel times. which can lead to lost productivity. Delays in transportation also affect logistics and supply chains. Ultimately increasing costs for businesses and consumers alike[4].studies have shown that traffic congestion can result in thousands of dollars in lost productivity due to time wasted In the presence goods and employees in traffic congestion[5].

**Environmental Impact:** Traffic congestion is a major contributor to urban air pollution, Vehicle emissions significantly degrade air quality. posing health risks to residents and contributing to broader environmental issues such as climate change And causes global warming[3,6,7].

**Quality of Life:** Prolonged congestion affects daily life by increasing stress levels among commuters. reducing time available for personal activities, and contributing to a general decline in urban livability. The frustration associated with traffic delays can diminish community satisfaction and overall well-being[3,7].

Traffic congestion and road safety are pressing issues in modern urban environments, necessitating advanced traffic monitoring systems. To combat the challenges posed by increasing traffic congestion there is a pressing need for intelligent traffic management systems (ITMS) These

systems utilize advanced technologies such as artificial intelligence (AI), sensor networks, and data analytics to optimize traffic flow and enhance safety.

Key features of ITMS include:

1. **Real-Time Traffic Monitoring:** Continuous data collection through sensors and cameras allows for real-time analysis of traffic conditions, This information can be used to adjust traffic signals dynamically based on current flow patterns[8].
2. **Predictive Analytics:** By analyzing historical traffic data these systems can predict congestion trends and implement proactive measures to alleviate potential bottlenecks before they occur[9].
3. **Adaptive Signal Control:** Unlike traditional static traffic signals adaptive systems adjust their timings based on real-time conditions, thereby improving traffic flow efficiency and reducing delays[10].
4. **Vehicle-to-Infrastructure Communication:** This technology facilitates direct communication between vehicles and traffic management systems enhancing coordination and response times during peak congestion periods[8].
5. **Accident Reduction:** By understanding the dynamics of traffic conditions these systems can help reduce accident rates by identifying high-risk scenarios before they escalate. For example, predictive modeling can provide insights into factors influencing accidents, enabling timely interventions[11].
6. **Emergency Response Optimization:** Intelligent systems can prioritize emergency vehicles during congested conditions by adjusting traffic signals based on real-time vehicle density, This ensures quicker response times for ambulances and fire trucks, ultimately saving lives[12].
7. **Sustainability:** Intelligent systems can contribute to reducing environmental impacts by optimizing routes that minimize fuel consumption and emissions. By managing traffic more effectively cities can improve air quality and promote healthier urban environments[13].

Traditional roadside units (RSUs) have been integral to traffic monitoring infrastructure but face several challenges in cost, scalability, and operational efficiency.

Conventional road traffic monitoring systems rely on a combination of static sensors, camera-based surveillance, and specialized roadside units (RSUs) to gather real-time data. These systems focus on detecting and managing traffic flows, identifying congestion, and aiding in traffic signal optimization. Table 1 shows traffic monitoring technology, its description, advantages and disadvantages.

**Table 1.** Traffic monitoring technology comparison.

Technology	Description	Advantages	Disadvantages
Inductive Loops	Embedded in road surfaces to detect vehicles.	Reliable and accurate under normal conditions.	Expensive to install, prone to damage, difficult to maintain.
Camera-Based Systems	Mounted on infrastructure to capture images and videos of traffic flow.	Provides visual data for detailed traffic analysis.	Expensive equipment, high data bandwidth, affected by lighting/weather.

Microwave Radar	Detects vehicle movement and speed using microwave signals.	Operates well in various weather conditions.	Limited by range and detection accuracy in complex traffic situations.
RFID Sensors	Tags vehicles for tracking movement over large distances.	Effective for monitoring vehicle speeds and route patterns.	Requires specialized equipment and setup, not real-time for all traffic.
Roadside Units (RSUs)	Fixed units placed along roadways that collect traffic data and provide communication between vehicles and infrastructure.	Real-time data collection and communication with vehicles.	High cost, complex installation, limited coverage, scalability issues.

While these systems have proven effective in specific applications, they also exhibit key limitations:

- a. **Cost and Maintenance:** Traditional RSUs are expensive to deploy, maintain, and upgrade, requiring specialized hardware and regular servicing.
- b. **Scalability Issues:** Expanding RSU networks across large urban areas is not feasible due to the high cost and infrastructure requirements.
- c. **Data Processing Delays:** Many systems struggle to provide real-time data due to slow processing times or network bottlenecks.

Since, we focus on Overcoming RSU Challenges in Traffic Monitoring , SO RSU-Related Issues we will discuss it in detail. RSUs play a critical role in Intelligent Transport Systems (ITS), particularly in enabling vehicle-to-infrastructure (V2I) communication. However, despite their potential, RSUs face several operational challenges that limit their effectiveness. Table 2 shows Challenges associated with traditional RSUs.

**Table 2.** Challenges associated with traditional RSUs.

RSU Challenge	Description
High Deployment Costs	Traditional RSUs involve significant upfront costs, including specialized hardware, labor, and installation expenses.
Limited Scalability	Due to their cost and reliance on fixed infrastructure, traditional RSUs are difficult to scale across large urban areas or changing environments.
Maintenance Complexity	Regular maintenance is necessary due to environmental wear and tear, often requiring road closures and costly labor.

Limited Data Coverage	Fixed RSUs cover limited sections of roads, creating blind spots in large areas or complex urban environments.
Network Latency	RSUs are dependent on network bandwidth and stable connections, which can degrade under high-traffic conditions, leading to delays in data analysis.

Addressing these challenges is crucial for improving traffic management systems and creating smart cities that rely on efficient, scalable, and cost-effective solutions. This paper proposes an innovative approach utilizing Android-based mobile devices as smart RSUs to overcome these limitations, while enhancing real-time traffic management capabilities.

In this paper, we cover the following: Section 2 discusses the related works. Section 3 provides an overview of the proposed system. Section 4 outlines the methodology. Section 5 presents the results. Section 6 discusses the findings, and Section 7 concludes the paper.

## II. Related Works

Several innovative approaches have been explored to enhance RSU-based traffic monitoring systems, focusing on cost-effectiveness and improved data collection methods.

1. **Low-Cost Devices:** They are used as a prominent strategy to reduce costs in traffic monitoring systems. These devices include various types of sensors that may be less expensive than traditional monitoring equipment, making their implementation easier in urban areas.

Examples of Low-Cost Solutions:

- 1.1 Wi-Fi Signal Monitoring: A study proposed a system that uses variations in Wi-Fi signal strength to detect vehicles. This method achieved classification accuracy ranging from 83% to 100% for vehicles [14].
- 1.2 Embedded Neural Networks: A device was developed that uses an embedded neural network to determine vehicle type and speed, achieving 96% accuracy for vehicle classification and 89% for speed with energy efficiency [15].
- 1.3 Mobile Device Integration: The use of mobile devices for collecting traffic data at a low computational cost was proposed, enhancing data accuracy through error correction algorithms [16].
- 1.4 IoT and Fog Computing: A low-cost monitoring system employs IoT and fog computing to record vehicle locations using GPS and analyze data to provide insights into traffic behavior, proving effective during peak seasons [17].
- 1.5 Magnetic Sensing Technology: The SenseMag system used magnetic sensors to classify vehicles with high accuracy by analyzing magnetic signals [18].
- 1.6 WiFi-Based Monitoring: The WiTraffic system utilized WiFi Channel State Information (CSI) for non-intrusive traffic monitoring, achieving 96% accuracy in vehicle classification along with effective speed estimation capabilities [19].
- 1.7 Edge Computing and LoRaWAN: This system combines low-cost devices with edge computing for real-time video analytics in traffic monitoring using the lightweight YOLO v3 model, enabling effective management even in resource-constrained environments [20].
- 1.8 BLE-Based Vehicle Detection: A system using low-cost Bluetooth Low Energy (BLE) devices for vehicle detection achieved 97.9% accuracy with a false-positive rate below 4.5% [21].

However, the low cost can lead to trade-offs in data accuracy and reliability, as the quality of measurements can vary between different types, necessitating careful selection based on application requirements. Some sensors may struggle with environmental factors or exhibit drift, resulting in unreliable data.

2. **Mobile Sensing:** Several innovative strategies have been explored to enhance the effectiveness of roadside units (RSUs) in traffic monitoring systems. Among these, mobile sensing has emerged as a key approach, leveraging vehicle-installed sensors to expand coverage and improve data collection.

The benefits of Mobile Sensing is:

- a. **Broader Coverage:** Mobile sensors can cover extensive areas without the need for a dense network of RSUs, allowing for dynamic monitoring of traffic conditions.
- b. **Cost Efficiency:** Utilizing existing vehicles as mobile sensors reduces the need for additional infrastructure investments.

Case Studies and Implementations

- 2.1 **Drive-By Sensing:** Utilizing bus fleets for sensing by integrating sensors with scheduled routes, providing extensive spatial-temporal data collection while maintaining operational efficiency [22].
- 2.2 **Vehicle-to-Vehicle (V2V) Communication:** Protocols developed for vehicular ad hoc networks (VANETs) allow vehicles to share traffic information directly, reducing reliance on RSUs [23].
- 2.3 **Cooperative Multi-Agent Systems:** Innovative methods that combine edge computing with multi-agent systems to estimate traffic density using data from various sources, improving decision-making [24].
- 2.4 **On-Demand Mobile Sensing:** A framework allowing vehicle owners to offer their mobile devices' sensing capabilities as services, reducing energy consumption and network strain [25].

Despite the benefits, mobile sensing faces challenges regarding vehicle density in monitored areas. In low-density scenarios, data collection may become unreliable, and the accuracy of data collected can vary based on environmental conditions and sensor calibration, necessitating ongoing research to ensure high-quality data..

3. **Cloud-Based RSU Systems:** represent a significant advancement in traffic monitoring technology, allowing for efficient data processing and storage.

Benefits of Cloud-Based Systems:

- a. **Reduced Local Processing Demands:** Centralizes data processing, easing the computational load on individual RSUs.
- b. **Scalability:** Easily adapts to growing traffic monitoring needs without significant hardware upgrades.
- c. **Enhanced Data Storage:** Provides extensive storage for long-term data retention and historical analysis.

Examples of Cloud-Based RSU Systems:

- 3.1 **Cloud-Assisted Mobile Crowd Sensing:** Utilizes smartphones to collect traffic data from citizens, improving congestion estimates and allowing for proactive driver guidance [26].

However, the effectiveness of cloud-based systems heavily relies on stable network connections. In areas with poor connectivity, data transmission delays may occur, leading to potential gaps in monitoring. Additionally, storing sensitive data in the cloud raises security concerns that require robust encryption and secure communication protocols.

4. **Fog Computing and Green Technology:** Fog computing has been explored as an effective method for managing traffic monitoring systems by reducing latency and enhancing real-time processing capabilities. Green technologies, particularly solar-powered solutions, are also gaining attention to enhance the sustainability of such systems.

Examples of Fog-Based and Green Solutions:

- 4.1 **Fog-Based Green VANET Infrastructure:** A robust fog-based VANET infrastructure was proposed, enhancing vehicle-to-vehicle and vehicle-to-RSU communication while reducing energy consumption. This system proved to be efficient in ensuring reliable and sustainable traffic monitoring solutions by utilizing green energy sources[27].
- 4.2 **Solar-Powered Smart Camera-RSU Integrated Platform:** This system integrates solar-powered smart cameras with RSUs, providing a low-cost, energy-efficient platform for traffic monitoring. The power management strategy ensures continuous operation, even in fluctuating weather conditions, making it highly effective in urban areas[28].

While previous research has made notable progress, key gaps remain, such as the lack of affordable solutions that match the performance of traditional RSUs, limited scalability for large urban areas, and challenges in real-time data processing. Additionally, few studies have explored the use of commercially available mobile devices as RSUs, which could greatly reduce costs and enhance flexibility.

This paper aims to develop an innovative Android-based RSU system to provide a scalable, cost-effective solution for traffic monitoring. The system will be designed using mobile devices from various cost categories (high, medium, low) and integrate advanced object detection algorithms like YOLOv8 and SAHI to enhance detection accuracy, especially for small or distant objects. In addition to using chat gpt-4o to describe road conditions. The paper proposes an alternative to traditional RSUs, demonstrating how Android-based devices can process real-time data over 4G networks, offering continuous traffic updates to users.

### III. Proposed System Overview

In this section, we present the design and development of a smart roadside unit system aimed at enhancing urban traffic monitoring through advanced AI technologies.

#### A. System Description

This project aims to design and develop smart roadside units as part of an Intelligent Transportation System (ITS) for smart city applications. The system focuses on providing low-cost traffic monitoring solutions using artificial intelligence technologies to enhance urban transportation efficiency. These units rely on low-, medium-, and high-end Android devices to collect images from the roads, which are then sent to dedicated servers for analysis using technologies such as the YOLOv8 model and the SAHI algorithm to detect vehicles and analyze traffic congestion.

The ChatGPT system plays a pivotal role in this analysis, functioning as a server to analyze the images sent from the smart roadside units and determine whether there are traffic accidents, maintenance work, or traffic congestion. After image analysis in the Control and Monitoring Center (CMC) using artificial intelligence, detailed reports are generated, including road status information, such as whether there is congestion, an accident, or maintenance work. These reports are periodically sent to the Android devices installed as smart roadside units, which, in turn, transmit them to the end users via Wi-Fi.

End users benefit from this information through a dedicated application that provides an interactive map showing the status of traffic congestion, accident locations, and maintenance areas, helping improve time management, reduce pollution, and alleviate traffic jams. The system features an open-source infrastructure, allowing for future development to include additional features such as pedestrian monitoring, license plate recognition, and vehicle speed measurement. This system contributes to improving the infrastructure of intelligent transportation in cities and offers

comprehensive solutions to urban transportation issues. Figure 1 shows the proposed system architecture.

### *B. System Process Sequence*

1. Image Capture: Smart RSUs capture images of the roads.
2. Image Transmission: Images are sent via the 4G network.
3. Image Reception: An HTTP server receives the images and forwards them to Servers 1 and 2 for processing.
4. Data Processing: Images are processed by Servers 1 and 2.
5. Data Aggregation: Data is sent to Server 3 to generate comprehensive reports.
6. Report Transmission: Road status reports are sent via the 4G network.
7. User Updates: Reports are sent to smart RSUs.
8. End-User Notification: RSUs broadcast updates via Wi-Fi to users through a dedicated application displaying the city map.

### *C. System Structure*

*1- Smart RSUs:* RSUs considered the backbone of the ITS [29]. infrastructure, providing effective solutions for traffic monitoring. The use of open systems such as Android and open-source detection models like YOLOv8 in addition to using chat gpt-4o to describe road conditions enhances transportation efficiency and reduces the need for complex systems like V2X.. The proposed device is closest to being represented by a mobile device Where it represents integrated platform. It has a processor, memory, GSM, Wi-Fi and a battery. We will conduct the experiment using three smartphones of different quality levels to determine the best one to use as a smart roadside unit. The selected phones are:

- Samsung Galaxy S21 Ultra: A high-quality phone equipped with a quad-camera system, featuring a 108 MP main sensor with 3x and 10x optical zoom capabilities. priced at \$257[30].
- Samsung Galaxy A32 5G: A mid- quality phone with a quad-camera system, including a 48 MP main sensor. priced at \$85[31].
- ZTE Blade A71: A low-quality phone with a triple-camera system, including a 16 MP main sensor. priced at \$78[32].

The devices were mounted on stands to stabilize the image and control the shooting angle, and performance will be compared based on image accuracy, quality, and how it affects vehicle detection.

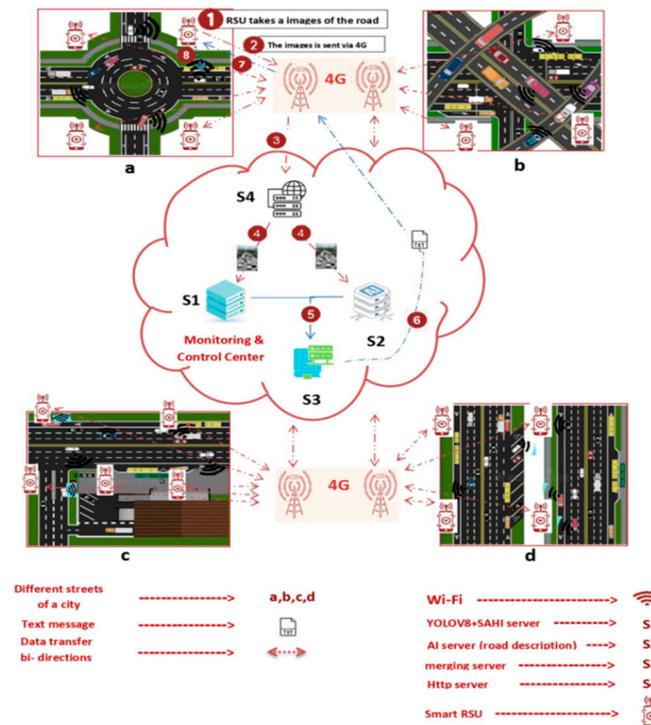
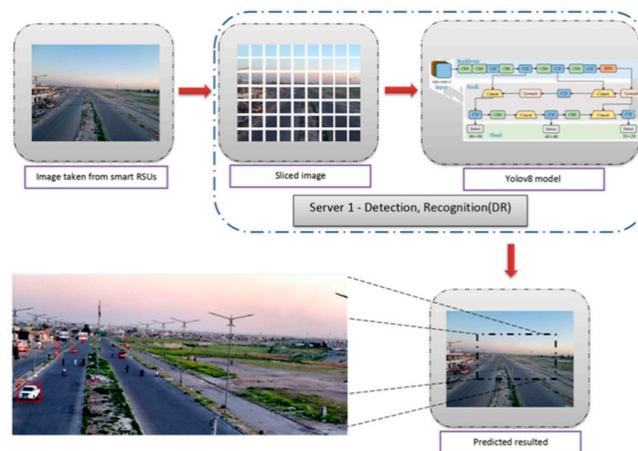


Figure 1. PROPOSED SYSTEM architecture.

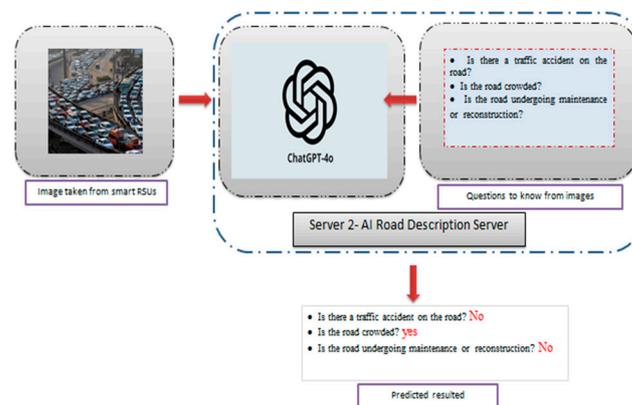
## 2. Control and Monitoring Center (CMC):

The CMC serves as the core of the system for collecting and analyzing traffic data. we assumed It comprises four servers:

- Server 1 - Detection and Recognition: The server employs advanced techniques for real-time traffic monitoring, relying on the YOLOv8 (You Only Look Once) deep learning model[33] and the SAHI (Slicing Aided Hyper Inference) algorithm [34]to analyze images received from smart roadside units (RSUs). YOLOv8 is distinguished by its high efficiency and speed in object detection, making it ideal for applications involving traffic flow analysis and congestion detection. It has been trained on extensive datasets such as COCO, ensuring robust performance in diverse scenarios. SAHI enhances the capabilities of YOLOv8 by dividing large images into smaller slices for more detailed object detection. This technique improves detection accuracy, particularly for small objects, while efficiently utilizing computational resources. The server processes each sliced image independently and then reassembles the results to ensure comprehensive coverage and precise identification of vehicles. This integrated approach allows for efficient and accurate traffic monitoring. Figure 2 shows how server processes images, applying the SAHI algorithm and YOLOv8 model to extract and analyze data.
- Server 2 - AI Road Description: Server 2 - AI Road Description uses artificial intelligence and deep learning, especially ChatGPT-4o[35], to learn about the road conditions in smart cities. ChatGPT-4, an advanced natural language processing model developed by OpenAI, analyzes images of city roads received from smart roadside units based on Android. The image analysis process focuses on three main aspects: identifying traffic accidents, detecting road maintenance works, and evaluating traffic congestion. The summary results from ChatGPT-4o are then combined with data from the first server in the third server to produce a report on the city's road conditions as a whole. Figure 3 illustrates the image processing flow as images enter and exit the second server.



**Figure 2.** Comprehensive depiction of image processing as they enter and exit the initial server.



**Figure 3.** Image processing flow diagram as images enter and exit the second server.

- **Server 3 - Statistical Reports:** is responsible for compiling and analyzing traffic data from Servers 1 and 2. It integrates vehicle data from Server 1, such as the number of vehicles detected within a certain range (e.g., 500 meters for high-cost devices), and calculates traffic density and occupancy rates. From Server 2, it retrieves information on road conditions (traffic accidents, congestion, and maintenance work) with simple yes/no responses. Using this data, Server 3 generates detailed periodic and real-time statistical reports. These reports are sent through the 4G network to roadside units, which then broadcast the information to end users via Wi-Fi. This helps users access updated maps, showing congested areas and reducing time wasted in traffic or accidents.
- **Server 4- HTTP server :** is responsible for collecting image packets from smart Roadside Units (RSUs) . It then sends these images to Server 1 and Server 2 for processing and analysis. Figure 4 shows the CMC.

#### D. Experimental Setup

##### 1. Hardware and Software

- **Hardware:**
  - Laptop: Lenovo with an Intel Core i5 processor (11th generation) and 8GB of RAM.
  - Cameras: Three cameras based on the Android system (high-cost, medium-cost, low-cost).
- **Software:**
  - Operating System: Windows 11 Pro.
  - AI Framework: YOLOv8 and SAHI in Visual Studio Code; AI model ChatGPT-4.

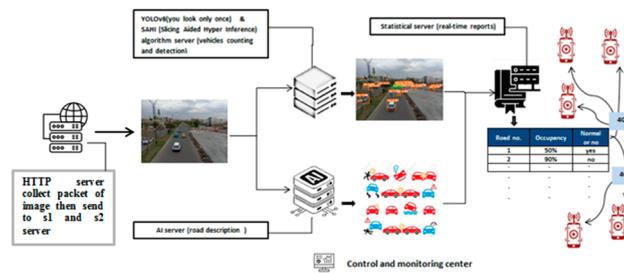


Figure 4. control and monitoring center(CMC).

## 2. Experimental locations

Location 1: Birth Bridge, cameras were positioned at a height of 9.5 meters, overlooking a four-lane street with two secondary streets. This location typically has low traffic, allowing for clear data collection and a wide field of view, as shown in Figure 5.

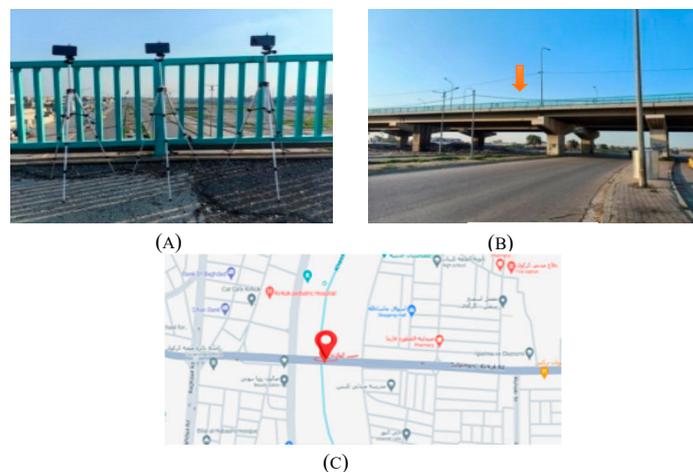


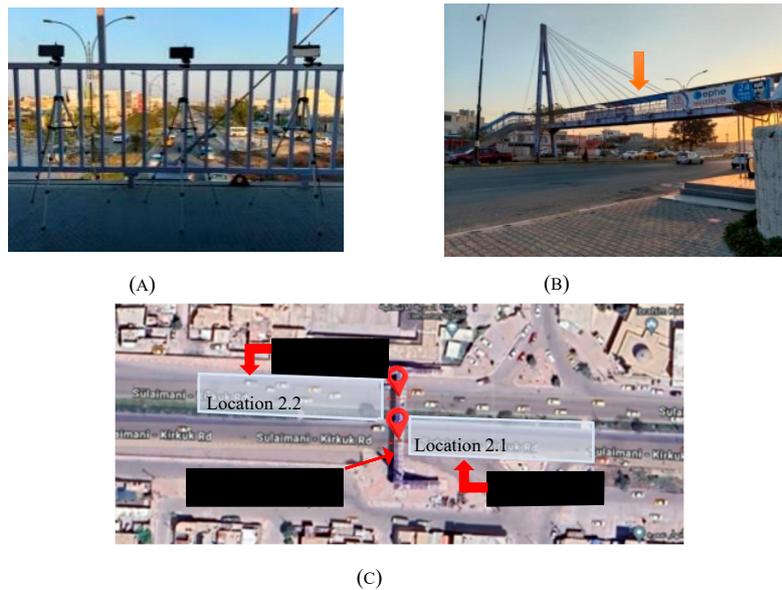
Figure 5. first location (a) How to install Android based devices (b) Real Photography shooting location (c) Photography location on map.

Location 2: Pedestrian Bridge ,, cameras were positioned at a height 6.95 meters height, this bridge spans a busy four-lane street, connecting the Birth Bridge and Shorja Bridge. It includes two sub-sites:

Site 2.1: Cameras directed towards Shorja Bridge for daytime photography during heavy traffic.

Site 2.2: Cameras facing Birth Bridge for nighttime photography, utilizing available lighting. Refer to Figure 6.

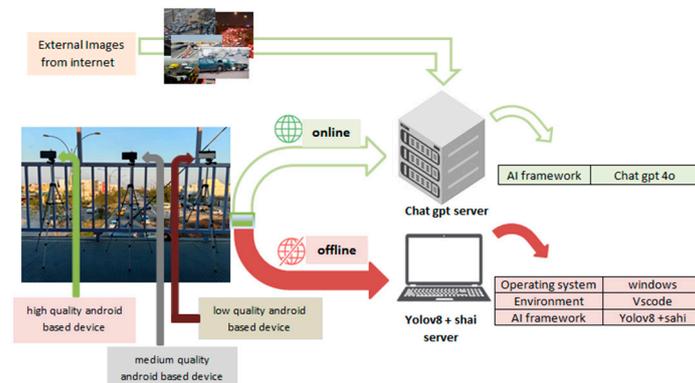
These locations were selected to capture diverse data under varying traffic and lighting conditions.



**Figure 6.** second location (a) How to install Android based devices (b) Real Photography shooting location (c) Photography location on map.

### 3. Data Collection

A total of 270 images were captured at both locations, managed by two individuals. Each device collected 30 images per minute. Figure 7 shows experimental setup.



**Figure 7.** experimental setup.

## IV. Methodology

This section outlines the methodology used in the experiment, including distance measurement methods,

detection rate calculations, and image processing. Advanced AI models and algorithms were utilized to analyze images captured from different locations and devices, with the aim of evaluating the system's performance in detecting road conditions, whether in cases of traffic congestion, maintenance work, or traffic accidents.

### A. Distance Measurement

The distances at the experimental sites were measured using basic tools, relying on the spacing between lighting poles. For example, at the first site, the distance from Pillar No. 4 is about 150 meters, with similar methods applied at other locations. Figure 8 illustrates the locations and distance measurements.



**Figure 8.** locations and distance limits. (a) first location (b) second location 2.1 (c) second location 2.2.

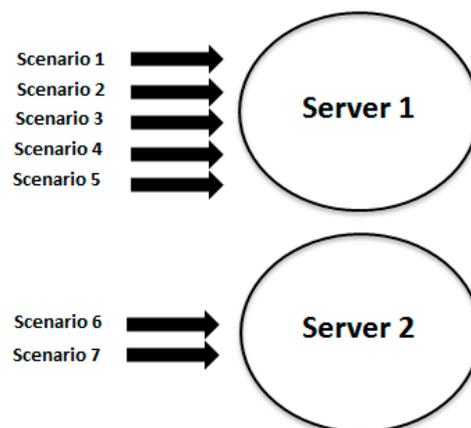
### B. Detection Rate (DR) Calculation

The Detection Rate (DR) is calculated by dividing the number of detected vehicles by the total number of vehicles (detected and undetected) within a specific distance. For example, to find the detection rate within 200 meters, divide the number of detected vehicles from (0 to 200 meters) by the total vehicles within that range. This gives a percentage indicating the system's detection effectiveness. The Detection Rate (DR) formula is

$$DR = \frac{\text{Number of detected vehicles}}{\text{Total number of vehicles (detected + undetected)}} \quad (1)$$

### C. Image Processing and Input Scenarios

This section presents image processing from various devices to assess road conditions, such as accidents, maintenance, or congestion. chat gpt-4o was used to describe road conditions. In addition, YOLOv8 models (large and small) were used to detection vehicles: the large model alone, and the small model combined with SAHI. The SAHI improves object detection by slicing large images into smaller segments (e.g., 640x640), allowing for the detection of small objects at long distances. Images were tested from high, medium, and low-cost Android devices. External images were also analyzed to test the model's performance in diverse scenarios. Figure 9 illustrates the image analysis process for the proposed system.



**Figure 9.** Image input scenarios for servers.

The input scenarios will be as follows:

1- samples images from the three devices (android based) were entered into the Yolov8 model ( size of model :large ) only.

2- same samples images from the three devices (android based) were entered into the Yolov8 ( size of model :small ) model and SAHI algorithm using mask as shown in Figure 10 .

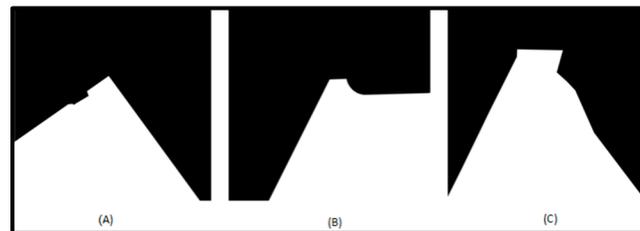
3- Inserting images for the first location , 30 daytime images. For the second location, 30 daytime images for location 2.1, and 30 night images for location 2.2. from the low-cost device Android based to processed using the Yolov8 model ( size of model :small ) with the Sahi algorithm With the following settings slice\_height=640, slice\_width=640, overlap\_height\_ratio=0.2, overlap\_width\_ratio=0.2) and the use of masks.

4- Inserting images for the first location , 30 daytime images. and For the second location, 30 daytime images for location 2.1, and 30 night images for location 2.2. from the medium-cost device Android based to processed using the Yolov8 model ( size of model :small ) with the Sahi algorithm With the following settings slice\_height=640, slice\_width=640, overlap\_height\_ratio=0.2, overlap\_width\_ratio=0.2) and the use of masks.

5- Inserting images for the first location , 30 daytime images. and For the second location, 30 daytime images for location 2.1, and 30 night images for location 2.2. from the high-cost device Android based to processed using the Yolov8 model ( size of model :small ) with the Sahi algorithm With the following settings slice\_height=640, slice\_width=640, overlap\_height\_ratio=0.2, overlap\_width\_ratio=0.2) and the use of masks.

6- Enter a set of 16 images from the first and second locations, captured by three devices—high-cost, medium-cost, and low-cost—into the CHAT-GPT4o model to describe and analyze them. The analysis will determine whether there is a traffic accident, maintenance work, or traffic congestion on the road.

7- Input a set of 34 external images into the CHAT-GPT4o model to describe and analyze them. The analysis will determine whether there is a traffic accident, maintenance work, or traffic congestion on the road.



**Figure 10.** Masks used to remove unwanted image parts (a) mask for first location ,(b) mask for second location 2.1 ,(c) mask for second location 2.2.

## V. Results

This section presents the results of the experiments conducted to evaluate the system's performance in vehicle detection and road condition assessment. The findings highlight the performance of different models across various devices and conditions, with a detailed analysis of detection rates and accuracy. Additionally, it addresses the challenges encountered and how advanced techniques were utilized to improve outcomes.

### A. Detection accuracy analysis for all scenario

This section examines detection accuracy in many settings, emphasizing the influence of image size, device type, and environmental conditions on the efficacy of object detection algorithms.

1-Scenario 1: Large Image Size and Object Detection Challenges involves using large images (min. 3456 x 4608) that are rich in detail. However, resizing them to 640 x 640, the training size for YOLOv8, results in detail loss and reduced accuracy, particularly for small, distant objects. The detection range during the day is up to 150 meters with 90% accuracy, but this accuracy declines significantly beyond that range. At night, the detection range further reduces to 140 meters, with accuracy dropping below 90%. Issues such as overlapping vehicles and natural obstacles like trees negatively impact detection accuracy, while vehicle headlights complicate nighttime detection. Proposed solutions include the SAHI algorithm, which divides large images into 640 x 640 segments to better match training data and preserve details, and the application of masks before processing to mitigate the effects of overlapping vehicles and obstacles.

2-Scenario 2: Image Slicing for Better Detection involves using the SAHI algorithm to divide images into 640 x 640 slices, with the number of slices varying by device resolution: 63 for low-cost (4608 x 3456), 252 for medium-cost (9284 x 6936), and 432 for high-cost devices (12000 x 9000). Results

showed that daytime detection accuracy reached up to 500 meters at the first location, while the second location was limited to 300 meters due to geographical constraints. At night, the detection range dropped to 300 meters due to reduced visibility. The use of masks effectively mitigated issues related to overlapping vehicles and natural obstacles, further enhancing detection accuracy.

The first and second scenarios clearly demonstrate the enhanced effectiveness of using SAHI in conjunction with YOLO compared to using YOLO alone. Figure 11 shows The difference between using the SAHI algorithm and not using it.



**Figure 11.** SAHI algorithm impact on detection accuracy.

3- Scenario 3: Detection accuracy with low-cost devices. After applying the SAHI algorithm and masking technique to images from low-cost devices, results show varying success across locations. In the first location, daytime detection accuracy was high at 150 meters 97% and 300 meters 90%, but dropped to 60% at 500 meters due to the camera sensor limitations. In the second location 2.1, during daytime in a crowded scenario, accuracy was 95% at 100 meters and 86% at 200 meters, decreasing to 77% at 300 meters, again affected by sensor quality and vehicle occlusion. For second location 2.2, nighttime detection accuracy fell to 72% at 100 meters and 56% at 300 meters, primarily due to poor image quality and vehicle headlights.

#### 4-Scenario 4:Detection accuracy with medium-cost devices

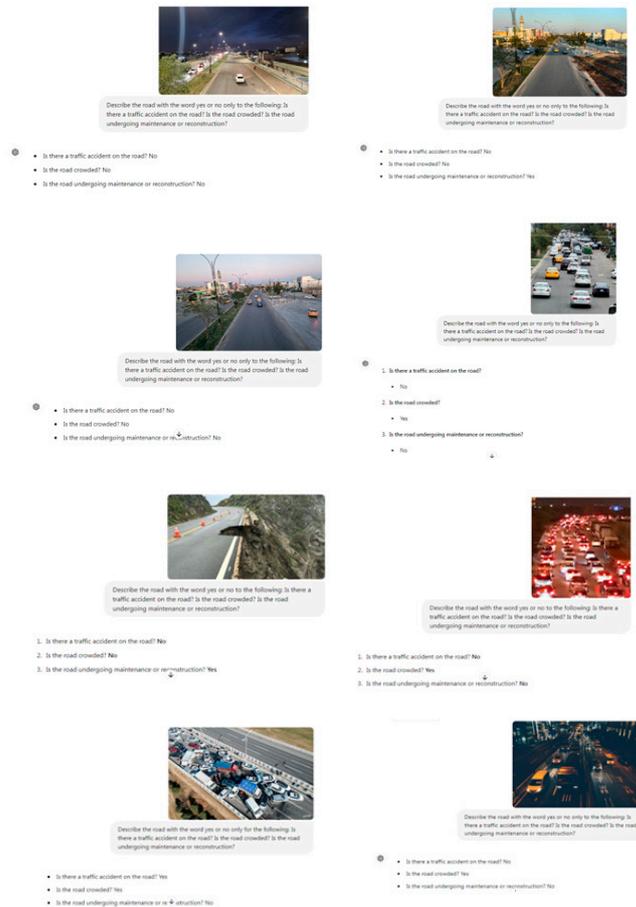
Using medium-cost devices, the SAHI algorithm and masking technique significantly enhanced detection accuracy. In the first location, daytime accuracy was nearly perfect at 150 meters 100% and high at 300 meters 92.8%, but dropped to 83% at 500 meters due to the medium-quality camera sensor. In second location 2.1, during crowded traffic, accuracy was 95% at 100 meters, decreasing to 86% at 200 meters and 77% at 300 meters due to vehicle occlusion. In second location 2.2, nighttime accuracy was high at 100 meters 98% but fell to 87% at 200 meters and 45% at 300 meters, reflecting the challenges posed by night conditions and sensor quality.

#### 5- Scenario 5: Detection accuracy with high-cost devices.

Using high-cost devices, detection accuracy showed the best performance. In the first location, daytime accuracy reached 100% at 150 meters, 95% at 300 meters, and 85% at 500 meters, with minor accuracy drops due to resolution limits. In second location, detection was highly accurate in crowded traffic 97% at 100 meters, 98% at 200 meters, and 95% at 300 meters, with occlusion being the main challenge. Nighttime accuracy remained high at 98.42% at 100 meters, 96% at 200 meters, and 90% at 300 meters, demonstrating the effectiveness of SAHI and masks, despite night conditions slightly reducing detection performance.

#### 6- Scenario 6: Assessing traffic accidents, maintenance work, and congestion with local images.

Images from high-cost, medium-cost, and low-cost Android devices were analyzed to detect traffic accidents, maintenance work, or congestion. The model accurately identified road conditions in all cases with 100% precision. Figure 12 shows the AI road description results using local images and external images.



**Figure 12.** the AI road description results using local images and external images.

7-Scenario 7: Assessing traffic accidents, maintenance work, and congestion with external images.

External images were analyzed to detect traffic accidents, maintenance work, or congestion. The model accurately described road conditions in all cases with 100% precision. Figure 12 shows the AI road description results for using local images and external images.

### B. Detection accuracy analysis (by device type)

This section evaluates the detection accuracy of Android devices under various lighting and traffic conditions across different distances. Table 3 summarizes the analysis, covering day/night settings and traffic scenarios like high, no, and average congestion. The experiments showed that using the YOLOv8 model with the SAHI algorithm significantly improved detection accuracy by splitting images into smaller segments, preserving detail across various devices. Masking also helped reduce issues with overlapping vehicles and obstacles. High-cost devices consistently performed better due to superior image resolution and sensor quality, offering higher detection accuracy in both day and night scenarios. Challenges included reduced accuracy from image quality, lighting conditions, overlapping vehicles, natural obstacles, and vehicle headlights. Figure 13. shows overlapping vehicles, natural obstacles, and vehicle headlights challenges.

**Table 3.** Device Performance Rates Under Various Traffic Conditions and Distances.

Device type	Condition Day/Night	100 m	150 m	200 m	300 m	500 m
Low cost	Low traffic (Day)	N/A	97%	N/A	90%	60%
	Moderate traffic (Night)	72%	N/A	80%	56%	N/A
	High traffic (Day)	95%	N/A	86%	77%	N/A
Medium cost	Low traffic (Day)	N/A	100%	N/A	92.8%	83%
	Moderate traffic (Night)	98%	N/A	87%	45%	N/A
	High traffic (Day)	95%	N/A	85%	56%	N/A
high cost	Low traffic (Day)	N/A	100%	N/A	95%	85%
	Moderate traffic (Night)	98.42%	N/A	96%	90%	N/A
	High traffic (Day)	97%	N/A	98%	95%	N/A

**Figure 13.** Issues resolved using image masks (a) overlapping vehicles parked on the side of the road,(b) natural obstacles, such as trees,(c) cars headlights.

After conducting all the experiments, it was found that the factors affecting detection accuracy include: image quality, where higher-quality sensors improved accuracy, especially for distant objects; lighting conditions, with nighttime photography and vehicle headlights reducing accuracy; overlapping vehicles and natural obstacles, which were mitigated using masks; distance, as accuracy generally decreased with increased distance, particularly in low-cost devices; and finally, the SAHI algorithm, which enhanced accuracy by splitting images into smaller segments while preserving details.

The final report, detailed in Table 4, will be distributed to all smart roadside units and broadcasted to end users. It covers 54 road sections, each 500 meters long, where smart units are deployed. The report includes four key columns: 1) Occupancy rate, indicating road congestion to help users choose alternate routes, saving time and fuel; 2) Congestion status; 3) Construction and maintenance updates, helping drivers avoid affected areas; and 4) Traffic accident reports for safety and route adjustments. The report is 53K bytes and is sent in 37 packets to roadside units for dissemination.

**Table 4.** broadcast message content that sent to end users .

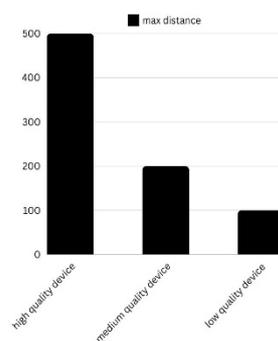
Road no.	Occupancy	crowded	maintenance or reconstruction	traffic accident
1	%80	yes	yes	No
2	%20	No	No	No
3	%60	yes	No	yes
4	%80	yes	yes	No
5	%20	No	No	No
6	%60	yes	No	yes
7	%20	No	No	No
8	%80	yes	yes	No
.	%20	No	No	No
.	%60	yes	No	yes
.	%80	yes	yes	No
.	%20	No	No	No
.	%60	yes	No	yes
.	%20	No	No	No

## VI. Discussion of Results

This section covers vehicle detection experiments and RSU deployment strategies, focusing on maximum distance coverage by Android devices and factors affecting RSU placement. It also addresses RSU distribution methods, balancing cost, performance, and environmental challenges, and provides an overall assessment of the system's real-world effectiveness.

### A. Maximum Distance Coverage

The detection range and accuracy of Android-based devices vary by quality and lighting conditions. High-quality devices detect up to 500 meters with 85% accuracy, improving to 90% at 300 meters, performing well both day and night. Medium-quality devices detect up to 200 meters with 85% accuracy, rising to 95% at 150 meters, also effective in both lighting conditions. Low-quality devices detect up to 100 meters with 95% accuracy during the day, but accuracy drops to 72% at night. These results highlight the need to select devices based on required distance and accuracy, particularly for day vs. night performance. Figure 14 provides details on these results.

**Figure 14.** maximum coverage for the three devices.

### B. RSU Placement Policy

Based on the specifications of the proposed devices and their vehicle detection ranges, we will plan the deployment of roadside units (RSUs) according to established criteria tailored to our devices, differing from previous methods. Our goal is to cover all streets in the Nineveh Governorate, particularly the city of Mosul, chosen for its strategic urban diversity. This map selection ensures comprehensive coverage across various road types, facilitating effective RSU deployment in different metropolitan areas (see Figure 15). The road network in Mosul is ideal for deploying RSUs to enhance traffic management and safety. After analyzing high-cost, medium-cost, and low-cost devices, considering factors like cost, network load, installation complexity, and maintenance, the high-cost device was selected. Covering 500 meters with two devices (One device per side) at \$514, it minimizes installation time, maintenance effort, and network burden, offering the best balance across all factors.

The medium-cost device (\$510) requires six devices(three device per side) for the same coverage, increasing complexity and network load, while the low-cost device (\$780) needs 10 devices(five device per side), further raising costs and maintenance challenges. Thus, the high-cost option is the most efficient.

Smart roadside units were deployed across key streets in the proposed map, covering arterial roads, expressways, commercial roads, and collector roads. Figure 16 illustrates the selected map and the distribution method.

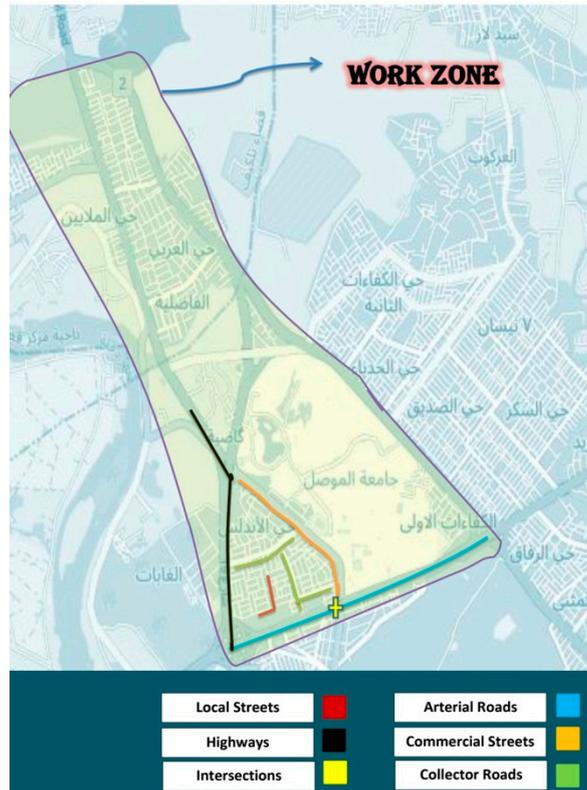


Figure 15. selected map.

The methodology used two algorithms for optimal coverage: one for two-sided roads and another for one-sided roads. Additionally, an anomalous case was considered for roads with slopes and curves. In anomalous cases, such as roads with curves, deploying smart road units at fixed 500-meter intervals may create blind spots. To prevent this, the placement of smart RSUs is adjusted to account for road curvature, ensuring comprehensive coverage and eliminating any blind zones. Figure 17 and Figure 18 show the pseudocode for each algorithm.

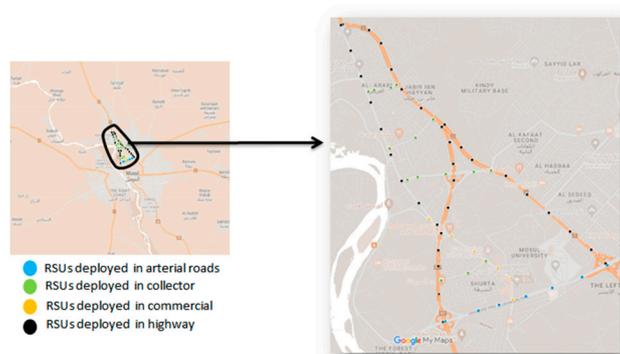


Figure 16. the selected map and the distribution method followed.

```
Input: Length_of_road
Output: Positions of Roadside Units (RSUs)

BEGIN

    // Deploy units at the beginning of the road

    Place RSU at position 0 on the left side of the road

    IF Length_of_road <= 500 THEN
        Place RSU at position Length_of_road on the right side of the road
    ELSE
        // Deploy the units at a distance of every 500 meters
        position = 500
        WHILE position < Length_of_road DO
            // Deploy two units, one on each side.
            Place RSU on the left side of the road at position
            Place RSU on the right side of the road at position

            // Increase distance by 500 meters
            position = position + 500
        END WHILE

        // Deploy the unit at the end of the road on the opposite side of the first unit.
        Place RSU at position Length_of_road on the right side of the road
    END IF
END
```

**Figure 17.** Pseudocode for deploying smart RSUs on two-sided roads.

```

Input: Length_of_road
Output: Positions of Roadside Units (RSUs)

BEGIN

// Deploy the first unit at the beginning of the road

Place RSU at position 0 on the side of the road

IF Length_of_road <= 500 THEN

// If the road length is less than or equal to 500 meters, we suffice with one unit at first

// No need to deploy other units.

RETURN

ELSE

// Deploy units at a distance of every 500 metres

position = 500

WHILE position <= Length_of_road - 500 DO

Place RSU at position on the side of the road

position = position + 500

END WHILE

// At the end of the road, we do not put a unit in the last 500 meters.

END IF

END

```

**Figure 18.** Pseudocode for deploying smart RSUs on one-sided roads.

The deployment covered 7.98 km<sup>2</sup> over 15.4 km using 117 smart RSUs across 65 locations, costing \$30,069, aligning with projections. Despite prior concerns about feasibility, strategic planning and optimal device selection ensured comprehensive coverage without excessive costs. By selecting the high-cost device, we achieved a balance between cost, maintenance, installation, and network integration. The deployment, guided by two algorithms, effectively handled both standard and anomalous cases, ensuring reliable coverage across various road types. Table 5 shows the Main features of the selected map and result details.

**Table 5.** Main features of the selected map and result details.

Selected city map	Mosul (Iraq)
Total length of area in km	15.4 km
Total area in km <sup>2</sup>	7.98 km <sup>2</sup>
Total of location of smart RSUs	65
Total of number of smart RSUs	$(65 \times 2) - 13 = 117$
Total of prices of smart RSUs	$117 * \$257 = \$ 30,069$

### C. Performance Evaluation

After describing the proposed system, it is essential to evaluate its performance. To do so, we need to measure the time associated with the sequence of operations, as illustrated in Figure 19. We will assess the proposed system's performance on HSPA-3G, 4G, and 5G networks using TCP and UDP protocols, with a focus on comparing latency across these technologies. To analyze the HSPA-3G network in Iraq, download and upload speeds were measured using the Speedtest app by Ookla over a 24-hour period. The results are shown in Figure 20.

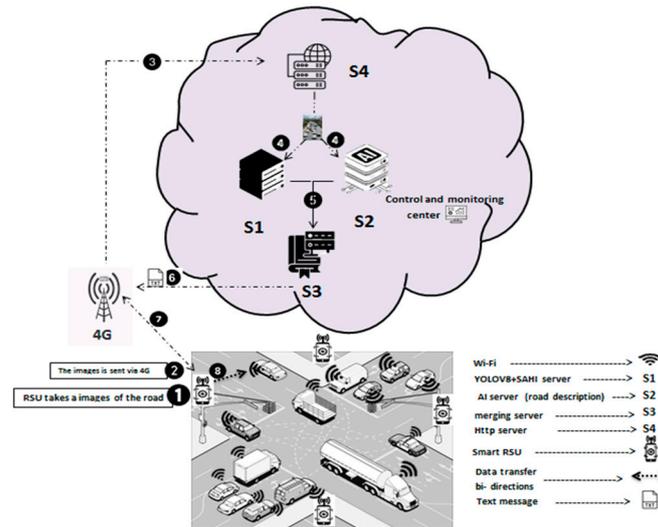


Figure 19. sequence of operations of proposed system.

Transmission and reception times vary throughout the day due to fluctuations in Iraq's HSPA-3G network stability. Network traffic peaks during the day, leading to slower transmission times. The average upload speed is 10.65 Mbps, and the average download speed is 29.17 Mbps, based on 24-hour measurements. To perform accurate calculations, key parameters are outlined in Table 6, with the protocols employed in the system are specified in Table 7. Additionally, In addition to compression rate of smart RSUs, the image compression by 76.47% to the images transmitted from the smart roadside units to the CMC.

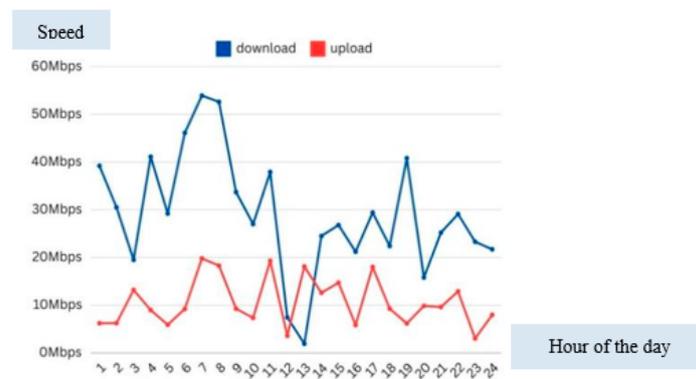


Figure 20. The quality of the download and upload of the HSPA-3G network.

Table 6. Network parameters and values.

Parameter	Value
Avg. Data rate (download)-HSPA-3G	29.17 Mbps
Avg. Data rate (upload)-HSPA-3G	10.65 Mbps
Avg. Data rate (download)- 4G	85 Mbps
Avg. Data rate (upload)- 4G	30 Mbps
Avg. Data rate (download)- 5G	200 Mbps
Avg. Data rate (upload)- 5G	75 Mbps
Avg. Distance	5 Km
Avg. image size	4 MB
Report size	53 KB
Processing rate for smart RSUs	30000 packet/sec[36]
Processing rate for http server	100000 packet/sec[37]

Processing rate for YOLO+SAHI server	55 image/sec per server[38,39]
Processing rate for chat GPT 4O server	55 image /sec per server[38,39]
Processing rate for statistics server	No. of smart RSUs*( <i>deay yolov8 + sahi and chatgpt servers</i> )
Number of smart RSUs	117

**Table 7.** Protocols used in the proposed system.

Protocols	Types
Application layer	HTTP
Transmission layer	TCP and UDP
Network layer	IPV4
Data link layer	Packet Data Convergence Protocol (PDCP) Radio Link Control (RLC)
Physical layer	Radio Resource Control (RRC)

To calculate the time for data transmission and reception, it's essential to determine the delay encountered by the data. Equation (1) expresses the total delay during data transmission and reception via the TCP protocol [39–50].

$$D_{\text{total}} = D_{\text{TCP handshake}} + D_{\text{transmission}} + \text{ACK delay} \quad (1)$$

The TCP three-way handshake establishes a connection between the smart RSUs and the CMC server. It involves three steps: the RSU sends a SYN request, the server replies with SYN-ACK, and the RSU confirms with ACK. To complete the three-way handshake termination, the process involves three steps: FIN, FIN-ACK and ACK. Figure 21 shows timeline of Packet Exchange Using the TCP Protocol. Each step takes one round-trip time (RTT), so with an RTT varies depending on the technology used HSPA-3G, 4G, 5G. Table 8 shows possible RTT parameters and values.

TCP acknowledgments (ACKs) are sent for every 10 packets, with an associated delay calculated as shows in Equation (2)[39–50]:

$$\text{ACK delay} = \frac{\text{number.of packet}}{10} \times \text{RTT}. \quad (2)$$

Transmission delay consists of several components, represented by Equation (3) [39–50]:

$$D_{\text{transmission}} = D_{\text{packet}} \times N \quad (3)$$

where : $D_{\text{packet}}$ : The delay for each packet is calculated using Equation (4):

$$D_{\text{packet}} = D_t + D_p + D_{\text{queuing}} + D_{\text{processing}} + D_{\text{radio scheduling}} \quad (4)$$

where:

$D_t$ : Transmission Delay per Packet , calculated using Equation (5):

$$D_t = \frac{\text{packet size}}{\text{data rate}} \quad (5)$$

$D_p$ : Propagation delay, representing the time taken for a signal to travel from sender to receiver, calculated using Equation (6):

$$D_p = \frac{\text{distance}}{\text{Speed of the signal in the medium}} \quad (6)$$

$D_{\text{queuing}}$ : Queuing delay, the time a packet waits in a queue before being processed or transmitted, calculated using Equation (7):

$$D_{\text{queuing}} = \frac{\lambda}{\mu(\mu-\lambda)} \quad (7)$$

$\lambda$ : Average packet arrival rate (packets/second).

$\mu$ : Average packet service rate (packets/second).

$D_{\text{processing}}$ : Processing delay, the time taken by a network device to process a packet, calculated using Equation (8):

$$D_{\text{processing}} = \frac{1}{\text{packet processing rate}} \quad (8)$$

$D_{\text{Radio scheduling}}$ : Delay due to radio scheduling and channel allocation .  $D_{\text{Radio scheduling}}$  varies depending on the technology used HSPA-3G, 4G, 5G. Table 8 shows possible  $D_{\text{Radio scheduling}}$  parameters and values.

The total number of packets (N) is determined using Equation (9):

$$N = \frac{\text{data size}}{\text{data rate}} \quad (9)$$

$D_{\text{compression}}$ : Delay due to compression image before send to CMC , which typically adds around 1s per image .This delay is added to the transmission equation only.

Table 8. Latency metrics for 5G, 4G, and HSPA-3G technologies.

Delay	HSPA-3G	4G	5G
RTT	10 ms	5 ms	1 ms
TCP handshake	60 ms	30 ms	6 ms
Radio scheduling	2 ms	1.5 ms	1 ms
Queuing	5 ms	3 ms	1 ms

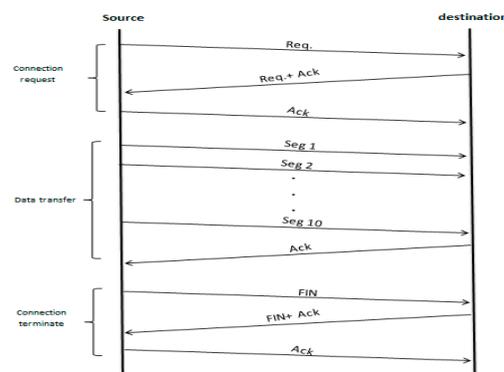


Figure 21. timeline of Packet Exchange Using the TCP Protocol.

The delay that occurs in CMC is represented by Equation (10) :

$$\text{CMC delay} = D_{s1} + D_{s2} + D_{s3} + D_{s4} \quad (10)$$

where: The delay in each server is determined using Equation (11):

$$D_{s1} = \frac{1}{\text{processing rate}} \quad (11)$$

Only the delay in server 3 is determined using Equation (12):

$$D_{s3} = D_{s1} + D_{s2} * \text{No. of smart RSUs} \quad (12)$$

Considering that the selected map almost represents 25% of the city of Mosul and required 117 smart RSUs along roads , we need to cover the city with 468 smart RSUs along roads. The total delay of proposed system is determined using Equation (13):

$$\text{total delay} = \text{delay transmission} + \text{delay CMC} + \text{delay receiving} \quad (13)$$

The difference in finding delay values using the UDP protocol is the elimination of the effect of delay, TCP handshake and acknowledgments. After applying the equations and based on the data , we obtained the results shown in Table 9.

**Table 9.** Total latency for 5G, 4G, and HSPA-3G technologies.

Type of protocols	No. of servers used	HSPA-3G	4G	5G
	for running YOLOv8 + SAHI and ChatGPT			
TCP	3 servers	32.23057 s	21.70112 s	12.95698 s
	6 servers	29.38841 s	18.85895 s	10.11482 s
	9 servers	28.44129 s	17.91184 s	9.167709 s
UDP	3 servers	29.33357 s	20.25265 s	11.67698 s
	6 servers	26.49141 s	17.41052 s	8.83482 s
	9 servers	25.54429 s	16.46341 s	7.88770 s

## VII. Conclusion

this research presents a comprehensive framework for vehicle detection and smart roadside unit (rsu) deployment, specifically focusing on enhancing traffic management and safety in the nineveh governorate, particularly the city of mosul. through rigorous experiments, we established that the detection capabilities of android-based devices significantly vary based on quality, lighting conditions, and placement strategy. high-quality devices achieved up to 500 meters of detection range with an accuracy of 85%, improving to 90% under optimal conditions. the deployment strategy for rsus was meticulously crafted, utilizing tailored algorithms to ensure maximum coverage with minimal cost and complexity. our analysis indicated that selecting high-cost devices is the most efficient choice, achieving significant coverage while balancing installation and maintenance efforts. the successful deployment across 15.4 km of road, utilizing 117 smart rsus at a cost of \$30,069, underscores the feasibility of our approach. performance evaluations across hspa-3g, 4g, and 5g networks highlighted the system's reliability and effectiveness in real-world conditions, achieving satisfactory data rates and low latency. overall, this research not only demonstrates the practicality of smart roadside systems but also contributes valuable insights into optimizing vehicular monitoring in urban environments.

## References

1. Eason, G., Noble, B., & Sneddon, I. N. (2014). On certain integrals of Lipschitz-Hankel type involving products of Bessel functions. Phil. United Nations, Department of Economic and Social Affairs. Retrieved from <https://www.un.org/en/development/desa/news/population/world-urbanization-prospects-2014.html>.
2. Balwan, M., Varghese, T., & Nadeera, S. (2021). Urban traffic control system review - A Sharjah City case study. Proceedings of the 2021 9th International Conference on Traffic and Logistic Engineering (ICTLE), 46-50.
3. Vishwakarma, S., Goliya, D., & Mehar, D. R. (2020). Intelligent transportation systems and its tools as a solution for urban traffic congestion: A review. Journal of Emerging Technologies and Innovative Research.
4. Bista, Raghu Bir & Paneru, Surendra. (2021). Does Road Traffic Congestion Increase Fuel Consumption of Households in Kathmandu City?. Journal of Economic Impact. 3. 67-79. 10.52223/jei3022102.
5. M, A. K., T, N. K., K, A. J., P, A., & P, M. I. (2023). Economic evaluation of traffic congestion & design of traffic signal with simulation at Ottapalam. International Journal for Research in Applied Science and Engineering Technology.
6. Cherednichenko, O., & Valackienė, A. (2022). Intelligent transport systems as traffic flow management tool (The case of Kyiv). Urban Development and Spatial Planning.

7. Subair, S. O., Ibitoye, B. A., & Kuranga, A. T. (2024). Evaluation of traffic congestion in urban roads: A review. *ABUAD Journal of Engineering and Applied Sciences*.
8. Abdelati, H. M. (2024). Smart traffic management for sustainable development in cities: Enhancing safety and efficiency. *International Journal of Advanced Engineering and Business Sciences*.
9. Ramana, K., Srivastava, G., Kumar, M. R., Gadekallu, T. R., Lin, J. C., Alazab, M., & Iwendi, C. (2023). A vision transformer approach for traffic congestion prediction in urban areas. *IEEE Transactions on Intelligent Transportation Systems*, 24(5), 3922-3934.
10. Kumar, A., Sadiqulla, M., Kumar, R., & Bhuiya, P. (2023). Smart traffic management system. *International Journal of Advanced Research in Science, Communication and Technology*.
11. Golhar, Y., & Kshirsagar, M. M. (2021). Emerging technologies for driving road safety and traffic management for urban areas. *Journal of Computer Science*.
12. Kamuni, D., Remala, L., Reddy, M., & Krishnaiah, T. R. (2022). Density-based traffic congestion control system and emergency vehicle clearance. *Proceedings of the 2022 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS)*, 1-6.
13. Li, S. (2024). Real-time traffic congestion detection technology in intelligent transportation systems. *Proceedings of the 2024 IEEE 13th International Conference on Communication Systems and Network Technologies (CSNT)*, 1029-1033.
14. Kanschat, R., Gupta, S. P., & Degbelo, A. (2022). Wireless-signal-based vehicle counting and classification in different road environments. *IEEE Open Journal of Intelligent Transportation Systems*, 3, 236-250.
15. Nguyen, J., Grimsley, R., & Iannucci, B. (2021). TrafficNNode: Low power vehicle sensing platform for smart cities. *Proceedings of the 2021 IEEE International Conference on Smart Internet of Things (SmartIoT)*, 278-282.
16. Martín, J., Khatib, E., Lázaro, P., & Barco, R. (2019). Traffic monitoring via mobile device location. *Sensors*, 19(10), 4505. <https://doi.org/10.3390/s19204505>.
17. Vergis, S., Komianos, V., Tsoumanis, G., Tsipis, A., & Oikonomou, K. (2020). A low-cost vehicular traffic monitoring system using fog computing. *Smart Cities*.
18. Wang, K., Xiong, H., Zhang, J., Chen, H., Dou, D., & Xu, C. Z. (2021). SenseMag: Enabling low-cost traffic monitoring using noninvasive magnetic sensing. *IEEE Internet of Things Journal*. <https://doi.org/10.1109/JIOT.2021.3074907>.
19. Won, M., Zhang, S., & Son, S. H. (2017). WiTraffic: Low-cost and non-intrusive traffic monitoring system using WiFi. *Proceedings of the 2017 26th International Conference on Computer Communication and Networks (ICCCN)*, 1-9.
20. Seid, S., Zennaro, M., Libsie, M., Pietrosemoli, E., & Manzoni, P. (2020). A low cost edge computing and LoRaWAN real time video analytics for road traffic monitoring. *Proceedings of the 2020 16th International Conference on Mobility, Sensing and Networking (MSN)*, 762-767.
21. Maus, G., & Brückmann, D. (2020). A non-intrusive, single-sided car traffic monitoring system based on low-cost BLE devices. *Proceedings of the 2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, 1-5.
22. Ji, W., Han, K., & Liu, T. (2023). Trip-based mobile sensor deployment for drive-by sensing with bus fleets. *Transportation Research Part C: Emerging Technologies*, 157. <https://doi.org/10.1016/j.trc.2023.104404>
23. Martuscelli, G., Boukerche, A. F., Foschini, L., & Bellavista, P. (2016). V2V protocols for traffic congestion discovery along routes of interest in VANETs: A quantitative study. *Wireless Communications and Mobile Computing*, 16, 2907-2923.
24. Guastella, D. A., & Pournaras, E. (2023). Cooperative multi-agent traffic monitoring can reduce camera surveillance. *IEEE Access*, 11, 142125-142145.
25. Rahman, S. A. (2017). On-demand mobile sensing framework for traffic monitoring.
26. Yılmaz, Ö., Görgü, L., O'Grady, M. J., & O'Hare, G. M. (2021). Cloud-assisted mobile crowd sensing for route and congestion monitoring. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2021.3074907>.
27. Ali, Q. (2022). Realization of a robust fog-based green VANET infrastructure. *IEEE Systems Journal*. <https://doi.org/10.1109/JSYST.2022.3215845>

28. Ali, Q. (2022). An efficient power management strategy of a solar-powered smart camera-road side unit integrated platform. *International Journal of Sensors, Wireless Communications and Control*, 13. <https://doi.org/10.2174/2210327913666221024160809>
29. Ibrahim, Q. (2016). Enhanced power management scheme for embedded road side units. *IET Computers & Digital Techniques*, 10(4), 174-185.
30. GSMArena. (n.d.). *Samsung Galaxy S21 Ultra 5G*. GSMArena.com. Retrieved October 17, 2024, from [https://www.gsmarena.com/samsung\\_galaxy\\_s21\\_ultra\\_5g-10596.php](https://www.gsmarena.com/samsung_galaxy_s21_ultra_5g-10596.php).
31. GSMArena. (n.d.). *Samsung Galaxy A32 5G*. GSMArena.com. Retrieved October 17, 2024, from [https://www.gsmarena.com/samsung\\_galaxy\\_a32\\_5g-10648.php](https://www.gsmarena.com/samsung_galaxy_a32_5g-10648.php).
32. GSMArena. (n.d.). *ZTE Blade A71*. GSMArena.com. Retrieved October 17, 2024, from [https://www.gsmarena.com/zte\\_blade\\_a71-11240.php](https://www.gsmarena.com/zte_blade_a71-11240.php).
33. Ultralytics. (n.d.). YOLOv8 models. Ultralytics. Retrieved October 14, 2024, from <https://docs.ultralytics.com/models/yolov8/>.
34. Akyon, F. C., Altinuc, S. O., & Temizel, A. (2022). Slicing aided hyper inference and fine-tuning for small object detection. arXiv. <https://doi.org/10.48550/arXiv.2207.10316>.
35. OpenAI. (2023). GPT-4: The latest milestone in OpenAI's effort in scaling up deep learning. Retrieved from <https://openai.com/gpt-4/>.
36. Ali, Q. I. (2008). An efficient simulation methodology of networked industrial devices. Proceedings of the 2008 5th International Multi-Conference on Systems, Signals and Devices (SSD), Amman, Jordan, 1-6. <https://doi.org/10.1109/SSD.2008.4632835>.
37. Ultralytics. (2023). YOLOv8 TensorRT integration and performance improvements. Retrieved from <https://docs.ultralytics.com/integrations/tensorrt/?h=yolov8#what-are-the-performance-improvements-observed-with-yolov8-models-exported-to-tensorrt>.
38. NVIDIA Corporation. (2020). NVIDIA DGX A100 system datasheet. Retrieved from <https://images.nvidia.com/aem-dam/Solutions/Data-Center/nvidia-dgx-a100-datasheet.pdf>.
39. Q. I. Ali, "GVANET project: an efficient deployment of a self-powered, reliable and secured VANET infrastructure," *IET Wireless Sensor Systems*, vol. 8, no. 6, pp. 313-322, 2018.
40. Q. I. Ali, "Design and implementation of an embedded intrusion detection system for wireless applications," *IET Information Security*, vol. 6, no. 3, pp. 171-182, 2012.
41. Q. I. Ali, "Green communication infrastructure for vehicular ad hoc network (VANET)," *Journal of Electrical Engineering*, vol. 16, no. 2, p. 10, 2016.
42. Q. I. Ali, "Performance evaluation of WLAN internet sharing using DCF & PCF modes," *Int. Arab. J. e Technol.*, vol. 1, no. 1, pp. 38-45, 2009.
43. H. M. Mohammed and Q. I. Ali, "E-proctoring systems: A review on designing techniques, features and abilities against threats and attacks," *Quantum Journal of Engineering, Science and Technology*, vol. 3, no. 2, pp. 14-30, 2022.
44. Q. I. Ali, "Design, implementation & optimization of an energy harvesting system for VANETs' road side units (RSU)," *IET Intelligent Transport Systems*, vol. 8, no. 3, pp. 298-307, 2014.
45. Q. I. Ali, "An efficient simulation methodology of networked industrial devices," in *Proc. 5th Int. Multi-Conference on Systems, Signals and Devices*, 2008, pp. 1-6.
46. Q. I. Ali, "Security issues of solar energy harvesting road side unit (RSU)," *Iraqi Journal for Electrical & Electronic Engineering*, vol. 11, no. 1, 2015.
47. Q. I. Ali, "Securing solar energy-harvesting road-side unit using an embedded cooperative-hybrid intrusion detection system," *IET Information Security*, vol. 10, no. 6, pp. 386-402, 2016.
48. Q. Ibrahim, "Design & Implementation of High-Speed Network Devices Using SRL16 Reconfigurable Content Addressable Memory (RCAM)," *Int. Arab. J. e Technol.*, vol. 2, no. 2, pp. 72-81, 2011.

49. M. H. Alhabib and Q. I. Ali, "Internet of autonomous vehicles communication infrastructure: a short review," *Diagnostyka*, vol. 24, 2023.
50. Q. I. Ali and J. K. Jalal, "Practical design of solar-powered IEEE 802.11 backhaul wireless repeater," in *Proc. 6th Int. Conf. on Multimedia, Computer Graphics and Broadcasting*, 2014.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.