Article

IoT-Enabled Vehicle Speed Monitoring System

Shafi Ullah Khan [1,2], Noor Alam [2,3], Sana Ullah Jan [4] and In Soo Koo [1,*]

1   Department of Electrical Electronic and Computer Engineering, University of Ulsan, Ulsan 44610, Korea;
    16pwele4945@uetpeshawar.edu.pk
2   Department of Electrical Engineering, University of Engineering and Technology Peshawar,
    Peshawar 25120, Pakistan; mailtonooralam@gmail.com
3   Department of Computer Software Engineering, University of Engineering and Technology Mardan,
    Mardan 23200, Pakistan
4   School of Computing, Edinburgh Napier University, Edinburgh EH10 5DT, UK; s.jan@napier.ac.uk
*   Correspondence: iskoo@ulsan.ac.kr

Abstract: Millions of people lose their lives each year worldwide due to traffic law violations, specifically, over speeding. The existing systems fail to report most of such violations due to their respective flaws. For instance, speed guns work in isolation and cannot measure speed of all vehicles on roads at all spatial points. They can only detect the speed of the vehicle the line of sight of the camera. A solution is to deploy a huge number of speed guns at different locations on the road to detect and report vehicles that are over speeding. However, this solution is not feasible because it demands a large amount of equipment and computational resources to process such a big amount of data. In this paper, a speed detection framework is developed to detect vehicles' speeds with only two speed guns, which can report speed even when the vehicle is not within the camera's line of sight. The system is specifically designed for an irregular traffic scenario such as that of Pakistan, where it is inconvenient to install conventional systems. The idea is to calculate the average speed of vehicles traveling in a specific region, for instance, between two spatial points. A low-cost Raspberry Pi (RPi) module and an ordinary camera are deployed to detect the registration numbers on vehicle license plates. This hardware presents a more stable system since it is powered by a low consumption Raspberry Pi that can operate for hours without crashing or malfunctioning. More specifically, the entrance and exit locations and the time taken to get from one point to another are recorded. An automatic alert to traffic authorities is generated when a driver is over speeding. A detailed explanation of the hardware prototype and the algorithms is given, along with the setup configurations of the hardware prototype, the website, and the mobile device applications.

Keywords: Internet of Things (IoT); artificial intelligence; Raspberry Pi (RPi) module

1. Introduction

The Internet of Things (IoT) is a network of electronic devices (usually referred to as things in this context) that communicate with each other through the Internet [1], also sometimes called cyber-physical systems [2]. These devices collect data from the environment and store it in a cloud that can be easily accessed and retrieved whenever needed. Generally, this technology incorporates sensors and actuators to capture and transmit data over the Internet. On-demand service provision, resource sharing, and flexibility are all fundamental features of the cloud-based IoT infrastructure. Smart energy, smart city, health monitoring systems, and smart vehicle monitoring systems are some of the crucial applications of IoT.

The present age transportation industry is associated with huge costs from disasters, large numbers of accidents, and injuries that lead to loss of life. All over the world, the major reasons for such accidents, injuries, and loss of life include violating traffic laws against speeding and dangerous driving. Despite the reality that dangerous driving is a

major problem, the current detection system of patrol officers is inefficient [3]. Increased speed multiplies the risks of collisions and injuries. At a higher speed, the driver needs a more distance to stop. The most important requirement is to create a system that can track, report, and record vehicle speeds.

According to a World Health Organization (WHO) report [3], over 1.2 million people around the globe each year lose their lives due to road accidents. National Highway & Motorway Police officials in Pakistan lament that, on average, 15,000 to 16,000 people die each year due to traffic accidents, mostly occurring due to speeding and violations of other traffic laws. For ensure secure and safe journeys on motorways, it is necessary to identify areas where speed and traffic violations occur. An isolated speed detection system does not provide such information, therefore resulting in continuously increasing numbers of motorway accidents in Pakistan. Previously, Abbas [4] published a method for measuring speed based on license number plates; however, this method has a flaw in that it is hard to distinguish the number plate of the second car when two vehicles appear in a frame following each other. Sina et al. [5] proposed using headlight detection to monitor a moving vehicle and then using the pinhole and Euclidian distance techniques to compute its speed. However, this method has a drawback if two cars appear in the same frame since detecting the rear vehicle becomes difficult. A genetic programming method for detecting vehicle speed was proposed by Song et al. [6]. Lin et al. [7] calculated vehicle speed using imaging geometry, image blur, and camera posture. For speed measurement, Rad et al. [8], employed a computer vision approach.

Fortunately, these issues can be resolved with proper application of the Internet of Things (IoT) targeting risk mitigation, loss prevention, protection, and cost reduction. Cameras integrated with Raspberry Pi (RPi) modules, along with deep-learning analytical tools and image processing algorithms, can gather information and make predictions and decisions that will make roads safer. Building such an IoT-based, intelligent, average speed–monitoring system has a range of advantages, including improving road conditions, reducing and managing costs, and increasing reliability and protections in traffic.

This paper develops an IoT-based vehicle speed monitoring system that can provide efficient results in an overcrowded and irregular traffic environment such as Pakistan's traffic situation. The employment of cutting-edge procedures and technologies has made the traffic system safer and more secure for all sorts of vehicles, stopping them from speeding and breaching traffic regulations. Using image processing techniques and RPi modules, we established a linked infrastructure network of IoT with RPi modules deployed for speed checks and accident detection on the high-speed highways.

The technology is critical for Pakistan's chaotic traffic where the conventional traffic monitoring system cannot be applied due to two main reasons: (1) continuous supply of power issues and (2) the highly irregular patterns in the traffic. These two issues restrict the existing methodologies to be applied in Pakistan and needs a robust system that can continuously check the speed of the vehicle irrespective of its trajectories even when the power is cut off for a period.

From an engineering standpoint, the suggested method is more automated and efficient, with an accuracy of greater than 95%. The prototype's vision system is tough and reliable, and it can function in a variety of temperature conditions, ranging from frigid cold to scorching heat, or fine dust from the desert to freezing rain from northern regions.

The organization of this paper is as follows: Section 2 presents a literature review to summarize existing research methodologies of speed monitoring systems. Section 3 provides a detailed description of the techniques applied in this project toward designing the prototype of SM&AD system for smart cities. This section further briefly explains the different pieces of equipment used in hardware design of proposed model, the various project modules and the optimal solution for the problem statement. The experimental results and the real-time functionality of project for testing the quality are given in Section 4. Finally, Section 5 concludes the paper, discusses some real-life use-cases and highlights the future works which can be carried out for further advancements in the field.

2. Literature Review

Pakistan is a populous developing country with a low per capita income. Vehicle usage for transportation has risen in recent years, and road safety is still an issue, with over 4000 persons dead and almost 10,000 wounded annually due to traffic accidents [9]. Over speeding of vehicles on highways is the leading cause of accidents and fatalities, according to statistics. As a policymaker, it is a challenge for all national leaders to give prompt attention to this growing issue. As a result, the study is required to determine a feasible remedy to this oversight.

Currently, speed guns are utilized in Pakistan to detect vehicle speed. These guns function in segregation and can detect the speed of vehicles located in front of a camera. The technology is designed to track a vehicle's speed even while it is out of view of the camera. In the current system, someone must always monitor the speed and then notify the traffic wardens, who will manually generate a citation for the violating car. The suggested approach can create an invoice to penalize cars that exceed speed limits without the assistance of traffic officials.

To overcome these problems, several techniques have been used to provide smart ways to control and manage traffic systems. Each of these solutions has taken its own path towards managing, controlling, and solving traffic problems. One of the major causes of accidents is speeding, which increases the risks of a crash and injuries [10]. In view of this, many attempts have been made to find a well-organized and reliable means to monitor speeding. This research is aimed at developing an over speeding detection system that can perform reliable speed detection. Since the over speeding detection system functions on a radio frequency, vehicle-related data is thus easier to collect. It can also be inferred that the IoT, as one of the innovations for the detection of speeding vehicles, would allow data to be conveniently transmitted and that this technique can be built into roads in the future.

In September 2009, the European Union (EU) approved the IoT Strategic Research Roadmap, proposed by the Cluster of European Research Projects (CERP). The Cluster of European Research Projects on the Internet of Things (CERP-IoT) promotes and shares research projects and relevant research findings on the IoT, highlighting the implementation of sensor technologies such as the Intelligent Transport System (ITS) [11], family-domain smart eHealth/mHealth, wearable sensing and computing devices, green buildings, smart homes, smart cities, and many more. The report formally proposes a new communication dimension for Information and Communication Technologies (ICT) that is maintained anytime and anyplace by anything and anyone, providing any service within any network. In fact, this is an extension and further development of the contact paradigm "anytime-anywhere-anyone" [11]. These ventures shall be considered multiple IoT-related programs in nearly 20 areas funded in the EU Seventh Framework Program (FP7) public tender announcement.

Isong et al. [12] proposed an approach to automatically monitor drinking-and-driving and speeding on South Africa's roads without having to deploy a manual system. They employed the IoT, vehicular ad-hoc networks (VANETs), and cloud computing to bring this innovation to the transportation sector.

In recent years, the VANET concept has received considerable attention from researchers in the field. It uses powerful technology that can support an ITS [13,14]. It has been implemented to increase road safety, boost driver visibility, and improve transportation quality through the provision of real-time road traffic information. It is also used to enhance the entertainment system of a vehicle in motion for the comfort of passengers [15]. VANETs currently use dedicated short-range communication (DSRC), which is the IEEE 802.11p [12] standard for improving road safety and remote traffic control.

In modern civilization, transportation plays a key role in the economic development and success of a country [16,17]. Today, control systems can manage certain situations, but not very effectively, because they require constant input from humans. There is a need for a system with regular satellite updates so that traffic can be managed smoothly. This could be done using IoT /internet protocols for transmitting data. A unified server must be

framed, so all humans can see after a complete traffic review. Diversion routes and waiting times would be calculated based on traffic density.

In [18], the authors suggested a new approach for handling a traffic grid that uses an IoT definition. Intelligent traffic controllers are framed with devices such as the PiCamera, Raspberry Pi, IR sensors, and radio frequency identification (RFID). IR sensors is used to determine the traffic density. RFID technology is also utilized to provide a green path (zero traffic) for emergency vehicles. This research suggests two types of machine management: manual (with human intervention) and automated (without human intervention).

Jain et al. [19] developed a speed detection system that uses a radar gun or speed gun to calculate a vehicle's speed, and if it exceeds the maximum limit, specifics of the vehicle are transmitted to the nearest control room. Speed is monitored by means of micro-electromechanical system (MEMS)-based sensors.

A proposed automatic license plate recognition (ALPR) algorithm uses the Hough transform to detect license plates by tracing the edges of the plates [20]. This transformation is a memory- and time-consuming process. Furthermore, in this wavelength-based technique, high-frequency coefficients are used to precisely detect license plates. Nevertheless, this algorithm does not perform efficiently.

Sawicki et al. [21] presented the use of RADAR for vehicle speed calculation. A RADAR works on the principle of bouncing a radio signal back from a moving object and the receiver receives the reflected signal. The receiver RADAR calculates vehicle speed on the basis of calculated difference in frequency. Given the high cost and susceptibility to target identification error, RADAR has not received widespread implementation for vehicles speed monitoring.

Song et al. [6] presented a genetic programming approach for detecting vehicle speed. Lin et al. [7] used an approach of imaging geometry, blur extent in the image and camera pose to calculate vehicle speed. Rad et al. [8] used an approach of computer vision techniques for speed measurement. These approaches suffer issues such as number plate occlusion, headlight occlusion, and/or being costly.

Xiao [22] built an intelligent traffic monitoring system that relies on automated detection of vehicles. However, in adverse weather conditions, this automated image-based identification of vehicle license plates has a low recognition rate. Research and design demonstrate that it is feasible and affordable to build an intelligent traffic monitoring system based on the IoT. An intelligent traffic monitoring system based on the IoT has a range of benefits, such as low cost, high reliability, and robustness to adverse weather conditions. Furthermore, it would also have a wide range of uses and would be easy to enforce.

In [23,24], the authors have developed an intelligent automatic luminous control system for streetlights according to the intensity of light in the environment. The system is intelligent enough to identify between day and night to switch mode of the system accordingly. This system has been equipped with surveillance system using cameras to control the traffic on road. This system translates audio-free video into text data to extract knowledge about the vehicle. The model presented here is related to controlling the traffic to avoid traffic jam and congestions. This does not offer the speed monitoring feature which is important to enforce the traffic rules.

3. Proposed IoT-Enabled Speed Monitoring System

This section describes the techniques, methods, algorithms, and tools used in this project in the IoT environment. It also covers all architectures used in the system. The description of project modules, algorithms, and components, along with their uses, configurations, and working details, are given below.

3.1. License Plate Detection

First, a system is developed to detect the license plate from a still image or a single video frame containing a vehicle. The system is developed by using contours with the help

of the Open-Source Computer Vision (OpenCV) library and Python. The captured vehicle image is converted to grayscale, and then, by using bilateral filters and edge detection algorithms, the useful information (i.e., the license plate) is extracted. After getting the plate from the image, all other information is excluded, and the area with the license is marked for character segmentation.

3.2. Character Segmentation

After detection, the registration from the plate is saved locally in the system as a new image for further processing to detect the numbers and characters used. These characters are detected by using a Google OCR algorithm and the Google Vision API.

API stands for application program interface, and it is a way for two platforms to communicate and exchange data. In our system, we used APIs such as Plate Recognizer API to get the number plate from a cropped image of the number plate from the vehicle, and Google Location API in the mobile App to trace the camera modules in different locations. The process is illustrated in Figure 1.
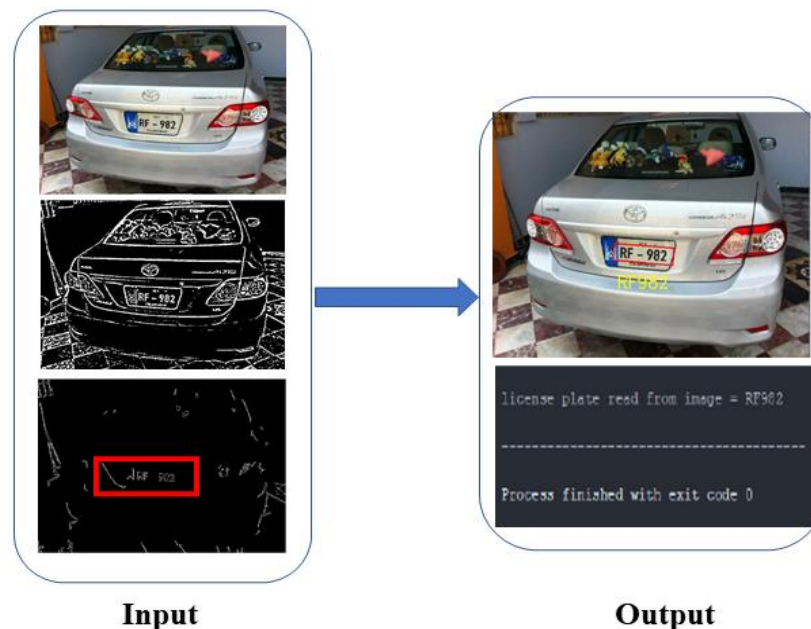

Figure 1. License plate detection and character segmentation.

3.3. Configuration for Live Streaming

Now the system needs to record and stream the vehicles passing by on the road. For this Raspberry Pi is configured by using PuTTY software to make the Raspberry Pi a live streaming web server. The recorded video (in MP4 format) is used at one camera point, and the live RPi-based webcam server is at another point to detect speeding vehicles. A vehicle's registration acquired from the recorded video is then used to extract further details on the vehicle and driver. Figure 2 illustrates the visual results of vehicle detection by subtracting the negative points from captured frame.
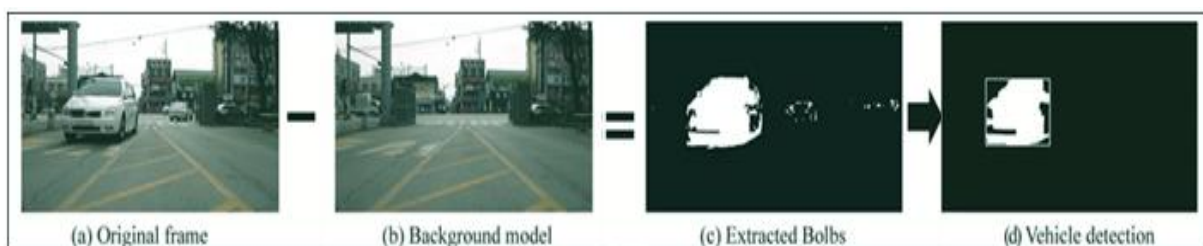

Figure 2. Background subtraction to remove negative points.

PuTTY is a third-party program that allows one to connect a PC to the Raspberry Pi module, which runs the Raspbian operating system. Before beginning the work, the Raspberry Pi module is set up using PuTTY to do certain necessary installations, preparing the pi module for actual implementation.

3.4. Training the System

Live videos were taken to train our system to detect vehicles from live traffic video instead of a still image. First, the OpenCV background subtraction algorithm is used to remove the areas other than the vehicle (negative points) from the video. Different datasets are used to train the model with the help of the Cascade Trainer GUI, a tool that can be used to train, test, and improve classifier models.

Now that the videos have been received from two sources, the interface must be chosen to feed the live recording to the system for additional training and processing in order to obtain a still image and generate vehicle data. We will choose the video file as the Entrance and the Raspberry PI webcam as the exit point, and the IoT infrastructure for receiving live video streaming on highways will be achieved using one of the following methods:

(1) IP camera (RPI CV2 camera) auto-detected locally or manually through the HTTP method.
(2) Web camera with USB
(3) Computer video file (.mp4, .avi, .mpeg)

3.5. Generating the Data and Getting the Required JSON File

After configuring and obtaining the live traffic stream on the highway, various video and image-processing algorithms such as the open-source Library OpenCV, PyTesseract, Open ALPR, Background Subtraction Algorithms, Plate Recognizer API's, haar cascade classifier and PixelLib object detection, and JSON libraries are used to generate the required JSON file. This file contains all vehicle information, such as a still photo, the license plate, and the vehicle's roadway entry and exit timings, as shown in Figures 3 and 4.
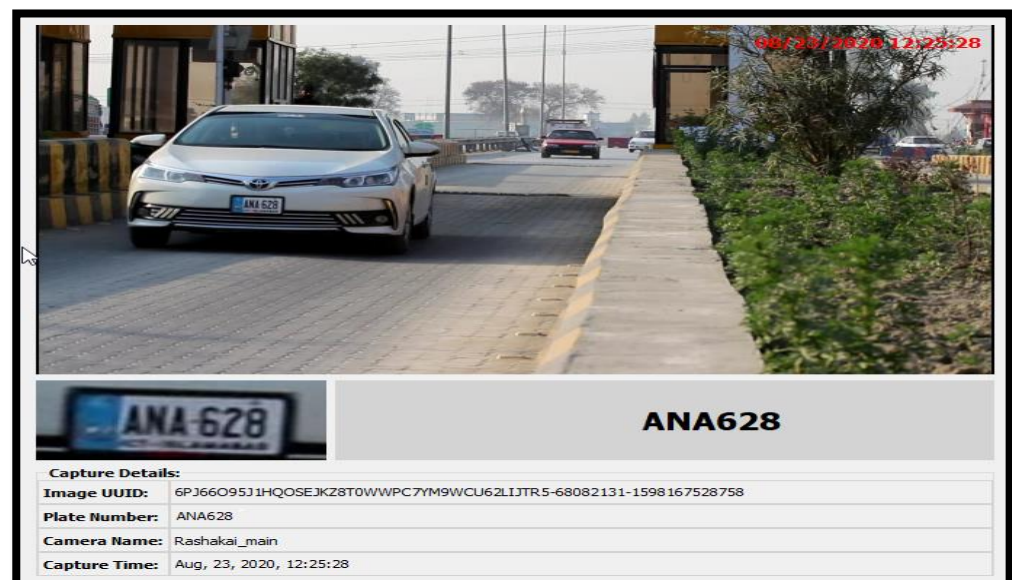

Figure 3. Data fetched by the system, including a vehicle photo, capture time, and the license plate.
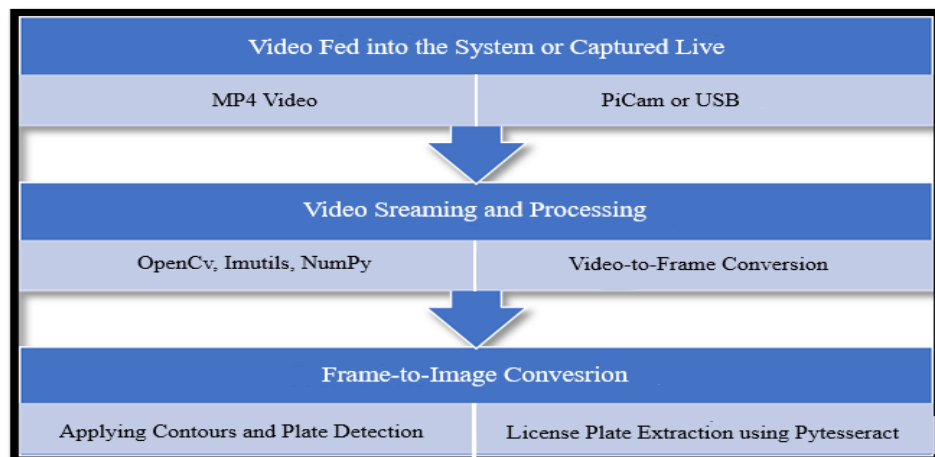
Figure 4. Overall system process for detecting license plates.

3.6. Shifting to AWS Cloud

The whole system is implemented over the cloud using Amazon Web Services (AWS), which offers a list of services. The AWS cloud is used to organize and manage large amounts of data obtained from the system. The cloud service is also responsible for the management of cross-communication among Firebase, the website, and the mobile application, and it also provides backup to store large amounts of data from the system.

3.7. Creating S3 Buckets for Individual Cameras

Three buckets are created in Amazon Simple Storage Service (Amazon S3), which could be increased with increases in the number of cameras connected to the system. One of the buckets is used for the entrance point data receiver, and another for generated results from the exit point, while the third bucket is used to back up all captured license plate data. A list of the buckets is presented in Figure 5.
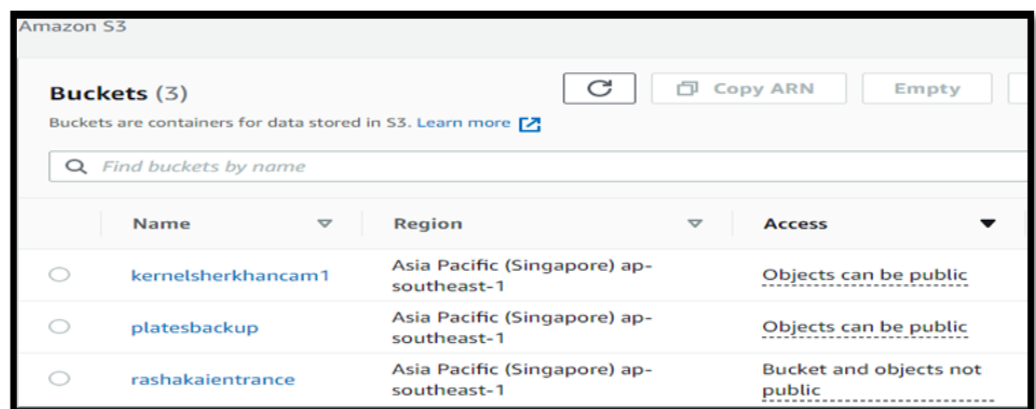

Figure 5. Launching three buckets.

3.8. Scripting and Making the Website Live Globally

Because our proposed system resides in the cloud, some scripts are created with Python and Boto3, which help us to create, configure, and manage AWS storage and services. With the command prompt, some scripts were also created just to copy the newly generated plates to a folder and then move it to the S3 reserved buckets.

3.9. Smart Speed–Monitoring Website

The website of a Smart Speed Monitoring System (SSMS) is a platform that provides easy average speed calculations by allowing users to automatically find a vehicle that has

exceeded the speed limit while moving between two terminals. It provides cheap and efficient speed monitoring management that comprises more automation and less manual activity. The website is user-friendly, which allows live information about speeding vehicles on motorways between any two fixed points.

To present the data and show the results, the website was created using HTML, CSS, and PHP, with Firebase-comprehensive APIs for use on the live website. To access the website remotely, Elastic Beanstalk assigns it a public IP address.

All the vehicle data, such as the entry point, exit point, license plate and time taken to travel between two points, are shown on the website. The website automatically compares the data with an already installed dataset and decides whether the vehicle violated traffic laws or not. A screenshot of the website's home page is in Figure 6. The site map for the SSMS website is in Figure 7.
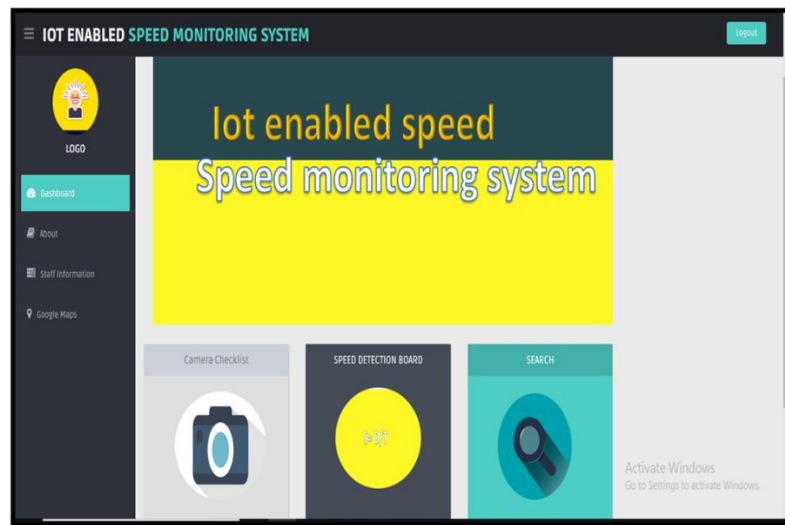

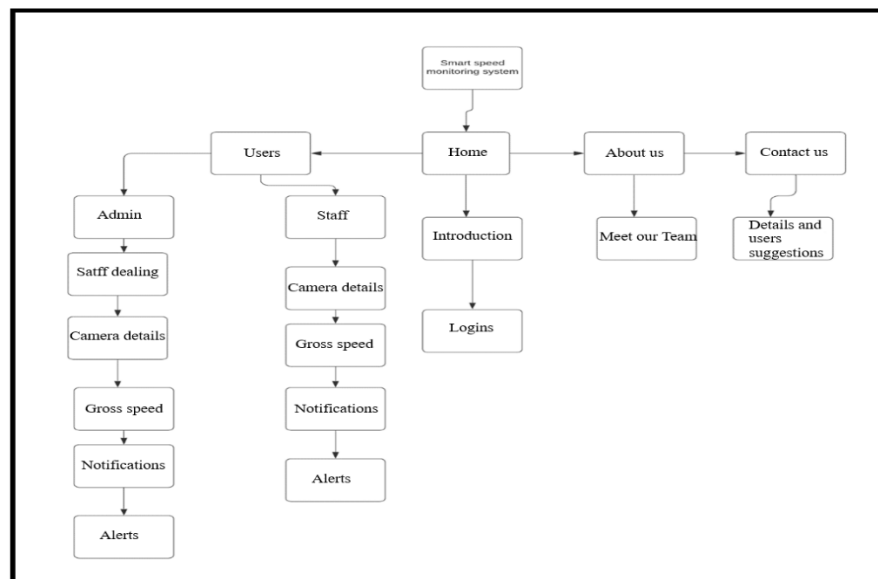**Figure 6.** IoT-enabled speed monitoring system home page.
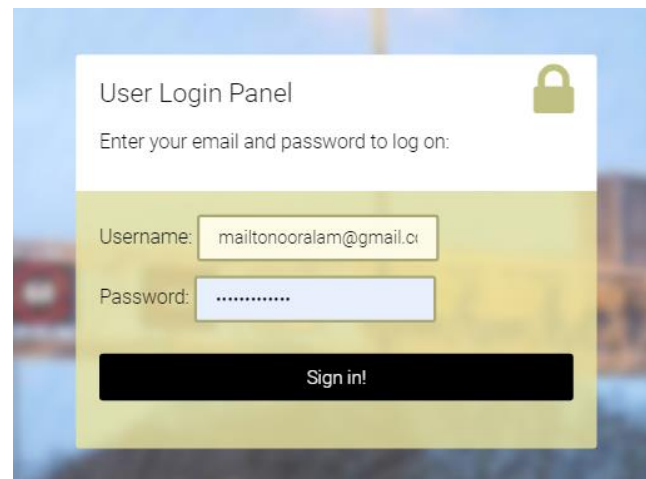

Figure 7. Site map for the SSMS website.

The SSMS website provides all the necessary information and details regarding the speed monitoring system. It comprises the following main pages:
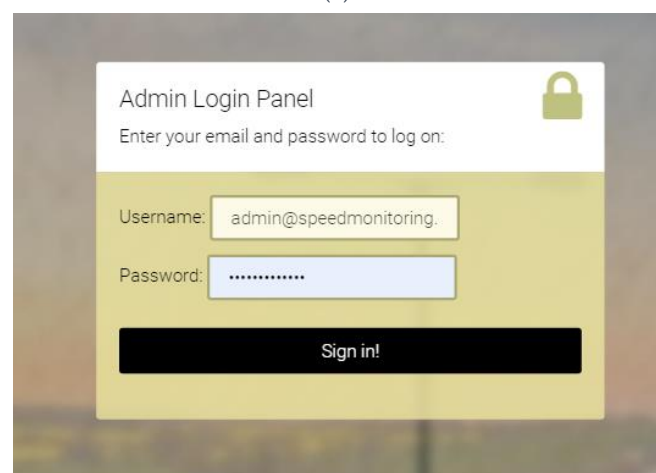
(1)   Home Page

(2) Notifications
(3) Admin Panel
(4) Alerts
(5) User panel
(6) Gross speed
(7) About us
(8) Contact us

The SSMS website consists of six modules; each module is interactive and shares information with other modules. The two user sides (admin and staff) have their own attributes, consisting of authorized functionalities. Screenshots of user and admin login pages are presented in Figure 8. The admin side deals with the following functionalities.

(1) Create new staff; update, delete and assign user passwords and IDs.
(2) Add cameras, update, and delete them.
(3) Process queries from the Contact Us section.
(4) Fetch data sent from server to website
(5) Notifications
(6) Alerts on the website and the application.

(a)

(b)

Figure 8. Screenshots of (**a**) the User Login page and (**b**) the Admin Login page of the SSMS website.

The user side deals with the functionalities below:

(1) See staff details.
(2) Add camera, update, and delete them.

(3)   Fetch data sent from server to website
(4)   Notifications
(5)   Alerts on website and the application.

3.10. Backend Website Development

The website's back-end oversees saving and arranging records and ensures that everything on the client-side runs properly. It interacts with the front-end, transmitting and receiving data, which are viewed as web pages. It is divided into three sections: (1) apps for web servers, (2) the database, and (3) the rationale of the application.

3.11. Android Application Development

An Android application was developed with the help of the Flutter mobile application framework and a real-time Firebase database to generate real-time alerts. The mobile application also consists of a map, targeted locations of the cameras, and details about the traffic wardens. Traffic wardens receive notifications via their mobile phones if a vehicle violates traffic laws.

3.12. Application Design and Implementation

The application for the SSMS was designed using Flutter, which is the fastest medium to build native apps for both IOS and Android operating systems. It is a cross-platform application from a single codebase, although it is in a very early stage of development, but there is a higher demand for Flutter developers. We have used some of the key libraries from Flutter to develop our application.

The application stores information on the patrolling authority, which is saved in the profile section. It also contains information on the vehicles that travel on the motorway and generates alerts for those that exceed the speed limit.

The application is designed to receive alerts and hold car information for authorities. The website and application hold the same data store, and through the alert section, the application receives alerts. Implementation of the application is based on the following information, as illustrated in Figures 9–11:

(1)   Vehicle registration number
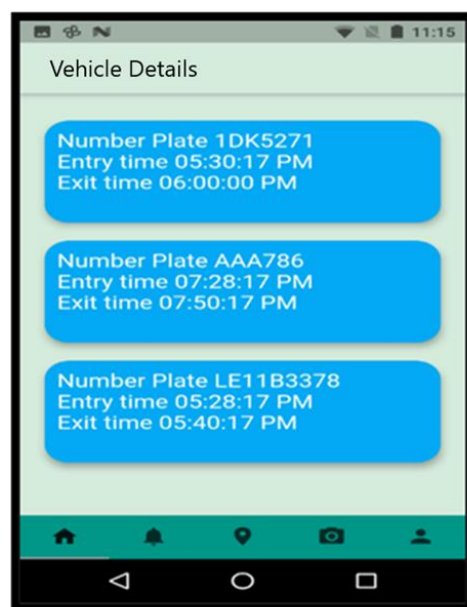(2)   Current location of the vehicle, and the camera location
(3)   Notification

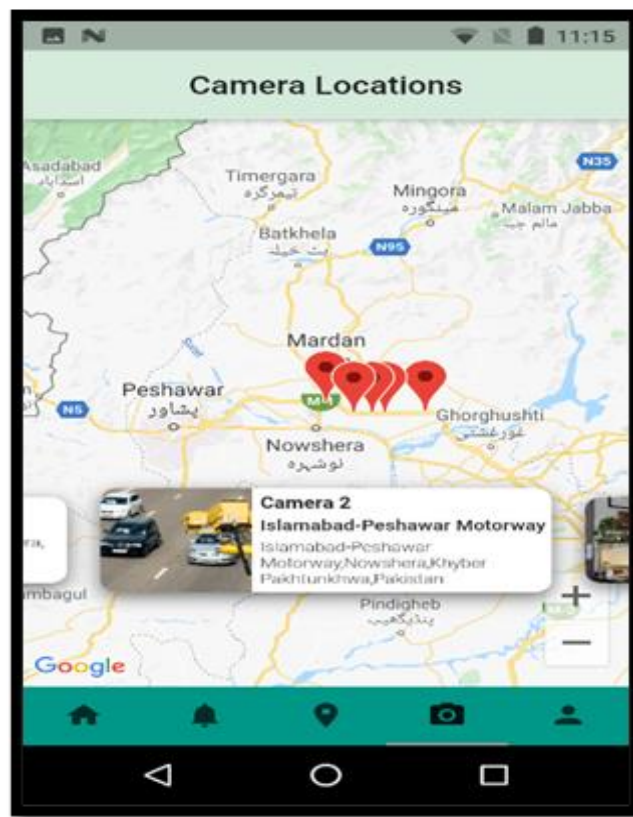**Figure 9.** Vehicle information fetched by the App.

Figure 10. Vehicle location detected by the App.
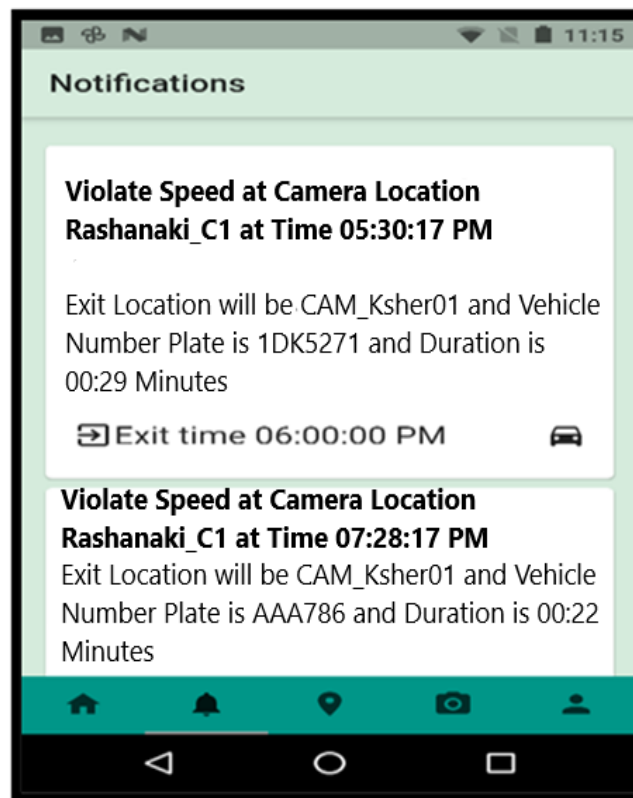

Figure 11. Vehicle speed violation notification shown by the app.

Figure 12 shows the flow diagram of the proposed IoT-enabled speed monitoring framework.
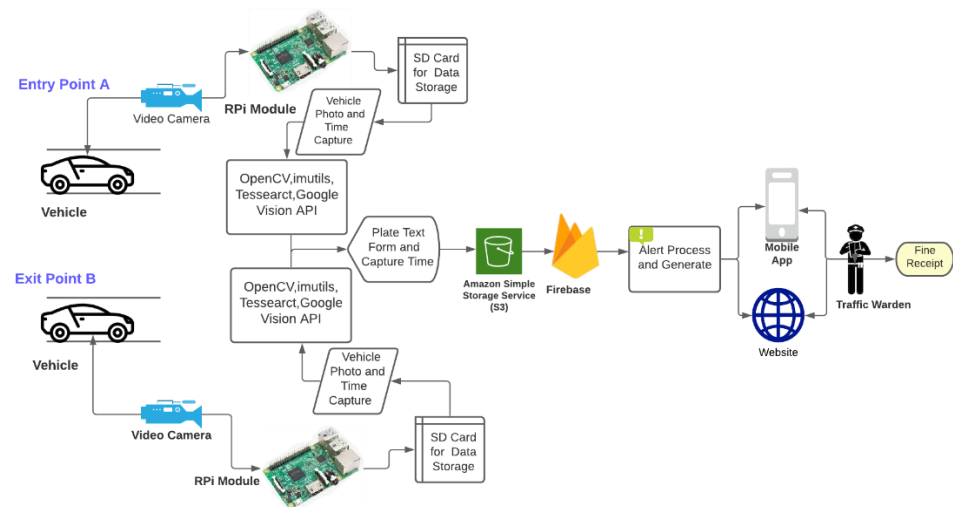

Figure 12. Flowchart of IoT-enabled Speed Monitoring System.

4. Testing

Testing is an important phase of the software development life cycle (SDLC) where value is added to the software system by enhancing customer confidence through promising test results. In order to ensure optimal resource utilization for development of a software system, an effective and efficient testing strategy is required that enables the development team to complete the testing phase in a timely manner within the allocated budget. The process comprises the following modules:

(1)   Hardware Testing
(2)   API Testing
(3)   Website Testing
(4)   Application Testing

Separate testing strategies are finalized for the modules, which are explained as follows.

4.1. Hardware Module

The main characteristics of the hardware module are safety, security, maintenance, and usability, but correctness, robustness, and interoperability are more crucial. Additionally, enough test cases are developed to test the presence of other quality attributes.

Checking and testing IoT devices is a challenging task, because each part, such as internal communications in the network, security check-ups, data mishaps, problems, or limitations in memory, processing power, and bandwidth, should be checked in detail. Gray-box testing is used to test cases and lets us know about the operating system. Effective test cases for IoT devices are given in Table 1.

Table 1. Effective test cases for IoT devices.

Testing Category	Sample Conditions
Validation of IoT Components	(1) Device hardware (2) Embedded software (3) Cloud infrastructure (4) Network connectivity (5) Cameras and sensors (6) Data formats (7) Robustness (8) Safety
Security, Performance and Validation of Data	(1) 5 Data packets (both sent & received) (2) Multiple request handling and interrupts (3) Device performance, data transmission frequency, and data values (4) Data values and data encryption/decryption
IoT functionality Validation	(1) Basic devices and connections with other IoT devices (2) Error handling, troubleshooting and calculations (3) Access control, data storage, and role management
Cloud Validation	(1) Cloud interface (2) IoT device-to-cloud connections and protocols (3) Device-to-cloud latency
Analytics and Communication Validations	(1) Broadcasting and protocols (2) Device-to-device (Pi-to-Pi) M2M functions (3) Interoperability and sensor data analytics

4.2. API Testing

API deployment and integration are among the crucial parts of a software system. Therefore, efficient testing is performed using specialized API testing tools, namely the Advanced REST Client (ARC), which is the de-facto standard for testing APIs.

Amazon Web Services was chosen as the deployment platform of the REST API for collecting and processing data from the sensors. Moreover, the most durable configuration of AWS is implemented for the deployment of the API. Standard storage S3 buckets and standard computer resource EC2 are created, as shown in Figure 13.
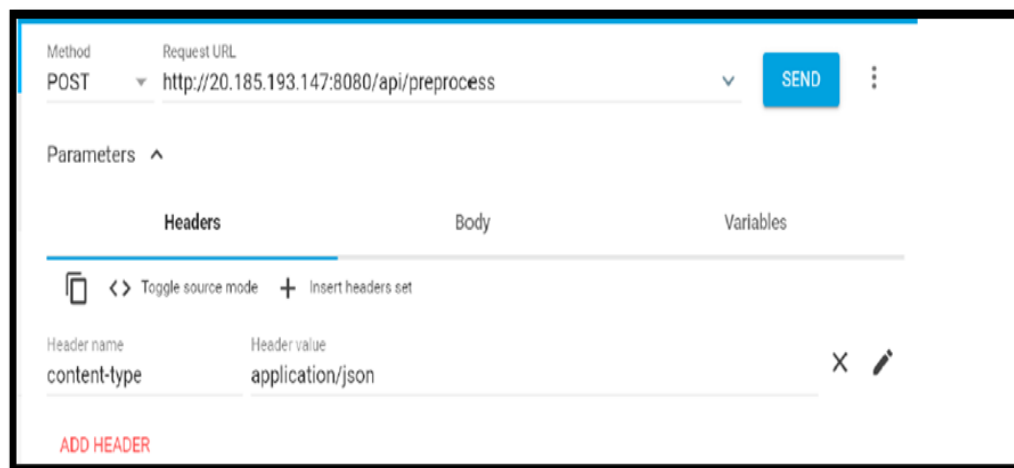

Figure 13. API testing for post-request response.

4.3. Website Module Testing

The website provides admin control, a user panel, receives data from the hardware module, and shares the same database with the Android application. Therefore, the website should be designed in such a way that it ensures safety, security, usability, and interoperability. Although the system was designed to take expert opinions as well as follow industry best practices, user testing was also performed to test and enhance the usability of the software system and obtain user feedback. Moreover, the website's interoperability was tested with several API testing tools, and the responses of the API were monitored.

The website's main testing strategy was Usage-Based Statistical Testing (UBST). Expert opinion was considered while assigning usage probabilities to the different functionalities. Usage probabilities are given in Table 2.

Table 2. Website feature usage probabilities.

Functionality	Usage Probability
Admin/User Login	20%
API Requests	20%
Camera Details	15%
Gross Speed	20%
Admin/User Profile	15%
Notifications/Log Outs	10%

4.4. Android Module Testing

The Test-Driven Development (TDD) paradigm is used to develop Android applications, because the Flutter framework is used for application development, which ships with a large number of tools for unit integration and system testing. Like the website module, usability, safety, security, and interoperability are the main attributes of the Android application. To enhance the usability of the application, Android material design was implemented in the front-end and validated with user testing results.

This module was also tested using UBST, where usage probabilities were assigned based on mutual consultation and expert opinion. Table 3 shows the usage probabilities of the application's features.

Table 3. The application's feature usage probabilities.

Functionality	Usage Probability
User Signup/Login	15%
User Panel	5%
In-app Notifications	10%
API Requests	15%
Camera Information	20%
Vehicle Information	20%
Location Information	15%

Effective testing was performed to confirm the agreed-upon requirements in the Android application. Moreover, extensive black- and white-box testing of the crucial functionalities was performed.

Comparison between Existing System (Speed Gun) and the Proposed System (IoT-Based Intelligent System)

(1) Because a speed gun can only identify a vehicle's speed when it is in front of a camera, a large number of such speed guns are required to detect the speed at various places. The suggested system is used to continuously monitor the vehicle's speed since it is based on the idea of monitoring the average speed between two places. The device can identify the speed even if the vehicle is not always visible to the camera.

(2) The current system is unable to monitor the vehicle's speed on several road lines at the same time, but the suggested system is capable of doing so.

(3) When comparing costs, the provided system is ten times less expensive than the current system. The suggested method requires a basic low-cost camera and a Raspberry Pi, whereas the speed gun is expensive and inaccurate.

(4) The existing system is a very manual and archaic system, whereas the proposed system is an automatic and up-to-date system, in which all processes are completed automatically with minimal human intervention.

(5) The proposed model addresses two critical issues in Pakistan's disorganized traffic environment: (1) need of continuous power supply and (2) irregular traffic trajectories that make speed monitoring more challenging.

5. Conclusions and Future Work

In recent years, we have witnessed great technological innovations and inventions in society, which have impacted lives substantially. Road safety and traffic violation surveillance systems were not left out. In this paper, we propose an IoT-based approach to automatically monitor speeding vehicles on highways without having to deploy traffic personnel. The traditional speed checking systems in the transportation industry fail to report the majority of speeding violations due to their respective flaws. For instance, speed guns on roads work in isolation, and cannot measure the speeds of all vehicles at all spatial points and can only accurately detect speeds of vehicles in front of the camera. Thus, a huge number of speed guns would be required to solve that problem, which would make the system far more costly. The proposed system, with the help of the IoT, measures the average speed between two detection points with surveillance cameras. The measured data are sent to the cloud for further calculations to enforce speed limits. With the help of entrance and exit points, information about any uncertainty in a given region is noticed. For example, a car passing an entrance point that does not reach the end point can be highlighted.

The system consists of a web network and a mobile phone application sharing real-time data, including details of the vehicles passing by, such as entrance time, photos, and registration numbers on license plates. Two different approaches (streaming a recorded MP4 video and installing Raspberry Pi with an IP camera) tested the efficiency of the proposed system. The advantages of such a system include installing comparatively few speed guns, monitoring vehicles even if they are not within the camera's line of sight, and minimum human intervention.

5.1. Future Work

Our future aim is to focus on the entrepreneurial process and add additional features to the speed monitoring system and to introduce it with a stronger business plan in future. Following are some of the key aspects that can be added to this project:

5.1.1. Vehicle Collision and Auto Accident Detection and the SOS Service

Accident detection systems can be used in connection with vehicle speed monitoring systems to identify accidents on highways. This may be enhanced by placing several sensors on the road that can find the car's location and timing. The system may be modified to send vehicle data to rescue centers in order to notify them to the need for first assistance.

5.1.2. Plate Detection of a Car Moving with More Specific Speed Limit

We used relatively low-cost devices and cameras in the proposed system, but the existing system can be improved further by using advanced cameras and IoT devices, as well as advanced software technologies and algorithms for image processing and deep learning, to improve the system and make it capable of detecting number plates and identifying passing vehicles travelling at speeds exceeding speed limit.

5.1.3. Vehicle Missing between the Endpoints

Other features for the vehicle missing between two points can be added, such as if the vehicle has just passed through one of the entrance points but has not reached or crossed the next points, some alerts to the traffic police will be generated to investigate why the vehicle is missing so that they can respond quickly if there is an accident, injury, or other type of mishap on the road.

5.1.4. Adding Some Other Features and the Use of Some Advance Technologies to Deal with the System Effectively and Handle Big Data of the System

Some advanced development platforms and services can be added to the existing system to enable it to deal with big data, such as large amounts of vehicle information. In addition to vehicle plate detection, we can detect the color, model, and direction of the vehicle by working further on the existing system, and we can implement the entire system in Amazon cloud storage to handle the system's functionality efficiently and effectively.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lombardi, M.; Pascale, F.; Santaniello, D. Internet of Things: A General Overview between Architectures, Protocols and Applications. *Information* **2021**, *12*, 87. [CrossRef]
2. Jan, S.U.; Young, D.L.; Soo, K. A distributed sensor-fault detection and diagnosis framework using machine learning. *Inf. Sci.* **2021**, *547*, 777–796. [CrossRef]
3. World Health Organization. *Global Status Report on Road Safety 2015*; World Health Organization: Geneva, Switzerland, 2015.
4. Abbass, A. Estimating vehicle speed using image processing. *AL-Mansour J.* **2010**, *14*, 127–141.
5. Sina, I.; Ari Wibisono, A.; Nurhadiyatna, B.H.; Wisnu, J.; Petrus, M. Vehicle counting and speed measurement using headlight detection. In Proceedings of the 2013 International Conference on Advanced Computer Science and Information Systems (ICACSIS), Sanur Bali, Indonesia, 28–29 September 2013; pp. 149–154.
6. Song, A.; Mengjie, Z. Genetic programming for detecting target motions. *Connect. Sci.* **2012**, *24*, 117–141. [CrossRef]
7. Lin, H.-Y.; Li, K.-J.; Chang, C.-H. Vehicle speed detection from a single motion blurred image. *Image Vis. Comput.* **2008**, *26*, 1327–1337. [CrossRef]
8. Rad, A.G.; Abbas, D.; Mohamed, R.K. Vehicle speed detection in video image sequences using CVS method. *Int. J. Phys. Sci.* **2010**, *5*, 2555–2563.
9. Mirza, F.H.; Qudsia, H.; Nadia, J. An autopsy-based study of death due to road traffic accidents in metropolis of Karachi. *JPMA* **2013**, *63*, 156–160.
10. Khan, M.A.; Khan, S.F.; BaOmar, T.; Rahman Mohammed Aqeel Ba Omar, A. Development & implementation of smart vehicle over speeding detector using IoT. *Adv. Sci. Technol. Eng. Syst.* **2019**, *4*, 170–175.
11. Rudin-Brown, C.M. Intelligent'in-vehicle intelligent transport systems: Limiting behavioural adaptation through adaptive design. *IET Intell. Transp. Syst.* **2010**, *4*, 252–261. [CrossRef]
12. Bassey, I.; Khutsoane, O.; Dladlu, N. Real-time Monitoring and Detection of Drink-driving and Vehicle Over-speeding. *Int. J. Image Graph. Signal Processing* **2017**, *9*, 1.
13. Xu, S.; Guo, P.; Xu, B.; Zhou, H. QoS evaluation of VANET routing protocols. *J. Netw.* **2013**, *8*, 132. [CrossRef]
14. Pascale, F.; Adinolfi, E.A.; Coppola, S.; Santonicola, E. Cybersecurity in Automotive: An Intrusion Detection System in Connected Vehicles. *Electronics* **2021**, *10*, 1765. [CrossRef]
15. Sergi, R.; Ganan, C.; Caubet, J.; Alins, J.; Mata, J.; Munoz, J.L. Analysis of video streaming performance in vehicular networks. In Proceedings of the First International Conference on Advanced Communications and Computation, INFOCOMP, Barcelona Spain, 23–29 October 2011; Volume 10, p. 2011.
16. Thakur, T.; Gogate, M. Traffic Controlling and Monitoring using Internet of Things. *Int. J. Latest Trends Eng. Technol. (IJLTET)* **2015**, *6*, 012017.

17. Perumalla, K.B.; Babu, M.S. An intelligent traffic and vehicle monitoring system using internet of things architecture. *Int. J. Sci. Res.* **2016**, *5*, 853–856.
18. Harshini Vijetha, H.; Dr Nataraj, K.R. IOT Based Intelligent Traffic Control System. *Int. J. Eng. Technol. Res. Manag.* **2020**, *4*, 59–63.
19. Jain, M.; Praveen, K.; Priya, S.; Chhavi, N.A.; Ankita, S. Detection of over speeding vehicles on highways. *Int. J. Comput. Sci. Mob. Comput.* **2015**, *4*, 613–619.
20. Du, S.; Mahmoud, I.; Mohamed, S.; Wael, B. Automatic license plate recognition (ALPR): A state-of-the-art review. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *23*, 311–325. [CrossRef]
21. Sawicki, D.S. *Traffic Radar Handbook: A Comprehensive Guide to Speed Measuring Systems*; Grove Enterprises: Brasstown, NC, USA, 1993.
22. Xiao, L.; Zhengxia, W. Internet of things: A new application for intelligent traffic monitoring system. *J. Netw.* **2011**, *6*, 887. [CrossRef]
23. Saifuzzaman, M.; Syeda Farjana, S.; Nazmun Nessa, M.; Fernaz Narin, N.; Mohammad Hanif, A. IoT Based Street Lighting Using Dual Axis Solar Tracker and Effective Traffic Management System Using Deep Learning: Bangladesh Context. In Proceedings of the 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, 1–3 July 2020; pp. 1–5.
24. Saifuzzaman, M.; Nazmun, N.M.; Fernaz, N.N. IoT based street lighting and traffic management system. In Proceedings of the 2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC), Dhaka, Bangladesh, 21–23 December 2017; pp. 121–124.