EAI Endorsed Transactions
on Scalable Information Systems Research Article EAI.EU

Internet-of-Video Things Based Real-Time Traffic Flow Characterization

Ali Khan[1], Khurram S. Khattak[1,*], Zawar H. Khan[2], T. A. Gulliver[2], Waheed Imran[1] and Nasru Minallah[1]

[1]National Center in Big Data and Cloud Computing, UET Peshawar, Pakistan.
[2]Department of Electrical and Computer Engineering, University of Victoria, Canada.

Abstract

Real-world traffic flow parameters are fundamental for devising smart mobility solutions. Though numerous solutions (intrusive and non-intrusive sensors) have been proposed, however, these have serious limitations under heterogeneous and congested traffic conditions. To overcome these limitations, a low-cost real-time Internet-of-Video-Things solution has been proposed. The sensor node (fabricated using Raspberry Pi 3B, Pi cameral and power bank) has the capability to stream 2 Mbps MJPEG video of 640x480 resolution and 20 frames per second (fps). The Camlytics traffic analysis software installed on a Dell desktop is employed for traffic flow characterization. The proposed solution was field-tested with vehicle detection rate of 85.3%. The novelty of the proposed system is that in addition to vehicle count, it has the capability to measure speed, density, time headway, time-space diagram and trajectories. Obtained results can be employed for road network planning, designing and management.

Keywords: Internet of Video Things (IoVT), Raspberry Pi (RPi), Video Streaming, Intelligent Transportation Systems (ITS), Camlytics.

Received on 03 April 2021, accepted on 20 September 2021, published on 21 October 2021

doi: 10.4108/eai.21-10-2021.171596

1. Introduction

According to the World Health Organization (WHO), the proportion of the world population in urban areas will reach 60% by 2030 [1]. While this urbanization can increase economic growth, it will adversely impact the environment, urban resources, and transportation network efficiency without proper planning and management. An inefficient transportation network results in increased congestion, pollution, and accident rates, as well as reduced productivity. Vehicle emissions such as carbon dioxide (CO_2), carbon monoxide (CO), sulfur dioxide (SO_2), nitric oxide (NO_X), and dust particulates are a major cause of cardiovascular and respiratory diseases [2], [3]. Improving the efficiency of transportation networks is essential in achieving liveable smart cities with a high quality of life. Intelligent Transportation Systems (ITSs) are required to achieve this goal.

The development of effective ITS solutions needs detailed traffic flow characterization. This includes parameters such as vehicle count, speed, headway, vehicle trajectory, and density [4]. Real-time traffic flow data is crucial for planning, designing, and efficient management of transportation networks. Further, it can be employed to validate and calibrate mathematical traffic flow models and traffic simulation tools such as Vissim, Paramics, and Corsim [5]-[10]. Typically, traffic characterization is done manually. This is not only labor-intensive and expensive but also inefficient as only limited data such as the number and

[*]Corresponding author. Email: khurram.s.khattak@gmail.com

type of vehicles can be recorded. Advances in technology have led to solutions that overcome the limitations of manual counting. These solutions employ either intrusive or non-intrusive sensors [11].

Intrusive sensors such as pneumatic tubes, inductive loops, piezoelectric and magnetic sensors are cumbersome and expensive to install and maintain. Their installation disrupts traffic and multiple sensors are needed on multilane roads. Intrusive sensors can only count and classify vehicles in homogenous traffic conditions when there is no congestion [12]. The performance of pneumatic tubes and inductive loops is hindered by high temperatures and they are prone to failure over time [13]. Further, magnetic sensors have small detection zones, so close proximity to vehicles is required [13]. Intrusive sensors cannot detect pedestrians and thus their impact on traffic.

Non-intrusive sensors have recently been developed. They provide greater functionality than intrusive sensors and are more robust. They are typically installed on or above roadways and provide accelerometer, ultrasonic, acoustic, infrared, radar, RFID, or wireless communication-based solutions. These sensors can measure parameters such as vehicle count, type, and speed, but are not effective in congestion and heterogeneous traffic. Further, performance can be affected by environmental and weather conditions [11]. Accelerometers are sensitive to vibrations and cannot detect stationary vehicles, while acoustic sensors are unable to detect bicycles and animal and human-driven carts. In addition, infrared and ultrasonic sensors are sensitive to light and temperature fluctuations [11], [13].

Accurate and detailed traffic characterization is important for planning, designing, and efficient management of transportation networks. Existing sensor-based solutions have limitations restricting their performance and they are less effective in congested and heterogeneous traffic conditions. Most of these solutions cannot detect or classify pedestrians, bicycles, bikes, three-wheelers, or human and animal-driven carts. However, advances in computer vision have made it possible to analyze traffic ranging from driver behavior (e.g physiological and psychological) to traffic behavior. As a consequence, several image processing based edge computing solutions have been proposed in the literature. However, these solutions are constrained by the computational limitations of single board computers (SBC) such as the Raspberry Pi (RPi). Thus, they are only able to provide vehicle count [14]-[16], vehicle count, and average speed [17], or vehicle count and type [18]-[21].

To overcome the drawbacks of existing solutions, an Internet-of-Video-Things (IoVT) based video streaming system is presented. This solution can provide vehicle count and speed as well as time headway, vehicle trajectories, heat maps and road capacity. Unlike sensor-based solutions, this system can characterize traffic flow under congested and heterogeneous traffic conditions. Furthermore, pedestrians, motorcycles, three-wheelers, and human and animal-driven carts can also be characterized. The solution presented here is low-cost and easy to install, configure, and maintain. Instead of using traditional Closed-Circuit Television (CCTV) [22], the system employs a real-time video transmission system based on an inexpensive Raspberry Pi (RPi) 3B single-board computer with a Pi camera. It can stream MJPEG video with 640×480 resolution at 20 frames per second (fps) over the internet. An i5 quad-core HP desktop computer with 4 GB of RAM is used to analyze the video using the Camlytics traffic monitoring software.

The rest of this paper is organized as follows. Section II presents the related work. The system architecture is given in Section III along with the video streaming methodology. Performance results are given in Section IV, and finally, some conclusions and suggestions for future work are given in Section V.

2. Related Work

Many wireless video streaming systems have been developed, but most do not provide real-time video streaming. IoVT based video streaming solutions suitable for real-time traffic flow characterization are fewer still [18], [23]. Raktrakulthum *et al.* [18] used machine learning for vehicle classification under low, moderate, and high traffic densities. The K-Nearest Neighbor (KNN) and Support Vector Machine (SVM) methods were used with a radial basis function to classify vehicles into two types, cars, and motorcycles. A low-cost vision system consisting of two synchronized RPis was used to generate a 3D point cloud as the classification input. A detection rate of 95.8% was achieved which is better than previous solutions. SVM provided better motorcycle classification in all traffic conditions. Balakrishna *et al.* [23] proposed a traffic monitoring system using Simulink. It transmits video to a ThingSpeak cloud platform for analysis and visualization, and data analytics are used to characterize real-time traffic patterns.

A video monitoring system was proposed by Rohadi *et al.* [24] which is based on an RPi and Pi camera. FFmpeg libraries and Google Drive were employed for video streaming and storage, respectively. The RPi CPU utilization was analyzed by varying parameters such as the video resolution, fps, and number of viewers. Tain *et al.* [25] proposed Cloud Live Video Streaming (CLVS) to stream video using an RPi to an Amazon S3 bucket. The video is segmented and encoded at the sensor node. CLVS is serverless by design which eliminates the need for an intermediate and continuously running streaming server. Filteau *et al.* [26] developed a cost-effective video streaming system using an RPi as a server with code written in Java. This system can stream video to multiple handheld devices while maintaining the Quality of Service (QoS). It was concluded that the processing power of end devices has a significant effect on the QoS.

Dursun *et al.* [27] proposed video streaming over secure data links for Unmanned Aerial Vehicles (UAVs). This was accomplished using MJPEG, TCP/IP, and TLS/SSL with the X.509 standard for user authentication and authorization.

This low-cost system is based on an RPi 3, Pi camera, and the HTTP protocol for video streaming. The reliability was evaluated experimentally. It was shown that video streaming efficiency is improved by reducing the number of frames per second (fps). Jennehag *et al.* [28] presented a low delay video streaming solution for resource-constrained IoT devices using the IoT platform SensibleThings. Six Mbps video with 1280×720 resolution and 25 fps was streamed over the internet in a Point-to-Point (P2P) manner using H.264 for video compression. The resulting delay was less than 200 ms. An RPi 2 and Pi camera were employed for video encoding and transmission, and also for rendering and display. The major bottleneck in this system is video encoding which accounts for 90% of the delay. Conversely, the distributed communication network had a minimal effect on delay.

Nguyen *et al.* [22] proposed an RPi based system for surveillance. Traditional CCTV cameras were not used as they have poor picture quality. Motion detection is employed so that video is stored only when motion occurs. This reduced the storage requirements by over 300%. Saha *et al.* [29] presented a system for remote monitoring and surveillance using an RPi and Pi camera. Real-time synchronized video and half-duplex audio were transmitted using both wired and wireless channels (LAN, Wi-Fi, and GSM). Video decoding, processing, and transcoding were done using an RPi while an SA828 module was employed for audio transmission. For security, variable IP addresses and password protection were used.

An augmented platform, Mini-CV, was proposed by Wallace *et al.* [30] as a low-cost solution for computer vision. A comparison was made with ROBOTIS-Ops which is a popular but 20 times more expensive solution based on a miniature humanoid. Instead of local computation, a network approach was used for video streaming so a more powerful computer running more complex algorithms could be employed. The system uses the FFmpeg libraries installed on an RPi Zero W with a Pi camera.

Komasilovs *et al.* [31] proposed a monitoring system for precision agriculture using an RPi for video streaming to YouTube for remote expert evaluation and analysis. This system was analyzed using different cameras and video resolutions to determine the best video streaming solution considering QoS, fps, latency, and power consumption. Gonzalez *et al.* [32] presented a low-cost agricultural monitoring system using an RPi to establish ad hoc communication links. The sensor node is comprised of an environmental sensor (BME280 providing temperature, humidity, and pressure), webcam, wireless card, and current sensor. A web interface developed using Node.js is used to access the sensor data and video.

Although several different solutions (both intrusive and non-intrusive sensors) have been proposed in the literature for traffic flow characterization, they are not effective in congested and heterogeneous traffic conditions. Hence, computer vision-based solutions are emerging that can provide detailed traffic flow parameters under all traffic conditions. In this work, a novel solution for traffic flow characterization is proposed. This system streams real-time traffic video to the commercial traffic analysis software Camlytics on a desktop computer. Camlytics is used to provide detailed traffic flow parameters from the video. The advantages of the proposed solution over others in the literature are as follows.

- Traffic can be characterized under all traffic conditions such as uncongested, congested, homogenous, and heterogeneous.
- Traffic can be characterized in varying weather conditions.
- Detailed traffic flow parameters can be obtained such as count, speed, density, time headway, time-space diagrams, and vehicle trajectories.

The traffic flow parameters obtained can be employed for validation and calibration of mathematical traffic flow models and traffic simulation software for better planning, design, and management of road networks [5]-[10].

3. System Architecture

Computer vision-based IoVT systems have emerged as a feasible solution for traffic flow monitoring because of their low cost, easy maintenance, and non-intrusive installation [12]. Moreover, these systems can work continuously or during user-defined intervals. Cloud-based platforms such as Microsoft Azure or Amazon Web Services (AWS) can be used to process the traffic video, but a P2P approach is employed here because of its advantages over cloud-based systems. P2P communication is directly between the source and sink without the proxy or intermediate nodes required with cloud-based systems. This results in low delay video transmission and avoids central points of failure [28]. The system is based on a Raspberry Pi (RPi) 3B, Pi Camera, Wi-Fi module, and power bank. It can stream video in real-time from the roadside to the desktop computer which has the Camlytics software installed. The system model is shown in figure 1. This system has been developed to provide vehicle count, speed, time headway, and road capacity in real-time. It has four major components: (1) sensor node, (2) video streaming, (3) power bank, and (4) Camlytics software (on a desktop computer). These components are described below.

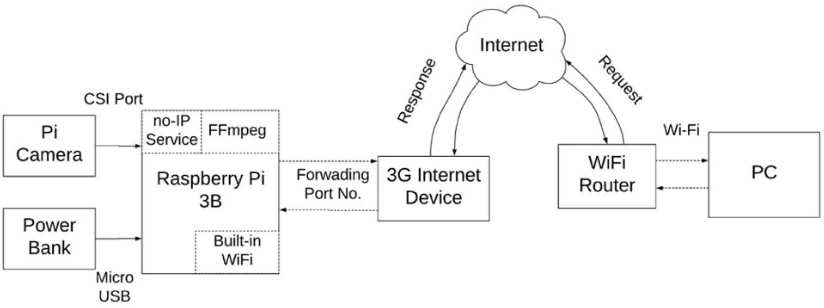


Figure 1. Schematic overview of proposed solution's system architecture implementation.

3.1. Sensor node

The sensor node is based on an RPi 3B with Pi camera for video capture, transcoding, compression, and transmission as can be seen in figure 2. An RPi is a general-purpose, low-cost Single-Board Computer (SBC) [33]. The Raspbian OS is used which is an optimized operating system for RPi. Programming is done using the C++ language. The RPi parameters are as follows [33].

- Quad-core 1.2 GHz Broadcom BCM2837 SoC processor
- 1 GB RAM
- VideoCore IV 400 MHz GPU
- MicroSD card slot, 3.5 mm analog audio-video jack, 4 x USB ports, HDMI, GPIO header, micro-USB power port, Display Serial Interface (DSI) and Camera Serial Interface (CSI)
- 40 GPIO (General Purpose Input/Output) pins,
- Built-in Wi-Fi and Bluetooth

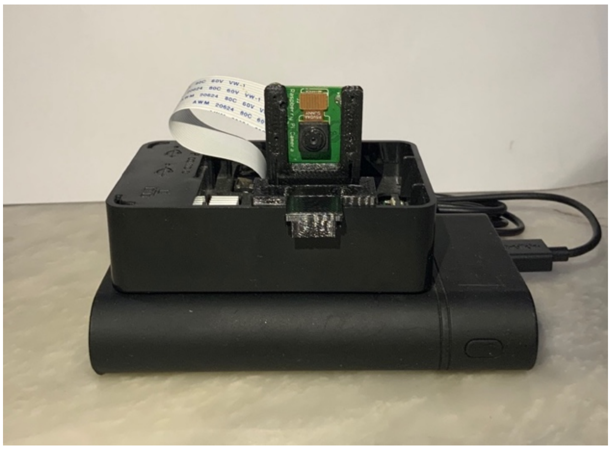


Figure 2. Sensor node of the proposed IoVT solution.

The Pi camera module v1.3 weighs just over 3 grams and has a high quality 5 MP OmniVision-5647 fixed focus lens. In addition to 5 MP still images, it also supports 1080p at 30 fps, 720p at 60 fps and 480p at 60/90 fps video. The camera is connected to the CSI port which is specifically designed for Pi cameras to provide high data rate transfers.

3.2. Video streaming

Computer vision algorithms have been shown to be very suitable for traffic flow analysis [12]. However, running these algorithms at edge locations is impractical using resource constraint SBCs such as RPi. Thus, video streaming is a better solution as then a powerful desktop computer can be used for traffic flow characterization. Video streaming applications range from streaming stored video to real-time streaming and broadcast and multicast streaming [22]-[24], [26], [27], [29]-[32]. The system presented here employs networked real-time streaming as can be seen in figure 3. With this approach, there are many opportunities along the network pipeline to reduce latency [29], [30]. These can range from the camera used to video recording quality (resolution and fps), camera port interface (CSI or USB) with the RPi, video encoding and streaming software, and transmission protocols [24], [27], [29]-[31]. The main advantages of a Pi camera over USB cameras are faster transfer rates and the graphics processing capability of the Broadcom CPU. Most systems proposed in the literature for video streaming employ a Pi camera [22], [27]-[29]. In [31], a comparison was done between Pi and Logitech cameras (hardware encoding versus software encoding), with respect to CPU utilization, latency, and power consumption. The Logitech camera provides significantly better video quality but at the cost of almost a 90% increase in power consumption [31].

The FFmpeg library [34], was written using the C language and is a free, fast, and powerful package for handling multimedia files. In the proposed system, these libraries are installed on the sensor node. They are employed to capture, encode, and stream video to the desktop computer. FFmpeg comes with the FFserver package which accesses the RPi through the internet. To increase the transfer speed, reduce delay, and overcome slow internet speeds, the video quality can be reduced by transcoding [29]. Field experimental results show that the proposed system can provide high-quality video (640×480 resolution at 24 fps). Compression can further improve the video transmission rate and reduce delay. Although the RPi has a hardware H.264 decoder which is more power-efficient than software decoders [28], the proposed system employs MJPEG. The advantage of MJPEG is individual frame-based compression which provides higher quality and better robustness to dropped frames. As the proposed system streams real-time video for traffic analysis, the combined video quality and robustness with MJPEG is preferred.

The Transmission Control Protocol (TCP) is slower than the User Datagram Protocol (UDP), but it provides better congestion control, pre-fetching and buffering capabilities [12]. TCP is a transport-layer protocol and HTTP is an application-layer protocol that runs over TCP. HTTP allows for efficient transfer of data from sensor nodes to the computer, thus ensuring a low latency [30]. As reported in the literature and validated by our field experiments, the end-to-end latency for 640×480 resolution video is under 50 ms [30].

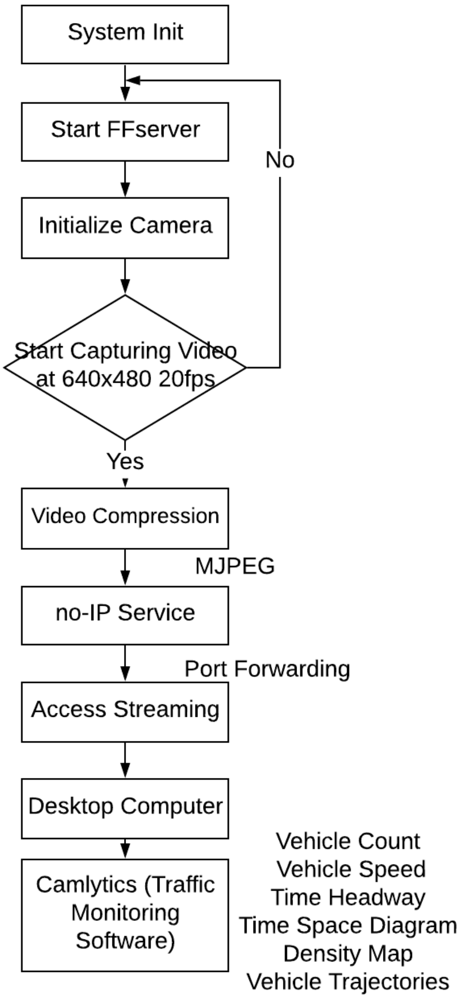


Figure 3. Software flow diagram of the proposed IoVT solution.

Because the system employs data streaming over the internet, parameters such as video resolution, fps, and transmission format must be carefully chosen [21], [24], [27]. Sending video data over the internet is the main contributor to data costs. The higher the resolution and fps, the higher the cost. figure 4 gives the hourly data rates obtained experimentally with the system for different resolutions and fps. This shows that increasing the fps from 10 to 24 for video with 640×480 resolution approximately doubles the data rate. An increase in video resolution and fps will also increase the demands on hardware resources and the end-to-end latency. For example, an increase of 6.8% and 0.3% in CPU and RAM utilization, respectively, was reported in [27] when the video resolution was increased from 640×480 to 1920×1080. Further, the RPi power consumption increased from 335 mAh to 425 mAh.

A permanent and unique IP address is necessary to access sensor nodes over the internet. NO-IP is a dynamic DNS service that is employed to access sensor nodes remotely. The remote host accesses video from a sensor node using the DNS name assigned to the node.

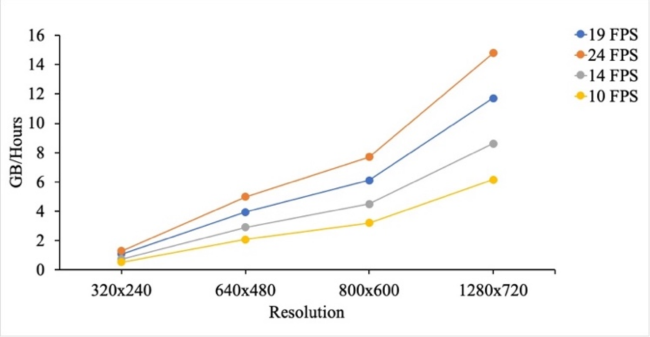


Figure 4. The effect of video resolution and frames per second (fps) on the MJPEG data transmitted per hour.

3.3. Current consumption

The system is powered by a Xiaomi Mi 2 10,000 mAh power bank with dual USB ports providing current at different voltages. The RPi when fully utilized requires 2.5 A/5 V. However, it provides the option to shut down unused modules for power conservation, for example, the Wi-Fi, HDMI, and LEDs. The current consumption for the system components were measured using a Keweisi KWS-V20 USB tester and the results are given in Table 1. This shows that when the LEDs, HDMI, and Wi-Fi are off, the RPi draws 200 mA. The current required for the system when streaming video is

$$I_{Streaming} = I_{Idle\_State} + I_{Pi\ Camera} + I_{Streaming\_State} + I_{Wi\text{-}Fi}$$

From the table, the system requires 900 mA for video streaming. With a 10,000 mAh battery, the streaming time is

$$\mathit{Streaming}\ \mathrm{Time} = 10{,}000\ mAh/900\ mAh = 11.11\ h$$

This estimate is in line with those in the literature [29], [31].

Table 1. Sensor node's current consumption breakdown

RPi Components	Current mA
Idle State (LEDs OFF, HDMI OFF, Wi-Fi OFF)	200 mA
Pi Camera	250 mA
Streaming State	420 mA
Wi-Fi	30 mA
Total Current	900 mA

3.4. Camlytics software

The Camlytics commercial software [35] is used to determine vehicle count, speed, time headway, heat maps, and vehicle trajectories. This software can be employed for multi-camera management, recording, and video analysis. It can detect objects, people, vehicles, movement in specific directions, camera blurring, and blackouts. Video captured from different sources such as prerecorded and real-time streamed videos recorded in formats such as .flv, .mkv, .mov, .wmv, .vob, and .avi can be analyzed. For real-time video analysis, access to video sources using the RTSP or HTTP protocol must be provided.

Camlytics currently supports both x86 and 64-bit Windows computers. Hardware requirements vary depending on the number of cameras and image resolution being streamed. A computer with 2 GB of RAM and 200 GB of hard disk space can analyze video streamed from 4 cameras with 320×240 resolution [35]. The system presented here is a Dell computer with an i5 quad-core processor and 4 GB of RAM running the Windows 10 operating system.

4. Results

For field testing, a single sensor node as shown in figure 2 was installed over a flyover bridge to analyze traffic on University Road in Peshawar, Pakistan. Video was streamed in real-time from the sensor node to the computer which was located at UET Peshawar. The duration of this video was 1442 s from 16:23:23 to 16:47:25 on Thursday, December 19, 2019. The data obtained shows that the traffic is heterogeneous.

The Camlytics software has a simple Graphical User Interface (GUI) which allows for the creation of event generation lines and zones through drag and drop functionality as shown in figure 5. These lines and zones can be used for video analysis from different viewing angles and video streams from multiple cameras in real-time. Two event generation lines denoted Enter (ingress) and Exit (egress) were drawn in figure 5 which shows a 50 m section of the road. When a vehicle crosses one of these lines, an event is generated and saved to .CSV file. The event consists of vehicle ID, line type (Enter or Exit), and the time at which the line was crossed, an example of which is shown in figure 6.

For traffic flow analysis, a vehicle must be detected at both the Enter and Exit lines. Vehicle data that does not satisfy this requirement is discarded and not included in the count, speed, time headway, and density calculations. In the 1442 s of real-time video, 632 vehicles were counted manually. However, the Camlytics software counted only 551 vehicles, so 81 vehicles were not detected at both the Enter or Exit lines. From these results, the detection rate of the Camlytics software is 85.3%. Note that there were no instances of a car being confused with another car.

The performance of a computer vision system depends on the algorithms employed and can be affected by environmental conditions such as the weather (e.g. rain) [36]. Improvements to computer vision algorithms using techniques such as Machine Learning (ML) will provide better results for ITS and other applications [36].

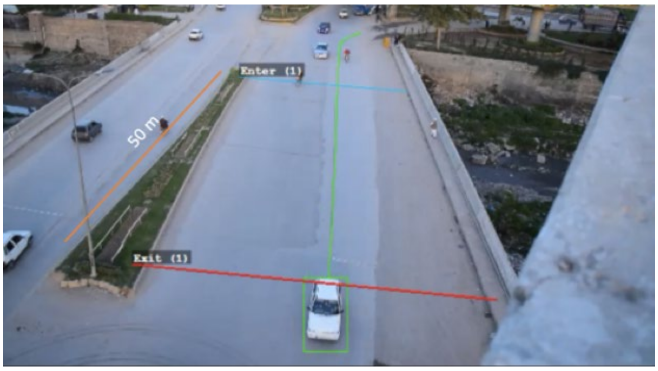
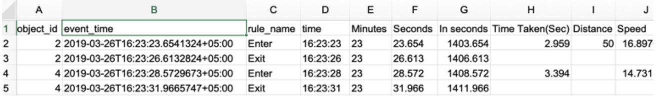


Figure 5. Camlytics setup for Traffic analysis on a 50 m road section.



	A	B	C	D	E	F	G	H	I	J
1	object_id	event_time	rule_name	time	Minutes	Seconds	In seconds	Time Taken(Sec)	Distance	Speed
2	2	2019-03-26T16:23:23.6541324+05:00	Enter	16:23:23	23	23.654	1403.654	2.959	50	16.897
3	2	2019-03-26T16:23:26.6132824+05:00	Exit	16:23:26	23	26.613	1406.613			
4	4	2019-03-26T16:23:28.5729673+05:00	Enter	16:23:28	23	28.572	1408.572	3.394		14.731
5	4	2019-03-26T16:23:31.9665747+05:00	Exit	16:23:31	23	31.966	1411.966			

Figure 6. Logged .CSV file of events generated on 50 m University Road section using Camlytics.

The time taken by a vehicle to traverse the 50 m road section is given by

$$Time\ Taken = Egress\ Time - Ingress\ Time$$

and

$$Speed = Distance / Time$$

The road capacity is given by

$$Road\ Capacity = Ingress\ Count - Egress\ Count$$

Figure 7 shows the speed of the vehicles for the 1442 s time period. The maximum speed observed was 82.229 km/hr at 16:42:25 and the minimum was 7.738 km/hr at 16:46:50. This shows that speed patterns can be obtained for use in traffic management or other applications.

Vehicle trajectories or time-space diagrams can be created to provide information about distances and the time taken to cover a distance, as well as identifying accidents on the road. With the Camlytics software, this was achieved by drawing lines over the 50 m road section at intervals of 10 m as shown in figure 8. The five lines are numbered 1,2,3,4 and 5, and they generate events that are saved in a .CSV file.

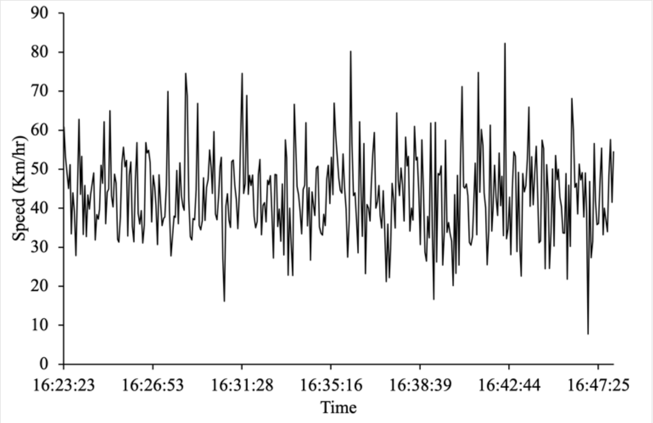


Figure 7. Vehicle speeds over a period of 1442s on 50 m road section

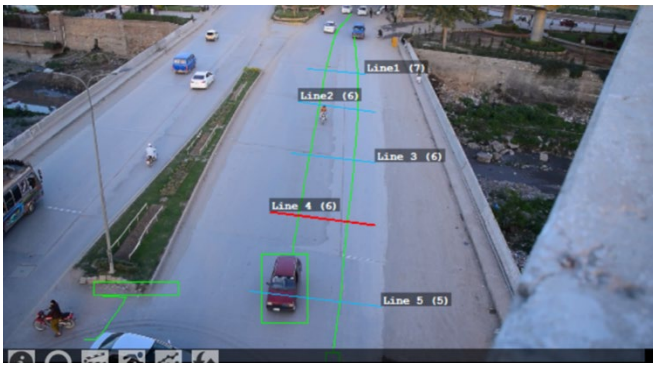


Figure 8. Vehicle trajectories on a 50 m road section obtained by drawing five lines of length 10 m.

In addition to vehicle count and speed, the video was used to extract time headways, time-space diagrams, density maps, and vehicle trajectories. Examples of the traffic data obtained are shown in Table 2. This gives the ingress time, egress time, elapsed time, distance, and speed. For example, vehicle 1 entered the road section at 23.654 s and exited at 26.613 s so the speed is 16.9 m/s, vehicle 2 entered at 28.572 s and exited at 31.966 s for a speed of 14.73 m/s, and vehicle 4 had the lowest speed of 12.52 m/s. The data obtained can be used to determine individual vehicle parameters such as the time and distance headway and macroscopic parameters such as density, speed, and flow. Traffic flow patterns can be determined such as when traffic peaks occur. This data can be used for other applications such as law enforcement by identifying vehicles exceeding the speed limit.

4.1. Time headway

Consider two consecutive vehicles in the same lane. The time difference between the front bumper of the lead vehicle and the front bumper of the following vehicle is the time headway $h_{t.}$ This can be determined from the data obtained. For example, consider vehicles 3 and 4 in Table 2. The ingress time of vehicle 3 is 30.014 s and for vehicle 4 it is 32.716 s. The corresponding time difference is

$$h_t = Ingress_4 - Ingress_3 = 2.716 - 30.014 = 2.702\ s$$

This data can be used to develop time headway distributions to analyze the impact of congestion and free-flow traffic on driver behavior.

4.2. Time-space diagram

A time-space diagram shows the vehicle trajectories and can be used to determine the time and space headways. The data for five vehicles in a single lane over the 50 m road section at intervals of 10 m is illustrated in figure 9. The time between two consecutive trajectories with respect to the *x*-axis gives the time headway, and with respect to the *y*-axis gives the distance headway. Figure 9 shows that at 20 m, the time headway between the car (dark blue) and van (orange) is 3 s, and between the van (orange) and car (gray) is 2 s. At 30 m, the time headway between the motorbike (yellow) and the car (gray) is 2.3 s and between the car (light blue) and motorbike (yellow) is 0.7 s. Similarly, the distance headway between vehicles can be obtained using a reference point on the *y*-axis. For example, at 3 s, the distance headway between the van (orange) and car (blue) is 28 m and this decreases to 20 m at 5.77 s. At 9.6 s, the distance headway between the motorbike (yellow) and car (light blue) is 10 m and this decreases to 7 m at 11.5 s. Headway data is important for traffic control such as signal optimization.

Table 2. The Traffic Data Obtained for Five Vehicles

Vehicle No.	Ingress (s)	Egress (s)	Elapsed Time (s)	Distance (m)	Speed (m/s)
1	23.654	26.613	2.959	50	16. 90
2	28.572	31.966	3.394		14.73
3	30.014	33.688	3.674		13.61
4	32.716	36.709	3.993		12.52

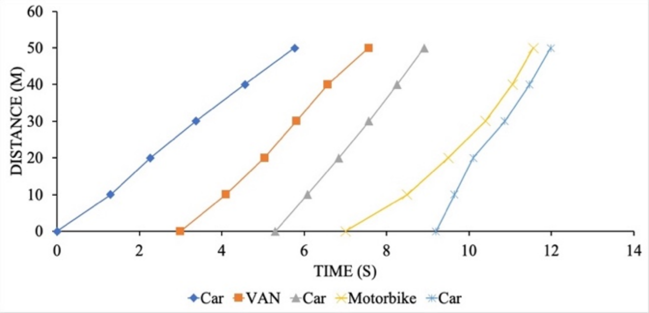


Figure 9. Trajectories for 5 vehicles on a 50 m University Road section.

4.3. Density map

A density map is used to illustrate the traffic density on a road. The traffic density is the number of vehicles passing a given location during the observation period. Figure 10 shows the density map for 1442 s on the 50 m road section. Warm colors indicate high density whereas cool colors are low density. Red, yellow, green, cyan and blue denote 193, 159, 136, 61 and 0 vehicles, respectively. This figure shows that the traffic is not homogenous as the density is highest at the center of the road. The edge of the road has low density and is sometimes devoid of vehicles. Thus, the road infrastructure is not efficiently used. Congestion and travel delays in Peshawar, Pakistan are often the result of road infrastructure and heterogeneous traffic.

4.4. Vehicle trajectories

The Camlytics software is capable of tracking and determining vehicle trajectories. Figure 11 shows the trajectories (white lines) for 1442 s on the 50 m road section. This shows that there are fewer vehicles on the right side of the road. These trajectories can be used to analyze vehicle dynamics and detect abnormal behavior to improve road safety [37].

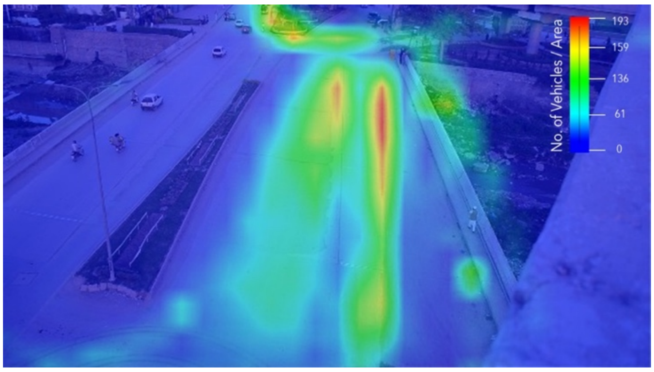


Figure 10. Vehicle density map on a 50 m University Road section over a period of 1442 s.

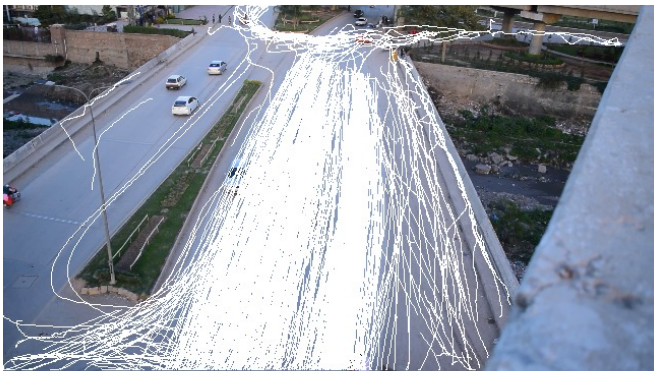


Figure 11. The vehicle trajectories on a 50 m road University Road section over a period of 1442 s.

5. Conclusion

In this research, an IoVT based solution has been proposed for real-time traffic flow characterization. This solution is a step towards exploiting embedded sensors to obtain real-time traffic statistics. The proposed solution has the capability to measure vehicle count, speed, density, time headway, time-space diagrams and trajectories. The sensor node is developed using an RPi 3B, Pi camera, Wi-Fi module and power bank. The sensor node has the capability to live stream MJPEG video of 640x480 resolution at 20 fps for over 11 hours with a fully charged power bank without any human intervention. It is a small and compact solution that can be easily moved and mounted anywhere as it weighs less than 500 g. With fabrication cost below $50, it is a low-cost solution as compared to intrusive and non-intrusive sensors. For traffic flow analysis, streamed video is analyzed using Camlytics installed on a Dell desktop located at our research lab.

For field testing, the proposed IoVT solution was evaluated in an urban setting. Traffic flow parameters such as vehicle count, speed, time headway, time-space diagrams, trajectories, and road capacity were measured. These results show that the proposed solution can characterize traffic under different traffic behavior including congested and heterogeneous traffic. Thus overcoming inherent limitations of both intrusive and non-intrusive sensor solutions. These statistics can be employed for validation and calibration of traffic simulation software and mathematical traffic flow models. These calibrated simulation software and mathematical models can in turn be used for better planning, designing and management of road networks.

In future, the proposed solution will be extended to a connected network of multiple sensor nodes to evaluate traffic flow on complex road configurations such as intersections, roundabouts, and bidirectional roadways with U-turns. This will require multiple sensor nodes streaming synchronized video streams. Furthermore, energy harvesting methodologies and solar energy will be explored to extend the battery life of the nodes.

Acknowledgements.

This work has been funded by the Higher Education Commission, Pakistan to establish a National Centre in Big Data and Cloud Computing, University of Engineering and Technology, Peshawar.

References

[1] World Health Organization (WHO) [Internet], [cited 2019 June 6]. Available from: https://www.who.int/gho/urban_health/situation_trends/urban_population_growth_text/en/

[2] Khan N, Khattak KS, Ullah S, Khan ZH. A low-cost IoT based system for environmental monitoring. In IEEE 2019 International Conference on Frontiers of Information Technology (FIT). 2019; Islamabad, Pakistan; 73-1735.

[3] Sohail AM, Khattak KS, Iqbal A, Khan ZH, Ahmad A. Cloud-based detection of road bottlenecks using OBD-II telematics. In IEEE International Multitopic Conference (INMIC). 2019; Islamabad, Pakistan; 1-7.

[4] Khan A, Khattak KS, Khan ZH, Khan MA, Minallah N. Cyber physical system for vehicle counting and emission monitoring. International Journal of Advanced Computer Research. 2020;10(50):181-193.

[5] Ullah MR, Khattak KS, Khan ZH, Khan MA, Minallah N, Khan AN. Vehicular traffic simulation software: A systematic comparative analysis. Pakistan Journal of Engineering and Technology. 2021;4(1):66-78.

[6] Imran W, Khan ZH, Gulliver TA, Khattak KS, Saeed S, Aslam MS. Macroscopic traffic flow characterization for stimuli based on driver reaction. Civil Engineering Journal. 2021;7(1):1-13.

[7] Iftikhar A, Khan, ZH, Gulliver TA, Khattak KS, Khan MA, Ali M, et al. A macroscopic traffic flow characterization at bottlenecks. Civil Engineering Journal. 2020;6(7):1227-42.

[8] Khan ZH, Gulliver TA, Imran W, Khattak KS, Altamimi AB, Qazi A. A macroscopic model based on relaxation time, Alexandria Engineering Journal. 2021;61(1):585-96

[9] Khan ZH, Gulliver TA, Azam K, Khattak KS. Macroscopic model on driver physiological and psychological behavior at changes in traffic. Journal of Engineering and Applied Science. 2019;38(2):1-9.

[10] Khan ZH, Imran W, Gulliver TA, Khattak KS, Wadud Z, Khan AN. An anisotropic traffic model based on driver interaction. IEEE Access. 2020; 8:66799-812.

[11] Guerrero-Ibáñez J, Zeadally S, Contreras-Castillo J. Sensor technologies for intelligent transportation systems. Sensors. 2018;18(4):1212.

[12] Tewolde GS. Sensor and network technology for intelligent transportation systems. In IEEE International Conference on Electro/Information Technology. 2012; Indianapolis, IN, USA;1-7.

[13] Jinturkar S, Pawar S. Vehicle detection and parameter measurement using smart portable sensor system. Communication and Applied Electronics.2016;4:5-9.

[14] Rashid MT, Hossain A, Hussain MN, Automated traffic measurement system based on FCD and image processing. In TENCON 2017 - IEEE Region 10 Conference. 2017; Penang, Malaysia;1381-1385.

[15] Gregor D, Cikel K, Arzamendia M, Gregor R. Design and implementation of a counting and differentiation system for vehicles through video processing. International Journal of Computer and Information Engineering. 2016;10(10):1771-1778.

[16] RiyazHussain S, Lokesh CR, Vamsikrishna P, Rohan G. Raspberry Pi controlled traffic density monitoring system. In International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET). 2016; Chennai, India;1178-1181.

[17] McQueen R. Detection and speed estimation of vehicles using resource constrained embedded devices [Internet]. 2018 [cited 2019 April 19]. Available from: https://people.ok.ubc.ca/rlawrenc/research/Students/RM_18_report.pdf.

[18] Raktrakulthum PA, Netramai C. Vehicle classification in congested traffic based on 3D point cloud using SVM and KNN. In 19th International Conference on Information Technology and Electrical Engineering (ICITEE); 2017; Phuket, Thailand;1-6.

[19] Sundararajan M, Khanaa V. Counting and classification of highway vehicles by using Raspberry Pi. International Journal of Pure and Applied Mathematics. 2018;118(18):193-201.

[20] Iqbal M, Yuwono RT, Fadhillah HM, Widiyanto S. Realtime stereo vision for vehicle detection, classification and counting using Raspberry Pi. Journal of Engineering and Applied Science. 2017;12(8):2207-2213.

[21] Espinoza FT, Gabriel BG, Barros MJ. Computer vision classifier and platform for automatic counting: more than

cars. IEEE Second Ecuador Technical Chapter Meeting (ETCM). 2017; Salinas, Ecuador;1-6.

[22] Nguyen HQ, Loan TT, Mao BD, Huh EN. Low cost real-time system monitoring using Raspberry Pi. In Seventh International Conference on Ubiquitous and Future Networks. 2015; Sapporo, Japan;857-859.

[23] Balakrishna S, Thirumaran M. Semantic interoperable traffic management framework for IoT smart city applications. EAI Endorsed Transactions on Internet of Things. 2018; 4(13):1-17.

[24] Rohadi E, Suwignjo SA, Pradana MC, Setiawan A, Siradjuddin I, et al. Internet of Things: CCTV monitoring by using Raspberry Pi. In 2018 International Conference on Applied Science and Technology (iCAST). 2018; Manado, Indonesia;454-457.

[25] Tian Y, Babcock R, Taylor C, Ji Y. A new live video streaming approach based on Amazon S3 pricing model. In 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC). 2018; Las Vegas, NV, USA;321-328.

[26] Filteau J, Lee SJ, Jung A. Real-time streaming application for IoT using Raspberry Pi and handheld devices. In 2018 IEEE Global Conference on Internet of Things (GCIoT). 2018; Alexandria, Egypt;1-5.

[27] Dursun M, Çuhadar I. Secure video streaming implementation for unmanned air vehicle (UAV) data link with Raspberry Pi 3 over https. Bilişim Teknolojileri Dergisi. 2018;11(1):23-28.

[28] Jennehag U, Forsstrom S, Fiordigigli FV. Low delay video streaming on the internet of things using Raspberry Pi. Electronics. 2016;5(3):60.

[29] Saha, JK, Ghuri MMA, Mamur MMR, Hossain MJ, Chowdhury TA, Paul B. A novel design and implementation of a real-time wireless video and audio transmission device. WSEAS Transactions on Computer Research. 2016; 4:161-172.

[30] Wallace D, Hament B, Vaz JC, Oh P. Augmenting a miniature humanoid platform with a low-cost networked computer vision framework. In 2018 IEEE 8th Annual Computing and Communication Workshop and Conferencce (CCWC). 2018; Las Vegas, NV, USA; 564-569.

[31] Komasilovs V, Zacepins A, Kviesis A, Nasirahmadi A, Sturm B. Solution for remote real-time visual expertise of agricultural objects. Agronomy Research. 2018;16(2):464-473.

[32] González S, Vargas TR, ArceP, Guerri JC, Energy optimization for video monitoring system in agricultural areas using single board computer nodes and wireless ad hoc networks. In 2016 XXI Symposium on Signal Processing, Images and Artificial Vision (STSIVA). 2016; Bucaramanga, Colombia; 1-7.

[33] Raspberry Pi [Internet], [cited 2019 April 25] Available from: https://www.raspberrypi.org/products/raspberry-pi-zero-w/.

[34] FFmpeg [Internet], [cited 2019 April 25] Available from: https://ffmpeg.org/.

[35] Camlytics [Internet], [cited 2019 April 25] Available: https://camlytics.com/.

[36] Alghyaline S, El-Omari NK, Al-Khatib RM, Al-Kharbshh HY. RT-VC: An efficient real-time vehicle counting approach. Journal of Theoretical and Applied Information Technology. 2019; 97:2062-2075.

[37] Jacob B, Violette E. Vehicle trajectory analysis: An advanced tool for road safety. Procedia-Social and Behavioral Science. 2012; 48:1805-1814.