Pakistan Journal of Engineering and Technology, PakJET
ISSN (p): 2664-2042, ISSN (e): 2664-2050
Volume: 5, Number: 2, Pages: 119- 127, Year: 2022

Vehicular Traffic Flow Characterization: An Edge Computing Perspective

Shehzad Ayaz[1], Khurram S. Khattak[1], Zawar H. Khan[2], Azhar Qazi[3]

[1]Department of Computer Systems Engineering, UET Peshawar, Pakistan
[2]Electrical Engineering Department, UET Peshawar, Pakistan
[3]Department of Electrical Engineering, CECOS University, Pakistan

Corresponding author: Khurram S. Khattak (e-mail: khurram.s.khattak@gmail.com).

Received: 15/03/2022, Revised: 20/05/2022, Accepted: 30/06/2022

Abstract- **Smart urban mobility solutions are being proposed to enhance the urban road network's efficiency. From a smart urban mobility perspective, real-world vehicular traffic flow parameters (such as vehicular count, flow, classification, speed, road capacity, time/distance headway, temporal/spatial densities, heatmaps, and trajectories) are imperative. In this context, a plethora of solutions have been proposed for traffic flow characterization in existing literature. These solutions can be categorized as either intrusive sensors, non-intrusive sensors or Internet-of-Video-Things based solutions. However, these solutions have serious limitations. In this context, compute vision based edge computing solutions have emerged as an optimum solution for traffic flow characterization. The objective of this work is to propose mathematical solutions for further enhancement of the already proposed edge computing solution's performance. A comprehensive literature survey of proposed edge computing solutions has been undertaken. Edge computing solutions are seriously constrained because of the compute resources of single board computers. At most only two traffic flow parameters (either count and classification or count and speed) can be measured. To overcome this limitation, mathematical traffic flow equations for eight additional traffic flow parameters have been reported. These additional eight parameters calculated mathematically range from traffic flow, density, sensitivity, equilibrium, critical density, time\distance headway and driver presumptions. Using these mathematical equations instead of compute heavy image processing can enhance already proposed edge computing solutions by 400%.**

Index Terms-- **Intelligent transportation system, Edge Computing, traffic flow characterization, Raspberry Pi, IoT.**

I. INTRODUCTION

With increasing urbanization, various challenges have emerged in developing future smart cities. One of the most pressing among these challenges is urban mobility. Associated problems with urban mobility range from congestion, accidents, productivity losses and ambient air pollution. With 25% share in world energy consumption, the transportation sector is responsible for 29% of overall greenhouse gas (GHG) emissions [1]. Furthermore, the transport sector is a major contributor of carbon dioxide (CO_2), nitric oxides, carbon monoxide (CO), and particulate matter (PM) in urban environments. This ambient pollution causes serious health issues such as cardiovascular, respiratory, pulmonary and cancer [1, 2]. Traffic congestion is the primary source of road network's inefficiency leading to time and labor productivity losses. According to a Texas A&M Transportation Institute report, an average American driver spent seven days in traffic congestion costing each driver on average $1000 in 2017 alone [3].

Intelligent transportation system (ITS) based solutions are gaining traction for providing smart urban mobility. ITS is integration of different technologies (such as compute boards, sensors, communication, cloud platforms, algorithms, and big data analytics) for better road network design, planning and management. One of the fundamental building blocks for providing ITS based solutions are real-world vehicular traffic flow parameters. These traffic flow parameters range from vehicle count, flow, speed, classification, density, sensitivity, time\distance headway, spatial\temporal densities, road capacity, and trajectories. Using these parameters, road network inefficiencies such as road bottlenecks can be identified. Furthermore, these parameters can be used for calibration and validation of traffic simulation software (such as VISSIM, Corsim and Paramics) and mathematical traffic flow models for better planning, designing and management of road networks [3].

In this context, varying solutions have been proposed for traffic flow characterization in existing literature. These solutions can be categorized as either intrusive sensors, non-intrusive sensors or Internet-of-Video-Things (IoVT) based solutions. However all of these have their limitations as detailed in section 2. With advancement in computing technology, edge computing solutions have emerged as the most optimum solution for traffic flow characterization. This research was undertaken with the following objectives:

- Limitation of intrusive\non-intrusive sensors and IoVT based solutions.

- Identifying the optimum solution for roadside traffic flow characterization.
- Proposing mathematical traffic flow characterization equations for optimizing edge computing-based solutions.

Edge computing solution's primary limitation is their compute resource constraints. Thus limiting them to measure at most only two traffic flow parameters (either count\classification or count\speed) as detailed in section 3. In this work we have reported mathematical traffic flow characterization equations for calculating eight traffic flow parameters. Compute resource requirements for mathematical equations are comparatively far less than image processing. Incorporating these mathematical equations can enhance the currently proposed edge computing solution's performance by 400%.

The rest of the work has been organized such that section II details the limitation of intrusive sensors, non-intrusive sensors and IoVT based solutions. In section 3, all edge computing solutions for traffic flow characterization in existing literature have been reported and analyzed. In section 4, mathematical equation-based solutions have been proposed for mitigating compute resource constraints of edge computing solutions. Lastly, conclusion and future work has been presented in section 5.

II. LIMITATIONS OF EXISTING PROPOSED SOLUTION

In existing literature, various solutions have been proposed for roadside traffic flow characterization. The second generation sensing technologies can be broadly categorized as either intrusive or non-intrusive sensors. Though marked improvement over manual counting, these sensors have serious limitations as tabulated in Table 1. Biggest limitation is their inability to characterize traffic under congested and heterogeneous traffic conditions. Moreover, these sensors can only measure vehicle count, speed and classification. as can be seen in Table 1.

To overcome inherent limitations of intrusive and non-intrusive sensors, compute vision based solutions (both IoVT and edge computing solutions) are emerging as the most optimum solution for traffic flow characterization. In compute vision based solutions, a single camera can act as a sensor by capturing roadside traffic flow video. Using image processing algorithms, a full range of traffic flow parameters (such as vehicle count, flow, speed, classification, road capacity, time/distance headway, temporal/spatial densities, heatmaps and trajectories) can be measured. Distinctive advantage of compute vision based solution is its capability to characterize traffic flow under all traffic conditions (such as congested, uncongested, homogeneous and heterogeneous). Unlike intrusive and non-intrusive sensors, this also includes the ability to count pedestrians, bicycles, three-wheelers, animal/human drive carts. Furthermore, performance of compute vision based solutions are less susceptible to meteorological conditions.

For better clarity, this section has been divided into four subsections: (A) intrusive sensors, (B) non-intrusive sensors, (C) IoVT based solutions, and (4) compute vision based edge computing solutions.

A. INTRUSIVE SENSORS

Intrusive sensors as the name suggests are mostly embedded in road surfaces. Though low cost, intrusive sensors are temporary solutions and cause traffic disturbances during installation and maintenance [4]. Intrusive sensors though highly accurate, can only measure vehicle count. To measure vehicle speed or classification, complex configuration of multiple intrusive sensors is required thus affecting their accuracy [4-6]. Major disadvantages of intrusive sensors are their inability to characterize traffic under congested and heterogeneous traffic conditions. Furthermore, these types of sensors are unable to detect pedestrians, animal/human driven carts, bicycles, and three-wheelers.

B. NON-INTRUSIVE SENSORS

Non-intrusive sensors are installed on or above road surfaces for traffic flow characterization. Though expensive as compared to intrusive sensors, they are easy to install, operate and maintain. Non-intrusive sensors are less damaging to the road surface as well as less detrimental to traffic flow during installation and maintenance. Non-intrusive sensors can provide more traffic flow parameters as compared to intrusive sensors. However, these are also incapable of characterizing traffic flow under congested and heterogeneous traffic conditions. Furthermore, accuracy of these sensors is highly sensitive to meteorological conditions such as rain, fog, wind, temperature, sound, and lightning conditions as tabulated in Table 1 [2, 4].

TABLE I
LIMITATIONS OF INTRUSIVE AND NON-INTRUSIVE SENSORS [4-6]

Sensor Type	Parameters	Disadvantages
	<u>Intrusive Sensors</u>	
Inductive Loop	Count	Traffic disturbance during installation and maintenance. Inability to detect non-metallic objects.
Pneumatic Tube	Count, Classification	High energy consumption, Susceptible to be torn under heavy traffic. Inability to operate under congested traffic.
Piezoelectric sensors	Count Speed	Affected by road surface temperature For heterogeneous traffic multiple sensors are required. Inability to operate under congested traffic.
Magnetic sensors	Count, Classification and Speed	Inability to detect non-metallic objects such as animal\human driven cart and pedestrians, Proximity to vehicles is required, Complex configuration of multiple sensors is required for classification and speed estimation.
	<u>Non-Intrusive Sensor</u>	
Accelerometers	Count, Speed	Sensitive to environmental vibrations, Incapable to detect stationary objects.
Acoustic sensors	Count, Classification	Sensitive to environmental sounds.

		Incapable to detect silent objects such as pedestrians, bicycles and animal/human driven carts.
Infrared sensors	Count, Speed, Classification	Reactive to sunlight, Susceptible to meteorological conditions such as fog, dirt, and wind.
Ultrasonic	Count, Speed	Performance affected by temperature and air turbulence.
Microwave Radar	Count, Speed, Classification	Poor performance under congested traffic conditions Vulnerable to electromagnetic interferences.
Bluetooth Beacons	Count, Classification	Poor performance in congestion when transmitter and receiver are not in line of sight.
Wi-Fi	Count, Classification	Static or slow-moving vehicles affects the performance of Wi-Fi due to weak signal strength received by receiver

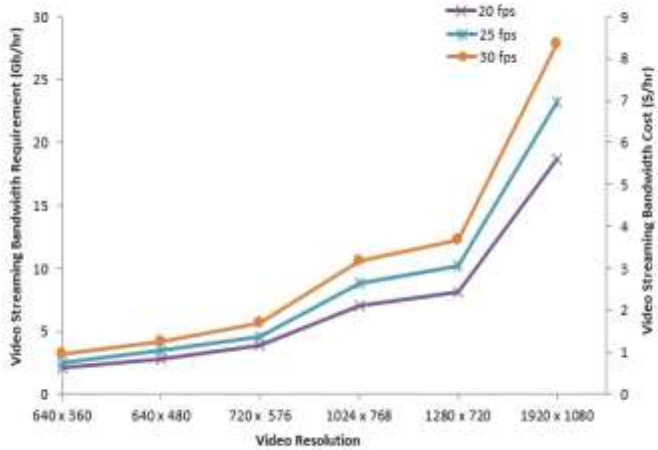

FIGURE 1. Data transmission bandwidth requirement for video streaming at different resolutions and fps.

C. IoVT SOLUTIONS

IoVT solutions are where roadside video is streamed wirelessly to a server for image processing [7, 8]. IoVT solutions are marked improvement over both intrusive and non-intrusive sensors. This improvement is both in terms of measured traffic flow parameters and ability to work under all traffic conditions. However, the biggest disadvantage of IoVT solutions is roadside video streaming's internet data bandwidth requirement.

Video streaming is uninterrupted transmission of video in the form of data packets over the internet (UDP/IP for faster transmission of data). Beside reliable transmission, energy-efficient real-time video encoding/decoding is an added requirement for such systems. For example, a 1920x1080 @ 30 frame per second (fps) video in raw form will require 1.49 Gbps internet data bandwidth as opposed to 10 Mbps if the same video is encoded using H.264 encoder [9]. For higher traffic flow parameter's measurement accuracy, higher video resolution and fps are imperative [10]. This can further increase video streaming's internet data bandwidth requirements. Roadside video streaming's internet data bandwidth requirements and costs per hour for different video formats are shown in Fig 1. The cost per hour for video streaming's internet data bandwidth requirements are calculated based on the average internet package (50 GB/ $15) available in Pakistan.

In addition to video streaming internet data bandwidth requirement, current consumption of the IoVT sensor node for video streaming has to be considered too. For example, Yuichiro et al. reported that as compared to software H.264 encoding, hardware H.264 encoders require 58 times less energy to encode a video of same resolution and fps [9]. In this context, Raspberry Pi (RPi) has emerged as the optimum single board computer (SBC) from a cost (both monetary and power consumption) efficiency perspective. RPi has built-in hardware H.264 encoder [9, 11].

D. EDGE COMPUTING SOLUTIONS

In light of limitations of above reported solutions (intrusive and non-intrusive sensors, IoVT, edge computing solutions have emerged as the most optimum solution. Salient point of such solutions are:

(i) Unlike intrusive and non-intrusive sensors, these solutions have the capability to characterize traffic under all traffic conditions (such as congested, uncongested, homogeneous and heterogeneous).
(ii) Unlike intrusive and non-intrusive sensors, these solutions have the capability to detect all types of vehicles (such as bicycle, bikes, animal/human driven carts) and pedestrians.
(iii) Unlike non-intrusive sensors, meteorological conditions (such as rain, fog, sunlight, temperature, noise, wind, electromagnetic interference) impact on compute vision-based solution's accuracy and performance is negligible.
(iv) Unlike IoVT solutions, edge computing solutions don't require high video streaming internet data bandwidth. With edge computing solutions, only 572 bytes are required to transmit 1 minute of roadside traffic flow parameters as opposed to 1 MB of video streaming [12].

However, performance of edge computing solutions is limited under the constraint of SBC's compute resources. This is because of the amount of computation resources required for real-time image processing. In edge computing, an SBC is employed to process roadside traffic video in real-time. However with technological advancement in SBCs, this limitation can be overcome as can be seen in Table 2. In [13], a comparative analysis was conducted using four different SBCs such as Raspberry (RPi) B+, TE6210 CITRIC platform, PAC Duo and S5PV210 Arm development kit. With respect to computation resources and cost, it was concluded that RPi performed better. In [10], four SBCs (RPi B+, Beagleboard Xm, RPi 2 and Odroid XU4) were compared from traffic flow characterization perspective. It was reported that with 98% traffic characterization accuracy, Odroid XU4 performance was the best. This was attributed to Odroid XU4's 2GHz quad-core ARM Cortex-A53f processor and 2GB RAM. However, with advanced Raspberry (RPi) 4 specifications as can be seen in Table 2, RPi has become a better choice. Accuracy rate of edge computing solutions employing RPi has improved over the years as can be seen in Table 3.

A. SINGLE BOARD COMPUTER (SBC)

Major limitation of edge computing solutions for traffic flow characterization is computation resource constraints of SBCs. Compute vision algorithms are compute heavy and as such needs high processing capability. However, with technological advancement in SBCs, it is hoped that in near future this constraint will be overcome. As can be seen in Table 3, RPi is fast emerging as a low-cost optimum SBC for edge computing solutions for traffic flow characterization.

RPi is a small credit card sized SBC with the ability to intercommunicate with other peripheral devices. Furthermore with inbuilt hardware H.264 encoder/decoder, RPi is the lowest cost solution available for IoVT based traffic flow characterization. RPi has a built-in 802.11n wireless module and specialized CSI port for integrating cameras. Continuous technological advancements are being made to make it even more compute optimum over the years as can be seen in Table 2. As such, RPi has become the most optimum SBC for edge computing solution in existing literature as can be seen in Table 3.

TABLE II
TECHNICAL SPECIFICATIONS OF DIFFERENT MODELS OF RASPBERRY PI

Specs	RPi 1 B+	RPi 2 B	RPi 3 B+	RPi 4 B
Year	2014	2015	2018	2020
SoC	Broadcom BCM2835	Broadcom BCM2836	Broadcom BCM2837B0	Broadcom BCM2711
CPU	ARM11 (32bit)	Cortex-A7(32bit)	Cortex-A53(64bit)	Cortex-A72
GPU	Video Core IV	Video Core IV	Video Core IV	Video Core VI
Clock Cycle	700MHz	900MHz	1.4 GHz	1.5GHz
Cores	Single-core	Quad-core	Quad-core	Quad-core
RAM	512 MB	1 GB	1GB	8GB
Power	5V	5V	2.5A/5V	2.5A/5V
Storage	MicroSD card	MicroSD card	MicroSD card	MicroSD card

Further computation efficiency while employing RPi can be achieved by using multiprocessing techniques. For example in [14], the image processing algorithm was divided into four steps. These were then processed stepwise using four cores of an RPi 2. Effects of video's resolution and fps on RPi's compute resources were studied in [14, 15].

B. SOFTWARE

For compute resource optimization of an SBC, the choice of programming language and image processing algorithms employed are imperative. In this context, OpenCV (Open-Source Computer Vision Library) is the most employed image processing library as can be seen in Table 3. OpenCV consists of over 2500 optimized algorithms for image processing such as face detection, tracking movements, video capturing, object's 3D models extraction and producing 3D point clouds from stereo cameras [15].

OpenCV runs much faster than similar programs written in MATLAB, with the added capability of further computational optimization [16]. With OpenCV, it is possible to analyze 30 fps as compared to MATLAB which can process only 3-4 fps. Further computational optimization can be achieved through the choice of programming language. C/C++ is recommended for edge computing solutions as OpenCV is written in it. This choice can be further collaborated from proposed solutions in existing literature. Nearly all edge computing solutions have employed OpenCV/C++ as can be observed in Table 3.

In existing literature, nearly all edge computing solutions proposed have employed RPi for traffic flow characterization as can be seen in Table 3. Advancement in compute vision algorithms is another area where innovation is being proposed for further improvement as can be seen in Table 3.

III. PROPOSED EDGE COMPUTING SOLUTIONS SURVEY

In light of discussion in section 2, compute vision-based edge computing solutions have emerged as the most optimum solution for traffic flow characterization. However, under the constraint of compute resources of SBCs, these solutions can provide only either vehicle count, count\classification or count\speed as can be observed in Table 3.

A. COUNT

A solution to measure traffic density was proposed in [16]. In the proposed solution, Pi camera was integrated with RPi through USB for capturing roadside traffic video. The traffic video is analyzed and in each frame the background is subtracted resulting in black and white pixels. The black pixels show the empty space on the road, while white pixels represent vehicles or pedestrians. The percentage of black and white pixels of all the frames were calculated to estimate traffic density and congestion on the road. Final traffic density report was displayed to travelers on public screens placed on roadsides.

Bhusari et al. [17] proposed a smart traffic control system using RPi 2, Pi camera and image processing. An AVR microcontroller was integrated with RPi for controlling LED traffic signals for traffic flow management. Traffic flow was analyzed using image processing techniques i.e., morphological operation and blob analysis. Vehicles were counted and compared with a user defined road density threshold. If road density was above user defined threshold, resulting traffic jam was cleared through traffic signals by giving priority to the roadside with traffic jam. The proposed solution had the added capability to detect ambulances and give right of way to such vehicles.

Mallikarjun et al. [18] designed a solution to detect, track and count vehicles in real time using RPi 3 with OpenCV installed.

Proposed solution's accuracy was reported at 96% from side and top view, 92.8% from front and rear view and 93% from multi view. Color features of vehicles were extracted to separate vehicles from the background. Vehicle's contours were detected in the image through conversion from RGB to HSV, due to its wide color space. The detected contours were then tracked through their calculated centroid points using Kalman filter. It was reported that on average RPi took 81.5ms to process one single frame [18].

In [19], Floating Car Data (FCD) was combined with image processing algorithms for vehicle counting. FCD was collected through detection of cellular phones in the vicinity through installed network towers. The vehicle count data through FCD, and image processing were compared individually and then with each other at a remote server. Accuracy of the proposed solution was reported at 93%.

TABLE III
PROPOSED RPI BASED EDGE COMPUTING SOLUTION FOR VEHICULAR FLOW CHARACTERIZATION SOLUTIONS

RPi 1 B+	RPi 2 B	RPi 3 B+	RPi 4 B	Ref
Hardware/ Software	Type/ Reported Accuracy	Parameters	Imaging Techniques	[16]
RPi 1, OpenCV/ Python Webcam 30 fps	Real time	Count	1. Background subtraction 2. Calculate black and white pixels count 3. Calculate percentage of free space on the road	[18]
RPi 3 OpenCV/ C++ Pi Camera 720x640 30 fps	1. Real time 2. Accuracy from Multi view, Side-top view and front-rear view was 93%, 96% and 92% respectively.	Count	1. RGB to HSV 2. Contour detection based on 8-connectivity, chain code method 3. Kalman filter	[19]
RPi OpenCV 10 fps	Real time 93% Accuracy	Count	1. Floating car data (FCD) 2. image processing	[10]
RPi 2B+ Pi camera 320x240 pixels 30 fps	Real time 70% Accuracy	Count	1. RGB to Gray conversion 2. Gaussian Blur and Thresholding 3. Blob detection 4. Cross correlation	[17]
RPi 2 OpenCV/ C++ Pi Camera	Real time	Count	1. Image Acquisition and Processing 2. Morphological Processing 3. Blob Analysis	[20]
RPi 3 B MATLAB Webcam	Real time	Count	1. Foreground detection 2. Gaussian Mixture models 3. Median filter to remove noise 2. Blob analysis	[11]
RPi 3 B OpenCV/ C++ Pi Camera	Real time 86.9% Accuracy	Count	1.Background Subtraction 2.Contour Detection 3.Convex Hall and tacking	[7]
RPi 3 B+ Python / Matlab Pi Camera	Streaming Accuracy for medium, heavy and nighttime at 91.08%, 97.47% and 88.16% respectively	Count Classification	1. Color image to grayscale 2. Histogram equalization 3. Cropping 4. Morphological operations	[21]
RPi 3 B OpenCV/ Python Webcam 1024x768 @ 30fps	Real time Stereo Average accuracy under different condition 90%	Count Classification	1. Haar-like features for real time segmentation 2. Random Sample Consensus Algorithm (RANSAC) 3. Histogram of Oriented Gradient 4. Scale Invariant Feature transform 5. Normalized-Sum of Squared differences (NSSD)	[14]
RPi 2 OpenCV/ C++ IP Camera 640x360 @ 15 fps	Real time Write accuracy 98%	Count Classification	1. Background subtraction (BGS) based on Gaussian mixture model (GMM) for Shadow removal 2. Morphological operations 3. Canny edge detector 4. Kalman filters 5. Hungarian algorithm for assignment	[12]
RPi 3 OpenCV/C++ Smartcam Samsung model SNH-E6440BN	Real time Accuracy =83%	Count, Classification	1. Support Vector Machine (SVM), 2. Kalman filter 3. Sensor Observation Service 4. K-Nearest Neigbors (KNN) and Gaussian Mixture Model (MOG2) for background subtraction 5. Dilation and erosion for foreground extraction and tracking	[8]
RPi OpenCV/ Python Pi Camera 480x360 @ 40 fps	Streaming Vehicle classification accuracy of SVM and KNN was reported at 95.8 % Motorcycle classification accuracy of SVM and KNN reported at 82.2% and 57.2%	Count, Classification	1. Support vector machine (SVM) 2. K-Nearest Neighbor (KNN)	[22]

	respectively .			
RPi OpenCV/ C++ Web camera @640x480	Real time	Count, Classification	1. Object detection, 2. Background subtraction 3. Kalman filter 4. Nearest neighbor algorithm	[23]
RPi model B OpenCV/ Python Pi camera	Real time	Count, Classification	1. Feature Extraction and Matching 2. Remove Unmatched feature 3. Scale Invariant Feature transform (SIFT), 4. RANSAC algorithms	[15]
RPi 2 OpenCV/ Python @ 320p, 540p, 720p	Real time	Count, Speed	1. Color conversion, 2. Motion detection 3. Get coordinates, 4. Speed calculation	[24]
RPi 3 B+ OpenCV/ C++ Pi camera	Real time Vehicle count and speed accuracy was reported at 100% and 80% respectively .	Count, Speed	1. Background Subtraction 2. Thresholding 3. Morphological processes 4. Contour detection & object tracking 5. Speed calculation	[25]

B. COUNT AND CLASSIFICATION

In [7], a solution was proposed for traffic counting and classification under different lighting conditions (day and night). The solution had the capability to classify vehicles either as small or large vehicles. First unnecessary portions in a frame were cropped, detecting bright and dark vehicles with maxima\minima transform and performing morphological operation to separate connected vehicles. Accuracy was reported under three types of traffic conditions with 91.08% in medium traffic, 97.47% in heavy traffic and 88.16% at nighttime traffic.

In [21], a solution was proposed for traffic counting and classification by employing two RPis with Pi cameras. The two cameras extracted Haar like features and made a bounding box around the detected vehicle. If the same detected bounding box is found in both systems, the two are compared to get matching features. Counting and classification was done by calculating width and height of the bounding boxes.

Gregor et al. proposed a solution for traffic counting and classification using RPi 2 [14]. To overcome computation resource constraints of RPi 2, the image processing algorithm was divided into four steps. These were inturn run parallely on RPi's CPU four cores. Four steps of the image processing algorithm were (1) image capturing, (2) Background Subtraction, (3) Detection and Tracking and, (4) Counting and Classification. For further compute resource optimization, the region of interest (ROI) was defined in each video frame. Regions outside ROI were discarded to help reduce computation needs. Experiments were conducted on videos of different resolutions and fps to find optimum resolution and fps from an accuracy perspective. It was concluded that the video with 640x360 resolution at 15 fps was the optimum choice from computation needs and accuracy perspective. Classification of vehicles was performed based on the bounding box size of detected vehicles.

Felipe Torres et al. proposed traffic counting and classification solution using RPi with an integrated Smartcam (SNH-E6440BN Samsung) [12]. Using a three parallel lines approach, the proposed solution has the capability to analyze bi-directional traffic flow. Measurement-based features (MBF) and intensity pyramid-based histogram-oriented gradients (IPHOG) were extracted from blobs. These blobs were in turn used to classify vehicles using support vector machine (SVM). Employing 5 minutes roadside traffic videos, an accuracy of 83% was reported under different weather conditions such as sunny, rainy, sunset and nighttime.

In [8], KNN and SVM algorithms were employed using viewpoint feature histogram (VFH) descriptors to count and classify vehicles as either cars or motorcycles. Image processing algorithm was a five-step process ranging from filtering, segmentation, tracking, feature extraction and Classification/count. Counting was measured after successful classification of a vehicle as either a car or motorcycle. For classification, two machine learning models SVM and KNN were compared and analyzed. An accuracy of 95.8% was reported for both models when classifying vehicles as a car. However for motorcycles, SVM with reported accuracy of 82.2% performed better than KNN with reported accuracy of 57.2% under high traffic conditions.

Suryatali et al. proposed a traffic counting and classification solution for toll collection [22]. The proposed solution employed RPi, Pi Camera and OpenCV libraries. Vehicles were counted and classified as either light or heavy vehicles in order to charge toll accordingly. Khaana et al. proposed an RPi based solution for traffic counting and classification [23]. The focus was on resolving resolution and image blurring issues. In the proposed solution, scale invariant feature transform (SIFT) and random sample consensus algorithm (RANSAC) were employed for vehicles counting and classification.

Balakrishna et al. proposed a solution for vehicle counting and classification. These traffic flow parameters were transmitted to a free and open-source cloud platform 'ThingSpeak' [20]. The proposed solution integrated an algorithm developed in the MATLAB tool called Simulink. For vehicle counting, image processing techniques (foreground detection, post-processing, and blob analysis) were performed on each frame.

Ali et al. proposed a real time solution for traffic flow monitoring such as vehicle count, traffic flow, density, and time headway [11]. Furthermore, associated roadside vehicle emissions (such as carbon dioxide, carbon monoxide and particulate matter) were also measured. Traffic flow monitoring was achieved through image processing techniques (such as Background subtraction, Contour detection, convex hull, and Tracking) deployed on RPi 3B with an accuracy of 86.9%. Measured and calculated parameters were transmitted to a cloud platform 'ThingSpeak' for storage, analysis, and visualization. Relationships between traffic flow parameters and roadside vehicular emissions were developed.

C. COUNT AND SPEED

Iszaidy et al. proposed a solution for traffic counting and speed estimation using RPi, Pi camera, OpenCV and Python [15]. Vehicles were detected, tracked, and assigned coordinates in each frame. Vehicle speed was estimated through coordinates of the detected vehicle in current and previous frames. To analyze computation resources, traffic videos of three different sizes 320p, 540p and 720p were captured. From the results, it was reported that RPi's CPU usage was almost the same for all three video resolutions. However, in the case of 720p video, memory requirement was higher than the other two.

In [24], an RPi with Pi camera was employed for vehicle counting and speed estimation. The video frames were processed for background subtraction through OpenCV library to isolate moving vehicles in frames. Morphological processes were carried out on the frames to enhance the detected objects and then the objects were tracked through contour detection. Individual vehicle's speed was estimated when the vehicle entered and exited from the calibrated region. A flag was set to monitor the distance covered by the vehicle in pixels and the number of frames it traveled from enter to exit lines. Accuracy for vehicle counting and speed estimation was reported at 100% and 80% respectively.

In [25], a solution for vehicle counting and speed estimation was proposed using RPi and Pi camera. Speed and orientation were estimated using the Gunnar Farneback method through calculation of optical flow. An Op matrix (16x16) was spread evenly throughout the image for two purposes: first to calculate optical flow between frames and second to reduce computational complexity. Measured traffic count and speed were transmitted to a local server using RPi's Wi-Fi module.

IV. DISCUSSION

Edge computing-based solutions for traffic characterization are seriously constrained under compute resources of SBCs. However, performance of these solutions can be markedly enhanced through employing mathematical traffic flow characterization equations. Mathematical traffic flow characterization has a lower computational cost as compared to compute visions algorithms-based traffic flow characterization. Ali et al. mathematically estimated traffic density, flow, and time headway from traffic count [11]. Distance headway, sensitivity, critical density, and driver presumptions can be estimated from traffic count and speed. Traffic flow is the number of vehicles passing over a unit length of road. The units of 'Traffic flow" are vehicles per unit time. Traffic flow ϒ can be calculated using (1) [26].

$$\Upsilon = \frac{N}{t} \tag{1}$$

Where N is the number of vehicles and t is the unit time. Time headway τ is required for velocity alignment between vehicles. This includes driver perception and reaction time. Perception time is required when a driver notices stimuli. While reaction time is required when vehicles align velocity between the forward and preceding vehicles. Time headway is the reciprocal of traffic flow and calculated using (2) [27],

$$\tau = \frac{1}{\Upsilon} \tag{2}$$

For a longer time headway (vehicles taking more time to align), the traffic flow is slower and thus congested. Conversely, when traffic flow is high then the time headway is shorter. Driver behavior also affects the traffic flow. For a shorter time headway, driver reaction is quick. Vehicle alignment quickly occurs in a shorter duration. Thus the driver response is aggressive. Conversely, a non-aggressive driver's reaction is slower thus taking longer time to align. Thus reducing the overall traffic flow [33]. For an equilibrium driver reaction, alignment is based on density, and flow is at equilibrium.

Traffic density δ is the number of vehicles over a length of road. Traffic density from the vehicle count can be calculated using (3) [28, 29]

$$\delta = \frac{N}{L}\,. \tag{3}$$

Where L is the length of the road segment under observation. Distance headway h is the road length covered by vehicles during time headway to align to conditions ahead [33]. Perception headway is road length covered when a driver perceives traffic conditions, while the time taken is perception time. A driver then takes action to align to conditions during reaction time headway. The distance covered during reaction is known as reaction distance headway. The distance headway is a sum of perception and reaction headway. While time headway is the sum of driver perception and reaction time. Distance headway is based on traffic density.

For high traffic density (i.e. congestion), the distance headway is smaller for drivers to align. Thus, it takes traffic longer to align. During free traffic flow, the distance headway is larger allowing quick alignment between vehicles as conditions are predictable. Distance headway can be calculated from traffic density using (4) [30]

$$h = \frac{1}{\delta}. \tag{4}$$

Traffic sensitivity ∇ is a measure of driver behavior [33]. A non-aggressive driver is less sensitive thus taking a longer distance headway to align to conditions ahead. An aggressive driver is more sensitive and takes a smaller distance headway to align. This results in a smaller change in distance headway. An equilibrium driver behavior is based on traffic density, taking an appropriate distance headway to align resulting in equilibrium traffic flow. Traffic sensitivity can be calculated using (5) [31]

$$\nabla = \frac{1}{\nabla h}. \tag{5}$$

Where Δ is the change in distance headway. For equilibrium traffic flow, equilibrium density δ_e can be calculated using (6)

$$\delta_e = \delta_m \left(1 - \frac{v(\delta)}{v_f}\right), \tag{6}$$

where δ_m is the maximum density, v_f is the velocity limit, and $v(\delta)$ is the equilibrium velocity distribution. Traffic sensitivity at equilibrium ∇_e is based on the distance headway at equilibrium, that is

$$h_e = \frac{1}{\delta_e}, \quad (7)$$

and sensitivity is

$$\Delta_e = \frac{1}{\Delta h_e}. \quad (8)$$

Fundamental diagram gives the characteristics of traffic flow on any given road. This diagram shows the critical flow, free flow and congested flow. The traffic flow congestion is due to either flawed infrastructure design\conditions or driver behavior. A driver's behavior is affected by the driver's ethnicity, age, psychological or physiological conditions, geographical location, traffic flow type, and meteorological conditions [34]. Beyond critical density, traffic flow becomes congested and results in stop and go traffic behavior. Critical flow is the maximum optimum utilization of road capacity and varies based on road conditions. Critical density $\propto$ from fundamental diagram is calculated as

$$\alpha = \frac{\mu}{v}. \quad (9)$$

Driver presumption σ is characterized by the traffic changes ahead and is obtained from the gradient of the fundamental diagram. For simplicity, driver presumption can be calculated using (10) [32]

$$\sigma = \frac{\Delta\gamma}{\Delta v}. \quad (10)$$

For smaller changes in velocity, a driver presumption is large, which causes stop and go traffic. Whereas for larger changes, a driver presumption is small, which causes smooth alignment between vehicles and the flow is smooth.

V. CONCLUSION

For sustainable and smart cities of the future, smart mobility has emerged as the primary challenge. With advancement in technology, ITS based smart mobility solutions are being proposed. In this regard, real-life traffic flow parameters are a fundamental building block for proposing ITS based solutions. In existing literature, varying solutions have been proposed for traffic flow characterization such as intrusive and non-intrusive sensors. However, these solutions are limited by their ability to provide all traffic flow parameters. Furthermore, these solutions have serious performance limitations under congested and heterogeneous traffic conditions. To overcome these limitations, compute vision-based solutions have emerged as the most optimum solution for traffic flow characterization. Its advantages range from their ability to provide a full spectrum of traffic flow parameters under all traffic conditions. As well as the ability to count all kinds of on-road objects such as pedestrians, bicycles, bikes, and animal/human driven carts.

In this work, compute vision-based edge computing solutions are compared and analyzed. Major advantage of edge computing solutions over IoVT solutions is elimination of video streaming bandwidth requirements. A detailed review of edge computing solutions in existing literature has been undertaken. It was concluded that the primary constraint is compute resources of SBCs. Because of which at most only two flow parameters such as count, count/speed or count/classification can be measured. To overcome this limitation, mathematical traffic characterization solutions have been detailed in this work. Using these mathematical equations, already proposed solutions in literature can be optimized to measure the full spectrum of traffic flow parameters such as traffic flow, density, critical density, time\distance headway, traffic sensitivity and driver presumptions.

In future, we plan to develop an edge computing solution for traffic flow characterization using RPi 4. To fully utilize the compute resources of RPi we will use multiprocessing techniques. Using measured traffic count and speed parameters, an attempt will be undertaken to provide all traffic flow parameters.

ACKNOWLEDGMENT

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

FUNDING STATEMENT

The author(s) received no specific funding for this study.

CONFLICTS OF INTEREST

The authors declare they have no conflicts of interest to report regarding the present study.

REFERENCES

[1] A. Zeb, K. S. Khattak, A. Agha, Z. H. Khan, M. A. J. Sethi, and A. N. Khan. "On-board diagnostic (OBD-II) based cyber physical system for road bottlenecks detection," *Journal of Engineering Science and Technology vol.17*, no. 2, p. 0906-0922, 2022.

[2] K. S. Khattak, N. Minallah, Z. Khan, M. A. Khan, A. N. Khan, and S. Ayaz, "Sensing technologies for traffic flow characterization: From heterogeneous traffic perspective," *Journal of Applied Engineering Science vol. 20*, no. 1, p. 29-40, 2022.

[3] M. R. Ullah, K. S. Khattak, Z.H. Khan, M. A. Khan, N. Minallah, and A. N. Khan, "Vehicular traffic simulation software: A systematic comparative analysis," *Pakistan Journal of Engineering and Technology vol. 4*, no. 1, p. 66-78, 2021.

[4] Guerrero-Ibáñez *et al*, "Sensor technologies for intelligent transportation systems," *Sensors vol. 18*, no. 4, p. 1212, 2018.

[5] M. Bernas *et al*, "A survey and comparison of low-cost sensing technologies for road traffic monitoring," *Sensors vol.18*, no. 10, p. 3243, 2018.

[6] J. Handscombe, and H. Q. Yu. "Low-cost and data anonymized city traffic flow data collection to support intelligent traffic system," *Sensors vol. 19*, no. 2, p. 347, 2019.

[7] T. Sorwar *et al*, “Real-time Vehicle monitoring for traffic surveillance and adaptive change detection using Raspberry Pi camera module,” presented at the *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, Dhaka, Bangladesh, pp. 481-484, IEEE, 2017.

[8] Raktrakulthum, Porn-anan, and C. Netramai. “Vehicle classification in congested traffic based on 3D point cloud using SVM and KNN,” presented at the *2017 9th International Conference on Information Technology and Electrical Engineering (ICITEE)*, Phuket, Thailand, pp. 1-6, IEEE, 2017.

[9] Y. Mori et *al*, “Reliable and energy-efficient transmission on the Internet-of-Video-Things,” presented at the *2017 17th International Symposium on Communications and Information Technologies (ISCIT)*, Cairns, QLD, Australia, pp. 1-4, IEEE, 2017.

[10] R. Wiska *et al*, “Vehicle traffic monitoring using single camera and embedded systems,” presented at the *2016 International Conference on Computer Science and Information Systems (ICACSIS)*, Malang, Indonesia, pp. 117-122, IEEE, 2016.

[11] A. Khan, K. S. Khattak, Z. H. Khan, M. A. Khan, and N. Minallah. "Cyber physical system for vehicle counting and emission monitoring." *International Journal of Advanced Computer Research vol. 10*, no. 50, p. 181, 2020.

[12] Espinoza *et al*, “Computer vision classifier and platform for automatic counting: more than cars,” presented at the *2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM)*, Salinas, Ecuador, pp. 1-6, IEEE, 2017.

[13] I. Iszaidy *et al*, “Implementation of raspberry Pi for vehicle tracking and travel time information system: A survey,” presented at the *2016 International Conference on Robotics, Automation and Sciences (ICORAS)*, Melaka, Malaysia, pp. 1-4. IEEE, 2016.

[14] Gregor *et al*, “Design and Implementation of a Counting and Differentiation System for Vehicles through Video Processing,” *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering vol. 10*, no. 10, p. 1771-1778, 2016.

[15] I. Iszaidy *et al*, “Video size comparison for embedded vehicle speed detection & travel time estimation system by using Raspberry Pi,” presented at the *2016 International Conference on Robotics, Automation and Sciences (ICORAS)*, Melaka, Malaysia, pp. 1-4. IEEE, 2016.

[16] S. Riyazhussain *et al*, “Raspberry Pi controlled Traffic Density monitoring system,” presented at the *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, Chennai, 2016, pp. 1178-1181, doi: 10.1109/WiSPNET.2016.7566322.

[17] S. Bhusari *et al*, “Traffic control system using Raspberry-pi,” *Global Journal of Advanced Engineering Technologies vol. 4*, no. 4, p. 413-415, 2015.

[18] M. Anandhalli, and V. P. Baligar, “A novel approach in real-time vehicle detection and tracking using Raspberry Pi,” *Alexandria engineering journal vol. 57*, no. 3, p. 1597-1607, 2018.

[19] M. T. Rashid, and M. S. Yasar, “Hybrid Traffic Sensor Technology Based on FCD and Image Processing,”

[20] S. Balakrishna, M. Thirumaran,” Semantic Interoperable Traffic Management Framework for IoT Smart City Applications,” *EAI Endorsed Transactions on Internet of Things 4*, 155482, 10.4108/eai.11-9-2018.155482.

[21] M. Iqbal *et al*, “Classification and Counting Using Raspberry Pi,” *Journal of Engineering and Applied Sciences vol. 12*, no. 8, p. 2207-2213, 2017.

[22] A. Suryatali, and V. B. Dharmadhikari, “Computer vision-based vehicle detection for toll collection system using embedded Linux,” presented at the *2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015]*, Nagercoil, India, pp. 1-7. IEEE, 2015.

[23] M. Sundararajan, “Counting and classification of highway vehicles by using raspberry Pi,” *International Journal of Pure and Applied Mathematics vol. 118*, no. 18, p. 193-201, 2018.

[24] R. McQueen, “Detection and Speed Estimation of Vehicles Using Resource Constrained Embedded Devices,” (2018).

[25] A. Jiménez, V. García-Díaz, and J. Anzola, “Design of a System for Vehicle Traffic Estimation for Applications on IoT,” presented at the *Proceedings of the 4th Multidisciplinary International Social Networks Conference (MISNC '17)*. Association for Computing Machinery, New York, NY, USA, Article 15, 1–6, https://doi.org/10.1145/3092090.3092103

[26] M. Umer, Z.H. Khan, W. Imran, S. Saeed, K.S. Khattak, K. Azam, M.A. Khan, “Behavioral analysis of LWR model under different equilibrium velocity distribution,” *Pakistan Journal of Science vol. 71*, no. 4, p. 194-199, 2019.

[27] A. Iftikhar, Z. H. Khan, T. A. Gulliver, K. S. Khattak, M. A. Khan, M. Ali, N. Minallah, “Macroscopic Traffic Flow Characterization at Bottlenecks,” *Civil Engineering Journal 6, no. 7* (2020): 1227-1242.

[28] Z. H. Khan, T. A. Gulliver, W. Imran, K. S. Khattak, A. B. Altamimi, and A. Qazi, "A macroscopic traffic model based on relaxation time." *Alexandria Engineering Journal* vol. 61, no. 1, p. 585-596, 2022.

[29] Z. H. Khan, T. A. Gulliver, and K. S. Khattak, "A Novel Macroscopic Traffic Model based on Distance Headway," *Civil Engineering Journal* vol. 7, p. 32-40, 2021.

[30] Z.H. Khan, W. Imran, S. Azeem , K. S. Khattak , T. A. Gulliver, M. S. Aslam. “A Macroscopic Traffic Model based on Driver Reaction and Traffic Stimuli,” Applied Sciences vol. 9, no. 14, p. 2848, 2019.

[31] Z. H. Khan and T. A. Gulliver and K. Azam and K. S. Khattak. “Macroscopic Model on Driver Physiological and Psychological Behavior at changes in Traffic,” *Journal of Engineering & Applied Sciences vol. 38*, no. 2, 2019.

[32] Z. H. Khan, W. Imran, S. Azeem, K. S. Khattak, T. A. Gulliver, M. S. Aslam, “A Macroscopic Traffic Model Based on Reaction Velocity,” *Iranian Journal of Science and Technology, Transactions of Civil Engineering (2019)*: 1-12.

[33] Z. H. Khan, “Traffic Modelling for Intelligent Transportation Systems,” PhD dissertation, Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC, Canada, 2016.

[34] Z. H. Khan *et al*, “A macroscopic traffic model based on weather conditions,” *Chinese Physics B vol. 27*, no. 7, p. 070202, 2018.