Check for
updates

A vision-based real-time traffic flow monitoring system for road intersections

Jahongir Azimjonov[1] · Ahmet Özmen[2] ⓘ · Metin Varan[3]

Abstract

In this study, a vision based real-time traffic flow monitoring system has been developed to extract statistics passes through the intersections. A novel object tracking and data association algorithms have been developed using the bounding-box properties to estimate the vehicle trajectories. Then, rich traffic flow information such as directional and total counting, instantaneous and average speed of vehicles are calculated from the predicted trajectories. During the study, various parameters that affect the accuracy of vision based systems are examined such as camera locations and angles that may cause occlusion or illusion problems. In the last part, sample video streams are processed using both Kalman filter and new centroid-based algorithm for comparative study. The results show that the new algorithm performs 9.18% better than Kalman filter approach in general.

Keywords Image based traffic flow monitoring · Vehicle detection · Vehicle tracking · Data association · Deep neural network

1 Introduction

Collecting and processing real-time traffic flow data is an essential component of dynamic traffic flow control systems. Traffic flow data can be collected from various types of sensors

✉ Ahmet Özmen
ozmen@sakarya.edu.tr

Jahongir Azimjonov
jahongir@chungbuk.ac.kr

Metin Varan
mvaran@subu.edu.tr

[1] School of Information and Communication Engineering, Chungbuk National University, Andijan State University, 28644, Cheongju-si, South Korea

[2] Department of Software Engineering, Sakarya University, Serdivan, Sakarya, 54054, Turkey

[3] Department of Electrical and Electronics Engineering, Sakarya University of Applied Sciences, Serdivan, Sakarya, 54187, Turkey

or cameras installed on roads within a region of interest. Accurate traffic predictions extensively depend on precise and timely collected traffic data. Because, the traffic coordination is driven by appropriate phase switching of traffic lights and routing information on variable message signs via prediction feedbacks.

As a result of the developments in image processing technologies, the use of low-cost cameras to obtain traffic data has become quite common recently. Radar, Lidar, and RFID sensors are the other devices for traffic data gathering [1]. These sensors are usually installed on roads for speed detection or vehicle counting and require less pre-processing to obtain digital data comparing to cameras. On the other hand, digital video cameras can be used both roads and intersections, and the data can be collected by wired or air-based network [2]. Many kinds of traffic and environmental information can be extracted from traffic data collected via multiple cameras by the help of novel image processing algorithms; such as object detection, object tracking, accident detection, speed detection, lane changes, registration plate identification, etc. These tasks can be performed with high success rates even on highways, where vehicles go very fast.

Early solutions for the automatic traffic control systems were not able to extract all this information in real-time due to the lack of both hardware and software tools. Recent advancements make video-based systems possible to use in online traffic monitoring and management systems. For example, Graphical Processing Units (GPU) with thousands of cores can process multiple video streams in parallel and produce expected results in real-time. Similarly, new artificial intelligence algorithms and methods including computer vision (CV), deep learning (DL), and machine learning (ML) extract desired information from live-stream videos. DL methods, which have recently been widely used in almost every field, have been the subject of many scientific studies, while at the same time, they have provided some practical outputs that previously seemed impossible in our normal lives. The prediction of Covid-19 cases, analysis of MR-images and developing image-based medical diagnosis systems can be given as examples to these studies [3–5].

Vehicle detection (identifying the type and location of a vehicle in an image or a certain frame of a video), and vehicle tracking (associating the detected vehicles to their correct peer through frames) topics have been enticing researchers of computer vision and image processing due to technological improvements in the area. Partial or full occlusion of objects, sensor or camera shaking, distinct weather conditions such as rain, snow, and wind, and extremely high or low-quality pictures complicates the vehicle detection, tracking and data association processes.

Considering these challenging problems, the research problems of the study are itemized as follows:

- Create a new image processing method to filter the noise from incoming video streams caused by camera shaking or improper light conditions.
- Find and adapt a new object detection method to extract and distinguish vehicles from incoming video frames.
- Develop a novel object tracking algorithm for vehicles crossing the intersection for collecting a high precision statistical information. The algorithm should not fail in case of occlusion.

Despite the aforementioned problems, the numbers, types, directions and speeds of vehicles passing through an intersection can be determined with acceptable accuracy by low-cost cameras and image processing without depending on the time and the weather conditions. In this paper, we propose a novel real-time traffic monitoring system based on a dynamic

traffic flow analyzer. The novelty of the proposed system comes from a vehicle tracking and data association algorithm which provides better solutions to the aforementioned issues.

In the study, YOLOv3 (You Only Look Once) is customized to create an object detection model by training our specific data. Then, the vehicle detection output is utilized to track and associate objects across successive frames. The Euclidean distance is used to measure the similarities between the detected objects and the best-matched peers are associated with each other. All the associated data of vehicle types and trajectories are stored in JSON format in the system for further use.

2 Related work

Intelligent Transportation Systems (ITS) have become one of the crucial parts of the smart or smart sustainable city concept due to its ability to solve potential problems such as traffic congestion, fuel waste, carbon emission, time delays and time lost on the traffic jams. "Advanced Traffic Management System(ATM)" or "Traffic Flow Monitoring (TFM)" is one of the most important components of ITS and these systems generally rely on real-time data. However, traditional data collection methods are no longer suitable for dynamic traffic flow control systems or simulation tools because they either suffer from the accuracy or large delays. A healthy real-time the traffic management system needs the collecting of data both from roads and intersections within accurate time intervals. It is much easier to collect data from roads using various sensors compared to intersections. But, it is very difficult to collect and process online data from multi-lane and multi-way intersections. However, development and deployment of computing devices such as high-performance hardware, camera sensors, recent advances in image processing and artificial intelligence has accelerated researches in this area.

Choosing correct sensor devices for real-time data collection is an important task for online traffic monitoring. There are various types of sensors in the market to harvest raw traffic data such as Blue-tooth, low energy radios, Wi-Fi technologies, GPS/GSM/GPRS, location proofs, Road Side Units (RSUs), camera, RADAR, LIDAR, LOOP, RFID, CCTV, and infrared or thermal cameras. However, the sensors other than camera have four main disadvantages including high installation costs, traffic disruption during installation, maintenance, and repairs. For these reasons, a vision-based system is proposed in this study that collects stream-data from installed cameras at intersections.

The other important issue is to locating the sensor in optimal positions to gather clean data at intersections or roads. There are plenty of research studies that report incorrect sensor (camera) placement causes occlusion and illusion. Incorrectly placed cameras cannot provide a wide field-of-view which is vital for traffic monitoring at intersections. Some experiments have been conducted in this study to show the effects of improper camera placement.

The next step is processing the stream data coming from the cameras in real-time to detect the objects (vehicles) and track them to generate statistical information. There are research studies about traffic flow monitoring that use classical and deep learning algorithms for object detection and object tracking. The challenging topics of traffic flow monitoring are high-performance requirements for vehicle detection, improperly trained vehicle detection models, and lack of robust vehicle tracking algorithms in terms of both theory and implementation. Table 1 shows existing studies and their objectives in the literature.

Table 1 Studies on vision based Intelligent Transportation Systems (ITS) that involve object detection and object tracking

Ref.	Objective of the study
Higuchi et al. [6]	This research focuses on pedestrian path forecasting in crowd for tracking the crowd
	from individual location information generated using mobile phone. Cell phone used as sensor and cryptology techniques are used to anonymize the data.
Zhao et al. [7]	This study is about vehicles tracking in closed environments, such as parking areas, using a mobile phone inertia sensor. Sequential Monte Carlo framework is used to represent vehicle states probabilistically.
Thiagarajan et al. [8]	In this work, a real-time transit tracking is studied by smart phone generated data.
Lee et al. [9]	Smartphone GPS/GPRS technology based vehicle tracking and its efficiency are studied.
Sivaraman and Trivedi [10]	This work is a literature survey on vision based vehicle detection, tracking and
	behavior understanding.
Guerrero-Ibáñez et al. [11]	In this work, available sensor technologies and their integration with the trans-
	portation infrastructure to achieve a sustainable ITS is discussed.
Zhou et al. [12], Ahmad et al. [13]	These papers present research findings on the effectiveness of wireless sensor systems and real-time communication technologies.
Geetha and Cicilia [14]	The use of IoT technologies such as GPS, RFID and controllers to reduce traffic
	congestion in ITS is discussed.
Ding and Huang [15]	A new method is presented and evaluated for spatial-temporal trajectories of
	network constrained moving objects in traffic flow analysis.
Zhang et al. [16]	This paper presents a survey of vision-based data-driven ITS, discussing the functionality of its key components and some deployment issues associated with it.
Kanhere et al. [17]	In this work, visually monitoring a highway is studied when the camera is relatively low to the ground and on the side of the road.
El Mokaddem et al. [18]	This is a review paper that presents current challenges and success factors of ITS
	applications in developing countries.
Datondji et al. [19]	This paper is a survey that addresses the issues related to vehicle monitoring with cameras at road intersections.

2.1 Vision-based vehicle detection studies

There are two different approaches in the literature for vehicle detection: machine learning-based and deep learning-based vehicle detection algorithms. Machine learning-based vehicle detection methods are built on handcrafted features and rules. In other interpretations, these methods usually work based on user-defined rules that manually extracted from vehicle features such as appearance-based (shape and disparity) and motion-based (pixel and blob movement) [10, 20].

Machine learning approaches The appearance-based methods generally use intensity and color like image features. Some example methods which use specific image features obtained from the appearance are HOG, Gabor, Haar-like, SURF and edge, clustering 3D points and v-disparity, clustering in the disparity space features [21]. Detection algorithms that classify vehicles based on appearance and shapes perform three well known processes: 1) background subtraction, 2) blob detection and recording the locations, 3) feature extraction using the blobs and their positions [10, 22, 23]. The extracted features become input for the classification algorithms in which proper identification tags are attached to their meta-data based on vehicle types. Although these approaches work well for off-line videos in the constant background and good weather conditions, they become short against rapid background changes, distinct light conditions, shape, and color problems [23].

In literature, a few motion based methods have been developed to overcome these issues. Dynamic background modeling, optical flow, occupancy grid, and optimal segmentation can be given as examples to these studies [24, 25]. The motion-based detection algorithms use the moving blobs or pixels in the scene to become robust against different light conditions [25]. However various occlusions different light conditions and background changes make it difficult to have an optimal feature descriptor which perfectly describes all kinds of vehicles [20].

Deep learning approaches Aside from machine learning approaches, new deep learning methods have also been studied to improve accurate vehicle detection solutions.Researchers have developed many algorithms using the three major models: Artificial Neural Networks (ANN), Deep Neural Networks (DNN), and Convolutional Neural Networks (CNN) [20]. The convolutional neural network architecture is the most widely preferred model due to its high-accuracy detection and classification properties [26]. There are many completed and ongoing research studies that use CNN in its origin effectively. Some of these architectures can be listed as Region-based CNN (RCNN), Faster RCNN (FRCNN), Single Shot Detector (SSD), Residual Networks (ResNet), Retina Networks (RetinaNet), and You Only Look Once (YOLO) [27]. Although the detection and classification accuracies of ResNet, RetinaNet,SSD, RCNN and FRCNN algorithms are slightly higher, YOLO is usually preferred in real-time traffic monitoring because of its high-speed detection. For example, YOLOv3 can process a maximum of 150 frame per second (FPS) while the other methods perform only 8 FPS at maximum [28]. Additionally, YOLOv3 gives good enough accuracy when it is trained with the dataset images obtained from the specific target such as an intersection or a region. Due to its high-speed detection, and sufficient accuracy in real-time, it is the only option for real-time traffic flow monitoring applications.

2.2 Vehicle tracking and data association

In vision-based systems, vehicle tracking means object re-identification and association of the objects with best-matched peers through consecutive frames [10]. Pixel, blob, or shape features of the objects or bounding-box information have been used to track objects frame by frame. Optical flow and geometric constraints, template matching, feature-based tracking methods have also utilized pixel, blob, or shape features to trace the vehicles [29]. In particular, Kalman filter and particle filter tracking algorithms use bounding-box coordinates [30, 31]. Although these two methods are good in tracking, they are not very successful in real-time traffic monitoring applications due to the poor response time fast changes of light conditions, occlusion, and their complex parameter structure [1].

However, in a recent study, a new tracking method is introduced with bounding box properties by adapting the Kalman filter and DeepSORT for vehicle tracking [32]. This study is tested with *AI City Challenge 2021* dataset, and successful results were obtained. The performance of the approach was measured using a formula contributing 70% from the accuracy and 30% from the execution time.

As seen from the literature, there is an increasing demand for robust vehicle detection, tracking, and data association algorithms in the field to overcome aforementioned issues. Both novel vehicle tracking and data association algorithms are the main contributions of this study for real-time intersection traffic monitoring. These algorithms improve existing vision-based real-time traffic flow monitoring systems by outputting high-accuracy results using high-performance YOLOv3 vehicle detection module. In the study, a new vehicle training/testing dataset has been prepared and used for training to improve YOLOv3 performance.

3 Proposed approach

In this study, a vision-based real-time traffic flow monitoring system has been developed that utilizes both deep learning and machine learning techniques effectively for vehicle
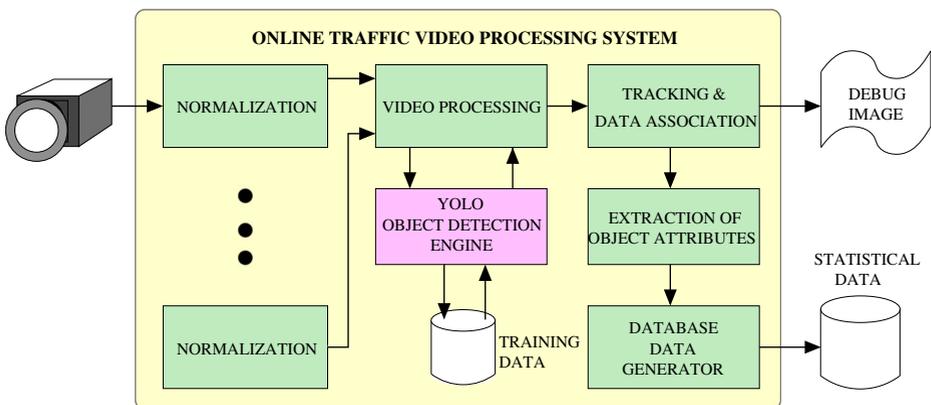

Fig. 1 Block diagram of online traffic video processing system. The input video stream is normalized by reducing the number of frames, and then sent to the object detection unit. After detecting the vehicles and their types in the frame, the tracking algorithm predicts/plots the trajectory and saves them in the database

detection and tracking at intersections. The proposed system can extract all the type of traffic flow information dynamically and cost-effectively. This information include directions, duration, total and categorical counting of vehicles in various circumstances such as rainy, snowy weather conditions, camera shaking, day/night time, partial or full occlusion and different illusion occurrence cases. The block diagram of the system is shown in Fig. 1.

Video streams are first parsed as frames and then the frame rate is dropped to a predefined ratio. This ratio depends on the traffic flow speed. Each selected frame is subjected to size reduction (downscaling) without losing much information. Then the frames are sent to the real-time object detection module which is YOLOv3. YOLOv3 is implemented using Darknet which is an open-source neural network framework written in C and Compute Unified Device Architecture (CUDA). However, its modules can be called from Python programs using Cython wrapper that converts the Python code into C/C++.

The video processing module developed using Python takes the downscaled frames and detects the vehicles first (if there is any) and classifies them with a certain confidence level. The module returns the predicted vehicle types, bounding-boxes of the vehicles, and the confidence level of predictions. Then, the detected vehicles are linked between frames to determine the vehicle's trajectories. Association of moving objects among independent frames can be called object tracking which is challenging work. In this study, a new centroid-based tracking algorithm has been developed to determine the trajectories.

3.1 Placement of cameras

In vision-based traffic analysis, choosing correct locations for the cameras have a key role in clean data acquisition which directly affects the success rate of the overall system. Incorrect placement may cause illusion due to sunlight or occlusion because of overlapped vehicles in the scene. For instance, low-angel video shots cause serious occlusion problems and "bird's eye view" with high-angle video capturing causes object classification issues. The best camera placement should provide a wide field-of-view of intersection or road zones where all vehicles are clearly identifiable. Figure 2 illustrates the proper and improper case pictures that were captured with the camera installed at the intersections in the different countries.

Pre-processing the image data includes collecting video streams, frame reduction and frame downscaling. Three criteria were taken into account for the system to give the desired response in time: 1) the task is real-time data processing, 2) the average speed of the vehicles at intersections, 3) the response time of the object detection algorithm which is a sub part of the overall traffic flow monitoring system.

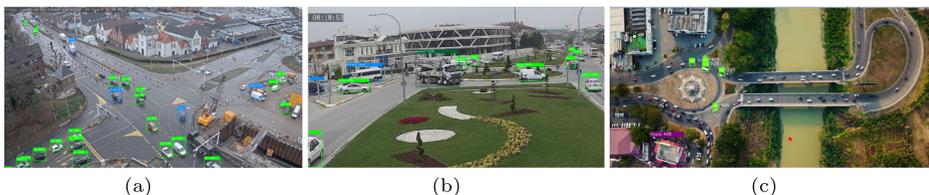(a)   (b)   (c)

Fig. 2 Various camera placements at different intersections: a) a proper camera placement with high-angle, b) a low-angle camera placement causes occlusion, c) bird's-eye view camera placement which causes object detection failure

3.2 Step by step the algorithm

- **Stream to frame conversion:** Stream-to-frame conversion is done using OpenCV library [33]. This library parses the incoming live video-stream into frames at every second and puts them into the BGR array (inverse order of RGB color codes). The array is a three-dimensional matrix in which each element corresponds to a pixel in a frame.
- **Frame reduction:** In literature 20 fps is found acceptable for real-time traffic monitoring; however, this study focuses on intersections only, and the average speed is slower than usual roadway traffic [22]. Generally, incoming video streams are 30 fps, however reducing it to 15 fps in the tests did not cause any information loss in terms of trajectory estimation. For this reason, 30 fps live streams that come from the cameras are immediately reduced to 15 fps throughout the study.
- **Down-scaling the selected frames:** Down-scaling or resizing an image can be done in many different ways. In this study, a function from OpenCV library (cv2.resize()) is used for down-scaling the frames to 608x608 for this study. This function uses the interpolation technique among neighboring pixels to reduce the size [33]. After resizing a frame, it is converted into a CType object which is necessary for YOLOv3.
- **Object detection in each frame:** In this study, YOLOv3 is used as a library for object detection. It runs on Graphical Processing Unit (GPU) card and called for every selected frame from the main program. In the beginning, some performance experiments were conducted and it was observed that YOLOv3 responds to requests over 30 fps when it runs on a GPU. Vehicle detection module returns three important parameters: 1) the detected vehicle type, 2) the bounding-box of the vehicle and 3) the confidence level of detection.
- **Data association and vehicle-trajectory estimation:** The centroids of the bounding-boxes are used for the vehicle-trajectory association. Sometimes a vehicle may disappear in some frames due to occlusion for example, and it gets difficult to associate it. Hence, the centroid values are also saved in an array to use for trajectory prediction. This dynamic array holds centroid values of 150 frames. When a vehicle disappears from the scene, then a new position for the disappeared vehicle is predicted using extrapolation technique. The association is set immediately if a new vehicle stays in the extrapolated trajectory (showing up of a disappeared vehicle). If a vehicle has not shown up in 150 frames around the predicted trajectory, then it is not tracked anymore. The flow-chart of the tracking and data association algorithm is shown in Fig. 3.
- **Statistical traffic flow information extraction:** After the trajectories of the vehicles are extracted from the video-stream data, the entry-exit timestamps and from which direction they came and in which direction they are going are resolved and recorded to the database. The throughput of the intersection and the other statistical information can then be calculated. For example, the number of vehicles passing through the junction in a time interval can be analyzed based on vehicle types such as car, bus, truck or speeds.

3.3 Object detection by YOLOv3

YOLOv3 is an object detection algorithm that is implemented with Fully Convolutional Neural Network (FCNN). When (nxn) sized input image is fed through the FCNN, the output becomes (mxm) bounding-box prediction with class probabilities. Although YOLOv3 accepts lower resolutions, 608x608 pixel image sizes are used both for training and test studies without affecting the prediction time. YOLOv3 divides the input image into SxS

```
                    ┌──────────────┐
                    │    START     │
                    └──────────────┘
                           │
                           ▼
                   ╱ Video stream ╱
                           │
                           ▼
                  ┌─────────────────┐
                  │  Normalization  │
                  └─────────────────┘
                           │
                           ▼
           No         ◇ frame ◇
                           │ Yes
                           ▼
                  ┌─────────────────┐
                  │ object <-- YOLO │
                  └─────────────────┘
                           │
                           ▼
       No    ◇ is new object ◇   Yes
              ◇  in lists?   ◇
              │
              ▼
        ◇ Prediction ◇  Yes
              │ No
              ▼
     ┌──────────────┐      ┌──────────────────┐
     │Create new list│     │  Update tracks & │
     └──────────────┘      │associate the obj │
                           └──────────────────┘
                                    │
                                    ▼
                           ┌──────────────────┐
                           │  Update database │
                           └──────────────────┘
```
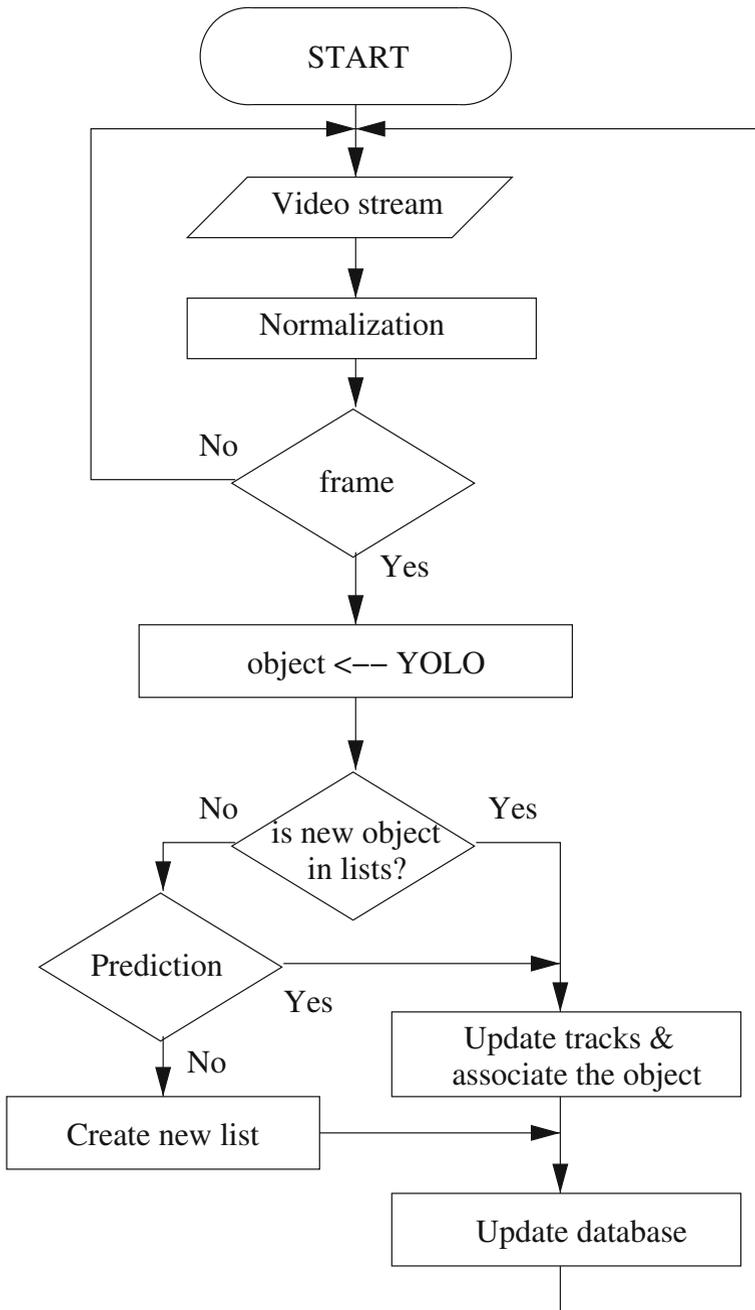
Fig. 3 Flow chart of centroid-based vehicle trajectory extraction algorithm. After normalizing the number of frames an object detection algorithm is used for vehicle recognition. The routes of the vehicles are determined by using the centroid based tracking algorithm

grid cells, then it runs a Convolutional Neural Network (CNN) on the image. In the last step, it thresholds the resulting detections by the model's confidence [28].

YOLOv3 has two different FCNN-based detection networks in its architecture which both rely on supervised learning methodology. The first architecture contains 24 layers, and the second network composes of 106 layers. The more layers produce better accuracy but require more processing power to get the output at the same time. Preparing a training data set plays a decisive role to build a successful vehicle detection model. The network must be trained in advance with the possible input set to get better predictions. The YOLOv3 network was trained with many intersection and vehicle images before it is used for real tests. It takes months to create a data set for an hour video (more than tens of thousands of images), so it is a time-consuming job. While searching an optimal way to make a robust data set for field-specific vehicle detection, a general-purpose pre-trained weights model that has been trained using the Microsoft-COCO data set is found suitable for our problem [34]. This pre-trained model is effectively used to create a new case-specific data set for vehicle detection problems at intersections.

To create a case-specific vehicle recognition model, the video stream is reduced to 1 fps and fed to the YOLOv3 in training mode. In this way, a newly trained model is created with the first-stage data set. The process is repeated with different sets until the value of the loss function drops under 10%. (1) shows the loss function used during the training. At the end of the training, an average of 80% of vehicle detection accuracy is achieved with 25 fps. During the study, training is repeated two times with different data sets: the first time Microsoft-COCO (this comes as default pre-trained), and the second time our case-specific data set is used.

$$L_{loc} = \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj}[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2]$$

$$L_{cls} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} (1_{ij}^{obj} + \lambda_{noobj}(1 - 1_{ij}^{obj}))(C_{ij} - \hat{C}_{ij})^2 + \sum_{i=0}^{S^2} \sum_{c \in C} 1_i^{obj}(p_i(c) - \hat{p}_i(c))^2$$

$$L = L_{loc} + L_{cls} \tag{1}$$

where x_i and y_i are the centroid, and w_i and h_i are the width and height of the bounding-box of the vehicle i in the dataset. Predicted i-th vehicle centroid location represented by $\hat{x}_i$, $\hat{y}_i$ and bounding-box sizes $\hat{w}_i$, $\hat{h}_i$. 1_{ij}^{obj} is an indicator function of whether the cell i contains an object. Similarly, 1_i^{obj} shows whether the j-th bounding-box of the cell i is *responsible* for the object prediction. C_{ij} is the predicted confidence score of cell i, which is calculated by *probability of containing an object × intersection over union (IoU)*, for this study $IoU > C$ is the set of all classes. $p_i(c)$ is the conditional probability of whether the cell i contains an object of class $c \in C$. $\hat{p}_i(c)$ is the predicted conditional class probability.

To improve vehicle detection accuracy as well as the performance of the system, a separate dataset for each intersection was prepared using an annotation tool. The training process was continued for 24 hours until the value of the loss function (error function) fell below 10%. Figure 4 shows the loss functions for the test intersections. A new training is applied for each intersection because varying factors such as camera position, light conditions. YOLOv3 uses binary cross-entropy loss for each label. This also reduces the computation complexity by avoiding the soft-max function.

The training parameters (hyperparameters) of the YOLOv3 model is shown in Table 2. The overall fps rate for vehicle detection was between 30-40, and the new YOLOv3 based
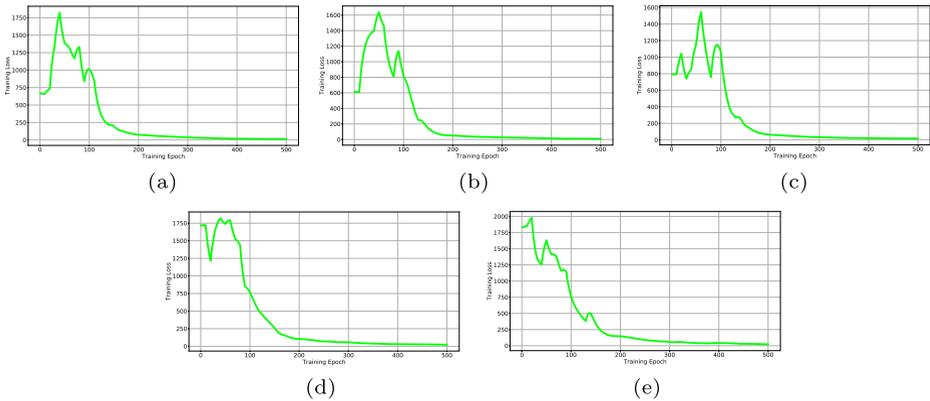
Fig. 4 The loss graphs obtained from training logs: a) Netherlands, b) Sweden, c) Turkey, d) Japan, e) Ukraine

vehicle detection system can process an image with 1920x1080x3 dimensions in around 33 - 25 milliseconds.

3.4 A centroid-based vehicle tracking and data association

Extracting the trajectories of vehicles from a video scene for the particular time interval provides highly valuable traffic flow information. The traffic condition of any desired region can be monitored and managed in a city using this online information. There are several pixel-based and bounding-box-point-based object tracking methods for finding the trajectories. These include optical flow and Kalman filters respectively. However, these approaches are not suitable to solve the tracking problem in intersections efficiently [31]. Optical flow and state based Kalman filter algorithm are not good for real-time applications since they start failing with increasing fps rate. Besides, Kalman filter tracker is not fully compatible with YOLOv3 detector due to YOLOv3's unstable object localization nature.

All tracking algorithms trace the same detected objects through successive image frames in slightly altered positions. Kalman filter tracking algorithm uses this dislocation information of bounding-boxes to estimate next location using the motion velocity ($v = v_0 + at$), coordinate ($x = x_0 + v_0 t + at^2/2, \quad y = y_0 + v_0 t + at^2/2$) formulas, and several other

Table 2 Main training hyperparameters used for YOLOv3 network

Hyperparameter	Value
Batch, subdivisions	24, 8
(width$X$height), channels	(608X608), 3
Momentum, decay	0.9, 0.0005
Learning_rate	0.001
Policy	steps
Steps	400000, 450000
Scales	0.1, 0.1
Burn_in	1000

state parameters. This makes Kalman filter algorithm very sensitive to any minor object dislocation. In addition to this, Kalman filter predicts the new position using just one previous frame parameters. Prediction a new position considering only one previous state parameters, in turn, creates uncertainty and ambiguity.

The new centroid based vehicle tracking algorithm includes two main stages: tracking and data association. The tracking part is responsible for registering newly detected objects, updating the existing objects, and de-registering the disappeared ones. Object centroid (x, y) points and Euclidean distance among the centroids are used to track the vehicles, hence the algorithm is called "Centroid Tracker".

The centroid based tracker algorithm receives bounding-boxes of detected vehicles and the confidence values of the classes from YOLOv3, and the distances between the centroids are calculated. If the distance between the objects of two consecutive frames are at between 0 and 100-pixel units, then the newly detected object is associated with the previous object id. This is called the association of a new centroid point to an existing trajectory by updating its status. If a correct pair cannot be established within 150 frames, then old trajectory points are all de-registered from the system considering it as a disappeared object. The object labels and trajectory points are stored in the JSON format for each frame. At the end, these JSON data are used to extract the trajectories of all objects using a simple data query algorithm.

4 Experimental setup

The test videos were selected based on different camera angles, installation altitudes, time of the day and weather conditions to extract weak and strong sites of the proposed method. They include alive intersection videos shot at different countries including the Netherlands, Sweden, Turkey, Japan and Ukraine. Sample images for the case study regions are illustrated in Fig. 5, and detailed information is given in Table 3 for accessing the videos.

In this study, two GPU based graphics cards (NVIDIA GeForce GTX 1050 TI 4 GB GDDR5 and NVIDIA GeForce RTX 2080 TI 11 GB GDDR6) are used on the host machine
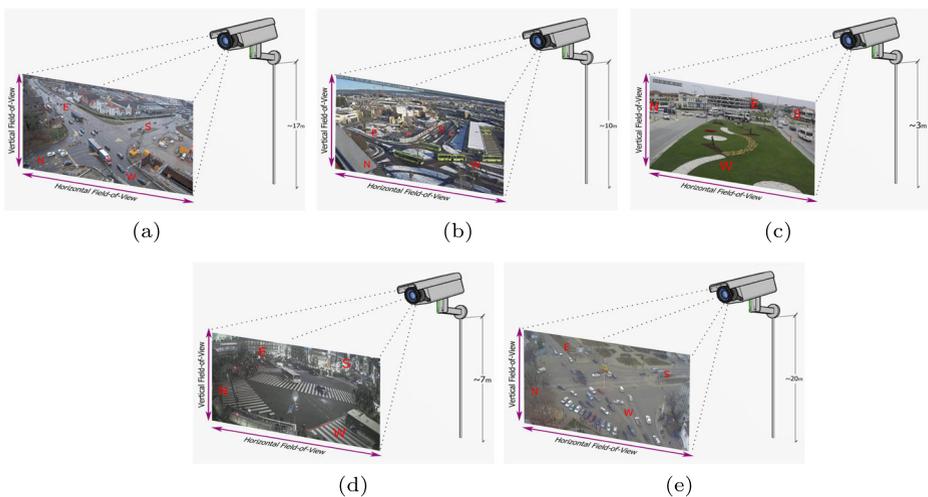

Fig. 5 The test videos examined in the study with different camera placements at different intersections: a) Netherlands, b) Sweden, c) Turkey, d) Japan, e) Ukraine

Table 3 The test videos can be obtained from YouTube

Ref.	The reason for selection	Time (min)	Country	YouTube-Link (https://www.youtube.com/)
a	Afternoon, high camera location	4	Nederlands	watch?v=R8t5F_nLYKU
b	Afternoon, distant video shooting	6	Sweden	watch?v=ariYGLo9T44
c	Morning time, low angle camera	3	Turkey	watch?v=9ZTLpKDzwNo
d	Evening time, close view	4	Japan	watch?v=kkryf5Z8UHw
e	Afternoon, high angle camera	9	Ukraine	watch?v=CthbczCzYys

Table shows selection criteria, video length in minutes, country and YouTube access links

with Intel I9 CPU, 32 GB RAM and 2 TB SSD hard drive. The host machine is loaded with Linux operating system and required software packages for development.

Accuracy calculations The statistical data produced by Kalman filter and the centroid based algorithm are recorded into a table separately. Then the ground truth data obtained by counting the vehicles by watching the video images carefully. The accuracy of the system is calculated using the ground truth data and measured values using (2) [35, 36]. However, different formulas must be used when the denominator value becomes zero in the accuracy calculations. These cases and the used formulas are explained in the following list:

- No vehicle cross the intersection and the system measured this correctly ($G = 0$, $M = 0$).
- Observed vehicle count is zero ($G = 0$), but the measured value is non-zero ($M \neq 0$).
- Observed vehicle count is non-zero ($G \neq 0$), but the measured value is zero ($M = 0$).
- Both observed vehicle count and measured values are non-zero ($G \neq 0$, $M \neq 0$).

$$A(G, M) = \begin{cases} 100 & if\ G = M\ or\ G = M = 0, \\ \frac{\min(G,M)}{\max(G,M)} & \begin{cases} let\ G = G + 1 & if\ G = 0 \\ let\ M = M + 1 & if\ M = 0, \\ if\ G, M > 0 \end{cases} \end{cases} \tag{2}$$

Two different average formulas were used to calculate the total accuracy values for the vehicle types and the regions. A simple average formula used for vehicle types is given in (3).

$$A_{avg} = \frac{\sum_{d=1}^{n} A_d}{n} \tag{3}$$

Since the number of vehicles in each class does not close to each other, the weighted average formula is used for accuracy calculations shown in (4).

$$A_{total} = w_{car} \times A_{avg-c} + w_{bus} \times A_{avg-b} + w_{truck} \times A_{avg-t} \tag{4}$$

where w_{car} is the weight for car type, and it is obtained by $w_{car} = \frac{n_{car}}{total\_vehicle\_count}$. The other weights are found in the similar way.

A form-based tool has been developed that outputs the real-time traffic flow statistics. Figure 6 shows a screen shot of the tool interface. Current video is shown in the middle and the statistical information on the left, the lower part shows the average speed and the average crossing time of the vehicle classes.
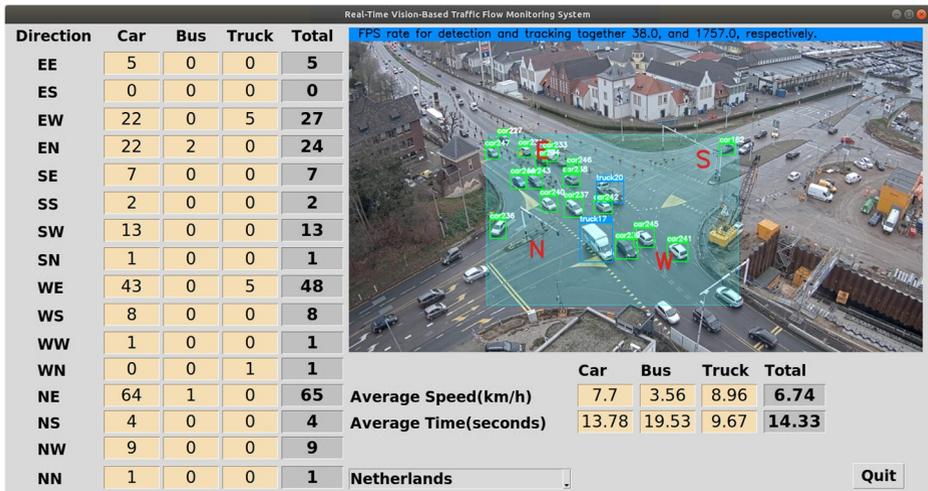
Fig. 6 A screen shot of form-based online tool that shows real-time traffic flow statistics acquired from the video stream

The comparative results between the centroid based tracking algorithm and Kalman filter approach are shown in Table 5. The results are presented as ground truth and measured quantities for both Kalman filter and the Centroid based method.

5 Results and discussion

This section presents discussion about comparative results between proposed system and Kalman filter approach on predicting vehicle directions, duration, total and categorical counting of vehicles in various circumstances such as rainy, snowy weather conditions, camera shaking, day/night time, partial or full occlusion.

In general, the centroid-based approach yielded better results than Kalman method. However, if a vehicle enters and leaves an intersection close points, then the centroid based algorithm is not doing well because it loses the intact in the middle of the intersection. This

Table 4 Statistical information about the intersections such as average speed and intersection crossing time generated by the online tool

Average Speed (km/h) and Duration (sec.)

Region	Car		Bus		Truck	
	km/h	sec.	km/h	sec.	km/h	sec.
Netherlands	6.5	18.2	3.9	18.7	7.4	19.6
Sweden	4.6	32.7	4.1	48.7	9.7	36.2
Turkey	6.2	38.3	5.8	37.5	4.3	33.0
Japan	12.0	29.9	14.6	29.9	6.6	34.3
Ukraine	6.4	79.0	5.1	79.0	6.1	70.9

is the case when a vehicle makes a U-turn at the intersection which can be seen from Table 5, in E-E, W-W, and N-N direction lines. However, Kalman method keeps the direction as a state information, and it does not lose the vehicle in such cases.

However, when there are more than 30 vehicles in a frame, Kalman filter has not been able to handle the same number of frames with centroid based approach. During the experiments, it is observed that Kalman based tracker cannot handle more than 5 fps when the vehicle count in the frame goes beyond 30 (Table 4).

It is seen that the height and angle of the camera are very effective on accuracy. If they are not set properly, the accuracy of the system decreases due to intense illusion or occlusion problems (see Fig. 4). Table 5 shows poor results for Video from Turkey due to the camera position is not set properly. On the other hand, better results are obtained in Japan since the angle of the camera is adjusted properly although the height of the camera is not very high.

It is also observed that the detection and association algorithm play crucial role to obtain high success rate. For instance, in the case of detection failure for some reason, such as heavy rain, snow, camera shaking, occlusion or illusion; a prediction process is carried out for the disappeared vehicles. Mispredictions reduces the overall accuracy of the system.

Interestingly, truck-type vehicles cross through the intersection faster in the Netherlands than other vehicle types (see Table 4). In the detailed examination of the video recording, it was seen that the trucks mostly flow in the direction of E-W and W-E, and since this direction is mainline the vehicles generally cross the junction faster than the other directions (see Fig. 5). Hence, the trucks are seen as the fastest crossing class although they are larger/heavier vehicles.

The intersection in Ukraine is slightly more irregular and difficult in terms of information extraction than others. Although the camera height and angle are quite proper, the tracking of the vehicles that cross the intersection is a challenging problem. Hence, both Kalman and centroid based algorithms did not perform as desired compared to other intersections.

Experimental results (qualitative and/or quantitative) illustrate that the proposed system can extract all the types of traffic flow information dynamically and cost-effectively.

6 Conclusion

In this work, a vision-based real-time traffic flow monitoring problem has been studied for road intersections. YOLOv3 is implemented as a model for vehicle detection module, and a new dataset with the vehicle images from the case study regions have been created for training. A novel vehicle tracking and data association algorithm has been developed using the vehicle bounding-box properties to estimate the vehicle trajectories. In the proposed solution, vehicle trajectories are estimated using a novel centroid-based vehicle tracking and data association algorithm. This algorithm works based on tracking the shifts of the bounding-box centroid points of the vehicles through the consecutive image frames. Euclidean distance is used to validate the estimated next position of the vehicles in the consecutive frames.

Experimental studies were designed to reveal the weak and strong sides of the developed system. In the five selected test cases, the proposed centroid based tracking method gave minimum 6.6%, maximum 12.9% and average 9.18% better results than Kalman method. The centroid based method also outperforms Kalman tracking by processing up to 30 frame per seconds while Kalman filter cannot go beyond 8 fps. In the study, it was also examined

Table 5 Comparative results of the experiments

	Car			Acc.(%)		Bus			Acc.(%)		Truck			Acc.(%)		Tot.Acc.(%)	
	G	K	C	K	C	G	K	C	K	C	G	K	C	K	C	K	C
Dir.	**Netherlands**																
E-E	0	1	6	50	14.3	0	0	0	100	100	0	0	0	100	100		
E-S	0	0	0	100	100	0	0	0	100	100	0	0	0	100	100		
E-W	37	12	29	32.4	78.4	0	0	0	100	100	7	6	6	85.7	85.7		
E-N	24	12	24	50	100	2	2	2	100	100	0	0	0	100	100		
S-S	0	0	0	100	100	0	0	0	100	100	0	0	0	100	100		
S-W	13	13	13	100	100	0	0	0	100	100	0	0	0	100	100		
S-N	1	1	1	100	100	0	0	0	100	100	0	0	0	100	100		
S-E	7	5	7	71.4	100	0	0	0	100	100	0	0	0	100	100	**77.0**	**84.5**
W-W	0	0	3	100	25	0	0	0	100	100	0	0	0	100	100		
W-N	0	0	0	100	100	0	0	0	100	100	1	1	1	100	100		
W-E	61	39	54	63.9	88.5	0	0	0	100	100	5	4	6	80	83.3		
W-S	7	0	8	12.5	87.5	0	0	0	100	100	0	0	0	100	100		
N-N	0	0	1	100	50	0	0	0	100	100	0	0	0	100	100		
N-E	65	56	64	86.2	98.5	1	1	1	100	100	0	0	0	100	100		
N-S	4	4	4	100	100	0	0	0	100	100	0	0	0	100	100		
N-W	16	4	17	25	94.1	0	0	0	100	100	0	0	0	100	100		
	Avg.:			**75.5**	**83.5**	Avg.:			**100**	**100**	Avg.:			**97.9**	**98.1**		
Dir.	**Sweden**																
E-E	0	0	0	100	100	0	0	0	100	100	1	0	1	50	100		
E-S	2	2	2	100	100	0	0	0	100	100	0	0	0	100	100		
E-W	9	8	9	88.9	100	0	0	0	100	100	0	0	0	100	100		
E-N	3	0	3	25	100	0	0	0	100	100	0	0	0	100	100		
S-S	0	0	0	100	100	1	0	1	50	100	0	0	0	100	100		
S-W	21	20	16	95.2	76.2	3	1	2	33.3	66.7	0	0	0	100	100		
S-N	22	22	21	100	95.5	2	1	3	50	66.7	1	1	1	100	100		
S-E	2	3	2	66.7	100	0	1	0	50	100	0	0	0	100	100	**81.7**	**88.4**
W-W	1	0	3	50	33.3	2	0	3	33.3	66.7	0	0	0	100	100		
W-N	23	23	22	100	95.7	0	0	0	100	100	0	0	0	100	100		
W-E	6	7	7	85.7	85.7	0	0	0	100	100	0	0	0	100	100		
W-S	21	0	23	4.5	91.3	2	0	2	33.3	100	1	0	1	50	100		
N-N	0	0	2	100	33.3	0	0	0	100	100	0	0	0	100	100		
N-E	0	0	0	100	100	0	0	0	100	100	0	0	0	100	100		
N-S	32	29	30	90.6	93.8	2	2	2	100	100	0	0	0	100	100		
N-W	13	13	13	100	100	0	0	1	100	100	0	0	0	100	100		
	Avg.:			**81.7**	**87.8**	Avg.:			**78.1**	**93.8**	Avg.:			**96.9**	**100**		

Table 5 (continued)

	Car			Acc.(%)		Bus			Acc.(%)		Truck			Acc.(%)		Tot.Acc.(%)	
	G	K	C	K	C	G	K	C	K	C	G	K	C	K	C	K	C
Dir.	**Turkey**																
E-E	0	8	2	11.1	33.3	0	0	0	100	100	0	2	0	33.3	100		
E-S	3	9	3	33.3	100	0	0	0	100	100	0	3	0	25	100		
E-W	2	2	3	100	66.7	10	1	7	10	70	13	0	7	7.1	53.8		
E-N	8	0	5	11.1	62.5	0	0	2	100	33.3	2	0	3	33.3	66.7		
S-S	0	0	0	100	100	0	0	0	100	100	0	0	1	100	50		
S-W	3	3	2	100	66.7	0	0	0	100	100	0	1	0	50	100		
S-N	26	1	19	3.8	73.1	0	0	0	100	100	11	9	7	81.8	63.6		
S-E	15	19	13	78.9	86.7	0	2	0	33.3	100	3	6	3	50	100	**65.8**	**78.7**
W-W	0	0	0	100	100	0	0	0	100	100	0	0	0	100	100		
W-N	1	0	2	50	50	0	0	0	100	100	1	0	0	50	50		
W-E	2	0	5	33.3	40	4	0	7	20	57.1	5	0	6	16.7	83.3		
W-S	2	3	2	66.7	100	0	0	0	100	100	0	0	0	100	100		
N-N	0	0	0	100	100	0	0	0	100	100	0	0	0	100	100		
N-E	23	19	25	82.6	92	0	0	1	100	100	3	2	0	66.7	25		
N-S	1	1	1	100	100	0	0	0	100	50	8	1	5	12.5	62.5		
N-W	3	3	2	66.7	66.7	0	0	0	100	100	0	0	0	100	100		
	Avg.:			**66.8**	**77.4**	Avg.:			**85.2**	**88.2**	Avg.:			**57.9**	**78.4**		
Dir.	**Japan**																
E-E	0	0	0	100	100	0	0	0	100	100	0	0	0	100	100		
E-S	17	16	17	94.1	100	0	1	0	50	100	0	0	0	100	100		
E-W	12	10	12	83.3	100	1	0	1	50	100	0	0	0	100	100		
E-N	0	0	0	100	100	0	0	0	100	100	0	0	0	100	100		
S-S	0	0	5	100	16.7	0	0	0	100	100	0	0	0	100	100		
S-W	8	6	7	75	87.5	0	0	0	100	100	0	0	0	100	100		
S-N	22	18	21	81.8	95.5	1	0	0	50	50	1	1	1	100	100		
S-E	2	2	2	100	100	0	0	0	100	100	0	0	0	100	100	**81.9**	**88.5**
W-W	0	0	0	100	100	0	0	0	100	100	0	0	0	100	100		
W-N	2	0	1	33.3	50	1	1	0	100	50	0	0	0	100	100		
W-E	17	15	17	88.2	100	2	0	2	33.3	100	0	0	0	100	100		
W-S	0	0	0	100	100	0	0	0	100	100	0	0	0	100	100		
N-N	0	3	0	25	100	0	0	0	100	100	0	0	0	100	100		
N-E	4	4	6	100	66.7	0	0	0	100	100	0	0	0	100	100		
N-S	19	14	15	73.7	78.9	0	0	0	100	100	0	0	0	100	100		
N-W	1	2	2	50	50	0	0	0	100	100	0	0	0	100	100		
	Avg.:			**81.5**	**88.1**	Avg.:			**86.5**	**93.8**	Avg.:			**100**	**100**		

Table 5 (continued)

	Car			Acc.(%)		Bus			Acc.(%)		Truck			Acc.(%)		Tot.Acc.(%)	
	G	K	C	K	C	G	K	C	K	C	G	K	C	K	C	K	C
Dir.	**Ukraine**																
E-E	4	16	6	25	66.7	0	0	2	100	33.3	0	1	2	50	33.3		
E-S	29	30	27	96.7	93.1	7	7	6	100	85.7	0	0	0	100	100		
E-W	89	76	88	85.4	98.9	5	4	4	80	80	2	4	3	50	66.7		
E-N	23	33	25	69.7	92	0	1	1	50	50	0	0	0	100	100		
S-S	10	5	19	50	52.6	0	0	0	100	100	0	0	1	100	100		
S-W	24	15	21	62.5	87.5	0	0	0	100	100	3	1	3	33.3	100		
S-N	101	81	83	80.2	82.2	2	1	1	50	50	2	0	0	33.3	33.3		
S-E	58	68	65	85.3	89.2	3	3	3	100	100	2	3	2	66.7	100	**70.7**	**82.9**
W-W	7	11	8	63.6	87.5	0	0	0	100	100	0	0	1	100	50		
W-N	16	16	13	100	81.2	1	1	1	100	100	0	0	0	100	100		
W-E	68	54	69	79.4	98.6	4	4	4	100	100	3	4	4	75	75		
W-S	13	1	16	7.7	81.2	0	0	0	100	100	0	0	0	100	100		
N-N	3	3	6	100	50	0	0	0	100	100	0	0	0	100	100		
N-E	15	20	16	75	93.8	0	0	0	100	100	2	1	0	50	33.3		
N-S	70	63	63	90	90	2	3	3	66.7	66.7	0	0	0	100	100		
N-W	44	32	36	72.7	81.8	3	2	2	66.7	66.7	0	0	0	100	100		
	Avg.:			**69.7**	**82.9**	Avg.:			**88.3**	**83.3**	Avg.:			**78.6**	**80.7**		

In the table, direction initials are used to represent the trajectory of the crossing vehicles; for example, a vehicle's path that comes from east inbound and leaves to south outbound is represented as "E-S". The other abbreviations are: "G" for Ground truth, "K" for Kalman filter and "C" for Centroid based tracking algorithms

how the camera position affects the results, and the most appropriate and correct places are reported with the examples.

A form-based software tool has been developed that utilizes the new algorithm to extract traffic flow information such as directional, classified total counting, instantaneous or average speed and crossing time of the intersection. This tool can also be used to generate real-time traffic data from on-site video streams. As future work, more vehicle features obtained from video images will be used to increase the accuracy of the tracking algorithm.

Data Availability Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

Declarations

Conflict of Interests Each of the authors confirms that this manuscript has not been previously published and is not currently under consideration for publication elsewhere. Additionally, we have no conflicts of interest to disclose.

References

1. Hilbert EE, Rennie PA, Kneidl WA (1980) A sensor for control of arterials and networks. IEEE Trans Veh Technol 29(2):208–215
2. Liu Y, Tian B, Chen S, Zhu F, Wang K (2013) A survey of vision-based vehicle detection and tracking techniques in ITS. In: Proceedings of 2013 IEEE international conference on vehicular electronics and safety, pp 72–77
3. Shoeibi A, Khodatars M, Alizadehsani R, Ghassemi N, Jafari M, Moridian P et al (2020) Automated detection and forecasting of COVID-19 using deep learning techniques: a review. CoRR. arXiv:2007.10785
4. Shoeibi A, Khodatars M, Jafari M, Moridian P, Rezaei M, Alizadehsani R, et al (2021) Applications of deep learning techniques for automated multiple sclerosis detection using magnetic resonance imaging: a review. Comput Biol Med 136:104697. https://doi.org/10.1016/j.compbiomed.2021.104697
5. Khodatars M, Shoeibi A, Ghassemi N, Jafari M, Khadem A, Sadeghi D et al (2020) Deep learning for Neuroimaging-based diagnosis and rehabilitation of autism spectrum disorder: a review. CoRR. arXiv:2007.01285
6. Higuchi T, Martin P, Chakraborty S, Srivastava M (2015) AnonyCast: privacy-preserving location distribution for anonymous crowd tracking systems. In: Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing. UbiComp '15. New York, NY, USA: Association for Computing Machinery, pp 1119–1130
7. Zhao M, Ye T, Gao R, Ye F, Wang Y, Luo G (2015) VeTrack: real time vehicle tracking in Uninstrumented indoor environments. In: Proceedings of the 13th ACM conference on embedded networked sensor systems. SenSys '15. New York, NY, USA: Association for Computing Machinery, pp 99–112
8. Thiagarajan A, Biagioni J, Gerlich T, Eriksson J (2010) Cooperative transit tracking using smart-phones. In: Proceedings of the 8th ACM conference on embedded networked sensor systems. SenSys '10. New York, NY, USA: Association for Computing Machinery, pp 85–98
9. Lee S, Tewolde G, Kwon J (2014) Design and implementation of vehicle tracking system using GPS/GSM/GPRS technology and smartphone application. In: 2014 IEEE world forum on internet of things (WF-IoT), pp 353–358
10. Sivaraman S, Trivedi MM (2013) Looking at vehicles on the road: a survey of vision-based vehicle detection, tracking, and behavior analysis. IEEE Trans Intell Transp Syst 14(4):1773–1795
11. Guerrero-Ibáñez J, Zeadally S, Contreras-Castillo J (2018) Sensor technologies for intelligent transportation systems. Sensors 18(4):1–24
12. Zhou Y, Dey KC, Chowdhury M, Wang K (2017) Process for evaluating the data transfer performance of wireless traffic sensors for real-time intelligent transportation systems applications. IET Intell Transp Syst 11(1):18–27
13. Ahmad F, Basit A, Ahmad H, Mahmud SA, Khan GM, Yousaf FZ (2013) Feasibility of deploying wireless sensor based road side solutions for intelligent transportation systems. In: 2013 international conference on connected vehicles and expo (ICCVE), pp 320–326
14. Geetha S, Cicilia D (2017) IoT enabled intelligent bus transportation system. In: 2017 2nd international conference on communication and electronics systems (ICCES), pp 7–11
15. Ding Z, Huang G (2009) Real-time traffic flow statistical analysis based on network-constrained moving object trajectories. In: Database and expert systems applications. Berlin, Heidelberg: Springer Berlin Heidelberg, pp 173–183
16. Zhang J, Wang F, Wang K, Lin W, Xu X, Chen C (2011) Data-driven intelligent transportation systems: a survey. IEEE Trans Intell Transp Syst 12(4):1624–1639
17. Kanhere NK, Pundlik SJ, Birchfield ST (2005) Vehicle segmentation and tracking from a low-angle off-axis camera. In: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05). vol 2, pp 1152–1157 vol 2
18. El Mokaddem Y, Jawab F, Saâd LE (2019) Intelligent transportations systems : review of current challenges and success factors: the case of developing countries. In: 2019 international colloquium on logistics and supply chain management (LOGISTIQUA), pp 1–6
19. Datondji SRE, Dupuis Y, Subirats P, Vasseur P (2016) A survey of vision-based traffic monitoring of road intersections. IEEE Trans Intell Transp Syst 17(10):2681–2698
20. Zhao Z, Zheng P, Xu S, Wu X. (2019) Object detection with deep learning: a review. IEEE Transactions on Neural Networks and Learning Systems 30(11):3212–3232
21. Peng C, Hirvonen D, Camus T, Southall B (2005) Stereo-based object detection, classifcation, and quantitative evaluation with automotive applications. In: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05) - workshops, pp 62–62

22. Fernández-Sanjurjo M, Bosquet B, Mucientes M, Brea VM (2019) Real-time visual detection and tracking system for traffic monitoring. Eng Appl Artif Intel 85:410–420
23. Mandellos NA, Keramitsoglou I, Kiranoudis CT (2011) A background subtraction algorithm for detecting and tracking vehicles. Expert Syst Appl 38(3):1619–1631
24. Zhu Y, Comaniciu D, Pellkofer M, Koehler T (2006) Reliable detection of overtaking vehicles using robust information fusion. IEEE Trans Intell Transp Syst 7(4):401–414
25. Franke U, Rabe C, Badino H, Gehrig S (2005) 6D-Vision: fusion of stereo and motion for robust environment perception. In: Kropatsch WG, Sablatnig R, Hanbury A (eds) Pattern Recognition. Berlin, Heidelberg: Springer Berlin Heidelberg, pp 216–223
26. LeCun Y, Bengio Y, Hinton G. (2015) Deep learning. Nature 521(7553):436–444
27. Girshick R (2015) Fast R-CNN. In: 2015 IEEE international conference on computer vision (ICCV), pp 1440–1448
28. Redmon J, Farhadi A (2017) YOLO9000: better, faster, stronger. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), pp 6517–6525
29. Zhu Y, Comaniciu D, Ramesh V, Pellkofer M, Koehler T (2005) An integrated framework of vision-based vehicle detection with knowledge fusion. In: IEEE proceedings. intelligent vehicles symposium, 2005, pp 199–204
30. Chang W, Cho C (2008) Real-time side vehicle tracking using parts-based boosting. In: 2008 IEEE international conference on systems, man and cybernetics, pp 3370–3375
31. Sivaraman S, Trivedi MM (2013) Integrated lane and vehicle detection, localization, and tracking: A synergistic approach. IEEE Trans Intell Transp Syst 14(2):906–917
32. Ha SVU, Chung NM, Nguyen TC, Phan HN (2021) Tiny-PIRATE: a tiny model with parallelized intelligence for real-time analysis as a traffic countEr. In: 2021 IEEE/CVF conference on computer vision and pattern recognition workshops (CVPRW), pp 4114–4123
33. Guido G, Gallelli V, Rogano D, Vitale A (2016) Evaluating the accuracy of vehicle tracking data obtained from unmanned aerial vehicles. International Journal of Transportation Science and Technology 5(3):136–151. Unmanned Aerial Vehicles and Remote Sensing
34. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D et al (2014) Microsoft COCO: common objects in context. Lect Notes Comput Sci, pp 740–755
35. Pu M, Mao J, Du Y, Shen Y, Jin C (2019) Road intersection detection based on direction ratio statistics analysis. In: 2019 20th IEEE international conference on mobile data management (MDM), pp 288–297
36. Szczodrak M, Dalka P, Czyżewski A (2010) Performance evaluation of video object tracking algorithm in autonomous surveillance system. In: 2010 2nd international conference on information technology, (2010 ICIT), pp 31–34