Soft Computing (2023) 27:13843–13859
https://doi.org/10.1007/s00500-023-08860-z

A nighttime highway traffic flow monitoring system using vision-based vehicle detection and tracking

Jahongir Azimjonov[1] · Ahmet Özmen[2] · Taehong Kim[1]

Accepted: 14 June 2023 / Published online: 6 July 2023

Abstract

Accurately estimated highway traffic flow info plays a decisive role in dynamic and real-time road management, planning, and preventing frequent/recurring traffic jams, traffic rule violations, and chain/fatal traffic accidents. Traffic flow information is extracted by processing raw camera images via vehicle detection and tracking algorithms. Object detectors including the Yolo, single-shot detector, and EfficientNet algorithms are used for vehicle detection; however, You only look once version 5 (Yolov5) has a clear advantage in terms of real-time performance. Due to this reason, the pre-trained Yolov5 models were utilized in the vehicle detection part, and in the vehicle tracking module, a novel tracker algorithm was developed using vehicle detection features. The performance of the proposed approach was measured by comparing it to the Kalman filter-based tracker. The evaluation results show that the proposed tracking approach outperformed the Kalman filter-based tracker with 5.82% (Buses), 2.24% (Cars), 36.50% (Trucks), and overall 2.58% better traffic counting accuracy for the 12 nighttime case study videos captured from the highways with different horizontal and vertical angle-of-views.

Keywords Nighttime vehicle detection · Nighttime vehicle classification · Nighttime vehicle tracking · Nighttime vehicle trajectory extraction · Nighttime highway traffic flow info extraction

1 Introduction

An essential component of traffic analysis and planning in urban areas is the road participant categorization (classifying vehicles such as bicycles, buses, cars, motorcycles, and trucks) on highways, as well as the estimation of statistical traffic flow information (traffic flow frequency and the proportion of each vehicle by its type and its trajectory data). However, in this contemporary era of expanding technology and population, real-time highway traffic flow monitoring is a difficult problem on urban areas. According to Luque-Baena et al. (2015), improper highway traffic management contributes to frequent traffic bottlenecks/congestion, traffic rule infringements or road ridges, and deadly road accidents. It takes a lot of time, money, and effort to solve this problem using conventional methods such as RADAR, LIDAR, RFID, or LASAR (Khan et al. 2018; Jeon et al. 2016). Human observers visit the area and tally the number of passing vehicles when these sensors are insufficient. Nevertheless, manual traffic data collecting is not a reasonable solution, and these techniques are unable to produce real-time traffic flow data. Additionally, they fall short when it comes to categorizing vehicles or getting data about the number of vehicles by their kind and direction of travel (Zhao et al. 2019).

However, state-of-the-art approaches to artificial intelligence (specifically machine vision techniques) have been utilized in real-time video/image data processing applications. Particularly, vision-based vehicle detection and vehicle tracking algorithms that are based on deep image processing and machine learning techniques are being used in a wide range of application areas of Intelligent Transportation Systems. These systems typically compose of vehicle detection, tracking (associating the best-matching peers in consecutive frames), spatio-temporal trajectory data, and

✉ Taehong Kim
taehongkim@cbnu.ac.kr

Jahongir Azimjonov
jahongir@chungbuk.ac.kr

Ahmet Özmen
ozmen@sakarya.edu.tr

[1] School of Information and Communication Engineering, Chungbuk National University, Chungdae-ro 1, Seowon-Gu, 28644 Cheongju, South Korea

[2] Department of Software Engineering, Sakarya University, Esentepe Campus, 54187 Sakarya, Turkey

estimation of traffic flow information. Traffic flow information includes speed, the categorical or total number of vehicles and vehicle entrances and exits in the specified area, and time period data. All this information is extracted from the vehicle trajectory data by utilizing vehicle recognition and vehicle re-identification approaches (Lu et al. 2020). Therefore, high-performance and accurate vehicle detection and tracker algorithms play an important role in this type of systems (Yang and Pun-Cheng 2018; Feng et al. 2019).

Vehicle detection is an approach for recognizing the label of interest (object class) and re-identifying the targeted objects through successive frames of a video stream. General-purpose object detection techniques such as conventional machine/deep learning methods are widely used for the vehicle recognition tasks. Conventional vision-based vehicle detection approaches include background subtraction and a classification algorithm like support vector machines (SVM), blob extraction via a classifier like K-nearest neighbor (KNN), edge-like feature extraction and a classification algorithm, or other kinds of classical detection techniques based on Scale-Invariant Feature Transform (SIFT) and Speeded Up Robust Features (SURF) utilize manually and hand-crafted feature vectors (Pillai et al. 2021). On the one hand, the aforementioned approaches necessitate in-depth knowledge and expertise to select representative features or feature vectors of a certain object type (Chauhan and Singh 2018). On the other hand, deep neural networks do not have to utilize manually extracted features or feature vectors, which are defined with the strict rules that require a low-level imaging expertise. Besides, they do not require deep knowledge about object content since the deep neural network architectures automatically extract deep and hidden features with user-defined activation functions such as Rectified Logic Unit (ReLU), Sigmoid, and Tanh, and the optimization functions like Adam, AdaDelta or Stochastic Gradient Descent (SGD). So these architectures are capable to extract deep and hidden vehicle features that help to recognize and re-identify objects on a video frame or an input image (Zhao et al. 2019).

Vehicle tracking is re-identifying the detected objects and associating them with their best-matched peers across a series of frames of an input video. In simple terms, all the detected vehicles are associated first; the detections are tracked with the assigned unique IDs in vehicle tracking part. In this way, unique vehicle trajectory data per vehicle are estimated. Trajectory data extraction algorithms use pixel, shape, color, and bounding box (Bbox) information to associate vehicles. Although pixel, shape, and color-based tracking techniques are considered to be reliable for tracking objects through subsequent frames, they are too sluggish for real-time video analysis applications. Even though the vehicle trackers that were built on pixel, shape, and color-like features are thought to be reliable for tracking objects through subsequent frames, they are extremely slow for real-time video analysis applications. On the other hand, vehicle tracking algorithms that are based on the Kalman filter or Particle filter approach simply process the coordinate information of the Bbox data of the recognized vehicles. Hence, these algorithms are a little faster than pixel, shape, or color-based approaches. Nevertheless, when the number of vehicles on a video frame increases, these methods also get failed to run in real time. Sudha and Priyadarshini (2020). As an example, the Kalman filter and particle filter-based tracking approaches struggle on high-speed roads where vehicles go very fast. In addition, if the number of vehicles is above 30 on a certain frame, the performance of these algorithms drops sharply. Moreover, vehicle recognition and re-identifying have been challenging tasks of classical machine-vision research due to the problems such as partial or full occlusion of vehicles, light illusion, camera shaking, extremely high- or low-quality picture, distinct weather conditions including rain, snow and wind, and especially nighttime issues, which can complicate the vehicle detection, tracking, and data association processes, and in some cases, these problems make such systems completely fail (Datondji et al. 2016). Due to these reasons, a robust and real-time tracking algorithm is necessary for the effective and efficient real-time video analysis tools.

General-purpose object detectors including Yolov(1–5), Single Shot Detector, EfficientNet, RetinaNet, Residual Networks (ResNets), Region-based Convolutional Neural Networks (R-CNN), and Faster R-CNN, which are based on CNN architectures, are utilized for real-time vehicle recognition, but, Yolov5 has clear advantages with high-performance and strong vehicle localization functionality. Therefore, the pre-trained Yolov5 models (nano, small, medium, large, X-large) were used for the vehicle detection task. For the vehicle tracking part, a novel Bbox-based and nighttime-adopted vehicle tracker algorithm was developed in this study. Besides, the Kalman filter-based vehicle tracker was implemented to compare the results of the proposed tracker in this paper. The main flow-chart of the system is illustrated in Fig. 1 Then, 10 traffic (vehicle) counting systems were developed using Yolov5 architectures with the Kalman filter-based and the new Bbox-based trackers, see Fig. 2. The evaluation results of the vehicle counting systems show that the proposed tracker algorithm outperformed with overall 2.58% better accuracy than the Kalman filter-based tracker.

2 Related works

Smart cities are built in the basement of several smart applications including intelligent transportation systems, intelligent medicine, smart energy, smart environment, etc. At the same time, intelligent transportation systems have seven sub-application areas such as dynamic traffic flow and road network management, traveler information systems, public

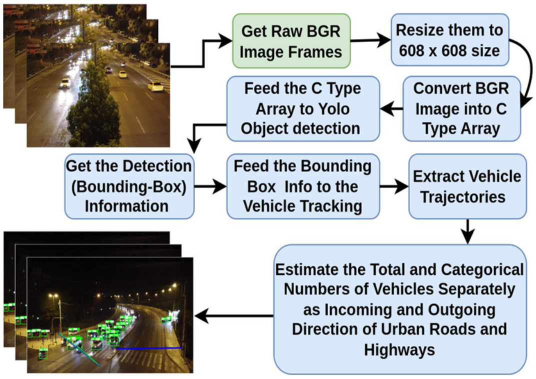
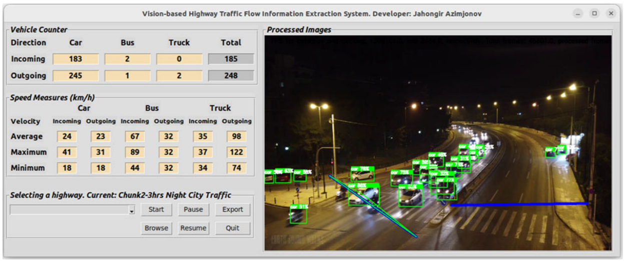

Get Raw BGR Image Frames
Resize them to 608 x 608 size
Convert BGR Image into C Type Array
Feed the C Type Array to Yolo Object detection
Get the Detection (Bounding-Box) Information
Feed the Bounding Box Info to the Vehicle Tracking
Extract Vehicle Trajectories
Estimate the Total and Categorical Numbers of Vehicles Separately as Incoming and Outgoing Direction of Urban Roads and Highways

Fig. 1 The illustration of the whole process of the main system


Vision-based Highway Traffic Flow Information Extraction System. Developer: Jahongir Azimjonov

Vehicle Counter

Direction	Car	Bus	Truck	Total
Incoming	183	2	0	185
Outgoing	245	1	2	248

Speed Measures (km/h)

	Car		Bus		Truck	
Velocity	Incoming	Outgoing	Incoming	Outgoing	Incoming	Outgoing
Average	24	23	67	32	35	98
Maximum	41	31	89	32	37	122
Minimum	18	18	44	32	34	74

Selecting a highway. Current: Chunk2-3hrs Night City Traffic

Fig. 2 The interface of the real-time traffic flow monitoring system

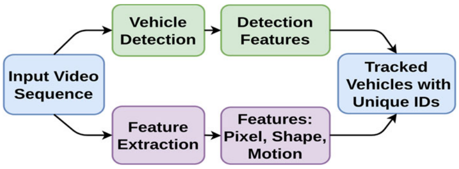


Fig. 3 Tracking by detection (up) and detection-free (down)

transport systems, commercial vehicle applications, vehicle safety applications maintenance & construction management applications, emergency management, and archived data management. Dynamic traffic flow and road network management are the most critical and crucial application field of the seven applications since it has a huge impact on reducing or preventing traffic congestion, managing roads & intersections in real-time, planning, and preventing frequent/recurring traffic jams, traffic rules violations, and chain/fatal traffic accidents (Azimjonov and Özmen 2021). Online traffic flow monitoring and control systems are developed using traffic flow information such as total and categorical traffic counts, traffic speed, and other traffic environmental variables. The system fully relies on vehicle recognition and extraction of vehicle trajectories. Recognizing vehicles and extraction of their trajectory data generally uses vehicle detection and tracking approaches (Song et al. 2018). Vehicle detection methods are divided into two main types, namely traditional machine learning and modern deep neural network-based approaches. Vehicle tracking approaches are also categorized into two groups such as trackers that utilize bounding box features (tracking by detection) and appearance-based tracking (tracking by detection free), see Fig. 3. In the following, we describe vehicle detection and tracking approaches in detail (Azimjonov and Özmen 2022).

Vehicle detection. Vehicle recognition (localization and classification) techniques, which utilizes classical machine learning algorithms, completely depend on hand-crafted and manually determined vehicle features/features vectors. The feature vector is extracted from the appearance or motion features of pixels in an image (Sivaraman and Trivedi 2013). The detection algorithms, which detect and classify vehicles based on the content (appearance and shape characteristics) of vehicle images on a video frame, mainly compose of three stages. In the first stage, background modeling (background subtraction) is performed by removing the background (Cabido et al. 2012). In the second stage, after background removal, the blobs and their locations are determined. Next, the desired (user-defined) features are extracted using the detected blobs and their locations (Fernández-Sanjurjo et al. 2019; Kavukcuoglu et al. 2010). In the third stage, the detected blobs and extracted features are fed to classification algorithms to form robust rules to identify vehicle labels. These algorithms perform well for offline videos with the constant background and good weather conditions. However, they are very weak against fast background changes, different weather conditions, color and shape problems; specifically, these algorithms fall short in processing of nighttime videos (Mandellos et al. 2011). Motion-based algorithms such as dynamic background modeling (Zhu et al. 2006), optical flow (Franke et al. 2005), occupancy grid (Badino et al. 2007), and tracking pixels (Erbs et al. 2011) have been developed to overcome the issues, which image/object content-based (appearance, shape, and disparity) algorithms faced in the past. The motion-based vehicle detecting relies on moving blobs/pixels, and they are robust against different weather conditions. However, due to the variety of partial or full occlusions, slowness, appearances, illumination, and background changes, it is difficult to manually design a robust feature descriptor to perfectly describe all kinds of vehicles (Rathore et al. 2018). Especially, it is very difficult to manually extract robust features from nighttime video images, which describes traffic objects features well. But, thanks to the recently emerged deep learning architectures, robust and accurate algorithms on object recognition have been developed to address the problems that the conventional machine vision techniques had in the past. These techniques include three major models including artificial neural networks (ANNs), deep neural networks (DNN), and convolutional neural networks (CNNs). These methods detect vehicles using automatically extracted deep and hidden features. CNN is a widely utilized and most representative model among the deep learning architectures (LeCun et al. 2015) due to its high-accuracy classification ability (Girshick et al. 2014). There have been conducted many research studies based on CNN architecture. For example, Region-based (RCNN) (Girshick 2015), Faster Region-based (FRCNN) (Ren et al. 2017), Single Shot Detector (SSD) (Liu et al. 2016), ResNet, RetinaNet, and Yolo (You Only Look Once, v1, v2, v3) (Redmon et al. 2016; Redmon and Farhadi 2017, 2018) are the most successful and widely utilized approaches. The detection and classification accuracy of the RCNN, FRCNN, SSD, ResNet, RetinaNet are slightly higher than Yolo; however, in real-time time video analysis tools such as traffic monitoring, only Yolo can be used because of its high-speed detection. The maximum fps rate of other methods is 8, whereas a minimum of 20 FPS and a maximum of 150 FPS can be obtained with Yolo (Redmon and Farhadi 2018). Due to its high-speed detection, and accuracy in real time, strong localization functionality unlike EfficientNet/Det, it is the only option for real-time traffic flow monitoring applications. However, the vehicle classification accuracy of this algorithm is below 57% since Yolo performs the object localization and classification tasks on an input image with a single neural network only one time. Hence, it causes such a low level of classification accuracy. Fifty-seven percentage of classification accuracy is not enough for traffic flow extraction systems on urban roads/highways. Thus, the classification accuracy of the Yolo-based vehicle detection algorithms should be improved by combing a robust classifier layer into the algorithm.

Vehicle tracking Extracting the trajectories of vehicle (*X* and *Y* trajectory vectors) from a roadway video scene for the particular time interval or in real time provides highly valuable traffic flow info such as vehicle speed, motion direction, total and categorical numbers, duration between entry and exit points of vehicles and other desired parameters. With these traffic data, one can easily monitor and manage the traffic condition of any desired region in a city or urban area. Although there exist several vehicle tracking methods which uses the pixel and bounding-box features such as optical flow, particle filter and Kalman filters, respectively, any of these approaches are not sufficient to solve the vehicle tracking problem accurately in real time on highways (Khalkhali et al. 2020). Optical flow and other pixel-based methods (Nguyen and Brilakis 2018) are very slow, which is inappropriate for real-time applications, while the Kalman filter algorithm struggles with many false predicted points. Additionally, the tracking performance of the Kalman filter-based trackers declines when the numbers of objects increase in a particular frame (Liu et al. 2019; Song et al. 2019). Besides, while testing Kalman filter with Yolo, it was observed that these two algorithms are not appropriate with each other, because Yolo does not determine the exact location of objects, but it estimates the bounding box information via minor shifts. In other words, the Yolo algorithm detects the same stationary or moving objects through successive frames in the slightly different positions from where they are exactly). The Kalman Filter tracking algorithm uses the motion information of Bboxes to predict new points such as motion velocity ($v_x = v_{0_x} + at$), $v_y = v_{0_y} + at$) and coordinate ($x = x_0 + v_{0_x}t + at^2/2$, $y = y_0+v_{0_y}t+at^2/2$), and several motion and direction state parameters (Xiao et al. 2020). These parameters make the Kalman filter too sensitive against minor movements/shifts of Bbox points. Additionally, the Kalman filter-based tracker predicts the new position using just one previous frame parameters when the observed object disappear from the certain frames. Predicting the new position of the disappeared object considering the state parameters from only one previous frame creates an uncertainty issue, which leads to failure of the tracker (Yang et al. 2019). Yolo's unstable object position estimation nature, and the Kalman filter-based tracker's motion-sensitive nature, and the uncertainty issue in vehicle tracking and data association make these two algorithms incompatible at some level. Consequently, they affect the accuracy and performance on the overall system negatively (Kanagamalliga and Vasuki 2018). Therefore, a robust vehicle tracking and data association algorithm, which is compatible with Yolo's unstable object localization nature, are a need for a highway traffic flow monitoring and management system.

The contribution of the study: (i) fine-tuning and decreasing the number of classes by three from 90 classes, (ii) creating a testbed dataset to measure and evaluate the proposed traffic counter system with the 12 nighttime case study videos captured from the highways using different cameras with horizontal and vertical angle-of-views, (iii) developing real-time and high-accuracy vehicle detection system by Yolov5 models, (iv) developing a novel Bbox-based vehicle tracking algorithm and the implementing of the Kalman filter-based tracking algorithm. Next, applying the vehicle detectors and trackers to the vehicle trajectories extraction task, (v) estimating the categorical and total numbers of vehicles on the twelve nighttime case study highway videos using the extracted vehicle trajectories, and comparing the accuracy of the vehicle counting systems.

3 Methodology

The main system was built on four main parts, namely vehicle detection, vehicle tracking, vehicle trajectory data extraction, and traffic flow information estimation from the extracted trajectory data. In the vehicle detection part, Yolov5 (general-purpose object detector) was utilized by fine-tuning the algorithm. In the vehicle tracking part, a novel vehicle tracker algorithm was developed using vehicle detection features by adopting them to nighttime videos. Additionally, the Kalman filter-based tracking algorithm was also implemented to compare the performance of the proposed tracker. In the third module, a total of 10 vehicle trajectory data extraction systems were created using five models (Nano, Small, Medium, Large, and X-Large weight models) of Yolov5, and both trackers by pairing with them. For instance, the first vehicle trajectory extraction system was formed using the Yolov5 Nano model with the Bbox-based tracker, and the second one is created using the Yolov5 Nano model with the Kalman filter-based tracker. Other 8 trajectory extraction systems are developed in the same way.

3.1 Vehicle detection with Yolov5 models

Yolov5 models are fine-tuned and adopted to three vehicle classes. The fine-tuned parameters:

- *lr0:* 0.009—initial learning rate (SGD=1E-2, Adam=1E-3)
- *lrf:* 0.19—final OneCycleLR learning rate (lr0 * lrf)
- *momentum:* 0.977—SGD momentum/Adam beta1
- *weight_decay:* 0.00053—optimizer weight decay 5e-4
- *warmup_epochs:* 3.0—warmup epochs (fractions ok)
- *warmup_momentum:* 0.81—warmup initial momentum
- *warmup_bias_lr:* 0.11—warmup initial bias lr
- *box:* 0.057—box loss gain, *cls:* 0.57—cls loss gain
- *cls_pw:* 1.0—cls BCELoss positive_weight
- *obj:* 1.0—obj loss gain (scale with pixels)
- *obj_pw:* 1.0—obj BCELoss positive_weight
- *iou_t:* 0.45—IoU training threshold
- *anchor_t:* 4.0—anchor-multiple threshold
- *anchors:* 3—anchors per output layer (0 to ignore)
- *fl_gamma:* 0.0—focal loss gamma (efficientDet default gamma=1.5)
- *hsv_h:* 0.0155—image HSV-Hue augmentation (fraction)
- *hsv_s:* 0.79—image HSV-Saturation augmentation (fraction)
- *hsv_v:* 0.43—image HSV-Value augmentation (fraction)
- *degrees:* 0.0—image rotation (± deg)
- *translate:* 0.17—image translation (± fraction)
- *scale:* 0.57—image scale (± gain)
- *shear:* 0.0—image shear (± deg)
- *perspective:* 0.0—image perspective (± fraction), range 0−0.001
- *flipud:* 0.01—image flip up-down (probability)
- *fliplr:* 0.51—image flip left-right (probability)
- *mosaic:* 1.0—image mosaic (probability)
- *mixup:* 0.01—image mixup (probability)
- *copy_paste:* 0.0—segment copy-paste (probability)
- *angle:* 0.15, *saturation:* 1.83, *exposure:* 1.85, *hue:* 0.5
- *nc:* 3 [number of classes]
- *depth_multiple:* 0.33 model depth multiple
- *width_multiple:* 0.50 layer channel multiple
- *anchors:* [10,13, 16,30, 33,23]-P3/8, [30,61, 62,45, 59, 119]-P4/16, [116,90, 156,198, 373,326]-P5/32
- *classes names:* 0—bus, 1—car, 2—truck
- *filters:* filters=(num_classes + 5)*3.
  So for 3 classes, filters = 24.
  max_batches = 6000, classes=3, filters=24

$$
\begin{aligned}
L_{\text{loc}} &= \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{obj}} [(x_i - \widehat{x}_i)^2 + (y_i - \widehat{y}_i)^2 \\
&\quad + (\sqrt{w_i} - \sqrt{\widehat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\widehat{h}_i})^2] \\
L_{\text{cls}} &= \sum_{i=0}^{S^2} \sum_{j=0}^{B} \big(1_{ij}^{\text{obj}} + \lambda_{\text{noobj}} (1 - 1_{ij}^{\text{obj}})\big)(C_{ij} - \widehat{C}_{ij})^2 \\
&\quad + \sum_{i=0}^{S^2} \sum_{c \in C} 1_{i}^{\text{obj}} (p_i(c) - \widehat{p}_i(c))^2. \\
L &= L_{\text{loc}} + L_{\text{cls}}
\end{aligned}
\tag{1}
$$

where L_{loc} and L_{cls} are the loss functions for the localization and classification tasks, respectively. 1_{i}^{obj}: An indicator function of whether the cell i contains an object. 1_{i}^{obj}: It indicates whether the j-th Bbox of the cell i is 'responsible' for the object prediction. C_{ij}: The confidence score of cell i, $Pr(containing\ an\ object) * IoU(pred, truth)$, IoU > 0.5 for this study. $\widehat{C_{ij}}$: The predicted confidence score. C: The set of all classes. $p_i(c)$: The conditional probability of whether cell i contains an object of class $c \in C$. $\widehat{p_i}(c)$: The predicted conditional class probability.

The Yolov5-based vehicle detection architectures have been developed by utilizing the aforementioned hyper-parameters. The Yolo-based vehicle detection algorithm sees the vehicle recognition task as a single regression problem, and it performs localization and classification tasks in a single network. So, the Yolov5-based vehicle detection system predicts the Bbox (localization), and label (classification) information of objects. At its core, this equation is utilized as a loss function, see Eq. 1.

Yolov5 is a robust object detector by nature; nevertheless, it may detect several duplicate bounding boxes for the same vehicle on any frame of any video. The maximum number of duplicated detections in a certain object depends on the number of classes. For instance, if there are 80 class labels in a trained model, so maximum of 80 so-called detections (Bboxes) can be detected in a particular location for a certain object. These all detections can be considered as 80 objects if they are not handled accurately. In the vehicle tracking part, a unique ID is assigned to each object. If 80 detections come to the vehicle tracking module, it assigns 80 IDs and gives

80 tracks for just one object. Consequently, in the vehicle counting part, one object can be counted as 80 objects. This case is a big issue for the development of an accurate vehicle counting system. The duplicate-detection problem was addressed using our misdetection filtering algorithm. In this algorithm, we performed the bounding boxes matching in the same location of an input image using intersection over the union. All Bboxes are checked against overlapping via IoU, then vehicle detections (bounding boxes) of two objects those IoU value greater than 0.90 are ranked based on their prediction confidence levels. The one with the highest confidence level was kept, other object with a lower confidence level was removed for all detection of each video frame, the algorithm has been given in Algorithm 1.

Algorithm 1 A duplicate detection filtering algorithm to remove misdetections of a vehicle in a certain frame.

```
Require: $len(Bboxes) \geq 0$
Ensure: $New\ set\ of\ Bboxes,\ after\ duplicate\ removal.$
  $DelIdList \leftarrow \{\}$                         ▷ Bbox IDs to be deleted.
  $NewDets \leftarrow \{\}$                           ▷ New set of Bboxes.
  $n \leftarrow len(Bboxes)$
  $k \leftarrow 0$                                    ▷ Indexes of bboxes to be deleted.
  for $i = 0$ to $n$ do
    $det1 = Bbox_i$                                   ▷ det1=[$label_i, confidence_i, bbox_i$]
    for $j = i + 1$ to $n$ do
      $det2 = Bbox_j$                                 ▷ det2=[$label_j, confidence_j, bbox_j$]
      $bb\_iou = bbox\_IoU(det1, det2)$
      if $bb\_iou \geq 0.9$ then                      ▷ $0.8 \leq threshold \leq 1.0$
        $det1 = Bbox_i$
        if $det1_1 \geq det2_1$ then
          if $j\ not\ in\ DelIdList$ then
            $DelIdList^{k+1} = DelIdList^k \cup \{j\}$
          end if
        end if
      end if
    end for
  end for
  $m \leftarrow 0$
  for $id = 0$ to $n$ do
    if $id\ not\ in\ DelIdList$ then
      $NewDets^{m+1} = NewDets^m \cup \{Bboxes_{id}\}$
    end if
  end for
```

3.2 Detection feature-based vehicle tracker approach

The proposed vehicle tracker approach uses the output of vehicle detection module. Vehicle detection module gets a video frame as an input, processes it. And it produces detection results. Next, the detection results are purified with the duplicate detection check algorithm. It removes the duplicate bounding boxes and produces output as in Eq. 2. Here, cl—class, conf—confidence and $X1$ and $Y1$—left and top, and $X2$ and $Y2$—right and bottom points.

$$Bboxes_i = \begin{array}{l}[(cl_i, conf_i, X1_i, Y1_i, X2_i, Y2_i),\\ (cl_{i+1}, conf_{i+1}, X1_{i+1}, Y1_{i+1}, X2_{i+1}, Y2_{i+1}),\\ \ldots,\\ (cl_n, conf_n, X1_n, Y1_n, X2_n, Y2_n)]\end{array} \tag{2}$$

The bounding box info as detection feature will be an input of the proposed tracker algorithm. The vehicle tracker generates time-dependent vehicle trajectory using Bbox information (X, Y, Width, Height) of two consecutive frames ($frame_i$ and $frame_j$). The tracker calculates all distances among all object pairs from both frames. It keeps all distance values to the 2-dimensional array D_{ij}, see Eq. 3. The structure of the array D_{ij} is given as Eq. 4.

$$Dist_{ij} = \sqrt{(Bboxes_i - Bboxes_j)^2} \tag{3}$$

The items by the rows of the Dist array correspond to the unique IDs of the vehicles on the ith (previous) frame and the items by the column of the Dist array indicate the newly assigned IDs of the same vehicles on the $(i + 1)$th frame on an input video, see Eq. 4.

$$D_{ij} = \begin{pmatrix} d_{11} & \ldots & d_{1n} \\ . & . & . \\ d_{m1} & \ldots & d_{mn} \end{pmatrix} \tag{4}$$

where d_{11} is the distance between the first vehicles of the previous and new frames, and m and n are the total numbers of vehicles in these successive frames, respectively.

The Dist array is sorted by it rows and minimized by its columns, Eq. 5. The tracker algorithm updates the states of the vehicles by assigning new IDs based on the sorted row and minimized column IDs of the previous states of vehicles (Bbox coordinates) when the sorted and minimized distance value of a certain vehicle resides within the manually defined circled area. The radius of the user-defined circular area is 60 pixel units for this study. The radius of the circular area is determined after the thorough observation of vehicles' behavior and velocity on a particular highway. If the speed is high, the radius of the area should be smaller than 2 times the fps (frames per second) rate of an input video. Otherwise, the algorithm cannot trace vehicles properly, because fast-moving vehicles cannot be detected with a high confidence level. Detections with less than a 60% confidence level were not considered as vehicles, and they were not included in the detection result array. A greater value should be assigned to the radius of a circular area if the velocity of the vehicles is low.

$$\begin{array}{l} Pre = argsort(Dist_i) = \{i|Dist_i\} \\ New = \underset{Pre}{\text{argmin}}(Dist_{Pre}) = \{Pre|Dist_{Pre}\} \end{array} \tag{5}$$

Table 1 Nighttime case study videos captured with distinct horizontal and vertical angle-of-view of the highways from different countries

Case Study	Video Name	YouTube Url www.youtube.com/watch?v=	Duration	Description/ Reason for selection	Country Address	Video ID
Chunk1-3h Night City Traffic	3h Night City Traffic Ambience for sleeping, relaxing | ASMR Ambient Sounds	C7Kez-mefBU	First 30 min 46 s of the video	Wide angle-of-view of both incoming and outgoing lanes	Palaio Faliro, Attica, Athens, Greece	Video1
Chunk2-3h Night City Traffic	3h Night City Traffic Ambience for sleeping, relaxing | ASMR Ambient Sounds	C7Kez-mefBU	Next 32 min 32 s of the video	Angle-of-view is not very good. The camera on the side of outgoing road	Palaio Faliro, Attica, Athens, Greece	Video2
Chunk3-3h Night City Traffic	3h Night City Traffic Ambience for sleeping, relaxing | ASMR Ambient Sounds	C7Kez-mefBU	Next 31 min 09 s of the video	Angle-of-view is not very good. The camera on the side of incoming road	Palaio Faliro, Attica, Athens, Greece	Video3
Chunk4-3h Night City Traffic	3h Night City Traffic Ambience for sleeping, relaxing | ASMR Ambient Sounds	C7Kez-mefBU	Next 31 min 53 s of the video	Angle-of-view is not very good. The camera takes incoming road with the better angle	Palaio Faliro, Attica, Athens, Greece	Video4
Chunk5-3h Night City Traffic	3h Night City Traffic Ambience for sleeping, relaxing | ASMR Ambient Sounds	C7Kez-mefBU	Next 11 min 58 s of the video	Angle-of-view is not very good. Outgoing lanes with better angle. There are many trees occluding the incoming lanes	Palaio Faliro, Attica, Athens, Greece	Video5
Chunk6-3h Night City Traffic	3h Night City Traffic Ambience for sleeping, relaxing | ASMR Ambient Sounds	C7Kez-mefBU	Last 32 min 32 s of the video	Angle-of-view is in the best position. Both directions were captured very well	Palaio Faliro, Attica, Athens, Greece	Video6
Part1-2h Night City Traffic	2h Night City Traffic Ambience for sleeping, relaxing | ASMR Ambient Sounds	iA0Tgng9v9U	First 29 min 39 s of the video	Angle-of-view is good. Both directions were captured very well	Kifisias, Katechaki, Attica, Athens, Greece	Video7
Part2-2h Night City Traffic	2h Night City Traffic Ambience for sleeping, relaxing | ASMR Ambient Sounds	iA0Tgng9v9U	Next 29 min 43 s of the video	Angle-of-view is not very good. The camera takes incoming road with the better angle	Kifisias, Katechaki, Attica, Athens, Greece	Video8
Part3-2h Night City Traffic	2h Night City Traffic Ambience for sleeping, relaxing | ASMR Ambient Sounds	iA0Tgng9v9U	Next 29 min 31 s of the video	Angle-of-view is good. Both directions were captured very well. Fast-speed	Kifisias, Katechaki, Attica, Athens, Greece	Video9

Table 1 continued

Case Study	Video Name	YouTube Url www.youtube.com/watch?v=	Duration	Description/ Reason for selection	Country Address	Video ID
Part4-2h Night City Traffic	2h Night City Traffic Ambience for sleeping, relaxing | ASMR Ambient Sounds	iA0Tgng9v9U	Last 29 min 38 s of the video	Angle-of-view is not very good. The camera captures both Directions with bad angles	Kifisias, Katechaki, Attica, Athens, Greece	Video10
Los Angeles Freeway	Los Angeles Freeway I-101 (HD 30fps traffic sound)	-59Mr4UTgNg	30 min	Angle-of-view is not very good for outgoing lanes. Vehicle speed is very slow in the incoming direction	Filipinotown (Los Angeles, CA), USA	Video11
Road traffic video	Road traffic video for object detection and tracking - 2020 Challenge	4WUTKcqV708	5 min 6 sec	Angle-of-view is not very good. The camera captures the incoming lanes with a bad angle	Unknown	Video12

Here, Prev (as previous) is a set that keeps the unique IDs of the vehicles from the ith (previous) frame with the minimized distance values, and New (as new) holds the IDs of the vehicles from the $(i+1)$th (new) frame with the minimized distance values. In case, any of the minimized distance values outside of the range of the manually defined circular area, the tracker algorithm computes the number of the appearance for the observed vehicles on the previous frames. If the appearance age for the particular vehicle satisfies the minimum appearance age (in our case, appearance and disappearance ages are 60), the current and next states of the vehicle are predicted by linear equations (Eq. 7) using the Bbox info from the 60 previous frames, then the predicted Bbox is assigned to the current vehicle, and at the same time the vehicle's age is incremented by one for the each prediction step, Eq. 6. The appearance and disappearance ages were determined after the observation of movement speed of vehicles on the selected roads/highways videos.

$$Bbox_{Prev_k} = \begin{cases} Bbox_{New_k}, & \text{if} \quad Dist_{Prev_k,\, New_k} \leq 60, \\ Bbox_{predict}, & \text{if} \quad Dist_{Prev_k,\, New_k} > 60,\ and\ age > 60, \\ \text{Remove the disappeared vehicle}, & \text{otherwise.} \end{cases} \tag{6}$$

If the disappeared object's age is greater than the user-defined disappearance age, then it is considered as lost (disappeared) and de-registered immediately. The appearance and disappearance age thresholds provide robustness for the Bbox-based vehicle tracking algorithm by predicting the current or next state of the disappeared vehicles when the vehicles are lost because of partial or full occlusion, and illusion issues, or camera shaking, or harsh weather conditions. For the Bbox-based tracking algorithm, the prediction step (process) is critical because of the challenges including bad weather conditions such as heavy rain, snow or winds, or camera shaking, or illusion and partial or full occlusion, vehicle detection algorithm struggle at some level, or fails completely. In such conditions, vehicle tracking activates its prediction system to estimate the next state of the disappeared vehicle(s). The prediction functions based on linear equations are illustrated in Eq. 7.

$$\begin{aligned} d_X &= \left(\sum_{i=0}^{n} X_i\right)/n,\ X_{n+1} = X_n + d_X, \\ d_Y &= \left(\sum_{i=0}^{n} Y_i\right)/n,\ Y_{n+1} = Y_n + d_Y, \\ d_W &= \left(\sum_{i=0}^{n} W_i\right)/n,\ W_{n+1} = W_n + d_W, \\ d_H &= \left(\sum_{i=0}^{n} H_i\right)/n,\ H_{n+1} = H_n + d_H. \end{aligned} \tag{7}$$

$$Bbox_{predict} = [X_{n+1}, Y_{n+1}, W_{n+1}, H_{n+1}]$$

where d_X, d_Y, d_W, d_H are the average difference distances between the first and last points of a disappeared object, $X_{n+1}, Y_{n+1}, W_{n+1}, H_{n+1}$ are the new predicted states, and X_n, Y_n, W_n, H_n are the last disappeared state of the trajectory of the lost vehicle. We have used linear equations to estimate the new state of disappeared objects since the motion of vehicles on highways are usually linear. For intersections, polynomial functions are better choice since the motion is nonlinear. To sum up, the proposed detection feature-based tracking algorithm is built on three main modules, namely new vehicle registering, vehicle state updating and vehicle deregistering. In the first module, all newly appeared vehicles are assigned with new unique IDs. Next, the vehicles with their unique IDs are tracked through the consecutive frames by updating their states in each step. In the last module, the disappeared vehicles are immediately removed from the tracks list. The algorithm is being illustrated via Algorithm.2.

Algorithm 2 A proposed detection feature-based tracker algorithm.

```
Require: $Bboxes,\ New\_Bboxes$
Ensure: $New\ set\ of\ Bboxes,\ after\ assigning\ unique\ IDs.$
  if $len(Bboxes_i) \geq 1$ then
      $New\_Bboxes \leftarrow register(Bboxes_i)$
      $New\_Bboxes \leftarrow update(Bboxes_i)$
      $New\_Bboxes \leftarrow deregister(Bboxes_i)$
  end if
```

3.3 Datasets and case study nighttime highway videos

In the vehicle detection part, the COCO128 dataset was utilized by extracting three classes including buses, cars and trucks. In the vehicle counting system, 12 nighttime case study videos captured from 4 different highways with distinct vertical and horizontal angle-of-view were used. All nighttime videos have been acquired from YouTube channels. The detailed info is illustrated in Table 1

3.4 Accuracy calculation

The accuracy of traffic (vehicle) counters was measured based on Eq. 8. The total weighted average accuracy formula (Eq. 9) was utilized to evaluate the overall accuracy of the vehicle counting systems due to the reason that the number of vehicles by their types was not distributed normally. For instance, the total number of buses, cars, and trucks for all incoming and outgoing lanes of the 12 highway videos are 106, 16,591, and 166, respectively. This kind of abnormally distributed number of vehicles does not allow measuring the true accuracy of the entire system. However, it strongly recommended utilizing simple total average accuracy formula if the number of vehicles by their types is distributed normally.

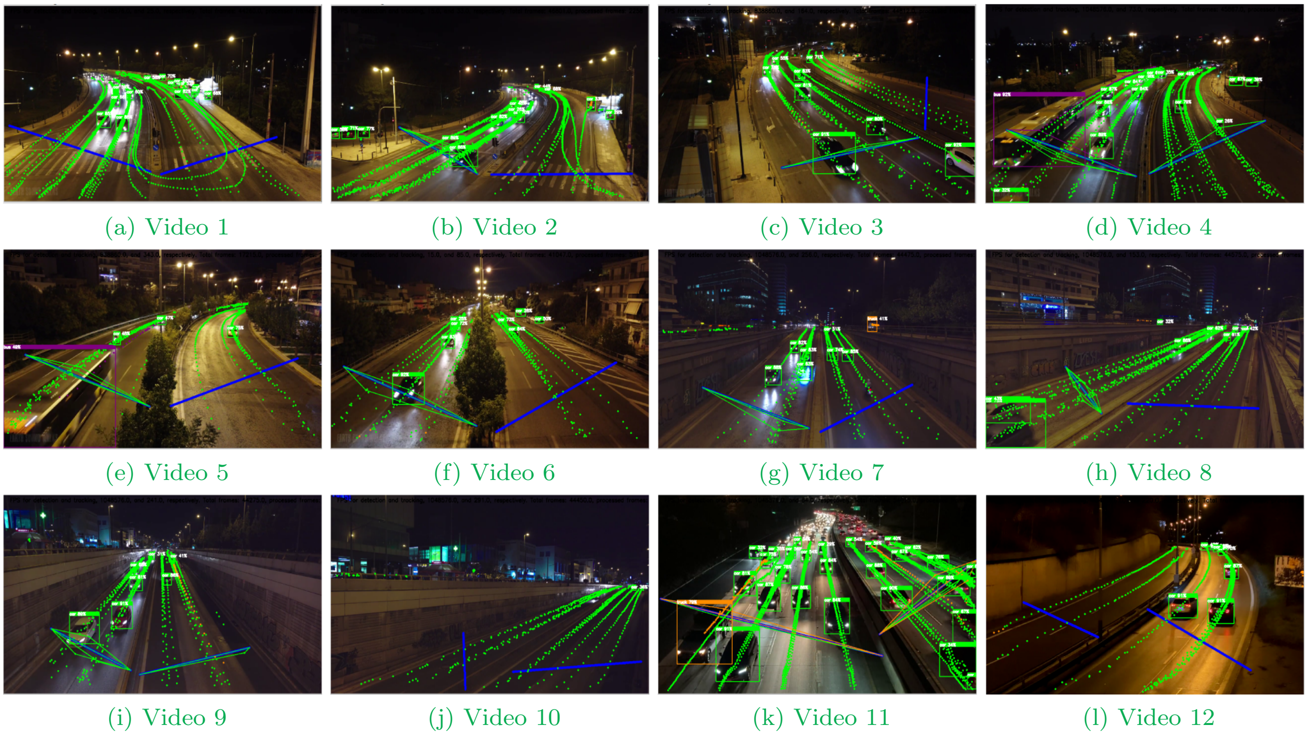

(a) Video 1 (b) Video 2 (c) Video 3 (d) Video 4
(e) Video 5 (f) Video 6 (g) Video 7 (h) Video 8
(i) Video 9 (j) Video 10 (k) Video 11 (l) Video 12

Fig. 4 Extracted vehicle trajectory data from the case study highways via the vehicle detection and the Bbox tracker

In future works, we are going to choose case study videos in which the number of vehicles is distributed normally.

$$\text{Acc} = 100 - 100 \times \frac{|\text{Ground Truth} - \text{Observed}|}{\text{Ground Truth}} \tag{8}$$

$$\text{Total\_Acc} = A_{\text{car}} \times w_{\text{car}} + A_{\text{truck}} \times w_{\text{truck}} + A_{\text{bus}} \times w_{\text{bus}} \tag{9}$$

where A_{car} is the accuracy of the estimated numbers of cars in incoming or outgoing direction. w_{car} is the weight for the cars and it is calculated in the following way, $w_{\text{car}} = N_{\text{cars}}/T_{\text{num\_veh}}$, N_{cars} is the ground truth number of cars, $T_{\text{num\_veh}}$ is the ground truth, total number of vehicles in a direction (incoming or outgoing). The other weights are calculated in the same way.

4 Results and discussion

4.1 Vehicle trajectory extraction results

Twelve nighttime highway case study videos have been processed with the Yolov5-based vehicle detection architectures and the developed vehicle tracking algorithm, and then, the time-dependent trajectory data of vehicles were extracted. The results can be seen in Fig. 4. Based on the figure and the vehicle counting results, it can be seen that the trajectories were extracted slightly accurately. Higher accurate trajectory data provide higher accurate traffic flow information such as total and categorical vehicle counts and speed, etc.

4.2 Traffic counting results of the entire system

4.2.1 Total weighted average accuracy results

The number of vehicles by their labels (class types) has been estimated with five Yolov5-based vehicle detectors and two vehicle trackers. In total, 10 traffic counters were developed, namely Yolov5n (Nano) vehicle detection model and the Bbox tracker, and Yolov5n (Nano) vehicle detection model and the Kalman filter-based vehicle tracker, etc. 12 case study videos were processed via the 10 vehicle counting systems. The vehicle count and accuracy results are illustrated in Table 2. The total weighted average accuracy (general accuracy) results of the traffic counting systems are shown in Fig. 5. Overall, the proposed vehicle tracker outperformed the Kalman filter-based tracker with 2.58% better total weighted average accuracy. The performance of the developed tracker for buses, cars, and trucks was 5.82%, 2.24%, and 36.5% better than the performance of the Kalman filter-based tracker. The main reason for this is the sensitivity of the Kalman filter approach.

The Kalman filter estimates the next state of a system based on a just previous state of it. This makes the Kalman filter very sensitive against instantaneous changes/minor shifts

Table 2 Vehicle counting results for all types of vehicles

Vehicle type	Buses										Cars										Trucks									
Road directions	Incoming			Acc. %		Outgoing			Acc. %		Incoming			Acc. %		Outgoing			Acc. %		Incoming			Acc. %		Outgoing			Acc. %	
Vehicle counters	G	K	B	K	B	G	K	B	K	B	G	K	B	K	B	G	K	B	K	B	G	K	B	K	B	G	K	B	K	B
Video 1																														
Yolov5n	4	17	10	23.53	40	6	3	3	50	50	605	577	495	95.37	81.82	684	630	608	92.11	88.89	7	1	6	14.29	85.71	5	0	0	0	0
Yolov5s	4	16	11	25	36.36	6	8	8	75	75	605	649	630	93.22	96.03	684	698	690	97.99	99.13	7	0	8	0	87.5	5	1	16	20	31.25
Yolov5m	4	9	3	44.44	75	6	4	4	66.67	66.67	605	682	600	88.71	99.17	684	581	672	84.94	98.25	7	2	1	28.57	14.29	5	0	1	0	20
Yolov5l	4	12	6	33.33	66.67	6	3	5	50	83.33	605	668	599	90.57	99.01	684	731	671	93.57	98.1	7	3	4	42.86	57.14	5	1	2	20	40
Yolov5x	4	7	4	57.14	100	6	6	5	100	83.33	605	645	604	93.8	99.83	684	746	673	91.69	98.39	7	7	7	100	100	5	1	4	20	80
Video 2																														
Yolov5n	7	8	3	87.5	42.86	6	2	1	33.33	16.67	679	692	710	98.12	95.63	767	772	721	99.35	94	2	7	0	28.57	0	9	0	0	0	0
Yolov5s	7	5	5	71.43	71.43	6	2	2	33.33	33.33	679	729	681	93.14	99.71	767	768	770	99.87	99.61	2	0	0	0	0	9	1	10	11.11	90
Yolov5m	7	6	5	85.71	71.43	6	3	3	50	50	679	860	709	78.95	95.77	767	777	749	98.71	97.65	2	0	1	0	50	9	0	6	0	66.67
Yolov5l	7	6	5	85.71	71.43	6	6	3	100	50	679	702	695	96.72	97.7	767	803	785	95.52	97.71	2	1	1	50	50	9	0	8	0	88.89
Yolov5x	7	6	5	85.71	71.43	6	5	5	83.33	83.33	679	681	676	99.71	99.56	767	787	780	97.46	98.33	2	0	3	0	66.67	9	1	9	11.11	100
Video 3																														
Yolov5n	3	3	3	100	100	3	0	0	0	0	721	652	678	90.43	94.04	429	352	365	82.05	85.08	4	0	0	0	0	3	0	0	0	0
Yolov5s	3	5	5	60	60	3	3	4	100	75	721	682	703	94.59	97.5	429	493	431	87.02	99.54	4	0	0	0	0	3	0	1	0	33.33
Yolov5m	3	2	4	66.67	75	3	2	2	66.67	66.67	721	666	682	92.37	94.59	429	490	440	87.55	97.5	4	0	3	0	75	3	0	2	0	66.67
Yolov5l	3	2	2	66.67	66.67	3	1	2	33.33	66.67	721	660	691	91.54	95.84	429	481	437	89.19	98.17	4	0	1	0	25	3	0	2	0	66.67
Yolov5x	3	2	4	66.67	75	3	2	0	66.67	0	721	708	733	98.2	98.36	429	429	433	100	99.08	4	1	4	25	100	3	0	5	0	60
Video 4																														
Yolov5n	5	21	7	23.81	71.43	3	0	0	0	0	530	515	506	97.17	95.47	524	450	429	85.88	81.87	7	0	0	0	0	10	0	0	0	0
Yolov5s	5	21	18	23.81	27.78	3	1	1	33.33	33.33	530	526	513	99.25	96.79	524	501	503	95.61	95.99	7	0	1	0	14.29	10	0	11	0	90.91
Yolov5m	5	6	6	83.33	83.33	3	1	0	33.33	0	530	540	550	98.15	96.36	524	528	524	99.24	100	7	1	2	14.29	28.57	10	0	5	0	50
Yolov5l	5	8	10	62.5	50	3	2	2	66.67	66.67	530	533	534	99.44	99.25	524	522	513	99.62	97.9	7	0	6	0	85.71	10	0	8	0	80
Yolov5x	5	5	8	100	62.5	3	3	1	100	33.33	530	532	528	99.62	99.62	524	525	524	99.81	100	7	2	3	28.57	42.86	10	0	6	0	60
Video 5																														
Yolov5n	2	0	0	0	0	3	2	3	66.67	100	253	161	143	63.64	56.52	180	160	159	88.89	88.33	5	1	1	20	20	3	1	0	33.33	0
Yolov5s	2	18	24	11.11	8.33	3	3	3	100	100	253	243	241	96.05	95.26	180	160	161	88.89	89.44	5	1	3	20	60	3	1	3	33.33	100
Yolov5m	2	1	1	50	50	3	1	3	33.33	100	253	275	242	92	95.65	180	166	167	92.22	92.78	5	1	3	20	60	3	1	2	33.33	66.67
Yolov5l	2	1	1	50	50	3	2	1	66.67	33.33	253	263	264	96.2	95.83	180	167	174	92.78	96.67	5	1	12	20	41.67	3	1	2	33.33	66.67
Yolov5x	2	1	1	50	50	3	1	1	33.33	33.33	253	265	264	95.47	95.83	180	170	181	94.44	99.45	5	1	5	20	100	3	1	2	33.33	66.67
Video 6																														
Yolov5n	3	4	13	75	23.08	11	8	14	72.73	78.57	479	370	348	77.24	72.65	525	428	424	81.52	80.76	3	1	0	33.33	0	5	1	7	20	71.43
Yolov5s	3	9	13	33.33	23.08	11	8	12	72.73	91.67	479	417	417	87.06	87.06	525	447	456	85.14	86.86	3	1	4	33.33	75	5	1	6	20	83.33

Table 2 continued

Vehicle type	Buses										Cars										Trucks									
Road directions	Incoming			Acc. %		Outgoing			Acc. %		Incoming			Acc. %		Outgoing			Acc. %		Incoming			Acc. %		Outgoing			Acc. %	
Vehicle counters	G	K	B	K	B	G	K	B	K	B	G	K	B	K	B	G	K	B	K	B	G	K	B	K	B	G	K	B	K	B
Yolov5m	3	2	3	66.67	100	11	5	8	45.45	72.73	479	467	469	97.49	97.91	525	549	546	95.63	96.15	3	1	1	33.33	33.33	5	1	4	20	80
Yolov5l	3	2	2	66.67	66.67	11	6	7	54.55	63.64	479	463	471	96.66	98.33	525	553	543	94.94	96.69	3	1	3	33.33	100	5	1	2	20	40
Yolov5x	3	1	3	33.33	100	11	8	9	72.73	81.82	479	468	474	97.7	98.96	525	532	520	98.68	99.05	3	1	3	33.33	100	5	1	8	20	62.5
Video 7																														
Yolov5n	3	1	1	33.33	33.33	6	5	5	83.33	83.33	571	374	458	65.5	80.21	555	502	511	90.45	92.07	3	0	0	0	0	2	0	1	0	50
Yolov5s	3	7	8	42.86	37.5	6	7	7	85.71	85.71	571	395	446	69.18	78.11	555	495	519	89.19	93.51	3	0	6	0	50	2	0	4	0	50
Yolov5m	3	1	2	33.33	66.67	6	6	5	100	83.33	571	547	576	95.8	99.13	555	542	555	97.66	100	3	0	0	0	0	2	0	1	0	50
Yolov5l	3	3	3	100	100	6	6	6	100	100	571	533	564	93.35	98.77	555	538	551	96.94	99.28	3	0	2	0	66.67	2	0	1	0	50
Yolov5x	3	1	2	33.33	66.67	6	6	6	100	100	571	543	572	95.1	99.83	555	540	558	97.3	99.46	3	0	1	0	33.33	2	0	1	0	50
Video 8																														
Yolov5n	5	1	2	20	40	7	1	1	14.29	14.29	576	525	518	91.15	89.93	493	471	476	95.54	96.55	4	0	0	0	0	6	0	0	0	0
Yolov5s	5	1	1	20	20	7	5	4	71.43	57.14	576	522	537	90.63	93.23	493	484	483	98.17	97.97	4	0	0	0	0	6	0	3	0	50
Yolov5m	5	0	1	0	20	7	3	3	42.86	42.86	576	549	561	95.31	97.4	493	502	498	98.21	99	4	0	0	0	0	6	0	4	0	66.67
Yolov5l	5	1	1	20	20	7	3	4	42.86	57.14	576	559	589	97.05	97.79	493	489	492	99.19	99.8	4	0	0	0	0	6	0	4	0	66.67
Yolov5x	5	0	1	0	20	7	0	4	0	57.14	576	550	578	95.49	99.65	493	487	490	98.78	99.39	4	0	0	0	0	6	0	5	0	83.33
Video 9																														
Yolov5n	4	3	9	75	44.44	5	2	1	40	20	522	149	81	28.54	15.52	524	273	266	52.1	50.76	4	0	0	0	0	7	2	0	28.57	0
Yolov5s	4	2	3	50	75	5	4	8	80	62.5	522	297	367	56.9	70.31	524	266	395	50.76	75.38	4	0	0	0	0	7	4	12	57.14	58.33
Yolov5m	4	3	1	75	25	5	3	2	60	40	522	561	534	93.05	97.75	524	503	513	95.99	97.9	4	1	0	25	0	7	3	2	42.86	28.57
Yolov5l	4	1	2	25	50	5	2	4	40	80	522	539	521	96.85	99.81	524	474	523	90.46	99.81	4	2	1	50	25	7	3	8	42.86	87.5
Yolov5x	4	0	0	0	0	5	1	1	20	20	522	568	516	91.9	98.85	524	501	535	95.61	97.94	4	0	0	0	0	7	0	4	0	57.14
Video 10																														
Yolov5n	2	0	1	0	50	2	0	0	0	0	463	190	255	41.04	55.08	523	401	453	76.67	86.62	3	0	0	0	0	4	0	0	0	0
Yolov5s	2	1	3	50	66.67	2	0	1	0	50	463	378	422	81.64	91.14	523	509	514	97.32	98.28	3	0	0	0	0	4	0	0	0	0
Yolov5m	2	2	1	100	50	2	0	0	0	0	463	434	469	93.74	98.72	523	517	525	98.85	99.62	3	0	2	0	66.67	4	0	2	0	50
Yolov5l	2	3	1	66.67	50	2	0	0	0	0	463	431	457	93.09	98.7	523	506	520	96.75	99.43	3	0	2	0	66.67	4	0	3	0	75
Yolov5x	2	4	1	50	50	2	0	0	0	0	463	442	463	95.46	100	523	508	524	97.13	99.81	3	0	1	0	33.33	4	0	3	0	75
Video 11																														
Yolov5n	8	11	6	72.73	75	6	4	3	66.67	50	2650	2094	2230	79.02	84.15	3253	2777	2722	85.37	83.68	17	1	2	5.88	11.76	41	1	1	2.44	2.44
Yolov5s	8	6	7	75	87.5	6	4	5	66.67	83.33	2650	2532	2616	95.55	98.72	3253	3190	3160	98.06	97.14	17	7	17	41.18	100	41	1	41	2.44	100
Yolov5m	8	3	5	37.5	62.5	6	6	6	100	100	2650	2575	2680	97.17	98.88	3253	3394	3282	95.85	99.12	17	13	15	76.47	88.24	41	1	32	2.44	78.05

Table 2 continued

Vehicle type	Buses										Cars										Trucks									
Road directions	Incoming			Acc. %		Outgoing			Acc. %		Incoming			Acc. %		Outgoing			Acc. %		Incoming			Acc. %		Outgoing			Acc. %	
Vehicle counters	G	K	B	K	B	G	K	B	K	B	G	K	B	K	B	G	K	B	K	B	G	K	B	K	B	G	K	B	K	B
Yolov5l	8	6	10	75	80	6	9	8	66.67	75	2650	2592	2703	97.81	98.04	3253	3256	3274	99.91	99.36	17	13	18	76.47	94.44	41	3	41	7.32	100
Yolov5x	8	7	9	87.5	88.89	6	9	7	66.67	85.71	2650	2537	2555	95.74	96.42	3253	3602	3234	90.31	99.42	17	23	19	73.91	89.47	41	4	49	9.76	83.67
Video 12																														
Yolov5n	1	2	1	50	100	1	0	2	0	50	39	20	15	51.28	38.46	46	33	31	71.74	67.39	6	0	4	0	66.67	4	1	1	25	25
Yolov5s	1	2	3	50	33.33	1	0	3	0	33.33	39	37	39	94.87	100	46	44	44	95.65	95.65	6	0	4	0	66.67	4	1	3	25	75
Yolov5m	1	2	1	50	100	1	1	1	100	100	39	41	44	95.12	88.64	46	45	44	97.83	95.65	6	0	7	0	85.71	4	1	3	25	75
Yolov5l	1	2	1	50	100	1	5	1	20	100	39	41	40	95.12	97.5	46	44	44	95.65	95.65	6	0	7	0	85.71	4	1	3	25	75
Yolov5x	1	2	1	50	100	1	0	1	0	100	39	40	40	97.5	97.5	46	45	45	97.83	97.83	6	4	8	66.67	75	4	1	3	25	75

G-Ground Truth, K-Kalman, and B-Bbox

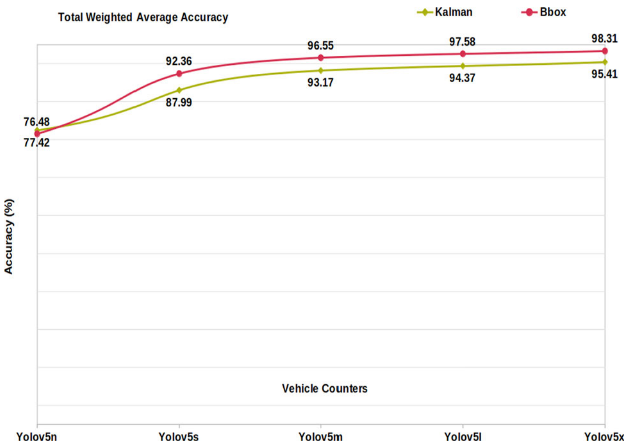


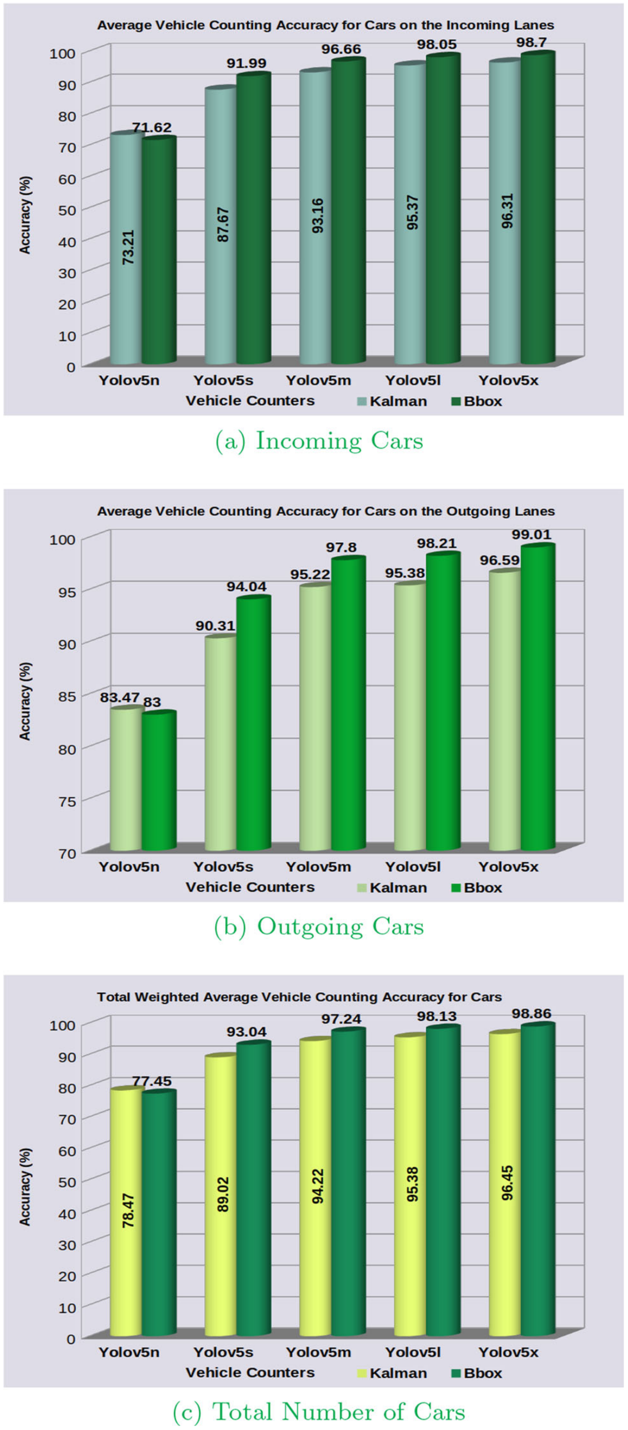
Fig. 5 Total weighted average accuracy of vehicle counters

in the state of the observed vehicle/object; however, in a vehicle tracking system, the next state of the vehicles/objects should be estimated based on several previous states of the observed vehicle/objects. Estimating the next state of a system based on several previous states of it enables a tracker to underestimate the unintentional minor and instantaneous changes. The Yolo algorithm predicts vehicle/object locations with minor shifts on an input video frame. Therefore, a stationary or moving vehicle is detected with minor shifts/differences from its true location of the observed vehicle through consecutive frames. This case is considered by the Kalman filter-based tracker as a movement of the observed vehicle, and the tracker starts estimations for a moving vehicle, even though the observed vehicle is in a stationary state. With the incorrect estimation, it starts to produce vehicle IDs for nonexisted vehicles, and mistracking (incorrect tracking) of vehicles occurs. Due to this reason, the Kalman filter-based algorithm might have performed with less accuracy than its peer. However, the proposed tracker predicts disappeared vehicles based on their 60 previous states. Because of this, the Bbox tracker does not produce fake tracks like other tracker does.

Furthermore, cars were detected accurately in all videos with all traffic counters, that is because, in the vehicle detection dataset (COCO128), the number of cars is far greater than the number of both buses and trucks. During training, vehicle detection had seen more car samples than the other two classes, which results in more accurate car detection than bus or truck detection. So, the vehicle detection module recognizes more cars with slightly higher confidence levels than the other two vehicle types. Consequently, it affects negatively both trackers. It can be seen from Table 2.

4.2.2 Traffic counting outcomes for cars

The bounding box-based tracker algorithm performed better with all Yolov5 models except the Yolov5n model. This is the lightest architecture via just 1.9 million parameters, so it is too prone to many false-positive vehicle detections, and its vehicle localization feature is the most unstable one than other Yolov5 models. Due to this reason, vehicles are detected with slightly different positions from their true locations on an input image. This leads the Kalman filter-based tracker to



(a) Incoming Cars

(b) Outgoing Cars

(c) Total Number of Cars

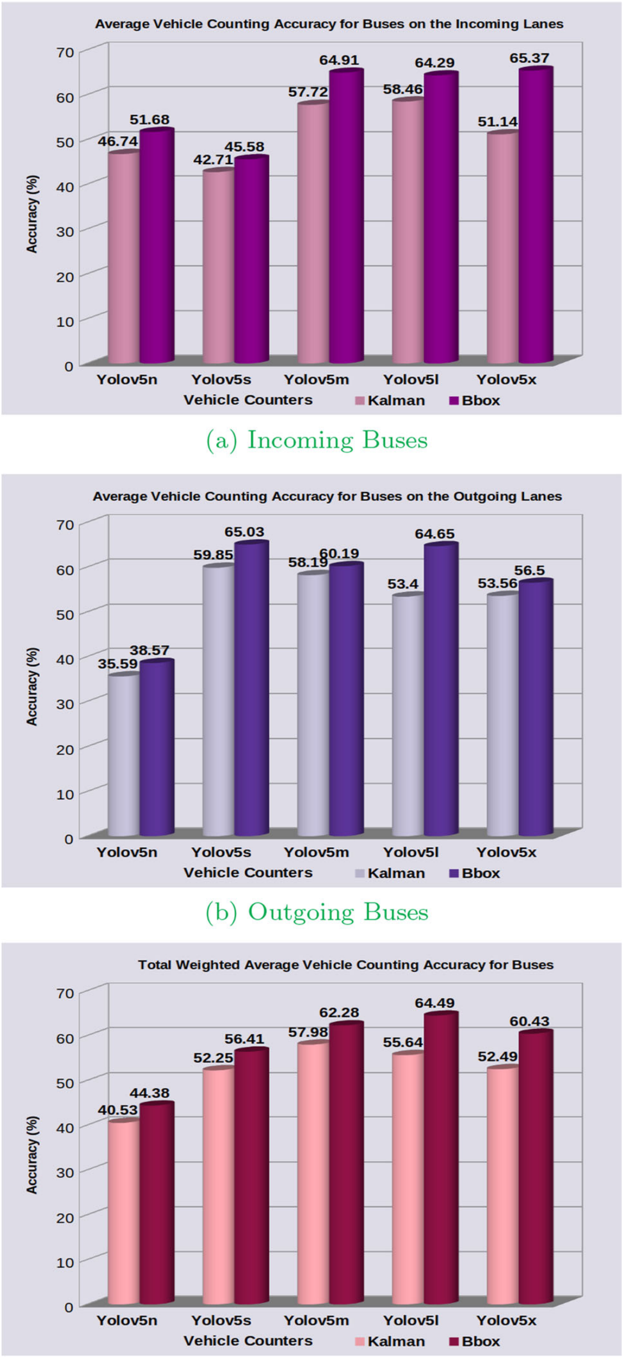
Fig. 6 Traffic counting accuracy outcomes for Cars

produce many false-positive tracks. The same cases occurred in almost all videos in both incoming and outgoing directions of the highways. Because of this reason, better accuracy outcomes than the Bbox-based tracker were obtained with the traffic counter system, which was built on Yolov5n and the Kalman tracker. This can be seen in Table 2 and Fig. 6.

However, vehicle counters that were built on the bounding box-based tracker and the other four models of Yolov5 outperformed the vehicle counters that were built on the Kalman filter-based tracker and four models of Yolov5. See for the detailed results Table 2 and Fig. 6. The vehicle detection based on Yolov5s architecture had also some false-positive car detections, which directly affected both trackers in which results slightly less traffic counting accuracy than other Yolov5 (medium, large, and X-large) architectures. However, the other three models with their better vehicle detection feature performed well; consequently, this case had a direct positive impact on the tracking accuracy of both trackers. However, the bounding box-based tracker outperformed its peer at a 2.24% better accuracy level for car counting.

4.2.3 Traffic counting outcomes for buses

There are a buses in all case study videos, in spite of the number of buses are few. Nevertheless, the vehicle detection models failed to detect them properly in most cases because of three reasons. The first reason is the COCO128 dataset. It includes less number of buses than the number of cars in it. It includes daytime and nighttime bus examples; however, the nighttime bus images are also very few in the dataset. So the nighttime bus image features could not be learned well by the Yolov5 architecture, which resulted in lower detection accuracy. The second reason is misclassification, in which buses were classified as cars and trucks that is because buses from the front side resemble car and truck objects at night time. The third one is the fast speed of vehicles in the video, because of the fast speed, the detectors failed. These issues led to the failure of the detectors. The failure of the detectors led to too many false positive tracks, which results in the failure or achieving lower accuracy of the entire traffic counting system for buses. See Table 2 and Fig. 7 for more details and insights. However, these misdetection and misclassification issues can be solved via the re-training of Yolov5 models on a new dataset that only contains nighttime vehicle images. In addition, the distribution of the samples of each class in the dataset should be distributed normally.



(a) Incoming Buses

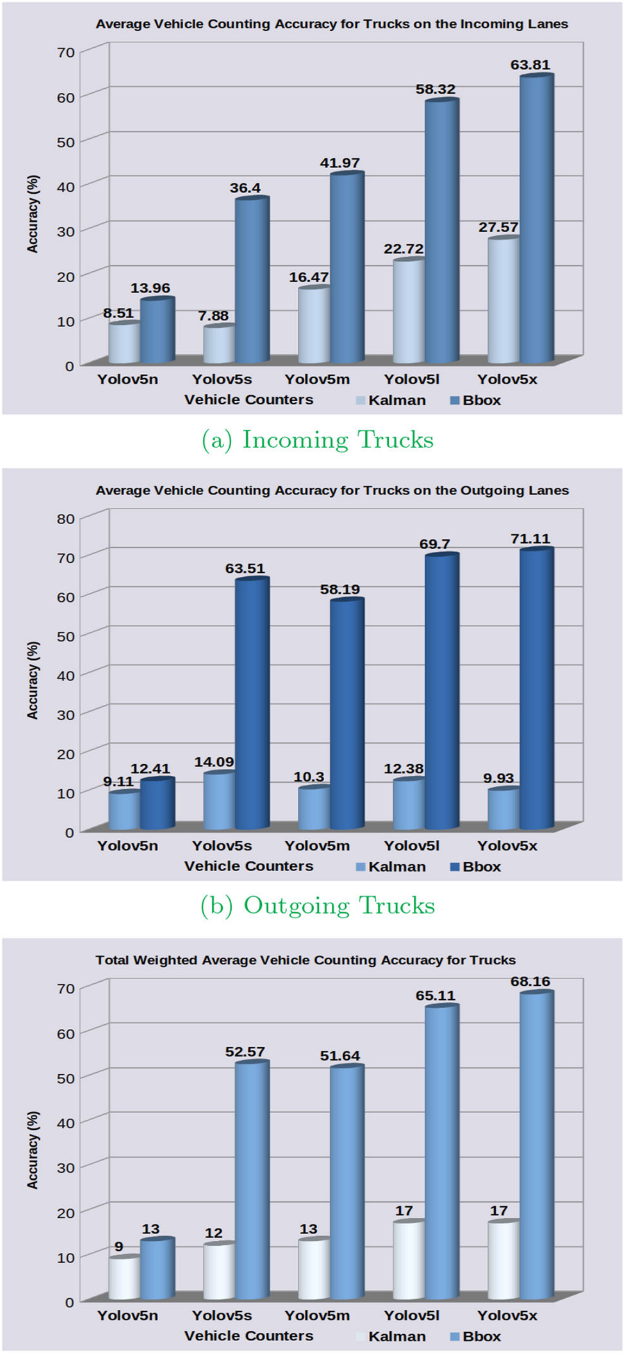
(b) Outgoing Buses

(c) Total Number of Buses

Fig. 7 Traffic counting accuracy outcomes for Buses

4.2.4 Traffic counting outcomes for trucks

For all video cases, there were trucks even though the number is few. But, because of the intense illusion cases of the headlights and taillights of vehicles, vehicle detectors performed with lower detection accuracy or failed. This led to the lower accuracy of vehicle trackers and traffic counters. However, the proposed algorithm performed with a 36.5% better traffic counting accuracy than the Kalman filter-based algorithm. Other details and more insight about the results can be seen in Table 2 and Fig. 8. The reason for the lower results in the Kalman filter-based tracker is its prediction algorithms. Kalman filter predicts the next state of an object based on



(a) Incoming Trucks

(b) Outgoing Trucks

(c) Total Number of Trucks

Fig. 8 Traffic counting accuracy outcomes for Trucks

only one previous state of the object. One state will not be sufficient for a proper estimation of an object's new state in case the object disappears from the several frames of an input scene due to intense headlight or taillight illusion or partial or full occlusion issues. However, the Bbox-based tracker approach predicts the new state of a disappeared object based on 60 previous states of the current object. It makes the tracker robust against disappeared object data processing. The solution to the problem is the improvement of vehicle detection accuracy. It can be improved by training Yolov5 models via only nighttime vehicle samples.

5 Conclusion

In this study, real-time vehicle (traffic) counting systems were developed using vehicle detectors, which are based on the pre-trained Yolov5 architectures, and vehicle trackers such as Kalman filter-based and Bbox-based. The evaluation results show that the proposed tracker algorithm outperformed the Kalman filter-based tracker with 5.82% (Bus), 2.24% (Car), 36.50% (Truck), and overall 2.58% better traffic counting accuracy for the 12 nighttime case study videos captured from the highways with different horizontal and vertical angle-of-views. For buses and trucks, both vehicle trackers performed with lower accuracy due to the reason that vehicle detectors produced detection results with lower confidence levels or failed in most cases. This is called misdetection. The misdetection cases led to lower vehicle tracking accuracy, which results in lower traffic counting accuracy. In addition to the misdetection cases, buses and trucks were misclassified as cars in most cases because of the intense illusion of headlights and taillights of vehicles. The misdetection and misclassification issues impacted negatively both vehicle trackers for buses and trucks, but the proposed tracking algorithm was less affected by these negative cases than the Kalman filter-based tracker method. Hence, more accurate trajectory data were estimated with the Bbox-based tracker, which results in higher and better traffic counting accuracy. The misdetection/misclassification issues can be solved by the headlight and taillight illusion removal techniques or with re-training of the vehicle detectors. In the future work, we are planning to develop a headlight and taillight illusion removal module to the Yolov5-based vehicle detection algorithm to address the misdetection and misclassification problems.

Funding This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. NRF-2022R1I1A3072355) and by the Scientific and Technological Research Council of Turkey under the Grant No. 119E077 and Title: "Development of a Customized Traffic Planning System for Sakarya City by Processing Multiple Camera Images with Convolutional Neural Networks (CNN) and Machine Learning Techniques".

Data availability Enquiries about data availability should be directed to the authors.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper. All authors of this research paper have directly participated in the planning, execution, or analysis of this study.

References

Azimjonov J, Özmen A (2021) A real-time vehicle detection and a novel vehicle tracking systems for estimating and monitoring traffic flow on highways. Adv Eng Inform 50:101393

Azimjonov J, Özmen A (2022) Vision-based vehicle tracking on highway traffic using bounding-box features to extract statistical information. Comput Electr Eng 97:107560

Badino H, Franke U, Mester R (2007) Free space computation using stochastic occupancy grids and dynamic. In: Programming, proceedings of the international conference computer vision, workshop dynamical vision

Cabido R, Montemayor AS, Pantrigo JJ (2012) High performance memetic algorithm particle filter for multiple object tracking on modern GPUs. Soft Comput 16:217–230

Chauhan NK, Singh K (2018) A review on conventional machine learning vs deep learning. In: 2018 international conference on computing, power and communication technologies (GUCON), pp 347–352

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

Erbs F, Barth A, Franke U (2011) Moving vehicle detection by optimal segmentation of the dynamic stixel world. In: 2011 IEEE intelligent vehicles symposium (IV), pp 951–956

Feng X, Jiang Y, Yang X, Du M, Li X (2019) Computer vision algorithms and hardware implementations: a survey. Integration 69:309–320

Fernández-Sanjurjo M, Bosquet B, Mucientes M, Brea VM (2019) Real-time visual detection and tracking system for traffic monitoring. Eng Appl Artif Intell 85:410–420

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. Springer, Berlin, pp 216–223

Girshick R (2015) Fast R-CNN. In: 2015 IEEE international conference on computer vision (ICCV), pp 1440–1448

Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: 2014 IEEE conference on computer vision and pattern recognition, pp 580–587

Jeon D, Kim D-H, Ha Y-G, Tyan V (2016) Image processing acceleration for intelligent unmanned aerial vehicle on mobile GPU. Soft Comput 20:1713–1720

Kanagamalliga S, Vasuki S (2018) Contour-based object tracking in video scenes through optical flow and Gabor features. Optik 157:787–797

Kavukcuoglu K, Sermanet P, lan Boureau Y, Gregor K, Mathieu M, Cun YL (2010) Learning convolutional feature hierarchies for visual recognition. In: Lafferty JD, Williams CKI, Shawe-Taylor J, Zemel RS, Culotta A (eds) Advances in neural information processing systems 23. Curran Associates, Inc, pp 1090–1098

Khalkhali MB, Vahedian A, Yazdi HS (2020) Vehicle tracking with Kalman filter using online situation assessment. Robot. Auton. Syst. 131:103596

Khan S, Ali H, Ullah Z, Bulbul MF (2018) An intelligent monitoring system of vehicles on highway traffic. In: 2018 12th international conference on open source systems and technologies (ICOSST), pp 71–75

LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521:436-444

Liu P, Wang G, Yu Z, Guo X, Lu W (2019) Vehicle tracking based on shape information and inter-frame motion vector. Comput Electr Eng 78:22–31

Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, Berg AC (2016) SSD: single shot multibox detector. In: Leibe B, Matas J, Sebe N, Welling M (eds) Computer vision—ECCV 2016. Springer, Cham, pp 21–37

Lu S, Wang Y, Song H (2020) A high accurate vehicle speed estimation method. Soft Comput 24:1283–1291

Luque-Baena RM, López-Rubio E, Domínguez E, Palomo EJ, Jerez JM (2015) A self-organizing map to improve vehicle detection in flow monitoring systems. Soft Comput 19:2499–2509

Mandellos NA, Keramitsoglou I, Kiranoudis CT (2011) A background subtraction algorithm for detecting and tracking vehicles. Expert Syst Appl 38(3):1619–1631

Nguyen B, Brilakis I (2018) Real-time validation of vision-based overheight vehicle detection system. Adv Eng Inform 38:67–80

Pillai MS, Chaudhary G, Khari M, Crespo RG (2021) Real-time image enhancement for an automatic automobile accident detection through CCTV using deep learning. Soft Comput 25:11929

Rathore MM, Son H, Ahmad A, Paul A (2018) Real-time video processing for traffic control in smart city using Hadoop ecosystem with GPUs. Soft Comput 22:1533–1544

Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR), pp 779–788

Redmon J, Farhadi A (2017) Yolo9000: better, faster, stronger. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), pp 6517–6525

Redmon J, Farhadi A (2018) Yolov3: an incremental improvement, pp 1–6. arXiv preprint arXiv:1804.02767

Ren S, He K, Girshick R, Sun J (2017) Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell 39(6):1137–1149

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

Song H, Wang X, Hua C, Wang W, Guan Q, Zhang Z (2018) Vehicle trajectory clustering based on 3d information via a coarse-to-fine strategy. Soft Comput 22:1433–1444

Song D, Tharmarasa R, Florea MC, Duclos-Hindie N, Fernando XN, Kirubarajan T (2019) Multi-vehicle tracking with microscopic traffic flow model-based particle filtering. Automatica 105:28–35

Sudha D, Priyadarshini J (2020) An intelligent multiple vehicle detection and tracking using modified vibe algorithm and deep learning algorithm. Soft Comput 24:17417–17429

Xiao X, Sun Z, Shen W (2020) A Kalman filter algorithm for identifying track irregularities of railway bridges using vehicle dynamic responses. Mech Syst Signal Process 138:106582

Yang Z, Pun-Cheng LS (2018) Vehicle detection in intelligent transportation systems and its applications under varying environments: a review. Image Vis Comput 69:143–154

Yang T, Cappelle C, Ruichek Y, Bagdouri ME (2019) Online multi-object tracking combining optical flow and compressive tracking in Markov decision process. J Vis Commun Image Represent 58:178–186

Zhao Z, Zheng P, Xu S, Wu X (2019) Object detection with deep learning: a review. IEEE Trans Neural Netw Learn Syst 30(11):3212-3232

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

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.