Traffic & Transportation

Analysing Headway Spacing and Calculating Passenger Car Equivalent Values Using Computer Vision and International Dataset

Burak ÇELİK[1], Ahmet TORTUM[2], Emre ÇİNTAŞ[3], Barış ÖZYER[4]

Original Scientific Paper
Submitted: 29 Nov 2024
Accepted: 7 Feb 2025

[1] burak.celik@atauni.edu.tr, Faculty of Engineering, Atatürk University, Erzurum, Turkey
[2] Corresponding author, atortum@atauni.edu.tr, Faculty of Engineering, Atatürk University, Erzurum, Turkey
[3] emrecintas@atauni.edu.tr, Faculty of Engineering and Architecture, Erzurum Technical University, Erzurum, Turkey; Faculty of Engineering, Atatürk University, Erzurum, Turkey
[4] baris.ozyer@atauni.edu.tr, Faculty of Engineering, Atatürk University, Erzurum, Turkey

Publisher:
Faculty of Transport and Traffic Sciences, University of Zagreb

ABSTRACT

Accurate traffic flow data are crucial for effective transportation planning and management. Different vehicle types impact traffic flow variably, requiring distinct passenger car equivalency (PCE) factors for calculating intersection and road capacity. Headway and spacing data are essential to assess traffic density and service level. Conventional data collection methods are time-consuming and often inaccurate. Unlike existing studies, this study employed computer vision to measure mixed traffic stream volume in terms of passenger car equivalent and collect headway-spacing data with high accuracy. The vehicle detection and counting procedures provide the mandatory infrastructure for measuring mixed traffic stream volume and collecting headway and spacing data. Novel approaches were introduced to gather comprehensive traffic data, including passenger car equivalent values, headway, spacing, flow rate, vehicle speed and traffic volume, using a single system. A custom and comprehensive international dataset was collected to analyse these approaches. Our trained model achieved a mean average precision (mAP) of 97.4%, with accuracies of 95% for headway, 93% for spacing and 99% for PCE values. The dataset can be downloaded at https://github.com/burak-celik/atavehicledataset.

KEYWORDS

intelligent transportation systems; traffic flow data; computer vision; headway-spacing data; passenger car equivalency.

1. INTRODUCTION

In recent years, intelligent transportation systems (ITS) have emerged as a key component for the development of smart cities, playing a critical role in the management, planning and security of public transportation [1-4]. Many developed countries utilise ITS to manage traffic conditions, integrate public transit systems [5, 6] and address traffic congestion and accidents effectively. These systems enable the determination of traffic flow characteristics and the collection of required information regarding traffic flow. This information encompasses the number, class, speed and length of vehicles passing at specific points. Besides these, the automotive industry has also seen a paradigm shift from traditional mechanical systems to intelligent systems, where artificial intelligence serves as the key element of advanced technologies [7]. Innovations that influence traffic flow, including semi-autonomous vehicles, autonomous vehicles, and advanced driver assistance systems such as adaptive cruise control, have emerged in this field. The implementation of these systems plays a crucial role in mitigating human factors that contribute to traffic accidents and enhancing passenger comfort [8, 9, 10].

For transportation planning, design and operational objectives, passenger car equivalency (PCE) factors have been employed for many years to convert mixed vehicle traffic into equivalent pure passenger car traffic streams [11, 12]. PCE factors account for the distinctive impacts of various vehicle types on traffic flow [13]. Essentially, this value serves as a coefficient indicating the number of passenger cars that would exert an equivalent influence on traffic flow as a heavy vehicle. Due to their dimensions and limited acceleration and deceleration capabilities, heavy vehicles (such as trucks, buses, recreational vehicles and, in some countries, vans) negatively affect traffic flow [14]. Consequently, detecting the presence of heavy vehicles within the traffic is essential for accurate calculations of intersection and road capacity in tandem with PCE factors.

Headway and spacing data are essential for various applications and analyses in traffic engineering, including capacity calculation, weaving areas, left turn at intersections, merging into the mainstream and merging or diverging into/from intersection arms and signalisation [15]. "The time, in seconds, between two successive vehicles as they pass a P_2 point on the roadway, measured from the same common feature of both vehicles" is the definition of headway. Similarly, spacing is defined as "the distance (between P_1-P_2 points), in metres, between two succeeding vehicles in a traffic lane, measured from the same common feature of the vehicles" [16, 17, 18]. A visual expression of these data, acquired through the proposed method, is presented in *Figure 1*.

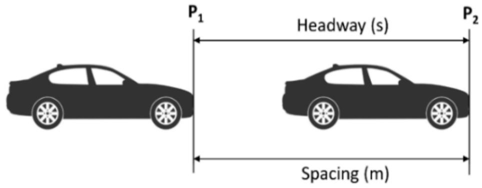

Figure 1 – Basic representation of headway and spacing

This study encompassed the determination of the number of vehicles by classes, employing a comprehensive dataset collected from 7 countries. The rationale behind this situation is that vehicle types differ across countries, and the goal is to enhance the model's accuracy by increasing data diversity. The vehicle detection and counting procedure creates a mandatory infrastructure for determining the mixed traffic flow in terms of equivalent passenger cars and collecting headway and spacing data. Diverging from previous research, passenger car equivalent values were derived using the PCE factors to characterise mixed traffic flows in terms of passenger cars. Additionally, headway-spacing data, flow rate and traffic volume (q) were obtained. The study collectively estimated vehicle count by class, passenger car equivalent values, flow rate, vehicle speed, headway-spacing data and traffic volume. The YOLOv5 algorithm's extra-large (XLarge) model was used for vehicle class determination. Training the YOLOv5 algorithm with international data was conducted using Google Colaboratory [19], a platform Google Research offers. This platform allows Python [20] code execution on GPUs, provided free of charge for machine learning and data analysis purposes.

Several key motivations underpin the initiation of this study:

1) **Creation of a comprehensive dataset and robust model for various conditions:** A remarkable contribution is the development of an expansive international dataset comprising 16,518 images with 140,293 annotated instances. This dataset, originating from 7 different countries and encompassing both daytime and nighttime images, addresses the scarcity of publicly available vehicle detection and counting datasets. The dataset used in this study differs from existing datasets in that it contains data from 7 different countries and many different vehicle types. Notably, this dataset surpasses prior literature's diversity, comprehensiveness and resolution level. We were able to collect data with greater accuracy than previous studies due to the complete dataset we used.
   Through our trained model, we achieve enhanced vehicle detection accuracy by mitigating challenges posed by variables such as daytime, nighttime and varying viewing angles. A trained model eliminates the difference between day and night and demonstrates efficacy across all conditions for vehicle detection and counting.
2) **A novel method for headway and spacing data:** We introduce a novel method for acquiring headway and spacing data with high accuracy relevant to diverse traffic engineering applications such as capacity calculation, weaving areas, left turn at intersections, merging into the mainstream and merging or diverging into/from intersection arms and signalisation.
3) **Utilisation of PCE factors and collection of flow rate data:** Computer vision was used to quantify mixed traffic flow in terms of equivalent passenger cars with PCE factors. Passenger car equivalent values are fundamental for the precise calculation of intersection and road capacity. Differing from existing research, computer vision was utilised to determine flow rate.

4) **Comprehensive system testing:** We rigorously assess our system's performance by conducting tests involving more than 10,000 vehicles under various conditions. System testing in existing studies is more limited and insufficient to measure the true accuracy of the model.

2. RELATED WORKS

The number of studies in the field of vehicle classification and counting has grown over the past few years, and the results of these studies have consistently improved thanks to the employment of various algorithms, datasets, models and scenarios. Engineering challenges are increasingly being solved with optimisation algorithms as computer computing power increases [21]. The primary issue that must be addressed in this field is ensuring that computer vision-based systems are as accurate as, or more accurate than, manually performed traffic counts [22]. New studies emerge in vehicle detection, classification and counting every day due to advances in deep learning and computer vision. Thus, the accuracy of computer vision-based automatic traffic counting systems can compete with the precision of manually performed traffic counts. This section explores the literature for vision-based vehicle detection, counting and traffic monitoring areas, highlighting data necessary for performance comparison.

In a study by Rajput et al. [23], the YOLOv3 algorithm was used to calculate charges according to vehicle class on toll roads. In the custom dataset, vehicles that do and do not have to pay tolls were divided into eight classes, as the Government of India recommended. There were 11,520 images in total in the dataset, and 80% were used for training and 20% for validation. With the YOLOv3 algorithm trained with the created dataset, 94.1% classification accuracy was obtained in the tests performed at highway tolls.

In the study by Song et al. [24], a vehicle detection and counting system based on computer vision was developed. In the custom dataset, vehicles were divided into three categories. The dataset contained a total of 11,129 images and 57,290 annotated instances. After road surface segmentation was performed on the images for the development of vehicle detection, the images were placed on the YOLOv3 network. Vehicle trajectories and the number of vehicles passing were calculated using the ORB algorithm. Following the training process, an 87.9 mAP score was obtained. In tests conducted on 1,509 vehicles, a counting accuracy rate of 92.6% was observed.

Bui et al. [25] researched various ways to enhance traffic analysis using video-based vehicle counting. Yolov3 and DeepSORT algorithms are used for vehicle detection, tracking and counting. Instead of virtual lines, distinguished regions for tracking and counting vehicles improve vehicle counting performance. The proposed method obtained an average of 88.5% vehicle count accuracy for different conditions.

In research carried out by Park et al. [26], an integrated system that utilised YOLOv4 and identified vehicles and licence plates was developed. A custom dataset was created using traffic images collected from South Korea at 4K resolution to train the YOLOv4 algorithm. According to their dimensions and passenger capacity, vehicles were divided into six classes in the dataset. There were a total of 12,044 annotated instances in the dataset. When the results of training the YOLOv4 algorithm with the collected dataset are examined, the mAP scores obtained in the tests that were carried out separately for 1-, 2-, 3- and 4-lane roads were 98.0, 94.0, 97.1 and 84.6, respectively.

An improved YOLOv4 detection method was suggested in research published by Xu et al. [27] to detect small and occluded objects more accurately. Following training the improved YOLOv4 network using the KITTI Dataset (Geiger et al. [28]), an 81.2 mAP score was recorded for the three primary object classes. After introducing the YOLOv4 network with the BDD100K Dataset (Yu et al. [29]) the mAP score was 61.6 for three primary object classes. Compared to the mAP scores achieved with the standard YOLOv4 network, the scores for the improved YOLOv4 network were 2.7 points higher.

An improved YOLOv5 network was proposed in research carried out by Zhang et al. [30] to reduce the rate of incorrect detections caused by occlusion. The dataset was created with traffic images collected from the Shandong region of China and images retrieved from BIT Vehicle_Dataset (Dong et al. [31]) to train the network. The database contains 2,844 images of 7 vehicle classes and approximately 3,200 annotated instances. Following the training process, the standard YOLOv5 network yielded an 89.8 mAP score, while the improved YOLOv5 network delivered a 90.5 mAP score for all vehicle classes.

Within the context of the research reported by Dinh et al. [2], a custom dataset was collected using traffic images gathered from the Vietnam region. The dataset comprised a total of 4,700 annotated images. Looking at the training results, it was determined that SSD MobileDet was the most successful model in terms of FPS and mAP scores on the Coral Dev Board (320 x 320). An average FPS of 26.8 and 92.1 mAP scores were obtained for this model. In vehicle counting tests, four distinct vehicle classes achieved an average accuracy of 83.5% toward the camera direction and an average accuracy of 82.7% away from the camera direction.

Harikrishnan et al. [32] proposed a method, a modified single-shot multi-box convolutional neural network named Inception-SSD, for vehicle detection and vehicle counting. Affinity propagation clustering (APC) is used instead of non-maximum suppression (NMS) to enhance the identification of nearby occluded cars. On the PASCAL VOC 2007 test dataset, the proposed ISSD achieved 79.3 mean average precision (mAP). The vehicles are counted class-wise with a weighted F1 of 98.5% and 93.3% vehicle counting accuracy obtained.

Khalifa et al. [33] created two distinct datasets consisting of 4,870 images for daytime and 5,338 images for nighttime, with traffic images collected from the Malaysian region in their study. In the research, the YOLOv5-small architecture was combined with the k-means algorithm to achieve anchor box optimisation in various illumination conditions. Existing mAP scores were improved by 5-6% approximately due to the k-means algorithm applied. After training the model, a 97.8 mAP score was obtained for the daytime dataset, and a 95.1 mAP score was obtained for the nighttime dataset. The average precision value was calculated as 97.8 for the daytime dataset, while the same was 95.2 for the nighttime dataset.

In the study by Neupane et al. [34], the YOLOv5-large network was trained using two datasets comprising images captured by Thailand's traffic surveillance cameras. There were seven vehicle classes and a total of 34,983 samples in the datasets. The transfer learning-based fine-tuning method was applied to solve the poor performance problem that arises when testing with images not included in the dataset for vehicle detection systems using deep learning. As a result of training the model, the mAP@50 score was determined to be 69.5%, and the precision and recall values were determined to be 96% and 95%, respectively. In vehicle counting tests, regardless of vehicle class, an average accuracy rate of 94% was recorded at various illumination levels.

In the research conducted by Djula et al. [35], YOLOv7 was employed in the vehicle detection and counting system. A custom dataset has been prepared for the vehicle types to vary widely in Indonesia and the trained model to be suitable for the place where the study is carried out. A dataset consisting of 1,105 images was created to train the model. An average of 94.5 mAP score was obtained for nine vehicle classes.

In the study of Kalva et al. [36], the YOLOv8 and DeepSORT algorithm was used to build a real-time vehicle identification and counting system. An open-source computer vision dataset was created to train the model. The dataset, which has 13 vehicle classes, consists of 627 images. As a result of model training, a 92.8 mAP score was obtained.

In the study by Chen et al. [37], a model was created using various sensors to collect data on traffic flow. CAV's hardware platform and software algorithm were used to create the model, and the Flow Project was used to optimise the model. CAV functions as a "super-extended floating car" in traffic flow, communicating with other intelligent cars and detecting any vehicle's driving status, distribution and density within its detection region. An established system can collect the mean headway of each lane and the mean spacing of each lane from the model.

Traffic data were effectively extracted from videos taken with legacy cameras by Liu et al. [38]. Vehicle classification, counting and speed measurement processes were made with images taken with these cameras. The vehicles were divided into classes according to their lengths in the study. The tests achieved 66% to 87% counting accuracy for SSD and 90% to 98% for YOLOv2 and Faster R-CNN. Headway and spacing data can be obtained from the time-space diagram produced by the proposed method.

Traffic Flow Load monitoring system, which is an essential data for bridge design, was installed by Ge et al. [39]. The study develops a monitoring system that combines a weigh-in-motion system with machine vision capabilities. The system is capable of obtaining vehicle length, axle spacing, vehicle speed and headway data. Target detection and tracking were performed with YOLOv3 and a matching algorithm. The proposed method offers improved time measurement resolution compared to methods that rely on the WIM system.

Du et al. [40] acquire time headway and passenger car unit (PCU) at junctions using the fusion algorithm of YOLOv3 and DSST. This fusion approach can more precisely extract the intersection's traffic flow metrics since it can fully reflect the traits of the two. The proposed method has been evaluated for three vehicle classes: buses, trucks and cars. Average 94.97% accuracy for PCU, average 78.87% accuracy for time headway was achieved.

3. METHODS

This section provides information about the collected dataset and the implemented traffic data collection system. The traffic data collection system established in this study, unlike the systems shown in similar studies in the literature, is comprehensive in terms of data diversity, versatility and collection of all data with a single system. The general architecture of the system can be seen in *Figure 3*. When it comes to summarising the workflow of the traffic data collection system, the test footage (image or video) was provided to the system in the first stage as the input. Using the YOLOv5x (Jocher et al. [41]) algorithm, which was subjected to transfer learning using the combined dataset, the classes of vehicles are detected, and their bounding boxes are drawn. The DeepSORT algorithm assigns a unique ID to each detected vehicle. The vehicle counting process is initiated when a vehicle with a unique ID enters the ROI (region of interest) with user-determined coordinates. ROI is shown in *Figure 2* and *Figure 6* as a white line.

3.1 Headway-spacing data and traffic volume

Simultaneously with the vehicle counting process, the system calculates headway and spacing data. For the headway calculation, the user drew green and red lines for each lane, as seen in *Figure 6*. If explained for the first lane, the time counter starts when the bounding box of the first vehicle moving on the lane touches the green line. To determine the correct time intervals in the calculation of the start time of the headway measurement, the real-time fps value and the frame ID (F_{id}) are used as in *Equation 1*. The counter stops when the following vehicle's bounding box touches the green line. Thus, the interval in time between two consecutive vehicles, that is, the headway, is calculated using *Equation 2*.

The average headway value for each lane (Avg TimeHW) is printed on the right in *Figure 6* by averaging the headways between all vehicles in the same lane. In this context, H_{st} represents the headway start time of each consecutive vehicle, calculated by

$$H_{st} = F_{id} x(1/\mathrm{fps}) \quad (1)$$

where F_{id} is the frame id and then headway constant (H) is calculated by

$$H = |V_c(H_{st}) - V_p(H_{st})| \quad (2)$$

V_c is the current vehicle, V_p is the previous vehicle and H is headway data. Vehicle speeds are used to calculate spacing. To explain the first lane, the distance between the white and green lines in *Figure 2* and *Figure 6* must be known to measure the speed. The average speed of the vehicles was determined by measuring the time the vehicles travelled the distance between these two lines in our system. *Figure 2* can be examined for a better understanding of the approach. V_1 and V_2 are the average speeds of the vehicles.

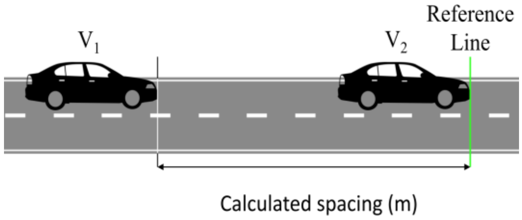

Figure 2 – Basic representation of the spacing calculation

In our method, the distance measurements given in *Figure 6* are calculated as in *Equation 3* by subtracting the y coordinate values (D_y) on the image axis of the points where each vehicle touches the white ROI line and own lane line. This length and the required lengths for different videos have been measured using satellite photos with GPS. To obtain the distance measurement from our method, the pixel distance must be chosen correctly to get the actual distance. Therefore, the distance measurement from the GPS was compared with the distance measurement from our method. D_y is the y coordinate distances, r_w is the white ROI line y-coordinate and l_n is the nth lane-line y-coordinate.

$$D_{y_i} = |r_w - l_{n_i}| \quad (3)$$

After obtaining the y distance in our method, the D_y expression is divided by the pixel distance as in *Equation 4* to obtain the real-world distance measurement. As in the headway calculation, the time in *Equation 1* is added to *Equation 5* according to the real fps.

$$D_{w_{y_i}} = D_{y_i} / P_{p_m} \tag{4}$$

$$\text{Displacement} = \text{Velocity}.\text{Time} \tag{5}$$

where, $D_{w_{y_i}}$ on the world axis of the points where each vehicle touches the white ROI line and own lane-line with y-coordinates. P_{p_m} represents metre per pixel, x is the distance (m), V is average speed (m/s) and t is the time (s). Also, the calculated spacing algorithm is shown in *Algorithm 1*. U_{id} is a unique vehicle ID, R_{st} is ROI line start times and S_t is headway start times array data.

When calculating the spacing, the green line was taken as a fixed reference line. Spacing is calculated with a fixed reference line approach as it changes constantly in moving traffic. The distance between the front bumpers of the first and second vehicles is calculated for the position of the second vehicle when it touches the green reference line in *Figure 2*. After calculating the average headway for a two-lane road, the traffic volume was calculated using *Equation 6* [42].

Algorithm 1 – Space headway calculation

```
SPACE HEADWAY (Hs)
 if (min(box) in r_w and U_id not in data), then
   H_st = F_id *(1/ fps)
   R_st [U_id] = H_st and data.append(U_id )
 end
 for i in range (0, len(l_n), 2) then
   if (min(box) in l_n) then
     D_y [i] = | r_w – l_n [i] |
     D_y [i] = D_y[i] / P_pm
     t = | S_t [U_id] - R_st [U_id]|
     V = D_y [i] / t
   end
 end
 CalculateHeadwaySpaces() :
   H = | V_c (H_st) - V_p (H_st) |
   H_s = V_p [V] * H
```

$$\text{Traffic Volume(veh / hr)} = \frac{3600\ \text{s / h}}{\text{headway (s / veh)}} \tag{6}$$

3.2 Collection of flow rate and passenger car equivalent values

In addition, the proposed system can compute the flow rate, which is the hourly calculation of the number of vehicles that pass through a particular road section in less than an hour [42], as well as the passenger car equivalent values, which are acquired by multiplying the number of vehicles that were counted for each vehicle class by the PCE factors pertaining to that vehicle class.

A GitHub repository content [43] was referenced as a traffic data collection system. This content was recoded and used in this study. In the collected dataset in Section 3.3, vehicles were classified as passenger cars, trucks, vans and buses, and then vehicle counts were performed across all four of these classes. The following are the reasons for dividing the vehicles into these four primary classes for the study:

1) A significant portion of the traffic flow comprises vehicles from these four classifications,
2) If the vehicles are classified into more than four classes, it may become difficult to identify visually similar vehicles, and as a result, the model may provide false detections,

3) The dimensions, start-up times, acceleration and deceleration capabilities, braking distances, headways and spacings and their ability to maintain high speeds are different in the four classes mentioned [44, 45].

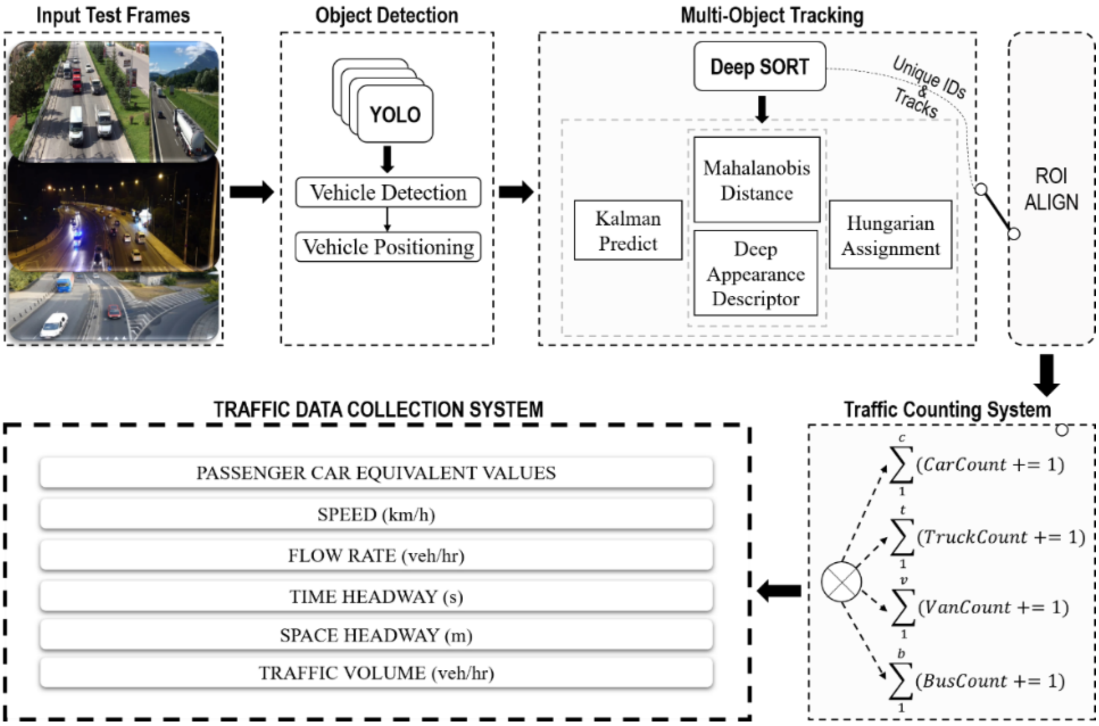

Figure 3 – Workflow diagram of traffic data collection system

Hence, vehicles were divided into four classes: passenger cars, trucks, vans and buses. PCE factors pertaining to each class can result in different values in different sources based on the features of the vehicles (dimensions, start-up times, acceleration and deceleration capabilities, braking distances, headways and spacings and their ability to maintain high speeds). These values may vary based on the standards established by different countries regarding this matter. *Table 1* displays the PCE factors [46] for Türkiye, that are utilised in the capacity analysis.

Table 1 – PCE factors in Turkish Standards TS 6407

Vehicle type	Urban road	Roundabout	Signalised intersection
Passenger car	1.00	1.00	1.00
Minibus, taxi	1.15	1.30	1.27
Commercial truck	2.00	2.80	1.75
Urban and interurban bus	3.00	2.80	2.25

PCE factors stated in the Highway Capacity Manual 2010 [42] are shown in *Table 2*. In HCM 2010, trucks and buses were assessed together. The PCE factors assigned based on the vehicle class also differ based on the terrain's topography.

Table 2 – Highway Capacity Manual 2010 PCE factors

Vehicle	PCE factors by type of terrain		
	Level	Rolling	Mountainous
Trucks and buses	1.5	2.5	4.5
Recreational vehicles	1.2	2.0	4.0

In this study, vehicle counting was performed, the number of vehicles passing through the monitored section was calculated based on the vehicle classes, and these numbers are presented in the results section both in terms of their vehicle class and passenger car equivalent values (for HCM and TS6407) to use them for capacity analysis.

System testing was conducted on a large variety of different scenarios and a vast number of vehicles. System tests with a limited number of vehicles may not be sufficient to determine the system's real success rate. Increasing the number of tests conducted helped to showcase the system's real success rate more accurately.

Figure 6 shows the system image in operation. The counting process is carried out when the centre of the bounding box passes the ROI specified by the user. The position and width of the ROI may be adjusted depending on the camera's angle and the distance between the camera and the vehicles. The optimum camera angle was set as 35-45° for the best vehicle counting accuracy.

3.3 Vehicle dataset

In the computer vision field, to improve the generalisation of the model, methods such as collecting more data, data augmentation and label smoothing are used [47, 48]. A large and diverse dataset was collected in this study to develop a generalised model. For the model to learn the vehicle classes more accurately, images of traffic from several countries were included in the dataset (since the variety of vehicle types in different countries is abundant). To improve the generalisation of the model, the dataset included:

- Traffic images with different resolutions (low, medium and high resolution)
- Traffic images collected from 7 different countries
- Traffic images collected in day and night time
- Vehicle images captured from the front, back and sides
- All annotation procedures were performed manually, and only double-checked images were used to prevent inaccurate annotation.

The dataset for this research contains 16,518 images and a total of 140,293 annotated instances. The number of annotated instances was high to increase vehicle detection accuracy. The distribution of annotated instances per vehicle type is presented in *Table 3*. The dataset contains an average of 8.5 annotated instances per image. *Figure 4* depicts the graph of the combined dataset compared to other datasets regarding the number of annotations per image.

Table 3 – Number of annotated instances for each class

Vehicle type	Number of annotated instances
Car	80,192
Bus	7,962
Truck	35,224
Van	16,915
Total	**140,293**

The dataset contains 11,129 images (57,290 annotated instances) from the study conducted by Song et al. [19] using images captured in the Hangzhou, China region. The annotations on the images from the research above were reviewed and re-created from scratch, and the van class was added. The remaining 5,389 images (83,003 annotated instances) were acquired from public traffic images from the United States, France, England, Poland, Thailand and Türkiye. The dataset was created with images collected from 7 different countries in total. Of the dataset, 75% of the images in the collected dataset were used for training, 15% for validation and 10% for testing. The dataset is universal since it covers various vehicle images from countries in different parts of the world. Therefore, it can be employed in other countries while offering high accuracy.

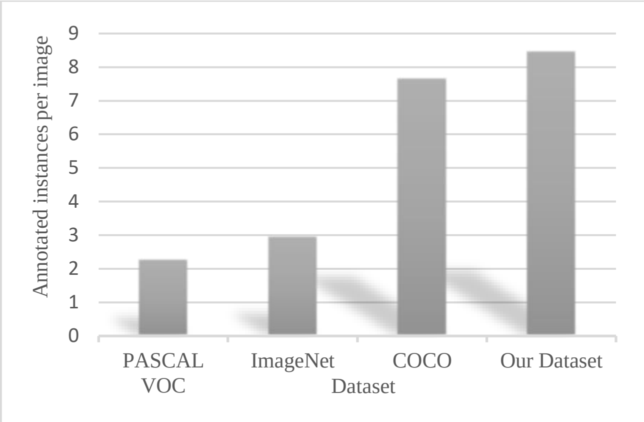

Figure 4 – Comparison of datasets in terms of annotated instances per image

3.4 Transfer learning

YOLOv5 provides pre-trained models on the COCO dataset (common objects in context), which consists of 80 classes. Instead of training the model from scratch, the YOLOv5x, among the default models of the YOLOv5 algorithm, was trained with the transfer learning technique and utilised in this study. Thus, both time and processing power were conserved, and a high level of training success was attained. The xLarge model of the YOLOv5 algorithm was applied because, as shown in *Figure 5* [41] it has a higher average precision (AP) value than the nano, small, medium and large models of the algorithm above. The training was carried out on the Google Colaboratory platform. While the YOLOv5x algorithm was trained for 500 epochs, the network size was set to 832x832, and the batch size was selected as 10. For all other training parameters, default values were utilised. The training process was completed in approximately 200 hours on the Colab platform. During the tests performed within the scope of the study, the best weights obtained after training were used. In Section 4, the results of the training process are explained in thorough detail.

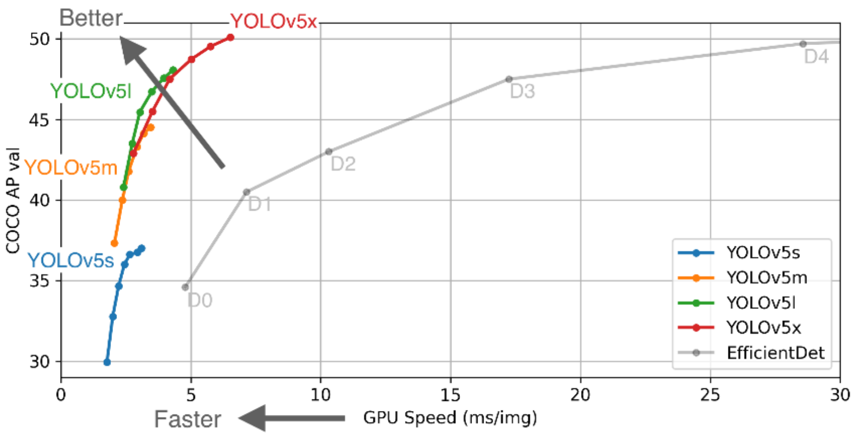

Figure 5 – Graph of AP – GPU speed values of models of YOLOv5 algorithm and EfficientDet algorithm

3.5 DeepSORT algorithm

The DeepSORT algorithm is a special object-tracking algorithm based on deep learning [49]. DeepSORT, widely used in the literature, can track the target object for a long time with the help of the Kalman filter it contains, even though an object comes in front of the tracked object. Thus, it is prevented from changing the unique IDs assigned to objects. A deep learning algorithm will be used in DeepSORT to decrease a large number of identity shifts and increase the effectiveness of tracking through SORT algorithm occlusions [50].

The Kalman filter is used iteratively for better attribution and can predict future locations based on the current location. After the unique ID assignment, it assigns the detections in a new frame to the existing traces using the Hungarian algorithm for the assignment cost function to reach a minimum [51]. Also, to include the uncertainties from the Kalman filter, the squared Mahalanobis distance is used, which gives a better result than Euclidean. To avoid the handicaps of the Kalman filter, which may fail in real-world problems (occlusion, dynamic movement), another distance metric based on the “deep appearance” of the object is used.

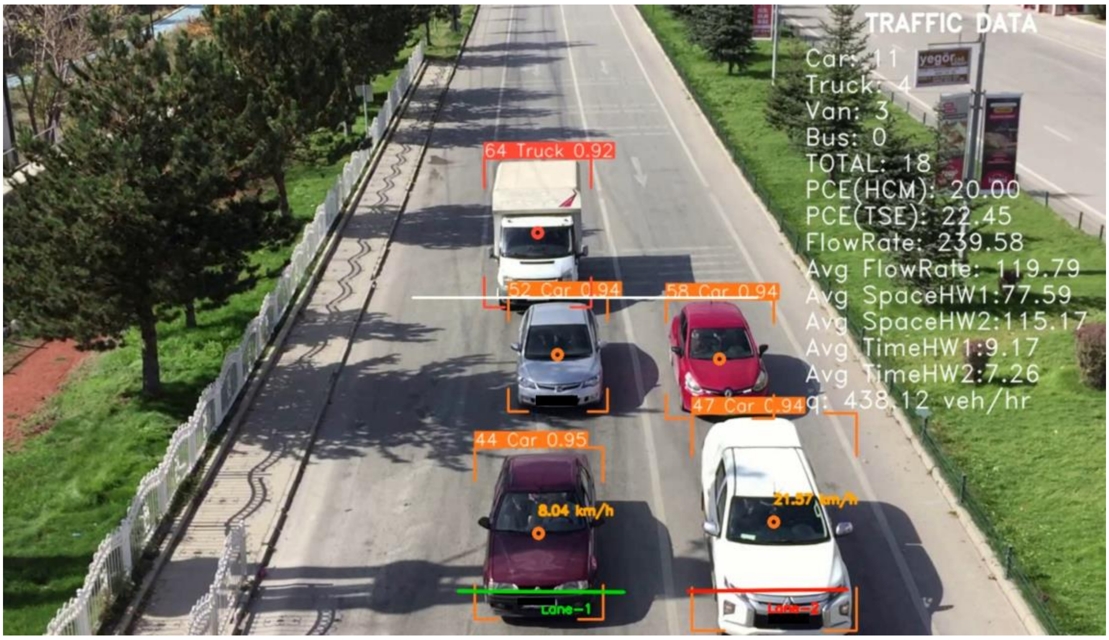

Figure 6 – Image of traffic data collection system in action

A classifier was created in this metric, and the last classification layer was removed. In this way, a single attribute vector (deep appearance descriptor) was generated, and the fast operation of the algorithm compared to classical architecture was ensured. The CUDA and cuDNN versions used in the tests of the model were 11.1 and 11.2, respectively.

4. RESULTS AND DISCUSSION

The training results for the study's model and the vehicle counting results utilising the trained model are provided in this section. Precision (P), recall (R), average precision (AP), mean average precision (mAP), F1-score and counting accuracy metrics, which are often employed in research using object detection algorithms, were utilised to compare the findings obtained in the study. P, R, AP, mAP, F1-score and counting accuracy were calculated as shown in *Equations 7, 8, 9, 10, 11 and 12*:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{7}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{8}$$

where TP, FN and FP are the number of true positives, false negatives and false positives, respectively. Precision is the proportion of all positive samples over the confidence threshold of 50%, whereas recall is the percentage of all positive samples found at the same confidence level.

$$\text{AP} = \frac{1}{11}\sum_{r=0}^{1} p_{max}(r), r \in [0, 0.1, ..., 1], \tag{9}$$

$$\text{mAP} = \frac{\sum \text{AP}}{n}, \tag{10}$$

The average precision on a group of 11 evenly spaced recall levels, known as AP, serves as a summary of the P-R curve. The mean of average precision of all classes is described by the mAP.

$$F1 = \frac{2}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}} \tag{11}$$

$$\text{Counting Accuracy} = \frac{\text{Number of correct detections}}{\text{Number of ground truth detections}} \tag{12}$$

F1 score calculates the harmonic mean of precision and recall, and counting accuracy is calculated by dividing the number of correct detections (the number of correct detections is found by subtracting the total number of errors from ground truth) by the ground truth [52].

Figure 7 depicts the loss, precision, recall and mAP graphs generated from the 500 epochs of training the YOLOv5x algorithm.

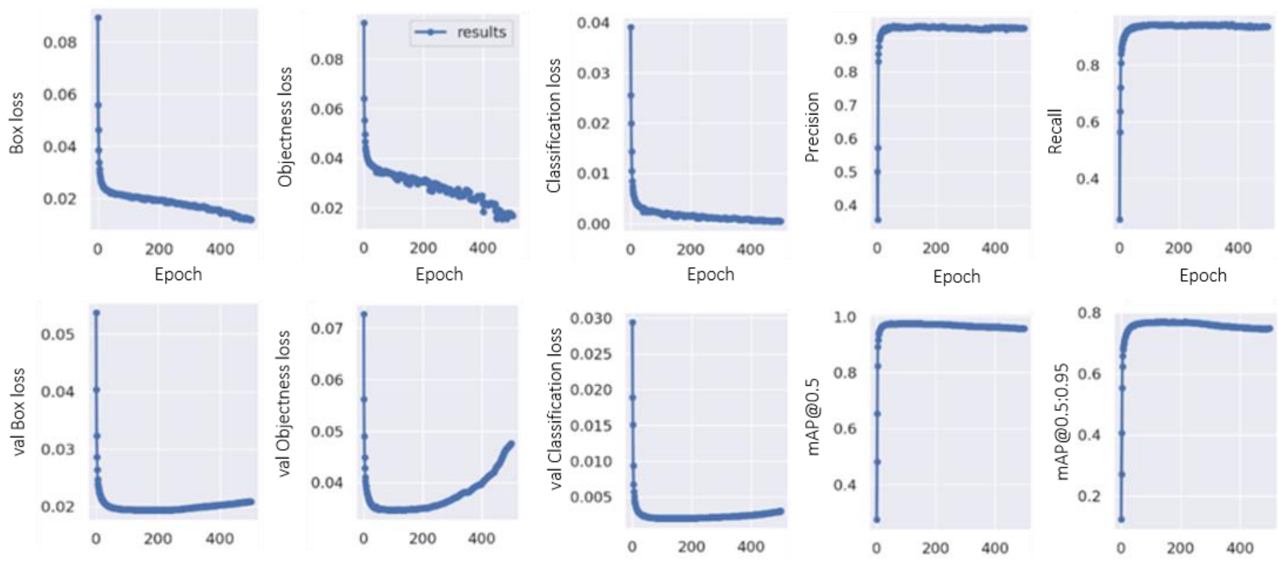

Figure 7 – Loss minimisation and P, R and mAP graphs of the model for the training and validation stage

To improve the generalisation of the model, the YOLOv5x algorithm was trained using data collected from seven different countries. The trained model obtained a score of 97.4 % mAP@0.5 for all classes in the dataset as shown in *Table 4*.

Table 4 – Training results for our model

Class	Precision	Recall	mAP@ 0.5	mAP@ 0.95
Car	0.94	0.94	0.98	0.74
Bus	0.95	0.95	0.97	0.77
Truck	0.95	0.93	0.98	0.77
Van	0.91	0.94	0.97	0.80
All classes	**0.94**	**0.94**	**0.97**	**0.77**

The results of the vehicle counts for daytime traffic images from *Table A1* are summarised in *Table 5*. In daytime conditions, the average counting accuracy for all vehicle classes was recorded as 96%.

Table 5 – Vehicle counting results for daytime

Vehicle type	Ground truth count	Method count	Counting accuracy
Car	4,421	4,328	0.98
Bus	105	118	0.88
Truck	516	508	0.98
Van	816	821	0.99
Total	**5,858**	**5,775**	**0.96**

Table 6 provides an overview of *Table A1* for nighttime vehicle counting data. In nighttime conditions, the average counting accuracy for all vehicle classes was recorded as 97%.

In contrast to other studies, a substantially more significant number of vehicles (10,064) were included for system testing in this study. Examining *Tables 5* and *6* and *Figures 8* and *10*, it is evident that the trained model eliminates the difference between day and night. In other words, with a well-prepared dataset and a single model, it becomes feasible to count traffic during both daytime and nighttime (minimum illumination) conditions. The results of the day and night circumstances were analysed independently since they may affect counting accuracy.

Table 6 – Vehicle counting results for nighttime

Vehicle type	Ground truth count	Method count	Counting accuracy
Car	3,417	3,411	0.99
Bus	85	86	0.99
Truck	514	489	0.95
Van	190	201	0.94
Total	**4,206**	**4,187**	**0.97**

Table A1 reveals that several vehicle classes in some videos have low counting accuracy rates. This is attributable to the fact that there is a limited number of vehicles belonging to the vehicle class specified in the video. As the number of vehicles counted using the traffic data collection system increases, the counting accuracy improves and converges its real value.

Table A1 shows the manual vehicle counting results derived from 29 traffic videos (21 different scenarios) and the vehicle count results performed with the proposed method. All videos used during the training and testing phases of the model are in 1080p resolution. Of the 29 videos utilised in the testing phase of the proposed method, 22 were recorded during daytime, while 7 were recorded at nighttime. However, an examination of *Table 5* and *Table 6* reveals that the vehicle count values for day and night are similar to each other. To transparently illustrate the system's vehicle counting performance, each test video's performance in the system is presented independently and in-depth. The ground truth count, method count, error, precision, recall, F1-score and counting accuracy values for each video can be seen in the mentioned table. The name of the video, duration of the video, side of vehicles seen by camera and time of the day are given in the first column, respectively. The total duration of the test videos was 406 minutes (6.8 hours). *Table 7* provides a summary of all vehicle counts conducted for the study. The ground truth count was determined by manually reviewing the vehicle counting videos one by one and recording the counts. In contrast, the method count was obtained using the proposed method for vehicle detection and counting.

Table 7 – Summary of vehicle counting results for all 29 videos

Vehicle type	Ground truth count	Method count	Counting accuracy
Car	7,838	7,739	0.99
Bus	190	204	0.93
Truck	1,030	997	0.97
Van	1,006	1,022	0.98
Total	**10,064**	**9,962**	**0.97**

Regardless of vehicle class, the counting accuracy rate was 99% when the ground truth count (10,064 vehicles) and total method count (9,962 vehicles) were examined. If the counting accuracy is calculated based on the average of counting accuracies per each vehicle class separately, then it was 97%. In another performance test, likewise to the study by Kang et al. [53], the trained model is tested on a test portion of other mainstream datasets or an appropriate subset of the dataset. These datasets have been created by Guerrero-Gómez-Olmedo et al. [54], Song et al. [24] and Neupane et al. [34] respectively. *Table 8* presents the results that were collected from performance tests.

Table 8 – Performance of our model on mainstream datasets

Dataset	Number of images	Number of vehicles	Counting accuracy
Olmedo et al.	1878	18524	0.93
Song et al.	2225	12490	0.99
Neupane et al.	467	2632	0.94

Headway evaluations (in terms of seconds) made manually and with the proposed method are given in *Table 9*. The 3rd column contains manual evaluation, and the 4th column contains the evaluation results made with the proposed method. Evaluations were conducted on 5 different videos and 1,548 vehicles, as this sample size was deemed sufficient for the study. As a result, headway time was obtained with a 95% accuracy.

In addition, in manual headway evaluations, the reaction time is also included due to the human factor. Therefore, the accuracy of the proposed method may be superior. The headway evaluation performance of the system is also shown in *Figure 9*.

Table 9 – Performance of the proposed method for headway measurement

Video	Vehicle count	Avg. headway (manual)	Avg. headway (method)	Accuracy
30.mp4	302	3.18	2.98	0.94
31.mp4	187	4.10	4.00	0.98
32.mp4	126	3.01	3.46	0.88
33.mp4	532	3.17	3.25	0.98
34.mp4	401	4.35	4.30	0.98
Total	**1548**	**3.56**	**3.60**	**0.95**

Table 10 compares the proposed method with other important studies in the field in recent years. As a result of the method used in the study and the large dataset, it is seen that the model is superior in terms of mAP scores, number of vehicles on which the model is tested and vehicle counting accuracy.

In the second column of *Table 11*, the passenger car equivalent values calculated according to HCM and Turkish Standard Institution (TSE) after manual vehicle counting for various videos are represented. These values were generated by multiplying the number of manually counted vehicles for each vehicle class by the PCE factors for that class.

Table 10 – Proposed method performance evaluation with other existing works

Method	Dataset	Algorithm used	Number of images	mAP@0.5	Vehicle count/counting accuracy
Song et al. (2019)	Custom	Yolov3 coupled with ORB algorithm	11129	87.8	1509/93.2
Harikrishnan et al. (2021)	Custom	Improved SSD coupled with APC	8000	84.6	1002/93.3
Dinh et al. (2021)	Vehicle detection dataset (custom)	SSD MobileDet 320 x 320	4700	92.1	415/83.3
Neupane et al. (2022)	Thai-vehicle-classification-dataset (custom)	Yolov5l coupled with multi-vehicle tracking algorithm	9333	69.5	4697/94.4
Proposed method	**ATA vehicle dataset (custom)**	**Yolov5x-DeepSORT**	**16518**	**97.4**	**10064/97.0**

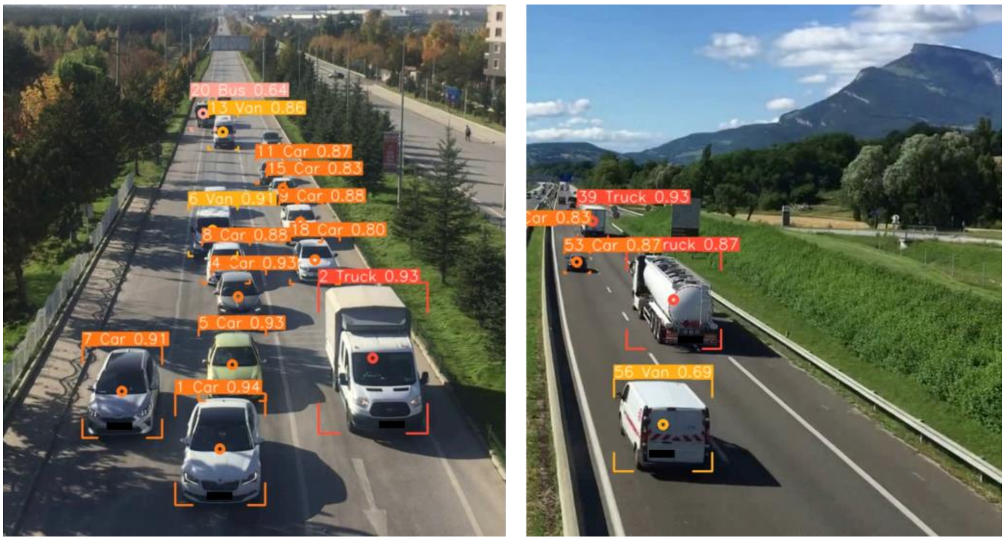

Figure 8 – Trained model object detection in the nighttime

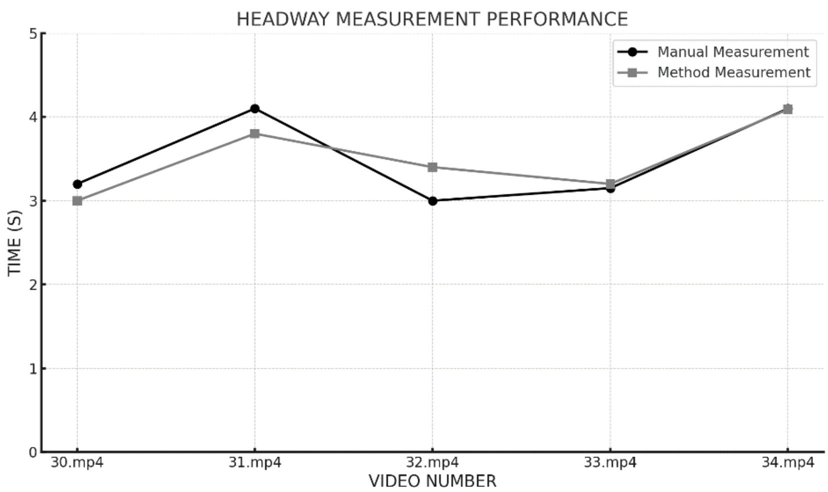

Figure 9 – Manual and proposed method headway measurement curves

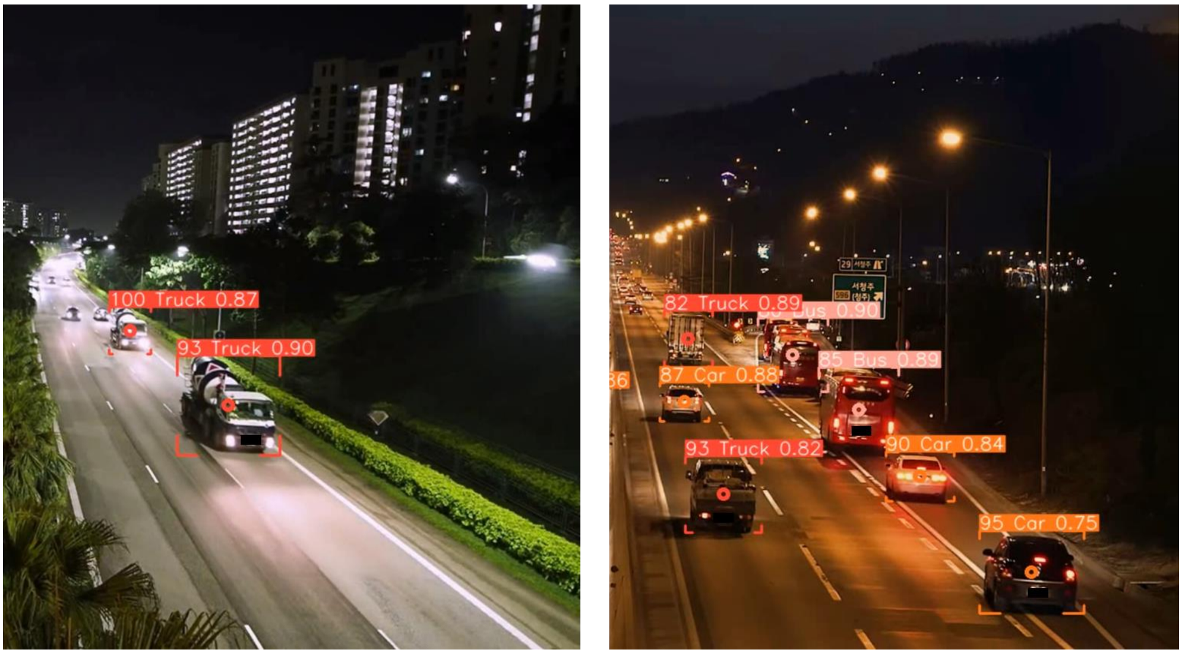

Figure 10 – Trained model object detection in the nighttime

The third column of *Table 11* contains passenger car equivalent values obtained with the traffic data collection system. The passenger car equivalent values represent the number of passenger cars that would have the same influence on traffic flow as the vehicles in the video. Using the PCE factors provided by HCM and TSE, the system's performance in measuring passenger car equivalent values was assessed for 1,587 vehicles, and more than 99% accuracy was attained for both. The vehicle counting data and passenger car equivalent values provide information about the relevant traffic flow and are also used to determine intersection and road capacity.

Table 11 – Passenger car equivalent value results for five videos

Video	Ground truth-passenger car equivalent values (HCM/TSE)	Proposed method passenger car equivalent values (HCM/TSE)
35.mp4	142/151.4	143.5/150.6
36.mp4	501/534.6	507/546.5
37.mp4	226.5/237.9	241.5/225.1
38.mp4	494/502	489.5/496.5
39.mp4	279.5/316.5	279/314.5
Total	**1643/1742.4**	**1633.5/1733.2**

100 observations were taken to assess the spacing data results. *Figure 11* displays the results of 100 spacing observations. The observed spacing data and the spacing values evaluated by the traffic data collection system were determined to be similar, as shown in *Figure 11*. Manual and system measurement data were evaluated, and spacing measurement accuracy was found to be 93%.

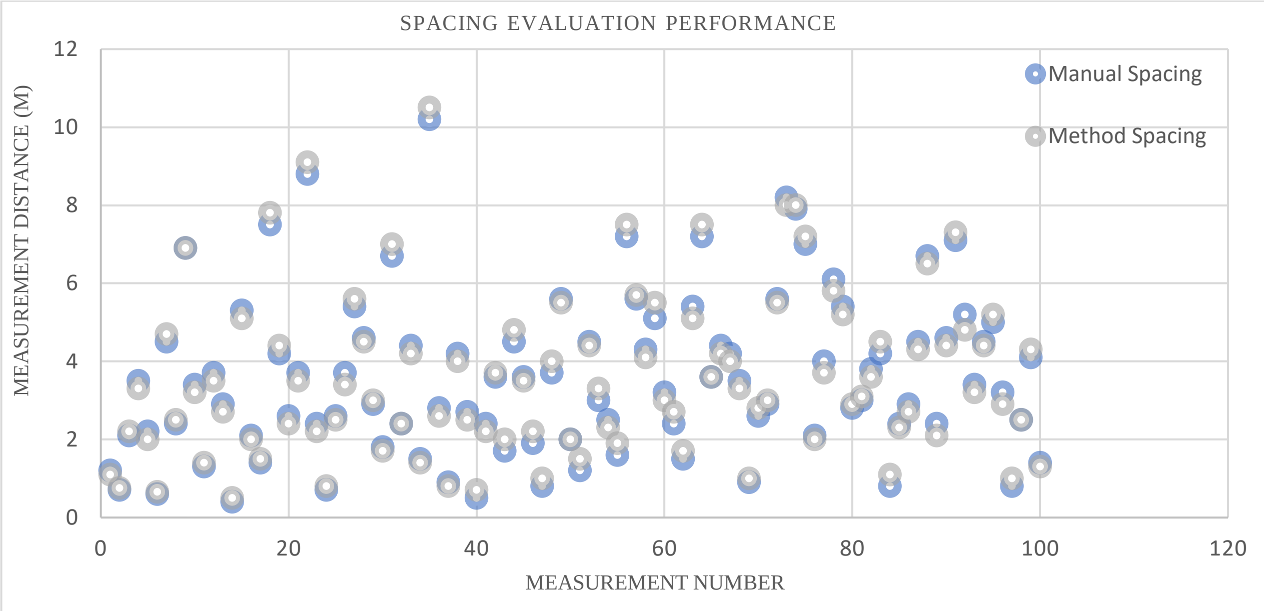

Figure 11 – Manual and proposed method spacing measurement scatter plot graph

5. CONCLUSIONS

The comprehensive traffic data collection framework introduced in this study demonstrates significant advancements in vehicle detection, counting and analysis across diverse conditions and environments. By leveraging a vast, international dataset and state-of-the-art computer vision algorithms, the model successfully maintains high accuracy in daytime and nighttime scenarios, achieving an impressive mean average precision (mAP) of 97.4% and consistent accuracy across headway, spacing and PCE metrics. Notably, this approach resolves issues found in similar studies – such as limited dataset diversity and restricted evaluation scenarios – by incorporating a broad range of vehicle types and environmental conditions, which enhances the generalisability of the model.

Compared to existing methods, this study's model not only streamlines the traffic data collection process but also provides high accuracy in estimating parameters critical for traffic management, including flow rate and intersection capacity through passenger car equivalency (PCE) factors. The testing conducted on over 10,000 vehicles showcases the model's robustness and reliability, highlighting its potential application for adaptive traffic systems in diverse urban environments. Such systems could benefit from enhanced traffic monitoring accuracy, which supports more effective congestion management and contributes to sustainable urban mobility goals.

Future research could build on this work by exploring integration with adaptive traffic control systems and extending the dataset to capture even broader global variability. As urban centres prioritise intelligent transportation systems, this study's approach lays a valuable foundation for further innovations that could make real-time, automated traffic data collection an accessible tool for city planners and transportation engineers globally.

REFERENCES

[1] Lin Y, Wang P, Ma M. Intelligent transportation system (ITS): Concept, challenge and opportunity. *IEEE 3rd International Conference on Big Data Security on Cloud (bigdatasecurity) 2017, 26-28 May. 2017, Beijing, CHINA.* 2017. p. 167–172. DOI: 10.1109/BigDataSecurity.2017.50.

[2] Dinh DL, et al. Towards AI-based traffic counting system with edge computing. *Journal of Advanced Transportation*, 2021:(1):1–15. DOI: 10.1155/2021/5551976.

[3] Abdulhai B, Kattan L. Reinforcement learning: Introduction to theory and potential for transport applications. *Canadian Journal of Civil Engineering*. 2003;30(6):981-991. DOI: 10.1139/l03-014.

[4] Theerthagiri P, Gopala Krishnan C. Vehicular multihop intelligent transportation framework for effective communication in vehicular ad-hoc networks. *Concurrency and Computation: Practice and Experience*. 2022;34(10), e6833. DOI: 10.1002/cpe.6833.

[5] Khan MA, et al. Integrating shared autonomous vehicles into existing transportation services: Evidence from a paratransit service in Arlington, Texas. *International Journal of Civil Engineering*. 2022;20(6):601-618. DOI: 10.1007/s40999-021-00698-6.

[6] Vahidi H, Sayed T. Using the Canadian ITS architecture for evaluating the safety benefits of intelligent transportation systems. *Canadian Journal of Civil Engineering*. 2003;30(6):970-980. DOI: 10.1139/l03-032.

[7] El Hajjami L, et al. A novel robust adaptive neuro-sliding mode steering controller for autonomous ground vehicles. *Robotics and Autonomous Systems*. 2023;170:104557. DOI: 10.1016/j.robot.2023.104557 .

[8] El Hajjami L, Mellouli EM, Berrada M. Robust adaptive non-singular fast terminal sliding-mode lateral control for an uncertain ego vehicle at the lane-change maneuver subjected to abrupt change. *International Journal of Dynamics and Control*. 2021;1-18. DOI: 10.1007/s40435-021-00771-x.

[9] El Hajjami L, et al. A robust intelligent controller for autonomous ground vehicle longitudinal dynamics. *Applied Sciences*. 2022;13(1);501. DOI: 10.3390/app13010501.

[10] El Hajjami L, Mellouli EM, Žuraulis V, Berrada M. Vehicle adaptive cruise controller based on an optimal super-twisting sliding mode control. *2nd International Conference of Smart Systems and Emerging Technologies (SMARTTECH)* 2022, 9-11 May. 2022, Riyadh, SAUDI ARABIA. 2022. p. 160-165. DOI: 10.1109/SMARTTECH54121.2022.00044.

[11] Yuksel E, et al. Data-driven computation of state-dependent passenger car equivalency for multiple truck lengths. *Transportation research record*. 2022;2676(11):194-205. DOI: 10.1177/03611981221092008.

[12] Radhakrishnan P, Mathew TV. Passenger car units and saturation flow models for highly heterogeneous traffic at urban signalised intersections. *Transportmetrica*. 2011;7(2):141-162. DOI: 10.1080/18128600903351001.

[13] Favero R, Setti JR. Factors affecting the ımpact of autonomous vehicles on freeway operations–An exploratory analysis using PCEs. *Promet-Traffic&Transportation*. 2024;36(1):55-68. DOI: 10.7307/ptt.v36i1.356.

[14] Çalışkanelli S, Tanyel S. Investigation of saturation flow value at signalized ıntersections. *TEKNIK DERGI*. 2018;29(1):8225-8248. DOI: 10.18400/tekderg.346592.

[15] Tunç A. *Yol güvenlik mühendisliği ve uygulamaları: Analytical studies in traffic engineering. Chapter II* . Ankara, Türkiye: Asil Yayın Dağıtım; 2003.

[16] Chen X, Li L, Zhang Y. A Markov model for headway/spacing distribution of road traffic. *IEEE Transactions on Intelligent Transportation Systems*. 2010:11(4); 773–785. DOI: 10.1109/TITS.2010.2050141.

[17] Transportation Research Board. *Highway Capacity Manual 2000*. Washington, USA: Transportation Research Board; 2000.

[18] Stepanović N, Tubić V, Zdravković S. Determining free-flow speed on different classes of rural two-lane highways. *Promet-Traffic&Transportation*. 2023:35(3);315-330. DOI: 10.7307/ptt.v35i3.195.

[19] Google Research. *Colaboratory: Frequently asked questions*. Google Research. https://research.google.com/colaboratory/faq.html [Accessed 24 November 2023]

[20] Guido VR, Drake Jr F. *Python 3 reference manual*. *California, USA. CreateSpace*; 2009.

[21] Hajjami LE, et al. Neural network optimization algorithm based non-singular fast terminal sliding-mode control for an uncertain autonomous ground vehicle subjected to disturbances. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*. 2024;238(7);1687-1697.

[22] Park D, et al. Investigating optimal aggregation interval sizes of loop detector data for freeway travel-time estimation and prediction. *Canadian Journal of Civil Engineering*. 2009:36(4);580-591. DOI: 10.1139/L08-129.

[23] Rajput SK, et al. Automatic vehicle identification and classification model using the YOLOv3 algorithm for a toll management system. *Sustainability*. 2022;14(15);9163. DOI: 10.3390/su14159163.

[24] Song H, et al. Vision-based vehicle detection and counting system using deep learning in highway scenes. *European Transport Research Review*. 2019;11(1);1–16.

[25] Bui KHN, Yi H, Cho J. A multi-class multi-movement vehicle counting framework for traffic analysis in complex areas using CCTV systems. *Energies*. 2020;13(8);2036. DOI: 10.3390/en13082036.

[26] Park SH, Yu SB, Kim JA, Yoon H. An all-in-one vehicle type and license plate recognition system using YOLOv4. *Sensors*. 2022;22(3);921. DOI: 10.3390/s22030921.

[27] Xu X, et al. A new deep model for detecting multiple moving targets in real traffic scenarios: Machine vision-based vehicles. *Sensors*. 2022;22(10);3742. DOI: 10.3390/s22103742.

[28] Geiger A, Len, P, Urtasun R. Are we ready for autonomous driving? The kitti vision benchmark suite. *IEEE Conference on Computer Vision and Pattern Recognition 2012, 16-21 Jun. 2012, Province, RI, USA*. 2012. p. 3354–3361. DOI: 10.1109/CVPR.2012.6248074.

[29] Yu F, et al. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020, Jun. 2020. p. 2636–2645. DOI: 10.48550/arXiv.1805.04687.

[30] Zhang et al. Real-time vehicle detection based on improved yolo v5. *Sustainability*. 2022;14(19):12274. DOI: 10.3390/su141912274.

[31] Dong Z, Wu Y, Pei M, Jia Y. Vehicle type classification using a semisupervised convolutional neural network. *IEEE transactions on intelligent transportation systems*. 2015;16(4):2247–2256. DOI: 10.1109/TITS.2015.2402438.

[32] Harikrishnan P, et al. Inception single shot multi-box detector with affinity propagation clustering and their application in multi-class vehicle counting. *Applied Intelligence*. 2021;51:4714–4729. DOI: 10.1007/s10489-020-02127-y.

[33] Khalifa OO, et al. Vehicle detection for vision-based intelligent transportation systems using convolutional neural network algorithm. *Journal of Advanced Transportation*. 2022;1–11. DOI: 10.1155/2022/9189600.

[34] Neupane B, Horanont T, Aryal J. Real-time vehicle classification and tracking using a transfer learning-improved deep learning network. *Sensors*. 2022;22(10);3813. DOI: 10.3390/s22103813.

[35] Djula EJS, Yusuf R. Vehicle detection with YOLOv7 on study case public transportation and general classification, prediction of road loads. *2022 2nd International Seminar on Machine Learning, Optimization, and Data Science (ISMODE), 22-23 Dec. 2022, Jakarta, Indonesia.* 2022. p. 7–11. DOI: 10.1109/ISMODE56940.2022.10180924.

[36] Kalva A, Chelluboina JS, Bharathi B. Smart traffic monitoring system using YOLO and deep learning techniques. *2023 7th International Conference on Trends in Electronics and Informatics (ICOEI), 11-13 Apr. 2023, Tirunelveli, India.* 2023. p. 831–837. DOI: 10.1109/ICOEI56765.2023.10126048.

[37] Chen D, et al. The scanner of heterogeneous traffic flow in smart cities by an updating model of connected and automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*. 2022;23(12);25361–25370. DOI: 10.1109/TITS.2022.3165155.

[38] Liu C, et al. A vision-based pipeline for vehicle counting, speed estimation, and classification. *IEEE transactions on intelligent transportation systems*. 2022;22(12);7547–7560. DOI: 10.1109/TITS.2020.3004066.

[39] Ge L, Dan D, Liu Z, Ruan X. Intelligent simulation method of bridge traffic flow load combining machine vision and weigh-in-motion monitoring. *IEEE Transactions on Intelligent Transportation Systems*. 2022;23(9);15313–15328.

[40] Du M, Tao G, Zhang W. Target detection and tracking of traffic flow at intersections with camera. 2022 34th Chinese Control and Decision Conference (CCDC), 15-17 Aug. 2022, Hefei, China. 2022. p. 725–730. DOI: 10.1109/CCDC55256.2022.10034187.

[41] Jocher G, Nishimura K, Minerva T, Vilariño R. *YOLOv5*. 2021. https://github. com/ultralytics/yolov5 [Accessed 20 March 2023]

[42] National Research Council. *Highway capacity manual 2010*. Transportation research board. Washington, USA: Transportation Research Board; 2000.

[43] M Broström . Real-time multi-object tracker using YOLOv5 and deep sort. 2020. https://github.com/mikel-brostrom/Yolov5_DeepSort_Pytorch [Accessed 5 July 2024]

[44] Tanyel S. *Capacity calculation method for roundabouts in Turkey*. Phd thesis. Istanbul Teknik Üniversitesi; 2001.

[45] Tanyel S, Çalışkanelli SP, Aydin, M, Utku S. The investigation of heavy vehicle effects on roundabouts. *Teknik Dergi*. 2013;24(4).

[46] Turkish Standards Institution. Türk Standartları TS6407:Equivalent car factors for different car types used in urban road transportation. 2013. https://intweb.tse.org.tr/Standard/Standard/Standard.aspx?081118051115108051104119110104055047105102120088111043113104073086075079047106069087102066090067 [Accessed 10 Aug 2023]

[47] Müller R, Kornblith S, Hinton GE. When does label smoothing help? *Advances in Neural İnformation Processing Systems*. 2019. Vol. 32.

[48] Çintaş, E, Özyer B, Şimşek E. Vision-based moving UAV tracking by another UAV on low-cost hardware and a new ground control station. *IEEE Access*. 2020;8:194601–194611.

[49] Wojke N, Bewley A, Paulus D. Simple online and realtime tracking with a deep association metric. *IEEE international conference on image processing (ICIP)2017, 17-20 Sep.2017, Beijing, China. 2017.* p. 3645–3649. DOI: 10.1109/ICIP.2017.8296962.

[50] Zuraimi MAB, Zaman FHK. Vehicle detection and tracking using YOLO and DeepSORT. IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), 03-04 Apr. 2021, Penang, Malaysia. 2021. p. 23–29. DOI: 10.1109/ISCAIE51753.2021.9431784.

[51] Host K, Ivasic-Kos M, Pobar M. Tracking handball players with the DeepSORT algorithm. Proceedings of the 9th International Conference on Pattern Recognition Applications and Methods, 22-24 Feb, 2020, Valletta, Malta. 2020. p. 593–599. 10.5220/0009177605930599.

[52] Rodríguez RC, et al. Mexican traffic sign detection and classification using deep learning. *Expert Systems with Applications*. 2022;202:117247. DOI: 10.1016/j.eswa.2022.117247.

[53] Kang L, Lu Z, Meng L, Gao Z. YOLO-FA: Type-1 fuzzy attention based YOLO detector for vehicle detection. *Expert Systems with Applications*. 2024;237:121209. DOI: 10.1016/j.eswa.2023.121209.

[54] Guerrero-Gómez-Olmedo R, et al. Extremely overlapping vehicle counting. *Pattern Recognition and Image Analysis: 7th Iberian Conference 2015, 17-19 Jun. 2015, Santiago de Compostela, Spain*. 2015. p. 423–431. DOI: 10.1007/978-3-319-19390-8_48.

Burak ÇELİK, Ahmet TORTUM, Emre ÇİNTAŞ, Barış ÖZYER

Bilgisayarlı Görü ve Uluslararası Veriseti Kullanılarak Headway-Spacing'in Analiz Edilmesi ve Eşdeğer Birim Otomobil Değerlerinin Hesaplanması

Öz

İsabetli trafik akışı verileri, etkili ulaşım planlaması ve yönetimi için çok önemlidir. Farklı araç tipleri trafik akışını değişken bir şekilde etkiler, kavşak ve yol kapasitesini hesaplamak için farklı Eşdeğer Birim Otomobil (PCE) faktörleri gerektirir. Headway ve spacing verileri, trafik yoğunluğunu ve hizmet seviyesini değerlendirmek için önemlidir. Geleneksel veri toplama yöntemleri zaman alıcıdır ve genellikle yanlıştır. Bu çalışma, mevcut çalışmalardan farklı olarak, karma trafik akışı hacmini eşdeğer otomobil birimi cinsinden ölçmek ve headway-spacing verilerini yüksek doğrulukla toplamak için bilgisayarlı görüyü kullanmıştır. Araç algılama ve sayım prosedürleri, karma trafik akışı hacmini ölçmek ve headway ve spacing verilerini toplamak için zorunlu altyapıyı sağlar. Tek bir sistem kullanılarak, eşdeğer birim otomobil değerleri, headway, spacing, akım oranı, araç hızları ve trafik hacmi dahil olmak üzere kapsamlı trafik verileri toplamak için yeni yaklaşımlar tanıtılmıştır. Bu yaklaşımları analiz etmek için özel ve kapsamlı bir uluslararası veri kümesi toplanmıştır. Eğittiğimiz model %97,4 mean Average Precision (mAP) değeriyle birlikte, headway için %95, spacing için %93 ve PCE değerleri için %99 isabet oranı elde etmiştir.

Anahtar Kelimeler

Akıllı Ulaşım Sistemleri; Trafik Akım Verisi; Bilgisayarlı Görü; Headway-Spacing Verisi; Birim Otomobil Eşdeğerliği.

Table A1 – Vehicle count results table

Video	Vehicle type	Ground truth count	Method count	Missing/multiple detection/error	Precision %	Recall %	F1-Score	Counting accuracy %
1.mp4 (2 min) Front Daytime	Car	151	153	0/2/2	1	0.99	0.99	0.99
	Bus	0	0	0/0/0	1	1	1	1
	Truck	9	6	3/0/3	0.67	1	0.8	0.67
	Van	13	12	1/0/1	0.92	1	0.96	0.92
	Total/Avg.	173	171	4/2/6	0.90	1	0.94	0.89
2.mp4 (2 min) Back Daytime	Car	96	94	2/0/2	0.98	1	0.99	0.98
	Bus	0	0	0/0/0	1	1	1	1
	Truck	6	6	0/0/0	1	1	1	1
	Van	5	5	0/0/0	1	1	1	1
	Total/Avg.	107	105	2/0/2	0.99	1	1	0.99
3.mp4 (1 min) Front Daytime	Car	47	46	1/0/1	0.98	1	0.99	0.98
	Bus	6	6	0/0/0	1	1	1	1
	Truck	1	1	0/0/0	1	1	1	1
	Van	8	10	0/2/2	1	0.8	0.89	0.75
	Total/Avg.	62	63	1/2/3	1	0.95	0.97	0.93
4.mp4 (6 min) Front Daytime	Car	148	148	0/0/0	1	1	1	1
	Bus	0	0	0/0/0	1	1	1	1
	Truck	15	16	0/1/1	1	0.94	0.97	0.93
	Van	29	28	1/0/1	0.97	1	0.98	0.97
	Total/Avg.	192	192	1/1/2	0.99	0.98	0.99	0.97
5.mp4 (5 min) Front Daytime	Car	58	58	0/0/0	1	1	1	1
	Bus	4	4	0/0/0	1	1	1	1
	Truck	14	13	1/0/1	0.93	1	0.96	0.93
	Van	18	19	0/1/0	1	0.95	0.97	0.94
	Total/Avg.	94	94	1/1/2	0.98	0.99	0.98	0.97
6.mp4 (14 min) Front Daytime	Car	374	372	2/0/2	0.99	1	1	0.99
	Bus	0	0	0/0/0	1	1	1	1
	Truck	50	51	0/1/1	1	0.98	0.99	0.98
	Van	116	116	0/0/0	1	1	1	1
	Total/Avg.	540	539	2/1/3	1	1	1	0.99
7.mp4 (14 min) Back Daytime	Car	256	259	0/3/3	1	0.99	0.99	0.99
	Bus	0	0	0/0/0	1	1	1	1
	Truck	46	45	1/0/1	0.98	1	0.99	0.98
	Van	100	97	3/0/3	0.97	1	0.98	0.97
	Total/Avg.	402	401	4/3/7	0.98	1	0.99	0.98
8.mp4 (6 min) Front Daytime	Car	52	52	0/0/0	1	1	1	1
	Bus	1	1	0/0/0	1	1	1	1
	Truck	8	8	0/0/0	1	1	1	1
	Van	18	16	2/0/2	0.89	1	0.94	0.89
	Total/Avg.	79	77	2/0/2	0.97	1	0.99	0.97

Video	Vehicle type	Ground truth count	Method count	Missing/multiple detection/error	Precision %	Recall %	F1-Score	Counting accuracy %
9.mp4 (5 min) Back Daytime	Car	38	41	0/3/3	1	0.93	0.96	0.92
	Bus	6	6	0/0/0	1	1	1	1
	Truck	5	2	3/0/3	0.4	1	0.57	0.4
	Van	22	23	0/1/1	1	0.96	0.98	0.95
	Total/Avg.	71	72	3/4/7	0.85	0.97	0.88	0.82
10.mp4 (5 min) Back Daytime	Car	191	178	13/0/13	0.93	1	0.96	0.93
	Bus	9	12	0/3/3	1	0.75	0.86	0.67
	Truck	32	40	0/8/8	1	0.8	0.89	0.75
	Van	38	41	0/3/3	1	0.93	0.96	0.92
	Total/Avg.	270	271	13/14/27	0.98	0.87	0.92	0.82
11.mp4 (9 min) Front Daytime	Car	138	135	3/0/3	0.98	1	0.99	0.98
	Bus	5	5	0/0/0	1	1	1	1
	Truck	11	11	0/0/0	1	1	1	1
	Van	33	35	0/2/2	1	0.94	0.94	0.94
	Total/Avg.	187	186	3/2/5	0.99	0.99	0.99	0.98
12.mp4 (1 min) Back Daytime	Car	146	146	0/0/0	1	1	1	1
	Bus	0	0	0/0/0	1	1	1	1
	Truck	20	23	0/3/3	1	0.87	0.93	0.85
	Van	8	9	0/1/1	1	0.89	0.94	0.88
	Total/Avg.	174	178	0/4/4	1	0.94	0.97	0.93
13.mp4 (22 min) Front Nighttime	Car	380	371	9/0/9	0.98	1	0.99	0.98
	Bus	5	7	0/2/2	1	0.71	0.83	0.6
	Truck	33	37	0/4/4	1	0.89	0.94	0.88
	Van	64	70	0/6/6	1	0.91	0.96	0.91
	Total/Avg.	482	485	9/12/21	0.99	0.88	0.93	0.84
14.mp4 (5 min) Front Daytime	Car	180	173	7/0/7	0.96	1	0.98	0.96
	Bus	3	3	0/0/0	1	1	1	1
	Truck	2	2	0/0/0	1	1	1	1
	Van	39	34	5/0/5	0.87	1	0.93	0.87
	Total/Avg.	224	212	12/0/12	0.96	1	0.98	0.96
15.mp4 (34 min) Back Nighttime	Car	982	1004	0/22/22	1	0.98	0.99	0.98
	Bus	45	49	0/4/4	1	0.92	0.96	0.91
	Truck	314	316	0/2/2	1	0.99	1	0.99
	Van	50	55	0/5/5	1	0.91	0.95	0.9
	Total/Avg.	1391	1424	0/33/33	1	0.95	0.97	0.95
16.mp4 (28 min) Back Nighttime	Car	477	472	5/0/5	0.99	1	0.99	0.99
	Bus	4	3	1/0/1	0.75	1	0.86	0.75
	Truck	2	2	0/0/0	1	1	1	1
	Van	8	10	0/2/2	1	0.8	0.89	0.75
	Total/Avg.	491	487	6/2/8	0.93	0.95	0.94	0.87

Video	Vehicle type	Ground truth count	Method count	Missing/multiple detection/error	Precision %	Recall %	F1-Score	Counting accuracy %
17.mp4 (10 min) Front Daytime	Car	109	110	0/1/1	1	0.99	1	0.99
	Bus	4	5	0/1/1	1	0.8	0.89	0.75
	Truck	5	5	0/0/0	1	1	1	1
	Van	40	40	0/0/0	1	1	1	1
	Total/Avg.	158	160	0/2/2	1	0.95	0.97	0.94
18.mp4 (10 min) Back Daytime	Car	121	120	1/0/1	0.99	1	1	0.99
	Bus	3	3	0/0/0	1	1	1	1
	Truck	6	5	1/0/1	0.83	1	0.91	0.83
	Van	43	46	0/3/3	1	0.93	0.97	0.93
	Total/Avg.	173	174	2/3/5	0.96	0.98	097	0.94
19.mp4 (3 min) Back Daytime	Car	194	198	0/4/4	1	0.98	0.99	0.98
	Bus	9	9	0/0/0	1	1	1	1
	Truck	46	43	3/0/3	0.93	1	0.97	0.93
	Van	3	3	0/0/0	1	1	1	1
	Total/Avg.	252	253	3/4/7	0.98	0.99	0.99	0.98
20.mp4 (3 min) Front Daytime	Car	158	148	10/0/10	0.94	1	0.97	0.94
	Bus	6	7	0/1/1	1	0.86	0.92	0.83
	Truck	54	45	9/0/9	0.83	1	0.91	0.83
	Van	15	19	0/4/4	1	0.79	0.88	0.73
	Total/Avg.	233	219	19/5/24	0.94	0.91	0.92	0.83
21.mp4 (8 min) Front Daytime	Car	225	251	0/26/26	1	0.90	0.95	0.88
	Bus	6	8	0/2/2	1	0.75	0.86	0.83
	Truck	31	30	1/0/1	0.97	1	0.98	0.97
	Van	20	19	1/0/1	0.95	1	0.97	0.95
	Total/Avg.	282	308	2/28/30	0.98	0.91	0.94	0.91
22.mp4 (23 min) Front Daytime	Car	130	132	0/2/2	1	0.98	0.99	0.98
	Bus	0	0	0/0/0	1	1	1	1
	Truck	3	3	0/0/0	1	1	1	1
	Van	20	20	0/0/0	1	1	1	1
	Total/Avg.	153	155	0/2/2	1	1	1	1
23.mp4 (23 min) Back Nighttime	Car	171	175	0/4/4	1	0.98	0.99	0.98
	Bus	0	0	0/0/0	1	1	1	1
	Truck	0	0	0/0/0	1	1	1	1
	Van	39	39	0/0/0	1	1	1	1
	Total/Avg.	210	214	0/4/4	1	0.99	1	0.99
24.mp4 (34 min) Front Nighttime	Car	519	549	0/30/30	1	0.95	0.97	0.94
	Bus	24	20	4/0/4	0.83	1	0.91	0.83
	Truck	165	134	31/0/31	0.81	1	0.90	0.81
	Van	34	29	5/0/5	0.85	1	0.92	0.85
	Total/Avg.	742	732	40/30/70	0.87	0.99	0.92	0.86

Video	Vehicle type	Ground truth count	Method count	Missing/multiple detection/error	Precision %	Recall %	F1-Score	Counting accuracy %
25.mp4 (29 min) Front Nighttime	Car	523	496	27/0/27	0.94	1	0.97	0.95
	Bus	1	1	0/0/0	1	1	1	1
	Truck	0	0	0/0/0	1	1	1	1
	Van	16	15	1/0/1	0.94	1	0.97	0.94
	Total/Avg.	540	512	28/0/28	0.97	1	0.99	0.97
26.mp4 (29 min) Back Nighttime	Car	536	519	17/0/17	0.97	1	0.98	0.97
	Bus	6	6	0/0/0	1	1	1	1
	Truck	0	0	0/0/0	1	1	1	1
	Van	18	22	0/4/4	1	0.82	0.9	0.78
	Total/Avg.	560	547	17/4/21	0.99	0.95	0.97	0.94
27.mp4 (30 min) Front Daytime	Car	332	295	37/0/37	0.89	1	0.94	0.89
	Bus	20	24	0/4/4	1	0.83	0.91	0.8
	Truck	39	34	5/0/5	0.87	1	0.93	0.87
	Van	29	30	0/1/1	1	0.97	0.98	0.97
	Total/Avg.	420	383	42/5/47	0.94	0.95	0.94	0.88
28.mp4 (30 min) Back Daytime	Car	535	544	9/0/9	0.98	1	0.99	0.98
	Bus	11	9	0/2/2	1	0.82	0.9	0.78
	Truck	40	43	3/0/3	0.93	1	0.96	0.93
	Van	53	43	0/10/10	1	0.81	0.90	0.77
	Total/Avg.	639	639	12/12/24	0.98	0.91	0.94	0.86
29.mp4 (15 min) Front Daytime	Car	509	562	53/0/53	0.91	1	0.95	0.91
	Bus	14	14	0/0/0	1	1	1	1
	Truck	79	70	0/9/9	1	0.89	0.94	0.87
	Van	107	117	10/0/10	0.91	1	0.96	0.91
	Total/Avg.	709	763	63/9/72	0.96	0.97	0.96	0.92