TOWARDS THE NEXT GENERATION INTELLIGENT TRANSPORTATION SYSTEM: A VEHICLE DETECTION AND COUNTING FRAMEWORK FOR UNDISCIPLINED TRAFFIC CONDITIONS

S.H. Ahmed[*], *M. Raza*[*], *M. Kazmi*[*], *S.S. Mehdi*[*], *I. Rehman*[*], *S.A. Qazi*[†]

Abstract: Modern development in deep learning and computer vision techniques, intelligent transportation system (ITS) has emerged as a useful tool for building a traffic infrastructure in smart cities. Previously, several computer vision techniques have been proposed for vehicle recognition, which were limited in handling undisciplined, dense and laneless traffic conditions. Moreover, these frameworks did not incorporate many of the local vehicle configurations common in South Asian countries such as Pakistan, Bangladesh, and India. Considering the limitations of previous frameworks, this paper presents efficient vehicle detection and counting model for undisciplined conditions including dense and laneless traffic, occulusion cases and diverse range of local vehicles. A dataset of more than 2400 images of vehicles has been collected comprising of six new categories of local vehicles, and considering undisciplined traffic conditions to ensure robustness in vehicle detection and counting system. Transfer learning based technique has been used, using faster R-CNN model with Inception V2 as underlying architecture. The experimental results show a precision of 86.14% in terms of mAP. The work finds its application in South Asian contexts as more smart cities are formed in this region. The proposed framework will enable traffic monitoring with higher reliability, accuracy and granularity, contributing in having next-generation ITS.

Key words: *intelligent transportation systems, vehicle detection, image classification, recurrent neural networks, computer vision, transfer learning*

Received: November 8, 2022
Revised and accepted: June 30, 2023

DOI: 10.14311/NNW.2023.33.011

[*]Syeda Hafsah Ahmed; Mehwish Raza; Majida Kazmi – Corresponding author; Syeda Shajeeha Mehdi; Inshal Rehman; Faculty of Electrical and Computer Engineering, NED University of Engineering & Technology, Karachi, Sindh, Pakistan, E-mail: syedahafsa007@gmail.com, mehwishraza@neduet.edu.pk, shajeeham@gmail.com, inshal.rehman@yahoo.com, majidakazmi@neduet.edu.pk

[†]Saad Ahmed Qazi; Neurocomputation Lab, National Centre of Artificial Intelligence, NED University of Engineering & Technology, Karachi, Sindh, Pakistan E-mail: saadqazi@neduet.edu.pk

1. Introduction

In urban cities, as the number of vehicles keeps on increasing with the inadequate infrastructure of traffic, vehicle accidents and traffic congestion become a serious issue to deal with. This causes a plethora of problems for today's society, including the degradation of urban traffic conditions, difficulty in vehicle tracking, and fatal accidents. The advanced solution for these problems is the intelligent transportation system (ITS) [12]. With the help of emerging technologies, modern cities can rely on ITS to reduce congestion issues and their negative effects as well as to gain additional benefits from the processed data. Vehicle detection and counting in road monitoring video streams are of great significance in ITS and are promising applications in the future technology of vehicle to vehicle (V2V) communication. It enables the extraction of vehicle images from image sequences or video, using different image processing algorithms [1, 50]. Vehicle detection and counting is a challenging task to perform due to occlusion, varying sizes, orientations, and types of vehicles. It is a crucial yet challenging part of ITS which can assist authorities in extracting useful information for making informed traffic decisions [45]. Moreover, it can be utilized in many other aspects including vehicle theft, large-scale parking lot management, safety management of highways, electronic toll collection (ETC), traffic investigation, etc.

Recent developments in deep learning (DL) technique make it interesting area of research for vehicle detection and classification. The DL techniques have shown promising results with better adaptability as compared to the classical machine learning approach. Convolutional neural network (CNN) based DL are very efficient in learning features automatically and are capable to perform various multiple tasks such as classification and bounding box regression [7, 16, 21, 51]. However, CNN based vehicle detection requires a huge amount of data to obtain good accuracy. The dataset must be general and cover the real traffic conditions to achieve generalizability. However, most of the existing research in vehicle detection has been conducted on datasets that are for disciplined traffic. Further, the available vehicle datasets like CompCar [49], Stanford [3] and PSU [34] are very small datasets based on very limited classes. Moreover, they do not include local vehicle categories of South Asian countries. To address the limitations of traditional vehicle detection systems, following main contributions have been made.

- The proposed framework is specifically tailored for the local transportation that is prevalent in South Asian countries like Pakistan, Bangladesh, and India. The research offers a new dataset that is more diversified and reflective of the actual traffic circumstances in this region. It consists of over 2400 images and six new categories of nearby local automobiles, namely cars, rickshaws, trucks, bikes, medium-sized vehicles (MSV), and buses
- The work presents an efficient solution for vehicle detection in dense and diverse traffic scenarios with occlusion and bumper-to-bumper traffic, which is a common problem in undisciplined traffic surveillance. The framework is capable to detect vehicles accurately even in undisciplined traffic conditions with visual occlusion and outperforms the existing models trained on limited classes of modern vehicles.

– A flexible vehicle counting framework is presented in three different modes of operation: total vehicle counts concerning the region of interest (ROI), frame by frame count, and single photo count.

The rest of the paper is organized as follows. In Section 2, the prevalent vehicle-detection methods are discussed briefly. In Section 3, the proposed methodology has been elaborated. The experimental setup and results are explained in Section 4 and Section 5 respectively. Finally, the article is concluded in Section 6.

2. Literature review

Vehicles are difficult to recognize in a range of road conditions and dynamic environments, thus detecting them in camera photos is always a challenge. Vehicle detection and recognition is a critical yet difficult task since the vehicle image is distorted and impacted by a number of factors. With each new car model, the number of vehicle kinds increases at first. Following that, there are many similarities between various vehicle types, and there are also significant differences across vehicle images due to differences in street circumstances, climate, lighting, and cameras. The field of vehicle detection can been divided into two categories depending on the techniques that are employed for feature extraction; manual feature extraction, and data-driven feature extraction approach.

In classical vehicle detection algorithms, the system usually finds features of an object or group of objects manually in the very beginning. After that, it tries to categorize these features into different classes via classification models. In the last step, the system decides which category the object belongs to. The biggest strength of this method is that the geometry and features of the targets are known before the detection. While it can be a weakness or limitation as well because it can only detect pre-learned objects. These traditional algorithms have three basic processes: region selection, feature extraction, and classification. Haar [28] and histogram of oriented gradients (HOG) [41] are the most commonly used feature extractors. The feature extractors are usually followed by classifiers like support vector machine (SVM) [35], decision trees, and AdaBoost [48].

The sums and differences of rectangles over an image patch are used to calculate Haar features [28]. It was best suited for real-time detection since it was efficient at computing the symmetric structure of objects to be detected. HOG features are extracted in two steps: evaluating edge operators over the image, discretizing, and discarding edge intensities' orientations into the histogram. Although this method is more time consuming, it has achieved good results for pedestrian detection [8]. The HOG feature vector has long been used in conjunction with the SVM classifier to determine the orientation of the object. In object detection, the HOG-SVM [8] had a lot of success. Authors in [10] have presented a comparative study of SVM and decision trees models, using the HOG algorithm for manual feature extraction, for vehicle detection and tracking. The dataset used is extracted from the Udacity website that consists of vehicle and non-vehicle samples of the KITTI benchmark suite. The models are trained with the same dataset and the evaluation result showed that SVM performs better for vehicle detection and tracking than decision tree. In [47], a vehicle detection system has been presented with occlusion handling, tracking, counting, and classification. Vehicle detection has been performed using the background subtraction technique using the adaptive gaussian mixture model. Further blobs were analyzed to extract geometrical features of vehicles. An algorithm has been proposed to handle the occlusion. Adaptive Kalman filter is used for vehicle tracking. For classification k-mean clustering, SVM, OC-SVM has been employed, classifying into three classes small, midsize and large vehicles. The vehicle detection results obtained range between 68.7%–99.5% mAP. The dataset used is highly disciplined and covers only a few vehicle types such as car, van, truck, and a big truck. In such ideal circumstances, traditional approaches demonstrated good accuracy but result deteriorates in the presence of shadows, occluded vehicles, complex scenarios, and settings.

In contrast to the manual feature extraction methods used in vehicle detection, the DL methodology extracts the features that are later employed in vehicle recognition using a data-driven algorithm. DL-based algorithms utilize the statistical data of images to learn the appearance of an object automatically. Furthermore, the CNN features are more prominent where feature learning process mimics the human visual mechanism [27]. DL object detection techniques are divided into two categories: two-stage object detection and one-stage object detection, owing to differences in network topology and feature learning progress. The two-stage object detection models are region based. These models perform image processing in two separate stages. In the first stage, the region of interest is retrieved from images and converted into a series of sparse candidate frames. Later in the second stage, the objects are classified and the location of the object is further refined. Faster R-CNN [36], mask R-CNN [20], and R-FCN [9] are widely used algorithms for two-stage detection model.

Region-based CNN (R-CNN) [16] is one of the initial extensions of CNN that uses a selective search method to extract features. But R-CNN was slow in computation and expensive in terms of storage; therefore it was not suitable for huge datasets [17]. As an alternative, another object detection technique is fast R-CNN [17], which showed a little improvement in terms of computation time than R-CNN. Faster R-CNN is a modified version of fast R-CNN. It uses region proposal network (RPN) for feature extraction instead of selective search. Another faster R-CNN breakthrough is by smoothing L1 loss; the loss function smoothens the training progress and improves detection outcomes. Mask R-CNN is an extension of faster R-CNN. In addition to the outputs like a class label and bounding box offset, mask R-CNN also gives an additional output of the object mask for each region of interest (ROI). Mask R-CNN is relatively simpler but due to the mask object, it adds a little computational overhead as compared to faster R-CNN. Region-based fully convolutional networks (R-FCN) [9], is a modified form of faster R-CNN to boost the speed and to incorporate translational variance into fully convolutional network (FCN). Although R-FCN has a better inference time, faster R-CNN gives a more accurate prediction than R-FCN [9].

In one-stage detection techniques, the prediction results are obtained in one step, directly from the image. The primary idea behind one-stage detection is to evenly undertake extensive sampling in multiple areas of the images, with varying sizes and aspect ratios, and then apply CNN to extract features before directly classifying and regressing, all in one step. As a result, while uniform intense sampling has the advantage of being quick, it also has the disadvantage of being more complicated to train, resulting in slightly lower model accuracy. Single shot detection [30], RetinaNet [29], and YOLO [6, 37–39] are the main algorithms of the one-stage detection model.

SSD [30] was proposed in 2015; to transform the inspection task into a uniform, end-to-end regression problem, and obtains the position and classification simultaneously by only one process. It extracts multi-scale features from a scene, where small-scale features help to detect a large object and vice versa. The main advantage of SSD is it works well with low-resolution images, which reduces the need of using expensive sensors to some extent. You only look once (YOLO) [6, 37–39] uses the CNN network to implement the detection, the training and prediction process is end-to-end, the algorithm is fast and straightforward, YOLO does the convolutional calculation of the whole picture, so it has the advantage of a larger field of view during the detection, and is not easy to misjudge the background. The full convolutional layer serves the purpose of the attention module. Besides, the generalization capability of YOLO is well, and the model robustness is high when migrating. Moreover, the new feature extraction network (DarkNet-19), adaptive anchor box, and multi-scale training make YOLO detection performance well. The one-stage algorithms work well with speed and simplicity due to end-to-end computation, but when it comes to accuracy, two-stage algorithms are considered a better choice. RetinaNet [29] solves this problem that the loss of simple samples can cover the loss of a large number of complicated cases, and it is a one stage algorithm model with accuracy comparable to the two-stage detection algorithm. However, compared with YOLO, SSD, RetinaNet's inference time is slow.

In [43], a comparative study, of vehicle detection algorithms based on a deep learning model, has been presented. The faster R-CNN and single shot detector (SSD) models were trained on India driving dataset (IDD), the dataset is constructed from the highly unstructured roads where lane discipline is not followed. It was observed that faster R-CNN outperforms the SSD in terms of accuracy. However, the amount of data collected is not diverse and not sufficient enough to achieve the desired accuracy. In [24] the researchers proposed a vehicle detection model based on tiny YOLO V2 and used a combination of CNN and SVM to classify objects. The proposed model has been trained on BIT-Vehicle produced by the Beijing Institute of Technology [11]. The vehicle dataset is restricted to big size cars and divided into six categories: bus, microbus, minivan, sedan, SUV, and truck. Detection results are found to be 0.82 IOU and an average recall of 94.45% with a threshold of 0.8. In [31], five major image processing models R-FCN, mask R-CNN, SSD, RetinaNet, and YOLO V4 have been compared on Berkley DeepDrive dataset [50]. The study concludes with YOLO V4 outperforming all the other algorithms with an average precision of 88.67% without occlusion and 73.09% with occlusion. The dataset used covers only disciplined traffic and does not contain local vehicle classes. In [32], a review of different vehicle detection algorithms has been presented. The paper compared the performance of five major deep learning models (faster R-CNN, R-FCN, SSD, YOLO V3, RetinaNet), for vehicle detection on the KITTI [14] dataset. The data is divided into three categories easy, moderate and hard based on occlusion. The RetinaNet algorithm works better in terms of precision than other algorithms but inference time is relatively slower. In [42], a vehicle type detection model has been proposed based on YOLO V2 with an enhanced feature extraction approach using a multi-layer fusion strategy. The study is conducted on two datasets obtained from BIT [21] and CompCars [49]. Although the model reached an accuracy of 94.78%, the data used was not diverse and the amount of data was very low. In [37], an improved model of faster R-CNN has been proposed, trained on the KITTI [14] dataset. The improved model depicted an accuracy of 83%. In [3], a comparative study was presented on vehicle detection using aerial images. The performance of three state-of-the-art CNN algorithms faster R-CNN, YOLO V3, and YOLO V4. The algorithms were tested on two different datasets Stanford dataset [3] and Prince Sultan University dataset [34]. The study highlighted the fact that the clarity of the features, the quality of annotation, and the representation of the learning dataset are more important than the actual size of the dataset, as the three algorithms gave better results on the PSU dataset than the Stanford dataset.

Most of the research discussed above has been done using the accessible public datasets that only reflect disciplined traffic conditions and include only vehicle types, which are common in well-developed countries. The greater part of the researchers' examination principally centers around the characterization of vehicles into general classes, like motorbike, vehicles, transports, or trucks [12, 40], yet this doesn't give adequate usefulness to fulfill requirements of Asian countries like Pakistan, Bangladesh, and India with local vehicles in undisciplined traffic conditions. In [43], the researchers target Indian traffic, however, their accuracy is low. This indicates the need for an efficient vehicle detection framework alongside the dataset that includes the local transportation, i.e., conventional trucks, local cars, rickshaws, and motorbikes in heavy undisciplined traffic conditions. In this paper, an efficient vehicle detection and counting framework is proposed based on transfer learning technique using faster R-CNN model with Inception V2 as underlying architecture, using a self collected dataset from Karachi, Pakistan.

3. Methodology

The proposed system is comprised of following steps: Data collection, data preparation, data pre-processing, detection, localization, and vehicle counting. As shown in Fig. 1, firstly, the dataset is collected by installing cameras on the pedestrian bridge. The footage obtained from the mounted camera is pre-processed and transformed into frames containing different classes of vehicles. The vehicles in each frame are annotated and converted into XML files which are then divided into train and testing datasets.

The model is trained on the training dataset and upon completion of the training, the inference graph is frozen that holds information like weight, graph into a single file. The loss is computed and the model is evaluated using object detection evaluation metrics, i.e., mAP. This trained model is then used to perform detection and counting on the given input in form of video and image. After passing through the detection phase, the information related to bounding boxes, detected objects is used to count the vehicle respective to their classes. The resulting output is written on the given video/image and the records are saved in a CSV file.

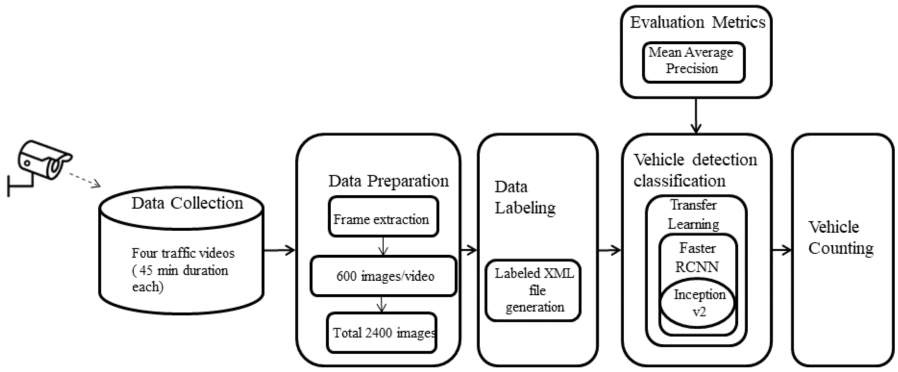

Fig. 1 *Proposed framework for local vehicle detection and counting.*

3.1 Data collection

The dataset is a critical input in DL based classification systems since it helps the algorithms learn the features and produce predictions based on the learnt information. There is currently no vehicle dataset available that covers full range of automobiles from Asian nations such as Pakistan, India, and Bangladesh to address detection and counting issues. Keeping in mind the aforementioned issues, the dataset was created by deploying cameras in various places of Karachi. The cameras were installed on pedestrian bridges to obtain a frontal view of the automobiles during the day. Traffic footage from several areas of Karachi were collected and then processed to extract vehicle images. Four 45 minutes long videos were obtained from the traffic footage. Each video produced 600 images to work on. In total, 2400 images were used. In Pakistan, SECAM standards are followed for which the standard frame rate for videos is 25 FPS. The frame rate for the video dataset used in this paper is therefore set to standard 25 FPS and the resolution of data set images is 1280×738 pixels. As shown in Fig. 2(a), the complexity of the dataset was medium to high due to the occlusion and overlapping of vehicles. Several instances of vehicles overshadowed following vehicles due to high density and lack of lane discipline.

3.2 Data preparation

Following data gathering, the next step is data preparation. To train the model, the videos were transformed into images using frame extraction at proper recording rate. Smaller recording speeds would result in labeling the same car many times because it would capture more photos than needed. Larger recording speeds, on the other hand, may result in fewer photos, resulting in the omission of many cars that required to be labeled.

To identify the entire image of vehicles without redundancy, a recording rate that adjusts to traffic flow was chosen. The videos were converted to images using a recording ratio of 90, which means that a snapshot of the traffic stream was

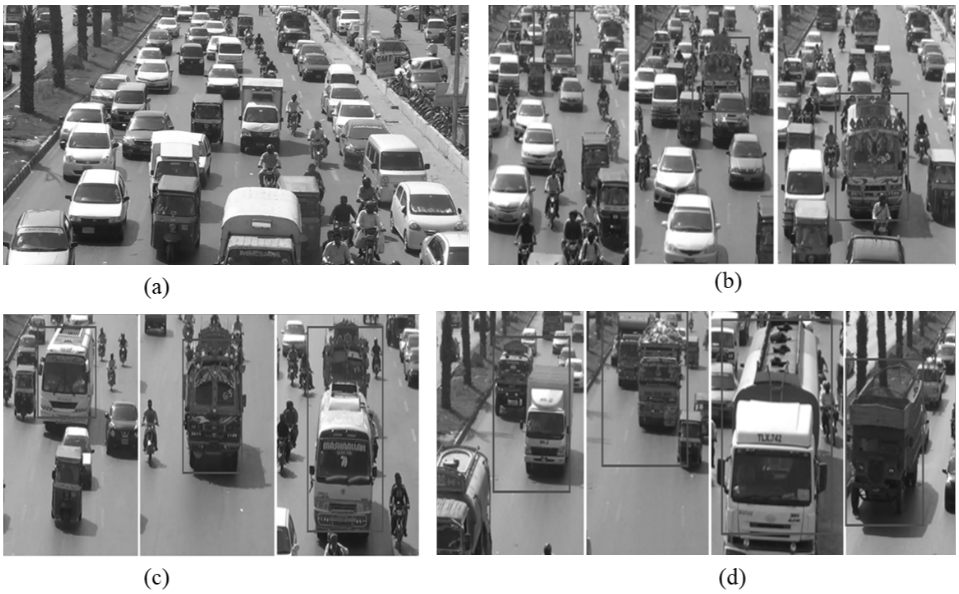

Fig. 2 *(a) A sample image from dataset, (b) a bus with the variation of appearance at varying distances, (c) different forms of bus, (d) different forms of trucks.*

collected and transformed to an image training dataset around every fourth second. With the right recording rate, the 45-minute video produced 600 pictures. The final collection contains around 2400 photos divided into six categories: bus, car, rickshaws, truck, bike, and MSV. There were several instances of each class in each image.

3.3 Data labeling

The dataset generated had fixed resolution. All the images in the dataset were of fixed size and had many variations of different vehicles. For the collected datasets, preprocessing involves annotation and labeling only. Each vehicle present in the image was labeled. There were numerous instances of different categories. For better accuracy, the vehicles at different angles and forms were annotated. For example, the car farther away from the mounted camera in a frame differed in its form when it was closest to the camera in the next few frames. Therefore, it was necessary to annotate the same car on different frames if they appear different. Fig. 2(b) illustrates the variation in the appearance of the traditional bus due to its varied distance from the camera in a different frame. The upper portion (roof) of the bus is not visible until it reaches closest to the mounted camera.

Annotation challenges: Annotations have a large impact on optimizing detection based on our criteria. Labeling presented a number of issues as well. One of the difficulties is categorizing the traffic, as there are many distinct classifications of traffic. The traditional buses and coaches are combined into a single category called "bus." Fig. 2(c) depicts many versions of the category "bus." The conventional bus and coach are widespread in the city, however their look varies substantially. Similarly, while being vastly different, traditional Pakistani trucks, water tankers, and huge delivery trucks were all classified as "trucks". Trucks differed in shape, size, and appearance, as illustrated in Fig. 2(d). There have even been reports of trucks with missing car pieces, giving them a one-of-a-kind

Rickshaws and bikes were virtually identical and exhibited minimal variance in the datasets; therefore they were simply categorized based on their classes. Jeeps, little and large automobiles, regardless of model, were considered "cars." MSV included HiRoof, HiAce, and other medium-sized automobiles. Donkey carts, bikes mounted to wheeled carts and chinchy with missing roofs or other vehicle pieces were among the unusual vehicles in the traffic. However, their frequency of occurrence in the traffic stream was so low that they could not be classified as a separate vehicle class. Such vehicles were also labeled as MSVs (medium sized vehicle). Another issue in the pre-processing step was bumper-to-bumper traffic and occlusion. Karachi's traffic is dense and diverse. Morever, in undisciplined traffic, vehicles may not follow a designated lane or may swerve in and out of lanes, making it difficult to track them accurately. Because the full image of the cars is not accessible, it is difficult to annotate the items completely. Fig. 2(a) depicts the occluded vehicle figures in the case of heavier traffic. In this circumstance, the annotations must be carefully modified while keeping future datasets for detection and classification in mind. Each vehicle in the frame was labeled, including any parked automobiles that were visible. Vehicle figures, both complete and fragmentary, were labeled such that the model could detect every version/form of the vehicle.

3.4 Classification and localization

The accuracy of the model was top priority for the proposed framework as the results carry important information necessary for applications like traffic controlling, monitoring, and map congestion calculation and visualization. The selected algorithm must exhibit good accuracy coupled with decent computational speed. When calculated in raw mAP, the faster R-CNN shows better performance than SSD, however, it is more computationally expensive.

The feature maps from the convolutional layers are considered as image features for the fully-convolutional RPN which takes the input of any size and outputs rectangular object proposals with object detection score. The features are extracted using the sliding window technique. Initially, RPN initiates $n \times n$ sliding window which has different scales and aspect ratios at every convolutional feature map. Every sliding window is then mapped to a low dimensional vector. These are sent as input to two layers: box classification layer and box regression which identifies the probability of being an object and coordinates of the bounding boxes respectively. After regions are proposed, regions of interest (ROI) are generated that pass through the pooling layer. ROI pooling handles the mapping of all proposals obtained from RPN into a single feature map. Each ROI is then passed through the pooling layer and fully connected layers to calculate the probability and refine the coordinates of bounding boxes [19].

3.5 Transfer learning

The implementation of the proposed framework is done using a pre-trained faster R-CNN TensorFlow model as given in [2]. Transfer learning involves using already built network architecture and models that have been developed for similar tasks and refining and returning it for the required purpose. Since the proposed framework is centered around the detection and counting of vehicles, pre-built models for object detection were deployed rather than building model from scratch. Deep neural networks require millions of samples of the dataset to be fully optimized [15]. Training these networks from scratch comes with the requirement of high computational resources and extensive datasets [15]. Moreover, using pre-built layers of CNN architecture which show good performance gives assurance for the better performance of the application that needs to be efficient. Adding custom layers and fine-tuning such networks increases the chances of trained models being accurate for the specific task and training these models with a custom dataset provides better efficiency. The proposed framework uses Inception V2 architecture for the faster R-CNN model.

3.6 Inception V2 architecture

The Inception V2 network model is pre-trained on the ImageNet dataset using high-computational and advanced computers. Inception V1 was introduced as GoogleNet in [44] which were further improved through different techniques including batch normalization [23] to provide Inception V2. The Inception V2 has several modules as shown in Fig. 3. Each of which performs four operations in parallel: 1×1 convolutional layer, 3×3 convolutional layer, 5×5 convolutional layer, and max pooling. 1×1 convolutional block performs depth reduction. The resulting outputs from mentioned four operations are concatenated together using a depth-wise approach and produce a filter concatenation block. Different inception module identifies different salient features at different level. The global features are captured by the 5×5 convolutional layers whereas the more distributed features are captured by the 3×3 convolutional layers. The low-level features are captured with max-pooling. The captured features are extracted and concatenated before passing down to the next layer.

3.7 Vehicle counting

The model's detection findings are used to accomplish the vehicle counting. It is necessary to configure parameters such as the path to the model and the inference graph through which counting is to be performed. The model receives input in the form of video or image, as well as information about its FPS, height, and width. The label map is then loaded, which contains information about various class names and associated indexes.

The object detection function generates and outputs detection boxes, detection scores, and detection classes for all extracted frames after the input is passed to the model for counting. Using this data, the visualization tool generates bounding boxes around each detected vehicle, along with its class and confidence score. The

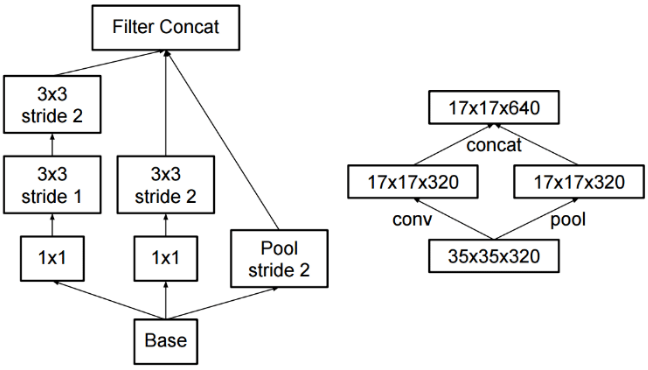

Fig. 3 *Inception V2 architecture [44].*

count per frame of each vehicle class is calculated, displayed, and recorded as entries in a.csv file using this information.

The concept of ROI (region of interest) line is used for total vehicle count or cumulative vehicle count. They-max coordinate of each detected vehicle of each frame is compared with the ROI position. If it overlaps, the function parameter that defines if the vehicle is detected or not is set to 1 and the visualization function returns the counter value is 1. For every detected vehicle on each frame, the process of visualization is performed. If the counter is 1, the count of the total passed vehicle is incremented and the color of the line for that frame is changed from red to green. All extracted frames are read iteratively until the end is reached. The output is saved in form of a video and record.

4. Experimental setup

The experiments were conducted on a graphics processing unit (GPU) server, Nvidia Tesla K80 having 2496 CUDA cores, with 12 GB GDDR5 RAM. The complete setup of the experiment is shown below in Tab. I.

To train the models, the efficient faster R-CNN-based deep learning framework was chosen. The main hyper parameter for faster R-CNN is the feature extractor, Inception V2 [23, 44] has been used for learning the features from the input image. The default values of momentum (0.9), weight decay (0.0005) and learning rate 2×10^{-4} has been used. The selection of a dataset is regarded as critical for utilizing distinct jobs in vehicle detection. Tab. II compares a few distinct datasets

Hardware	Environment
Computer	GPU Server
CPU	Intel(R) Xeon(R) CPU @2.3 GHz with 45 MB cache
GPU	NVDIA Tesla K80, 2496 CUDA cores
Memory size	RAM (∼12.6 GB), Hard Disk (∼320 GB)

Tab. I *The hardware environment.*

	KITTI	BDD100K	India driving dataset	Stanford	PSU	BIT	CompCars	Dawn	Our dataset
Number of images	15000	120000000	46588	8506	270	9850	186000	1000	2400
Image resolution	1242×375	1280×720	1920×1080	1409×1916	1920×1080	1920×1080	2100×1395	1920×1080	1280×738
No. of vehicle classes	4	6	10	4	–	6	12	5	6
Training/test split	80/20	70/30	67/33	80/20	80/20	80/20	–	–	80/20
Source of data collection	Vehicle onboard camera	Vehicle onboard camera	Vehicle onboard camera	3DR SOLO drone	3DR SOLO drone&web source	Road side fixed camera	Web-images	Web-images	Road side fixed camera
Diversity of vehicles	Limited	Wide	Wide	Wide	Limited	Limited	Wide	Limited	Wide
Occlusion	No	Yes	Yes	No	No	No	No	No	Yes
Bumper to bumper traffic	No	No	Yes	No	No	No	No	No	Yes

Tab. II *Comparison of datasets.*

for road item identification. In comparison to other current datasets, our data set includes local traffic vehicles such as animal carts and rickshaws, which are typically not included in any other dataset. The details of vehicular classes in other datasets are given in Tab. III.

Dataset	Classes (vehicles only)
KITTI	car, van, truck, tram
BDD 100K	bike, car, bus, truck, motor, train
IDD	bicycle, bus, train, motorcycle, car, truck, auto-rickshaw, caravan, animal and trailer
Stanford	bus, golfcart, car, skateboard
PSU	none
BIT	sedan, SUV, microbus, truck, bus, minivan
CompCar	MPV, SUV, hatchback, sedan, minibus, fastback, estate, pickup, sports, crossover, convertible, hardtop conv
Dawn	none
Our dataset	bus, car, rickshaw, truck, bike, medium-sized vehicle (MSV)

Tab. III *Dataset classes.*

5. Results and comparison

Performance of the algorithm is evaluated based on mean average precision (mAP), which is currently the most popular object detection measure, as described by PASCAL VOC Challenge [9]. The important evaluation parameters used are true positive (TP), false positive (FP), and false negative (FN). TP is the number of detections in both ground truth and results. FP is the number of detections in results only, excluding the ground truth. FN is the number of detections in ground truth only, excluding the results.

Precision (P) is the percentage of correct positive predictions. Recall (R) is the percentage of correct positive predictions among all ground truths. Precision and Recall cannot completely determine the performance of an object detection model, so mean average precision (mAP) is calculated as shown in Eq. 1. The value of mAP ranges from 0–1 with a higher value indicating better results.

$$\mathrm{mAP} = \frac{1}{classes} \sum_{c=1}^{classes} \frac{\mathrm{TP}(c)}{\mathrm{TP}(c) + \mathrm{FP}(c)}. \tag{1}$$

In Tab. IV, a summary has been presented based on the datasets, algorithms, and results of the previously proposed works on vehicle detection, compared to our work. It is evident from Table 5, that most of the research in ITS majorly focuses on disciplined traffic where the vehicles are strictly moving in one lane, with well-defined boundaries, and in a structured environment. There are very few frameworks reported that perform vehicle detection in a constrained environment like unstructured road, undisciplined traffic, and poor boundaries. Many researchers [11, 13, 31, 32, 42, 43] have proposed several deep learning models for vehicle detection but experiments are limited to a strict disciplined environment and include only vehicles that are common in well-developed countries.

Refs.	Algorithms	Dataset	Undisciplined traffic	Results	Limitations	Goals
[43]	Faster R-CNN, SSD	India Driving Data	yes	F-R-CNN outperforms SSD, the mAP value ranges from 28% to 32%	Low precision, inadequate data	1) Vehicle detection
[24]	Tiny YOLO V2	BIT	no	0.82 IOU and an average recall of 94.45% with the threshold of 0.8	Restricted dataset to big cars only	1) Vehicle detection 2) Vehicle classification
[31]	R-FCN, M-R-CNN, SSD, RetinaNet, YOLO V4	Berkley Deep Drive	no	YOLO V4 outperforms all other algorithms. mAP(wot)=88.67% mAP(wiot)=73.09%	Undiversified dataset	1) Vehicle detection
[32]	F-R-CNN, R-FCN, SSD, YOLO V3, RetinaNet	KITTI	no	The results of Hard are: Faster R-CNN (48.37)% R-FCN (66.01)% SSD (38.91)% YOLO V3 (38.23)% RetinaNet (68.73)%	Undiversified dataset	1) Vehicle detection
[42]	Improved YOLO V2	BIT, CompCars	no	94.78%	Inefficient in detecting far cars & small vehicles	1) Vehicle detection
[13]	F-R-CNN	KITTI	no	83.60%	Undiversified dataset	1) Vehicle detection

[3]	F-R-CNN, YOLO V3 YOLO V4	Stanford dataset, PSU dataset	no	YOLO V4: 34.4% (SD)94.6% (PSU) YOLO V3: 32.6% (SD) 96% (PSU) Faster R-CNN: 31.4% (SD) 84.5% (PSU)	dataset restricted to aerial images only	1) Vehicle detection
[4]	Fast R-CNN	Traffic videos downloaded from youtube	no	90%	Undiversified dataset	1) Vehicle detection 2) Vehicle tracking
[33]	Ensemble learning (faster R-CNN & SSD)	FLIR RGB, FLIR Thermal, KITTI, MB7500	no	86%	High time complexity	1) Vehicle detection 2) Traffic density estimation
[22]	Customized YOLO V4	Dawn dataset	no	81%	Does not address undisciplined traffic	1) Vehicle detection
Our paper	F-R-CNN	Traffic videos of Kaarchi	yes	86.14%		1) Vehicle detection 2) Vehicle classification 3) Vehicle counting

Tab. IV *Comparison with previously stated work.*

This doesn't fulfill the vehicular classification requirements of Asian countries like Pakistan, Bangladesh, and India with local vehicles in undisciplined traffic conditions. The paper [43] has presented a comparative study of different Vehicle detection algorithms on undisciplined traffic but the results showed quite low precision. The purpose of conducting this study is to bridge this gap by training the deep learning model on real undisciplined traffic. The dataset used in the study comprises real traffic videos of the city of Karachi, Pakistan. The traffic in Karachi is highly undisciplined and unstructured. Moreover, the study focuses on all the vehicle types that have not been covered in any of the existing datasets, like vehicle types rickshaws, animal cart, HiAce, traditional heavy trucks in addition to common vehicle classes like the car, motorcycle, truck, and bus. The experiments have been conducted on datasets that overcome the aforementioned shortcomings of previous datasets. The results show the average precision of vehicle detection attained is 86.14%. The experimental results verified that the proposed vehicle detection and counting method for real undisciplined traffic video scenes has good performance and feasibility. Moreover, vehicles in other countries of South Asia like in India, Bangladesh have the same shapes such as rickshaws, animal carts, and conventional trucks. Hence, the methodology and results of the proposed vehicle detection and counting model can be used as a reference for South Asian transport studies.

6. Conclusion

The suggested model implements vehicle detection and vehicle counting for heterogeneous traffic in cities where lane discipline is not enforced. By focusing on uncontrolled traffic situations for vehicle detection and counting, the system overcomes the limitations of previous research work. It detects and counts the number of vehicles passing on the road by adding up the total number of vehicles or counting them frame by frame. To train the detection and counting system, a collection of 2400+ photos of six classes (including local vehicles) is collected utilizing uncontrolled traffic circumstances. The framework has been specifically built to accommodate vehicle designs typical in South Asian countries such as Pakistan, India, and Bangladesh. It aspires to automate vehicle counting and detecting systems to facilitate V2V communication in smart cities by making educated judgments about infrastructure, traffic offences and other traffic related issues.

References

[1] ABADI M., BARHAM P., CHEN J., CHEN Z., DAVIS A., DEAN J., DEVIN M., GHEMAWAT S., IRVING G., ISARD M., KUDLUR M., LEVENBERG J., MONGA R., MOORE S., MURRAY D.G., STEINER B., TUCKER P., VASUDEVAN V., WARDEN P., WICKE M., YU Y., ZHENG X. TensorFlow: A system for large-scale machine learning. 21, doi: 10.1007/978-1-4842-5967-2_8.

[2] AHMED S.H., RAZA M., MEHDI S.S., REHMA, I., KAZMI M., QAZI S.A. Faster RCNN based Vehicle Detection and Counting Framework for Undisciplined Traffic Conditions. In: 2021 *IEEE 18th International Conference on Smart Communities: Improving Quality of Life Using ICT, IoT and AI (HONET)*. Online. Karachi, Pakistan: IEEE, 2021, pp. 173–178, doi: 10.1109/HONET53078.2021.9615466.

[3] AMMAR A., KOUBAA A., AHMED M., SAAD A., BENJDIRA B. Vehicle Detection from Aerial Images Using Deep Learning: A Comparative Study. *Electronics*. 2021, 10(7), pp. 820, doi: 10.3390/electronics10070820.

[4] ARORA N., KUMAR Y., KARKRA R., KUMAR M. Automatic vehicle detection system in different environment conditions using fast R-CNN. *Multimedia Tools and Applications*. 2022, 81(13), pp. 18715–18735, doi: 10.1007/s11042-022-12347-8.

[5] BAUTISTA C.M., DY C.A., MANALAC M.I., ORBE R.A., CORDEL M. Convolutional neural network for vehicle detection in low resolution traffic videos. In: 2016 *IEEE Region 10 Symposium (TENSYMP)*. Online. Bali, Indonesia: IEEE, 2016, pp. 277–281, doi: 10.1109/TENCONSpring.2016.7519418.

[6] BOCHKOVSKIY A., WANG C.-Y., LIAO H.-Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. Online, 22 April 2020, doi: 10.48550/arXiv.2004.10934.

[7] CHANG J., WANG L., MENG G., XIANG S., PAN C. Vision-Based Occlusion Handling and Vehicle Classification for Traffic Surveillance Systems. *IEEE Intelligent Transportation Systems Magazine*. 2018, 10(2), pp. 80–92, doi: 10.1109/MITS.2018.2806619.

[8] CHEN Z., CHEN K., CHE, J. Vehicle and Pedestrian Detection Using Support Vector Machine and Histogram of Oriented Gradients Features. In: *International Conference on Computer Sciences and Applications*. Online. Wuhan, China: IEEE, 2013, pp. 365–368, doi: 10.1109/CSA.2013.92.

[9] DAI J., LI Y., HE K., SUN J. R-FCN: Object Detection via Region-based Fully Convolutional Networks. Online. 2016, doi: 10.48550/arXiv.1605.06409.

[10] DIMILILER K., EVER Y.K., MUSTAFA S.M. Vehicle Detection and Tracking Using Machine Learning Techniques. In: ALIEV R.A., KACPRZYK J., PEDRYCZ W., JAMSHIDI M., BABANLI M.B., SADIKOGLU F.M. (eds.), *10th International Conference on Theory and Application of Soft Computing, Computing with Words and Perceptions* – ICSCCW-2019. Online. Cham : Springer International Publishing, Advances in Intelligent Systems and Computing, 2020, pp. 373–381, doi: 10.1007/978-3-030-35249-3_48.

[11] 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), pp. 22473–2256, doi: 10.1109/TITS.2015.2402438.

[12] FAZLI S. Neural Network based Vehicle Classification for Intelligent Traffic Control. *International Journal of Software Engineering & Applications*. 2012, 3(3), pp. 17–22, doi: 10.5121/ijsea.2012.3302.

[13] GAO Y., GUO S., HUANG K., CHEN J, GONG Q., ZOU Y., BAI T., OVERETT G. Scale Optimization for Full-Image-CNN Vehicle Detection. Online, 19 February 2018, doi: 10.1109/IVS.2017.7995812.

[14] GEIGER A., LENZ P., URTASUN R. Are we ready for autonomous driving? The KITTI vision benchmark suite. In: *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3354–3361, doi: 10.1109/CVPR.2012.6248074.

[15] GHAZI M.M. Plant identification using deep neural networks via optimization of transfer learning parameters. *Neurocomputing*, 235, 2017, pp. 228–235, doi: 10.1016/j.neucom.2017.01.018.

[16] GIRSHICK R., DONAHUE J., DARRELL T., MALIK J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In: textitIEEE Conference on Computer Vision and Pattern Recognition. *IEEE,* 2014, pp. 580–587, doi: 10.1109/CVPR.2014.81.

[17] GIRSHICK R. Fast R-CNN. Online. 27 September 2015, doi: 10.48550/arXiv.1504.08083.

[18] HE K., GKIOXARI G., DOLLÁR P., GIRSHICK R. Mask R-CNN. Online, 24 January 2018, available from: http://arxiv.org/abs/1703.06870.

[19] KOIRALA A., WALSH K.B., WANG Z., MCCARTHY C. Deep learning – Method overview and review of use for fruit detection and yield estimation. *Computers and Electronics in Agriculture*. 2019, 162, pp. 219–234, doi: 10.1016/j.compag.2019.04.017.

[20] HE K., GKIOXARI G., DOLLÁR P., GIRSHICK R. Mask R-CNN. Available from: http://arxiv.org/abs/1703.06870.

[21] HEDEYA M.A., EID A.H., ABDEL-KADER R.F. A Super-Learner Ensemble of Deep Networks for Vehicle-Type Classification. *IEEE Access*. 2020, 8, pp. 98266–98280, doi: 10.1109/ACCESS.2020.2997286.

[22] HUMAYUN M., ASHFAQ F., JHANJHI N.Z., ALSADUN M.K. Traffic Management: Multi-Scale Vehicle Detection in Varying Weather Conditions Using YOLOv4 and Spatial Pyramid Pooling Network. *Electronics*. 2022, 11(17), pp. 2748, doi: 10.3390/electronics11172748.

[23] IOFFE S., SZEGEDY C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *32nd International Conference on Machine Learning*, doi: 10.48550/arXiv.1502.03167.

[24] KARUNGARU S., DONGYANG L., TERADA K. Vehicle Detection and Type Classification Based on CNN-SVM. *International Journal of Machine Learning and Computing*. 2021, 11(4), pp. 304–310, doi: 10.18178/ijmlc.2021.11.4.1052.

[25] KENK M. DAWN. Online, 6 March 2020, Mendeley.

[26] KOIRALA A., WALSH K.B., WANG Z., MCCARTHY C. Deep learning – Method overview and review of use for fruit detection and yield estimation. *Computers and Electronics in Agriculture*. 2019, 162, pp. 219–234, doi: 10.1016/j.compag.2019.04.017.

[27] LECUN Y., BENGIO Y., HINTON G. Deep learning. Nature. 2015, 521(7553), pp. 436–444, doi: 10.1038/nature14539.

[28] LIENHART R., MAYDT J. An extended set of Haar-like features for rapid object detection. In: Proceedings. *International Conference on Image Processing*. Online. Rochester, NY, USA: IEEE, 2002, pp. I-900–I–903, [Accessed 27 September 2022], ISBN 978-0-7803-7622-9, doi: 10.1109/ICIP.2002.1038171.

[29] LIN T.-Y., GOYAL P., GIRSHICK R., HE K., DOLLAR P. Focal Loss for Dense Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, doi: 10.1109/TPAMI.2018.2858826.

[30] LIU W., ANGUELOV D., ERHAN D., SZEGEDY C., REED S., FU C.-Y., BERG A.C. SSD: Single Shot MultiBox Detector. In: LEIBE B., MATAS J., SEBE N., WELLING M. (eds.), Computer Vision – ECCV 2016. Online. Cham: Springer International Publishing, 2016, pp. 21–37.

[31] MAHAUR B., SINGH N., MISHRA K.K. Road object detection: a comparative study of deep learning-based algorithms. *Multimedia Tools and Applications*. 2022, 81(10), pp. 14247–14282, doi: 10.1007/s11042-022-12447-5.

[32] MENG C., BAO H., MA Y. Vehicle Detection: A Review. *Journal of Physics: Conference Series*. 2020, 1634(1), pp. 012107, doi: 10.1088/1742-6596/1634/1/012107.

[33] MITTAL U., CHAWLA P. Vehicle detection and traffic density estimation using ensemble of deep learning models. *Multimedia Tools and Applications*. Online. 27 August 2022, doi: 10.1007/s11042-022-13659-5.

[34] PSU Car Dataset.Available online: https://github.com/aniskoubaa/psu-car-dataset

[35] QUAN YUAN, THANGALI A., ABLAVSKY V., SCLAROFF S. Learning a Family of Detectors via Multiplicative Kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2011, 33(3), pp. 514–530, doi: 10.1109/TPAMI.2010.117.

[36] REN S., HE K., GIRSHICK R., SUN J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. Online. 6 January 2016, doi: 10.48550/arXiv.1506.01497.

[37] REDMON J., DIVVALA S., GIRSHICK R., FARHADI A. You Only Look Once: Unified, Real-Time Object Detection. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Online. Las Vegas, NV, USA: IEEE, 2016, pp. 779–788. [Accessed 27 September 2022], ISBN 978-1-4673-8851-1, doi: 10.1109/CVPR.2016.91.

[38] REDMON J., FARHADI A. YOLO9000: Better, Faster, Stronger[C]. textitComputer vision and pattern recognition, 2017, pp. 6517–6525, doi: 10.1109/CVPR.2017.690.

[39] REDMON J., FARHADI A. YOLOv3: An Incremental Improvement. Online, 8 April 2018, doi: 10.48550/arXiv.1804.02767.

[40] ROBICQUET A., SADEGHIAN A., ALAHI A., SAVARESE S. Learning Social Etiquette: Human Trajectory Understanding In Crowded Scenes. In: LEIBE B., MATAS J., SEBE N., WELLING M. (eds.), Computer Vision – ECCV 2016. Online. Cham: *Springer International Publishing*, 2016, pp. 549–565, doi: 10.1007/978-3-319-46484-833.

[41] RYBSKI P.E., HUBER D., MORRIS D.D., HOFFMAN R. Visual classification of coarse vehicle orientation using Histogram of Oriented Gradients features. In: 2010 *IEEE Intelligent Vehicles Symposium*. Online. La Jolla, CA, USA: IEEE, June 2010, pp. 921–928, doi: 10.1109/IVS.2010.5547996.

[42] SAN, J., WU Z., GUO P., HU H., XIANG H., ZHANG Q., CAI B. An Improved YOLOv2 for Vehicle Detection. Sensors, 2018, 18(12), pp. 4272, doi: 10.3390/s18124272.

[43] SEHGAL J., SHARMA M., CHATTERJEE J., MEHRA A. Comparative Study of Deep Learning Models for Vehicle Detection in an Unconstrained Road Scenario. In: *2020 International Conference on Communication and Signal Processing (ICCSP)*. Online. Chennai, India: IEEE, 2020.

[44] SZEGEDY C., WEI LIU, YANGQING JIA, SERMANET P., REED S., ANGUELOV D., ERHAN D., VANHOUCKE V., RABINOVICH A. Going deeper with convolutions. In: 2015 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Online. Boston, MA, USA: IEEE, 2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.

[45] TANG Y., ZHANG C., GU R., LI P., YANG B. Vehicle detection and recognition for intelligent traffic surveillance system. *Multimed Tools Appl.* doi: 10.1007/s11042-015-2520-x.

[46] TANG Y., ZHANG C., GU R., LI P., YANG B. Vehicle detection and recognition for intelligent traffic surveillance system. *Multimedia Tools and Applications*. 2017, 76(4), pp. 5817–5832, doi: 10.1007/s11042-015-2520-x.

[47] VELAZQUEZ-PUPO R., SIERRA-ROMERO A., TORRES-ROMAN D., SHKVARKO Y., SANTIAGO-PAZ J., GÓMEZ-GUTIÉRREZ D., ROBLES-VALDEZ D., HERMOSILLO-REYNOSO F., ROMERO-DELGADO M. Vehicle Detection with Occlusion Handling, Tracking, and OC-SVM Classification: A High Performance Vision-Based System. *Sensors*. 2018, 18(2), pp. 374, doi: 10.3390/s18020374.

[48] WEN-CHUNG CHANG, CHIH-WEI CHO. Online Boosting for Vehicle Detection. *IEEE Transactions on Systems, Man, and Cybernetics*, Part B (Cybernetics), 2010, 40(3), pp. 892–902, doi: 10.1109/TSMCB.2009.2032527.

[49] YANG L., LUO P., LOY C.C., TANG X. A Large-Scale Car Dataset for Fine-Grained Categorization and Verification. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, doi: 10.1109/CVPR.2015.7299023.

[50] YU F., CHEN H., WANG X., XIAN W., CHEN Y., LIU F., MADHAVAN V., DARRELL T. BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Online. Seattle, WA, USA: IEEE, 2020, pp. 2633–2642, doi: 10.1109/CVPR42600.2020.00271.

[51] ZAKRIA CAI J., DENG J., KHOKHAR M.S., UMAR AFTA, M. Vehicle Classification Based on Deep Convolutional Neural Networks Model for Traffic Surveillance Systems. In: 15th *International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*. Online. Chengdu, China: IEEE, 2018, pp. 224–227, doi: 10.1109/ICCWAMTIP.2018.8632593.