Journal of Theoretical and Applied Information Technology
15th April 2019. Vol.97. No 7

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

RT-VC: AN EFFICIENT REAL-TIME VEHICLE COUNTING APPROACH

[1]SALAH ALGHYALINE, [2]NIDHAL KAMEL TAHA EL-OMARI, [3]RA'ED M. AL-KHATIB, [4]HESHAM Y. AL-KHARBSHH

[1]Department of Computer Science, Faculty of Information Technology, The World Islamic Sciences and Education University, Amman – Jordan. Email: salah.alghyaline@wise.edu.jo

[2]Department of Software Engineering, Faculty of Information Technology, The World Islamic Sciences and Education University, Amman – Jordan. E-mail: nidhal.omari@wise.edu.jo

[3]Department of Computer Science, Faculty of Information Technology and Computer Sciences, Yarmouk University, Irbid-21163, Jordan. E-mail: raed.m.alkhatib@yu.edu.jo

[4]Department of Software Engineering, Faculty of Information Technology, The World Islamic Sciences and Education University, Amman – Jordan. E-mail: hesham.alkharabsheh@wise.edu.jo

ABSTRACT

This paper proposes and implements an efficient real-time vehicle counting (RT-VC) approach. This approach is based on the most efficient action detection and tracking methods in computer vision. YOLO is used for object detection, whereas Kalman filter with Hungarian algorithm are used for tracking. The road is divided into two zones of interest by the end-user, and any vehicle will be counted if its trajectory crosses these zones of interest. The experiments show that the proposed system is very accurate in comparison with other existing approaches. For comparative evaluation, our proposed approach obtained accuracy above 90% for most of the tested videos in the highway roads. Therefore, the proposed approach can efficiently work with many real-time surveillance systems and has the potential to be used in many real road applications.

Keywords: *Artificial Neural Networks (ANNs), Convolutional Neural Network (CNN), Artificial Intelligence (AI), Deep Learning Algorithms, Vehicle Counting, Surveillance Systems, Traffic Managements.*

1. INTRODUCTION

Machine learning is indeed a powerful technology that has been recently derived from Artificial Intelligence (AI), which is an important area of computer science. Basically, the AI is called 'machine intelligence', because AI designs many computer activities that can be used for speech recognition, planning, learning, and problem solving [1][2][3]. These activities are harnessed to make intelligent machines much capable to work, operate, behave, react, or conduct by itself like humans [4]. Even with its relatively short life span, one of the effective technique used machine learning systems is the deep learning. Recently, deep learning techniques deploy many efficient algorithms for image processing especially for image identifying. Deep convolutional nets is one of powerful deep learning algorithm that showed superior results in image processing such as manipulating video, audio, text and speech [5]. These kinds of deep learning algorithms can manipulate for the real-time videos.

Nowadays, roads are equipped with many types and different styles of cameras. Some cameras are used for capturing the vehicles' plate number in case of violating the speed limit. Since, these cameras are originally designed to measure the speed of the vehicle, radar systems are used for the speed monitoring process [6]. Likewise, there is another kind of cameras used to monitor the traffic related to the security relevance, like the surveillance cameras where the systems of these security cameras rely on direct human intervention in controlling and monitoring rooms.

For roads, inside cities or on highway, there are huge number of moving vehicles [7]. Monitoring or making any statistics by human is usually found to be prone to errors and, therefore, it is more efficient to use automated systems for monitoring and collecting statistics about the moving vehicles.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Overall, we find that the key philosophy behind monitoring and estimating the traffic flow is to improve the road signaling systems and redirecting the traffic flow especially in case of traffic jams. Furthermore, knowing the road load will provide the decision maker with accurate knowledge about the current roads' situations and conditions. This will help in making better plans for constructing new roads or making new projects near these roads [8].

Increasingly, there is a widespread of IT-based solutions that are based on Internet of Things (IoT) to count persons entering any place like a room [9]. Likewise, there are also some smart real-time systems that utilized the power of genetic algorithm for solving the school bus routing problem [10].

Practically, the existing car counting approaches are either video-based sensors [11][12][13][14][15], or non-video-based sensors used such as Inductive loop Detector, Radar, Infrared, Ultrasonic [16][17][18]. Meanwhile, the sensor-based methods are also efficiently used to count persons especially when they are entering rooms [9].

Vehicles counting based on computer vision requires less hardware equipment and maintenance compared with other traditional sensor-based approaches [8]. In practice, only cameras and general-purpose computers are just required to process the video's frames [8][19]. However, some computer vision counting approaches suffer from many issues specially the background subtraction based approaches[11][14]. The background of any video could be changed dramatically due to weather, lighting and moving objects (i.e., trees, flags) [20]. The detection accuracy of object detection counting based approaches significantly affects the final counting accuracy and so it is crucial to use an efficient object detection method in vehicles counting system [21].

This paper proposes a robust and accurate car counting system that works perfectly in real-time with either low or high speed roads. The proposed method starts by detecting the moving vehicles using a very accurate object detection approach, 'YOLO Darknet' [21]. Then, it tracks the moving object by using 'Kalman Filter' which is one of the most efficient algorithms in tracking moving-objects[15][22]. Finally, the proposed approach uses the zones of interest for counting the moving vehicles. These zones can be identified easily by the user during the running time. The results show that the proposed approach is very effective and accurate compared with other existing competitive approaches in the literature.

We organize the rest of the paper as follows. In the next section, comprehensive review to the related works and literature is presented. Section 3 fully describes the methodology of proposed real-time efficient vehicle counting approach. The used dataset is explained in section 4. Section 5 shows the experimental results with full comparative evaluation process. The paper conclusion is presented in section 6.

2. RELATED WORK

Viewed in a broad sense, vehicles' classification or detection is a crucial stage during vehicle counting systems. The accuracy rate of detection results is based on the extracted features of objects that have been detected from images and video frames [23]. There are two types of algorithms to extract features from images: the hand-crafted algorithms and the deep learning algorithms [23].

Related to the first type, there are many hand-crafted algorithms to extract the features such as: Scale-Invariant Feature Transform (SIFT) [24], Speeded Up Robust Features (SURF) [25], Histograms of Optical Flow (HOF) [26], Histograms of Oriented Gradients (HOG) [27], and Motion Boundary Histogram (MBH) [28]. In [29], the authors have recently proposed an approach based on multiple extracted features (SURF, HOF, HOG, and MBH).

Related to the second type, extracting features based on deep learning algorithms is recently presented in variety of surveillance systems [30][31]. The Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) are widely used for extracting features from detected objects that are founded in videos [1][32][33]. Compared with the traditional hand-crafted features, CNNs and RNNs have been proven as the most accurate algorithms in the field of deep learning features [21][23]. They are based on a set of convolutional layers that uses a set of parameters to train the detection model [1].

There is another methodology based on background subtraction model that is used to develop methods related to car counting system. In literature, these background subtraction are adapted by several researchers to find the moving objects in a specific scene using a static camera [34][35]. However, there are some difficulties in finding well-known datasets to be used in training and testing stages.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Biswas *et al.* (2017) introduced a car counting system based on background subtraction with OverFeat approach [8]. Daigavane and Bajaj (2010) used image subtraction to distinguish the moving objects from the background objects. Thus, they used a segmentation technique to make the final detection decision [36]. However, due to some cases, like moving objects, or due to some weather affects, some kind of variations might be occurred related to the backgrounds of these images. So, background subtraction-based approaches might enroll some failures in these cases [36].

Many vehicles counting approaches are based on object-detection; therefore, we review some of the object detection approaches in this section. For example, Karaduman *et al.* (2012) proposed a method based on detecting the edge of image [37]. Then, they used this edge-based detection method to calculate the optical flow of the moving vehicles.

Galoogahi et al. (2013) presented another method, which showed that learning multi-channel of detectors or filters significantly improved the vehicles' detection results [38]. Based on using the multiple scales of feature-maps of Deep Convolutional Neural Networks (DNNs), Chen et al. (2014) suggested a new Hybrid DNN method to find small objects in satellite images [31].

Liu and Mattyus (2015) used integral image features in multiple pyramids for vehicle detection [39]. Their approach was applied on aerial images to detect the type and the orientation of the vehicle. Sivaraman and Trivedi (2010) introduced a vehicles detection framework based on Harr-like features for describing the object [40]. They used Adaboost tool for the classification step, and then they implemented the particle filter for tracking the detected object [40].

Furthermore, Recurrent Neural Networks (RNNs) are implemented to track objects in [41]. Kahou *et al.* (2017) used RNNs to track and identify the object location [32]. Shaoqing *et al.* (2015) introduced an accurate object detection framework called 'Faster R-CNN' [33]. Then, the obtained results are compared with other related approaches that are based on the selective search for the region proposal networks. They proved that the region proposal stage in 'Faster R-CNN' can be reduced by sharing it with the convolutional features. The detection time is significantly reduced by this sharing [33]. Unfortunately, their approach cannot run in real-time even with a powerful Graphics Processing Unit (GPU). This drawback makes their proposed approach difficult to be used within real-time applications.

As a consequence of 'Faster R-CNN', Redmon *et al.* (2016) introduced an object detection approach called You Only Look One (YOLO) [42]. This approach has relatively a very fast speed as compared to the aforementioned one, 'Faster R-CNN'. While the frame rate speed of YOLO approach is 45 frames per second (fps), it is only 5 fps for 'Faster R-CNN'. However, the accuracy rate of YOLO approach is less than 'Faster R-CNN'. After that, based on Darknet-19 CNN architecture, Redmon and Farhadi (2017) introduced the second version of YOLO, namely YOLOv2. Besides that the speed of this version is more than 40 fps, its accuracy is slightly better than 'Faster R-CNN'. This, in turn, makes YOLOv2 more applicable to be used in real-time applications [43]. And therefore, the YOLO-based approaches with its high detection speed and accuracy are recently attracted many researchers to employee them in their proposed systems [44][45].

3. PROPOSED METHODOLOGY

In this section, the proposed real-time vehicle counting (RT-VC) approach is discussed in detail.

As a matter of fact, solving the vehicle counting problem manually is a tedious issue. For this reason, the main focus of our proposed approach is adapting the most efficient deep learning algorithm associated with solving the vehicle counting problem automatically.

Our proposed RT-VC framework developed a new mechanism for counting vehicles based on the most efficient approaches in object detection and tracking systems. Figure 1 illustrates the general workflow of our proposed real-time approach. As illustrated in this figure, the proposed approach starts by identifying the zone of interest. These zones of interest depend on the road and should be identified by the user of the system.

After that, this approach detects the set of positions for the moving vehicle. Then, it builds the trajectory of the moving vehicles over a period of time. This trajectory consists of a set of points that the vehicle visited over a period of time. Finally, our proposed real-time approach starts directly counting the moving vehicles as if their trajectories pass through the zones of interest. The counting results are updated automatically on the screen. Some samples of implemented results and videos can be downloaded from the website of our RT-VC

Journal of Theoretical and Applied Information Technology
15th April 2019. Vol.97. No 7

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

approach[1]. The obtained results show that our proposed approach efficiently counts vehicles. One more important thing, the proposed approach counts vehicles in multiple sides of the street, if they have been initially identified by the user at the drawing zones stage. The following subsections discuss thoroughly each stage of the proposed efficient real-time vehicle counting approach.

3.1. Vehicle Detection

Detecting the moving vehicle is a crucial task during vehicles counting. Our goal during the detection stage is to make the detection process to be accurate and run in real-time speed.

The YOLO is reported as the state-of-the-art detection tool [43]. Also, this investigation and analysis proved that the YOLO outperforms most of other existing object detection approaches like ‘Faster R-CNN’ and other approaches of hand-crafted object detection. Therefore, we adapted Darknet-19 CNN model to be used in vehicle detection stage. Then, this CNN model is trained on YOLO framework. A sample of 4,924 moving vehicles, extracted from 939 frames, was labeled randomly. Meanwhile, the whole used dataset consists of 10 videos that have nearly 200,000 frames. The number of classes is set to be only one class to represent any moving vehicle in roads.

As a matter of fact, the adapted Conventional Neural Networks are successfully learned by extracting knowledge from a small sample of labeled data [46]. After training phase, our proposed model is capable to conduct and evaluate a large sample of unlabeled outlet data. Thus, our proposed system should be trained enough according to specified learning rules until it learned how to achieve the desired results [46][47]. After enough training operation, the CNN will be capable to classify any new datasets of videos based on the similarity to the trained datasets.

Figure 2 shows a sample from the labeled pictures before the training stage. After 66,500 iterations, the average loss reached 0.245 error rate. After that, there was no significant change in the error rate. Consequently, we used the trained model after 66,500 iterations. The training step is finished at this point; this is depicted in Figure 3.

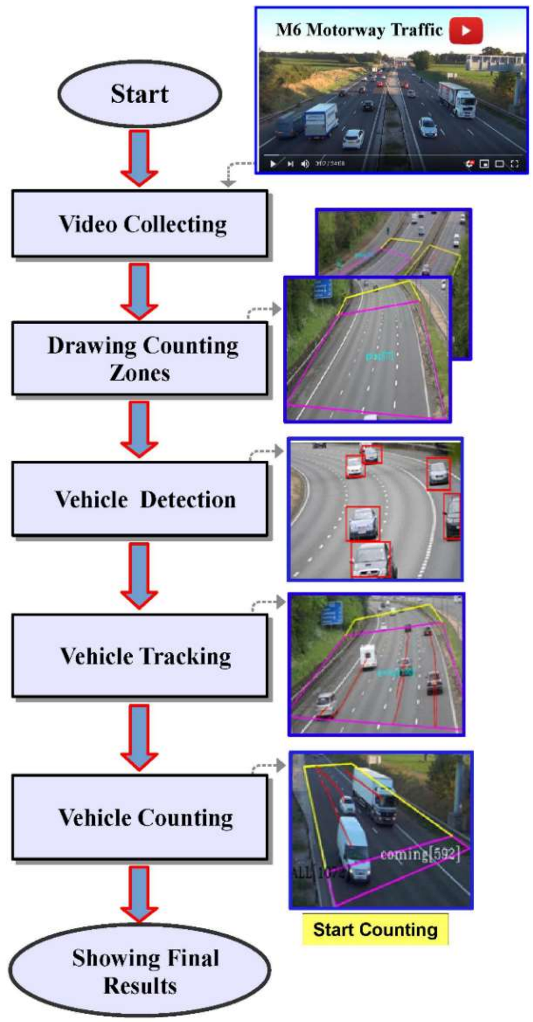


Figure 1: *Main workflow of the proposed vehicle counting approach.*

The second version of YOLO is called ‘YOLO9000’ and it is based on the Darknet-19 CNN architecture [43]. Darknet-19 is the most appropriate one to be deployed in this vehicle detection stage in our proposed approach. Therefore, we adapted the CNN architecture from YOLO9000 [43], to train our CNN model with some particular customizations. These customizations to the adapted CNN architecture from YOLO9000 [43], are as follows:

- The number of classes is reduced to one class called ‘Car’, which refers to any moving vehicle in the road.

[1] http://academic.wise.edu.jo/Salah.Ghyaleen/SitePages/vehicle-counting.aspx

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

- The network includes 19 convolutional layers in addition to 5 Maxpooling operations.
- As depicted in Figure 4 (a), the input image of 416 × 416 dimensions is divided into $G \times G$ grids of (13 × 13).
- After that, the trained CNN model predicts the bounding boxes and the confidences, as seen in Figure 4 (b).
- Finally, our approach makes its final detection based on the predicted confidences and bounding boxes, as illustrated in Figure 4 (c).

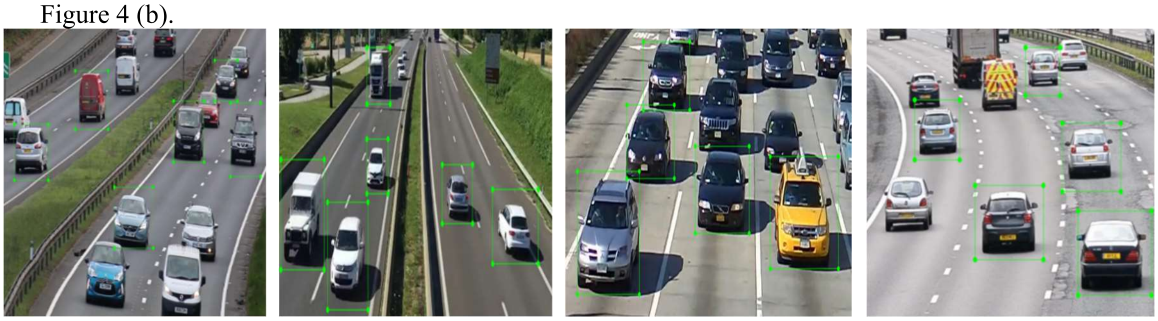


Figure 2: *A Sample of labeled objects.*

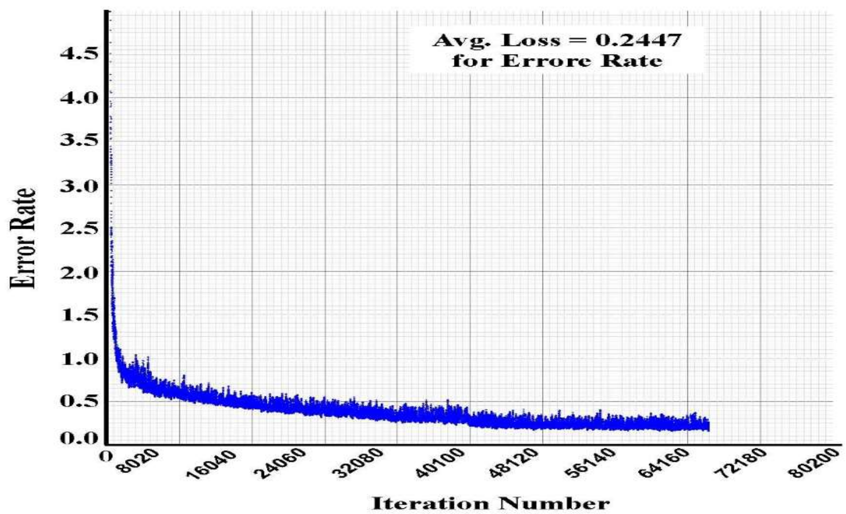
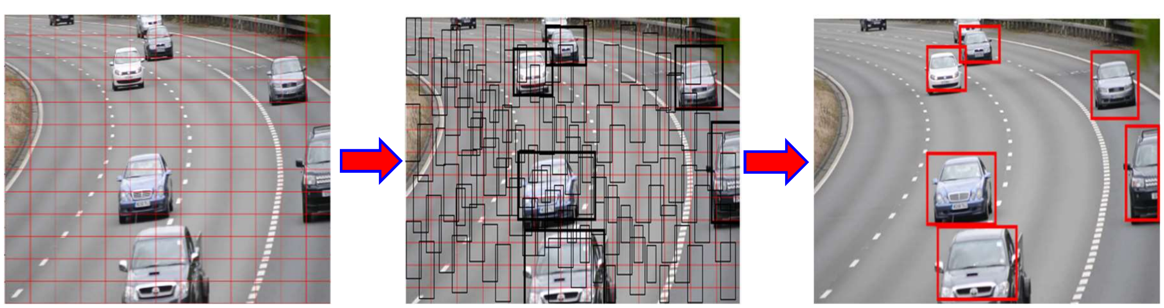


Figure 3: *Average loss or the error of the trained CNN model using YOLO architecture.*



(A) (B) (C)

Figure 4: *Main steps of the vehicle detection stage: A) Dividing the input image into $G \times G$ grid. B) Predicting the boxes with their confidences. C) Thresholding the boxes with the highest confidences.*

3.2. Vehicle Tracking

The accuracy of vehicles' detection in the previous stage is efficient by adapting the Darknrt-19 CNN model and train it in YOLO framework. Now, the main idea of this stage (i.e., Vehicle Tracking stage) is to keep tracking the detected vehicles while they are moving on the road. Even that the achieved detection accuracy rate from the adapted YOLO is relatively high, it is hard to detect the moving vehicles in all locations during their movement on the road. It is indeed necessary to keep tracking every vehicle at all locations as far as possible and to get the movement trajectory. To achieve this goal, we should predict the vehicle's position in case of missing the detection on some frames. Since Kalman filter is one of the most efficient and accurate tracking approaches [22][48], we adopted it to tackle the tracking problem. Then, we deploy Hungarian algorithm to find the optimal assignment between the detected vehicles' locations from one frame to another [49]. The details of adopting and implementing these two tracking algorithms can be explained as follows.

The main purpose of using Kalman filter is to predict the object positions based on the previous measurements and the motion models. The main phases of this deployed filter are summarized on the following mathematical formulas:

1. Current state prediction

$$X'_k = AX_{k-1} + BU_k + W_k \qquad (1)$$

Where:
A : Model of the transition state.
X_{k-1} : System previous state.
B : The model of the control-input.
U_k : Control vector.
W_k : Noise error covariance.

2. Error covariance prediction

$$P'_k = AP_{k-1}A^T + Q_k \qquad (2)$$

Where:
P_{k-1} : The estimated error covariance at time $k-1$
Q_k : Noise error covariance.

3. Kalman Gain

$$K_k = P'_k H^T (HP'_k H^T + R)^{-1} \qquad (3)$$

Where:
H : Measurement matrix.
R : Covariance of the measurement error.

4. Updating the state estimate

$$X_k = X'_k + K(Y_k + HX'_k) \qquad (4)$$

Where:
Y_k : Measurement value

5. Updating the estimate covariance

$$P_k = (I - K_k H)\, P'_k \qquad (5)$$

Where:
I : Identity matrix

Hungarian algorithm in [49] is used to make the optimal assignment between the detected vehicles' locations from one frame to another consecutive frame. Suppose K denotes a $n \times n$ matrix that includes the set of current trajectories and the new predicted points by Kalaman filter. Also, suppose $K_{i,j}$ is the cost of assigning the trajectory i to the predicted point j. Then, the main steps in Hungarian algorithm can be summarized as follows:

1. Let $R_{\min}$ denote the minimum value at each row in K, then subtract $R_{\min}$ from each row at K.
2. Let $C_{\min}$ denote the minimum value at each column in K, then subtract $C_{\min}$ from each column at K.
3. Cross the rows and columns that have zeroes values with the minimum number of lines.
4. If the number of used lines in the previous step is n, then the optimal solution is reached. Otherwise, move to step 5.
5. Let m denote the smallest value at K that is not crossed by any line as, then subtract m from each uncrossed row at K. Finally, add m to each crossed column, and return to step 3.

3.3. Vehicle Counting

At the beginning of the counting process or even before the detection and tracking stages, the user should select six points. These points are used to draw two polygons. The two polygons divide the region of interest into two successive zones: Zone 1 and Zone 2. These zones are denoted by Z_1 and Z_2, respectively. Z_1 is located at the beginning of the road and represents the first area of vehicle detection. On the other hand, Z_2 located at the end of the road and represents the last detection area. This is shown clearly in Figure 5. However, these two zones do not have a regular shape like a rectangle shape. Obviously, this is related to the camera view where the nearer to the camera has a longer line-length than the farther one. On the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

contrary, the farther from the camera has a shorter line-length than a nearer line.

Let $\{T_1, T_2, T_3, \ldots, T_n\}$ denote a set of n trajectories obtained from the tracking process; where T_i represents the trajectory for the i *th* car. The set of 2-dimensional points in trajectory T_i is indicated by $T_i = \{P_1, P_2, P_3, \ldots, P_L\}$.

The car i will be counted if it satisfies the following conditions:

1. $P_1 \in Z_1$
2. $P_L \in Z_2$
3. $L \geq L_{th}$

where L_{th} the minimum tracking frames.

The L_{th} is set to *10* frames during all experiments. Let $L_{\max}$ denote the maximum tracking length, where $L_{\max}$ is set to $S \times FPS$, and S is the maximum number of seconds that are needed for tracking. Noted that we set $S = 3$ during all experiments, and FPS is the frame per second for the tested video. Figure 6 shows samples from the proposed vehicle counting approach.

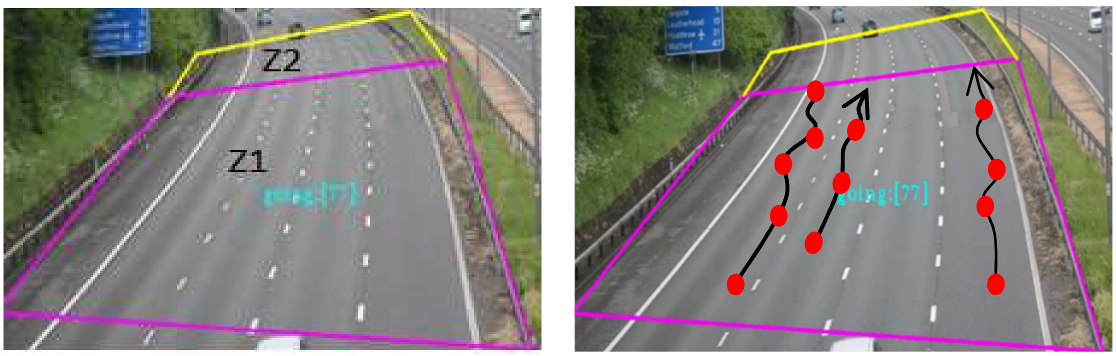
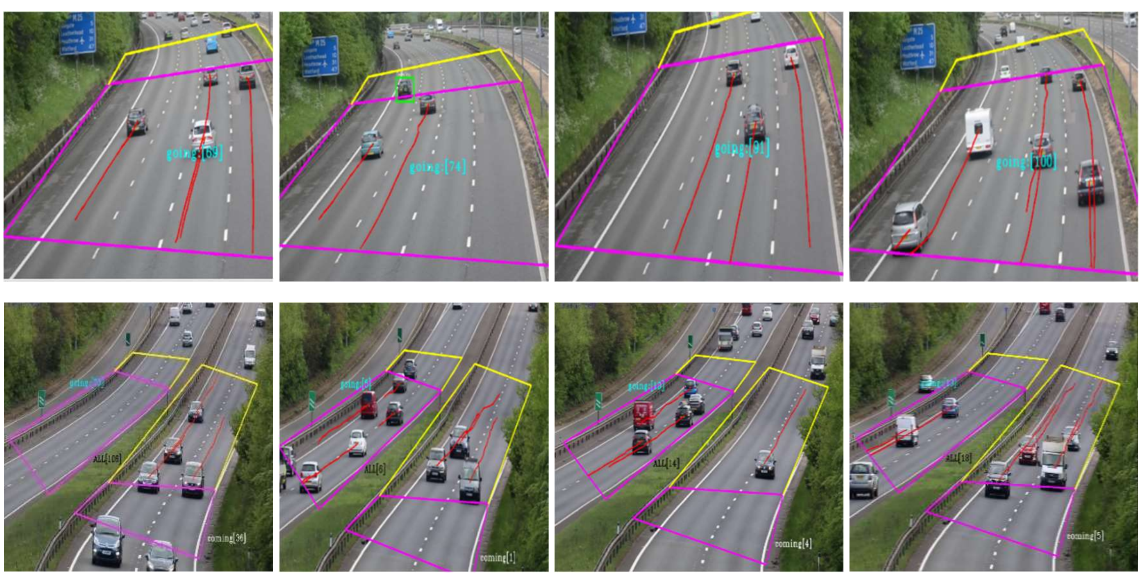


Figure 5: *Drawing a region of interest by the user.*



Journal of Theoretical and Applied Information Technology
15th April 2019. Vol.97. No 7

ISSN: **1992-8645** www.jatit.org E-ISSN: **1817-3195**

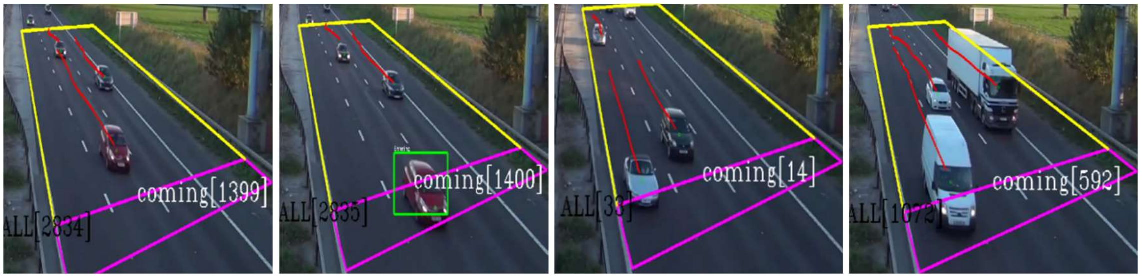


Figure 6: *Vehicle Counting Results from the proposed approach.*

4. DATASETS

We have noticed that most of the existing vehicle-datasets in the literature are only used for vehicle detection purposes and most of the counting systems are also using videos captured by traditional cameras above bridges. However, for training our adapted CNN model and testing the proposed approach, real high-resolution videos form YouTube are used as reported in Table 1. A sample of frames taken from six different videos is shown in Figure 7. The main characteristics of these videos are shown in Table 1. The total time of these collected datasets is equal to 90 minutes. Given that the total number of frames in this dataset is around 200,000 frames, a sample of 939 frames is used for training our proposed CNN.

Journal of Theoretical and Applied Information Technology
15th April 2019. Vol.97. No 7

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Figure 7: *Sample frames from the created dataset*

Table 1: *Main characteristics of the collected YouTube videos.*

Video #	Video Name (as named in YouTube)	Duration (minutes)	Format	Resolution (pixels)	Frame Rate (fps)
Video 1	M6 Motorway Traffic	34:08	mp4	1920×1080	25
Video 2	Relaxing highway traffic	14:00	mp4	1280×720	30
Video 3	M25 Motorway Traffic UK HD - rush hour - British Highway traffic, congestion.	10:00	mp4	1920×1080	24
Video 4	UK M25 London Heathrow / M4 Langley	09:09	mp4	1280×720	30
Video 5	A3 A-road Traffic UK HD - rush hour - British Highway traffic May 2017	06:24	mp4	1920×1080	30
Video 6	Traffic - A406 North Circular Road / B550 Colney Hatch Lane	06:20	mp4	1280×720	30
Video 7	UK Motorway M25 Trucks, Lorries, Cars Highway	04:35	mp4	1920×1080	24
Video 8	M25 Clockwise April 2015 Traffic UK Highway British Motorway	02:55	mp4	1920×1080	24
Video 9	Sample video cars	00:32	mp4	1280×720	30
Video 10	West Side Highway Traffic HD Close up in Manhattan, New York	01:40	mp4	1280×720	30

The proposed approach is evaluated on another dataset that contains six videos; this dataset is freely available for researchers on the site of Amirkabir Computer Vision Lab.[2]. These videos cover various periods of times (noon, afternoon sundown, and night) and the weather situations (day and night) [50][51]. This dataset is used to measure the performance of the proposed method and to compare it with the previous counting methods.

5. EXPERIMENTAL RESULTS AND ANALYSIS

To show the efficiency of our proposed approach, extensive experiments were performed using C language on processor Intel (R) Core (TM) i5-8600k, 8 GB RAM, and GTX1080 graphic card. These experiments were conducted on six real videos from YouTube as shown in Table 2. Videos with different lengths were used (0:32 to 34:08

[2] https://ceit.aut.ac.ir/~safa/Traffic%20Dataset%20(Kamkar-Safabakhsh).rar

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

minutes). The longest one is 'video 1' where its length is 34:08 minutes. The following videos are excluded from these experiments: 'video 4', 'video 6', 'video 8', and 'video 10'. Meanwhile, these videos are only included in the training phase. Related to these experiments, the following points should be taken into our considerations:

- As depicted in Table 2, 'video 1' represents a two-sided highway road where the counting accuracy is 90.9% on this road. The vehicles pass through this way with a high speed and they keep enough distance between each others. While the actual number of passing vehicles is 5090, our proposed real-time approach has counted 4629 of them correctly.
- '*video 2*' is also a two-sided highway road. Our proposed method achieved high accuracy 96.76% where there are 988 passing vehicles at this road.
- In 'video 7' only the vehicles on the left-side were counted because the camera was pointed in the direction of that side. The accuracy on this road is 96.97%.
- Since the vehicles in 'video 3' were closed to each other, the lowest accuracy was achieved with 'video 3' (69.70%). Many vehicles in this video were missed due to the occlusion problem, which in turns effects the tracking and counting stages.
- It is clear from these experiments that our real-time proposed method achieved high accuracy (above 90%) on the highway roads when the vehicles are not close to each other. In this case the occlusion problem is avoided. The same evaluation criteria that were used in OverFeat-based system in [8] are also used here to evaluate our proposed approach.

Table 2: *Performance comparison with CAMLYTICS system.*

Video #	Manual counts	CAMLYTICS		Our proposed approach	
		Counted vehicles	Accuracy	Counted vehicles	Accuracy
Video 1	5090	1449	28.5%	4629	**90.9%**
Video 2	988	671	67.91%	956	**96.76%**
Video 3	1264	295	23.34%	881	**69.70%**
Video 5	522	150	28.74%	417	**79.89%**
Video 7	330	88	26.67%	320	**96.97%**
Video 9	52	16	30.77%	51	**98.08%**
Average Accuracy	-	-	34.32%	-	88.72%

The following equation represents this evaluation metric:

$$Accuracy = 1 - \left(\frac{|\text{manual counts - algorithm's counts}|}{\text{manual counts}}\right) \quad (6)$$

To evaluate the performance of our proposed real-time approach with other competitive approaches, new experiments on the same datasets are performed. We compare our results with another real-time software called CAMLYTICS [52], which is a commercial software for vehicles counting. However, CAMLYTICS failed in many times to detect and track the vehicles, as shown in Table 2.

Another comparison process with OverFeat-based approach in [8] is performed in Table 3. The authors in [8], presented a car counting system based on OverFeat framework. Only one video of their tested videos is uploaded on YouTube, which is '*Video 9*'. This Video contains 52 vehicles. The experimental results showed that our proposed real-time approach successfully counted (51/52) vehicles in that video. On the other hand, OverFeat-based approach counted 50 vehicles from 52 vehicles.

Table 3: *Performance comparison with OverFeat-based approach* [8].

Video #	Manual counts	OverFeat [8]	Our approach
Video 9	52	50	**51**
Accuracy		96.15%	**98.08%**

In order to show the efficiency of our proposed approach, a new comparative evaluation

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

process with two previous vehicle counting approaches [50][51] is performed in Table 4. In this comparison we used six videos from Amirkabir Computer Vision Lab. In [50], the authors reported the best known results on this dataset. Table 4 summarize and compare the counting results from [50][51], with results of our proposed approach. The best results of vehicle-counting are highlighted in a bold font style.

As illustrated in Table 4, the counting result of our proposed real-time vehicle counting is the best among the previous counting methods[50][51]. Furthermore, our proposed RT-VC approach achieved impressive counting results even in challenging situations such as night and rainy weather where there are special and hard circumstances. It is worth noting that the method in [50], has obtained better results over the method in [51]. Finally, we can conclude that our proposed RT-VC approach introduced the best solution for vehicle counting problem by achieving improved outcome with less complexity.

Figure 8, shows the percentage of the correctly counted vehicles accuracies for [50][51]. This chart illustrates the superiority of our proposed real-time vehicle counting approach against the other competitive methods.

Table 4: *Vehicle counting results and comparison of our proposed approach against two methods* [50][51].

Measured Metrics	**Method & Reference**	**Sunny day**				**Rainy day**	
		Noon	**After noon**	**Sundown**	**Night**	**Day**	**Night**
Actual number of vehicle	-	228	413	261	277	128	247
Percentage of correctly counted vehicle	Our proposed approach	**94.7**	**89.8**	**96.9**	**97.1**	**96.9**	**95.6**
	Kamkar and Safabakhsh [50]	90.5	48.7	68.2	66.8	51.6	46.6
	Rashid et al. [51]	86.8	48.4	82.8	30.7	50	43.3
Percentage of falsely counted vehicle	Our proposed approach	1.3	2.2	1.5	4.3	1.6	1.2
	Kamkar and Safabakhsh [50]	7.9	2.9	4.2	21.7	4.7	1.6
	Rashid et al. [51]	182.9	146.5	84.7	141.2	84.4	72.1

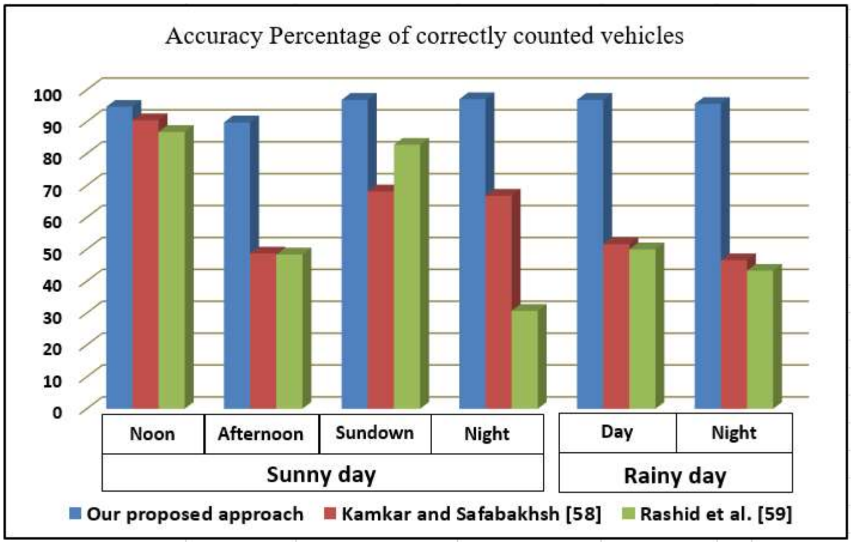


Figure 8: *Accuracy comparison of our proposed counting approach with* [50][51]

6. CONCLUSIONS

This paper proposes and implements a new accurate real-time vehicle counting (RT-VC) approach. The road is divided into zones of interest. Then, the moved vehicles are directly detected using YOLO framework, which is the most powerful deep learning detection framework. After that, the detected vehicles are tracked using Kalman filter and Hungarian algorithms. In order to obtain the final results, the vehicles' locations are checked and counted through the zones of interest.

The experimental outputs have proved that our proposed real-time approach achieves high accuracy for counting vehicles. Experiments on six real videos from YouTube show that our results are excellent and our proposed RT-VC approach outperformed the CAMLYTICS (commercial software) and OverFeat (research proposal) approaches. Another six videos from Amirkabir Computer Vision Lab. were used in a new experiments to compare the performance of our RT-VC approach against Kamkar and Safabakhsh method [50], and Rashid et al. method [51]. As reported in Table4, these new comparative experiments, also, proved that our proposed RT-VC approach achieved the best counting results even in challenging situations such as night and rainy weather where there are special and hard circumstances. In addition, many experiments have been applied in highway roads, where the vehicles move in a high speed and keep enough distance between them. The final achieved accuracy rate exceeded 90% for most of the tested videos. Meanwhile, the speed in average is around 32 fps, which is a normal real-time speed.

Furthermore, the speed of some videos is slightly more than the real-time speed where the real speed is equal 30 fps. Consequently, our proposed real-time vehicle counting approach is very fast and accurate and can be used in real world applications. Zones of interest can be manually defined by the user at the beginning of the counting process. Finally, the user can define more than one side of the road for counting coming and going cars. This means our proposed real-time approach has the ability to count vehicles in two-way traffic with both opposite directions simultaneously.

REFERNECES

[1] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] I. A. Doush *et al.*, "Harmony Search Algorithm for Patient Admission Scheduling Problem," *J. Intell. Syst.*, 2018.

[3] H. Rashaideh *et al.*, "A Grey Wolf Optimizer for Text Document Clustering," *J. Intell. Syst.*, no. July, 2018.

[4] S. Hawkins, Jeff and Blakeslee, *On intelligence: How a new understanding of the brain will lead to the creation of truly intelligent machines*. Macmillan, 2007.

[5] K. M. O. Nahar, R. M. Al-Khatib, M. Al-Shannaq, M. Daradkeh, and R. Malkawi, "Direct Text Classifier for Thematic Arabic Discourse Documents," *Int. Arab J. Inf.*

Technol., vol. 17, no. 3, 2020.
[6] S. Baumgartner and M. Gabele, "Airborne Road Traffic Monitoring with Radar," *World Congr. Intell. Transp. Syst. (ITS), Beijing, China*, no. 1, pp. 1–6, 2007.
[7] K. Garg, S. K. Lam, T. Srikanthan, and V. Agarwal, "Real-time road traffic density estimation using block variance," *2016 IEEE Winter Conf. Appl. Comput. Vis.*, pp. 1–9, 2016.
[8] D. Biswas, H. Su, C. Wang, J. Blankenship, and A. Stevanovic, "An automatic car counting system using overfeat framework," *Sensors (Switzerland)*, vol. 17, no. 7, pp. 1–13, 2017.
[9] K. M. O. Nahar and R. M. Al-Khatib, "EPSSR: Energy preserving system for smart rooms," in *Proceedings of 2nd International Conference on the Applications of Information Technology in Developing Renewable Energy Processes and Systems (IT-DREPS)*, 2017, pp. 1–6.
[10] R. M. Al-Khatib and K. M. O. Nahar, "SRT-GA: Smart real-time system using a powerful genetic algorithm for school bus routing problem," in *Proceedings of 2nd International Conference on the Applications of Information Technology in Developing Renewable Energy Processes and Systems*, 2017, pp. 1–8.
[11] G. Salvi, "An Automated Vehicle Counting System Based on Blob Analysis for Traffic Surveillance," in *Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV)*, 2012, p. 1.
[12] M. Lei, D. Lefloch, P. Gouton, and K. Madani, "A video-based real-time vehicle counting system using adaptive background method," *SITIS 2008 - Proc. 4th Int. Conf. Signal Image Technol. Internet Based Syst.*, pp. 523–528, 2008.
[13] X. Xiang, M. Zhai, N. Lv, and A. El Saddik, "Vehicle counting based on vehicle detection and tracking from aerial videos," *Sensors (Switzerland)*, vol. 18, no. 8, pp. 1–17, 2018.
[14] N. Seenouvong, U. Watchareeruetai, C. Nuthong, K. Khongsomboon, and N. Ohnishi, "A computer vision based vehicle detection and counting system," *2016 8th Int. Conf. Knowl. Smart Technol.*, pp. 224–227, 2016.
[15] H. Van Pham and B. R. Lee, "Front-view car detection and counting with occlusion in dense traffic flow," *Int. J. Control. Autom. Syst.*, vol. 13, no. 5, pp. 1150–1160, 2015.
[16] S. Taghvaeeyan and R. Rajamani, "Portable roadside sensors for vehicle counting, classification, and speed measurement," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 1, pp. 73–83, 2014.
[17] I. Pavlidis, V. Morellas, and N. Papanikolopoulos, "A Vehicle Occupant Counting System Based on Near-Infrared Phenomenology and Fuzzy Neural Classification," *IEEE Trans. Intell. Transp. Syst.*, vol. 1, no. 2, pp. 72–84, 2000.
[18] B. Yang and Y. Lei, "Vehicle detection and classification for low-speed congested traffic with anisotropic magnetoresistive sensor," *IEEE Sens. J.*, vol. 15, no. 2, pp. 1132–1138, 2015.
[19] K. Adi, A. P. Widodo, C. E. Widodo, A. Pamungkas, and A. B. Putranto, "Automatic vehicle counting using background subtraction method on gray scale images and morphology operation," *J. Phys. Conf. Ser.*, vol. 1025, no. 1, 2018.
[20] S.-Y. Chiu, C.-C. Chiu, and S. Xu, "A Background Subtraction Algorithm in Complex Environments Based on Category Entropy Analysis," *Appl. Sci.*, vol. 8, no. 6, p. 885, 2018.
[21] A. Asvadi, L. Garrote, C. Premebida, P. Peixoto, and U. J. Nunes, "Real-Time Deep ConvNet-Based Vehicle Detection Using 3D-LIDAR Reflection Intensity Data," *Adv. Intell. Syst. Comput.*, vol. 694, pp. 475–486, 2018.
[22] X. Li, K. Wang, W. Wang, and Y. Li, "A multiple object tracking method using Kalman filter," *2010 IEEE Int. Conf. Inf. Autom. ICIA 2010*, vol. 1, no. 1, pp. 1862–1866, 2010.
[23] S. Alghyaline, J. W. Hsieh, and C. H. Chuang, "Video action classification using symmelets and deep learning," *2017 IEEE Int. Conf. Syst. Man, Cybern.*, pp. 414–419, 2017.
[24] D. G. Lowe, "Distinctive Image Features from Scale Invariant Keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[25] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-Up Robust Features (SURF)," *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, 2008.
[26] J. Perš, V. Sulić, M. Kristan, M. Perše, K. Polanec, and S. Kovačič, "Histograms of optical flow for efficient representation of body motion," *Pattern Recognit. Lett.*, vol. 31, no. 11, pp. 1369–1376, 2010.
[27] N. Dalal, B. Triggs, O. Gradients, and D. Cordelia, "Histograms of Oriented Gradients for Human Detection," pp. 886–893, 2010.
[28] N. Dalal, B. Triggs, and C. Schmid, "Human Detection using Oriented Histograms of Flow and Appearance," in *European conference on computer vision*, 2006, pp. 428–441.
[29] S. Alghyaline, J. W. Hsieh, H. F. Chiang, and R. Y. Lin, "Action classification using data mining and Paris of SURF-based trajectories," *2016 IEEE Int. Conf. Syst. Man, Cybern.)*, pp. 2163–2168, 2017.
[30] M. Tiezzi, S. Melacci, M. Maggini, and A. Frosini, "Video Surveillance of Highway Traffic Events by Deep Learning Architectures Matteo," in *International Conference on Artificial Neural Networks*, 2018, pp. 584–593.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[31] X. Chen, S. Xiang, C. Liu, and C. Pan, "Vehicle Detection in Satellite Images by Hybrid Deep Convolutional Neural Networks," vol. 11, no. 10, pp. 1797–1801, 2014.

[32] S. E. Kahou, V. Michalski, R. Memisevic, C. Pal, and P. Vincent, "RATM: Recurrent Attentive Tracking Model," *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, pp. 1613–1622, 2017.

[33] J. Ren, Shaoqing and He, Kaiming and Girshick, Ross and Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[34] S. H. Shaikh, K. Saeed, and N. Chaki, "Moving object detection using background subtraction," *SpringerBriefs Comput. Sci.*, pp. 15–23, 2014.

[35] Y. Shen *et al.*, "Real-time and robust compressive background subtraction for embedded camera networks," *IEEE Trans. Mob. Comput.*, vol. 15, no. 2, pp. 406–418, 2016.

[36] M. Daigavane and P. Bajaj., "Real Time Vehicle Detection and Counting Method for Unsupervised Traffic Video on Highways," *IJCSNS Int. J. Comput. Sci. Netw. Secur.*, vol. 10, no. 8, pp. 112–117, 2010.

[37] O. Karaduman, H. Eren, H. Kurum, and M. Celenk, "Approaching car detection via clustering of vertical-horizontal line scanning optical edge flow," *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, pp. 502–507, 2012.

[38] S. Kiani Galoogahi, Hamed and Sim, Terence and Lucey, "Multi-channel correlation filters," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 3072–3079.

[39] K. Liu and G. Mattyus, "Fast Multiclass Vehicle Detection on Aerial Images," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 9, pp. 1938–1942, 2015.

[40] S. Sivaraman and M. M. Trivedi, "A general active-learning framework for on-road vehicle recognition and tracking," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 2, pp. 267–276, 2010.

[41] Q. Gan, Q. Guo, Z. Zhang, and K. Cho, "First Step Toward Model-Free, Anonymous Object Tracking with Recurrent Neural Networks," pp. 1–13, 2015.

[42] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[43] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.

[44] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2999–3007.

[45] J. Huang *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3296–3305.

[46] N. K. T. El-Omari and A. A. Awajan, "Document Image Segmentation and Compression using Artificial Neural Networks and Evolutionary Methods," in *International Conference on Information and Communication Systems (ICICS09)*, 2009, pp. 320–324.

[47] N. K. T. El-omari, A. H. Omari, O. F. Al-badarneh, and H. Abdel-jaber, "Scanned Document Image Segmentation Using Back-Propagation Artificial Neural Network Based Technique," *Int. J. Comput. Commun.*, vol. 6, no. 4, pp. 183–190, 2012.

[48] B. Kwolek, "Particle swarm optimization based object tracking," *Fundam. Informaticae*, vol. 95, no. 4, pp. 449–463, 2009.

[49] H. W. Kuhn, "The Hungarian Method for the Assignment Problem," *Nav. Res. Logist. Q.*, vol. 2, pp. 83–97, 1955.

[50] S. Kamkar and R. Safabakhsh, "Vehicle detection , counting and classification in various conditions," *IET Intell. Transp. Syst.*, vol. 10, no. 6, pp. 406–413, 2016.

[51] N. C. Mithun, N. U. Rashid, and S. M. Rahman, "Detection And Classification of Vehicles from A Video Using Time-Spatial Image," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 3, pp. 1215–1225, 2012.

[52] "Camera Analytics Software for Home & Business," *Camlytics*, 2018. [Online]. Available: Camlytics.com.