

Article

# A Multi-Class Multi-Movement Vehicle Counting Framework for Traffic Analysis in Complex Areas Using CCTV Systems

Khac-Hoai Nam Bui , Hongsuk Yi \* and Jiho Cho

Korea Institute of Science and Technology Information, Daejeon 34141, Korea;  
hoainam.bk2012@gmail.com (K.-H.N.B.); jhcho@kisti.re.kr (J.C.)

\* Correspondence: hsyi@kisti.re.kr

Received: 27 March 2020; Accepted: 16 April 2020; Published: 19 April 2020



**Abstract:** Traffic analysis using computer vision techniques is attracting more attention for the development of intelligent transportation systems. Consequently, counting traffic volume based on the CCTV system is one of the main applications. However, this issue is still a challenging task, especially in the case of complex areas that involve many vehicle movements. This study performs an investigation of how to improve video-based vehicle counting for traffic analysis. Specifically, we propose a comprehensive framework with multiple classes and movements for vehicle counting. In particular, we first adopt state-of-the-art deep learning methods for vehicle detection and tracking. Then, an appropriate trajectory approach for monitoring the movements of vehicles using distinguished regions tracking is presented in order to improve the performance of the counting. Regarding the experiment, we collect and pre-process the CCTV data at a complex intersection to evaluate our proposed framework. In particular, the implementation indicates the promising results of our proposed method, which achieve accuracy around 80% to 98% for different movements for a very complex scenario with only a single view of the camera.

**Keywords:** intelligent transportation system; deep learning; computer vision; vehicle detection and tracking; vehicle counting framework

---

## 1. Introduction

Traffic flow analysis is an important fundamental for urban planning and management of the Intelligent Transportation System (ITS). Recently, the advanced technologies in ITS (e.g., connected vehicles, edge computing, and wireless sensor networks) have enabled huge volumes of traffic data that are available from a variety of sources to provide smart traffic control [1]. However, when the connected environment is still far from reality and developing Wireless Sensor Network (WSN) faces expensive cost and transmission problems, analyzing traffic flow from low-cost video surveillance (CCTV) systems becomes a promising solution [2]. Specifically, by monitoring traffic flow from CCTV, we able to evaluate and verify the performance of the system. Moreover, various applications can be applied based on vehicle detection and tracking using machine vision techniques such as vehicle re-identification (Reid), vehicle classification, and abnormal detection [3]. In the case of vehicle monitoring, the video-based system is able to track the different movements of vehicles by a monocular camera instead of developing multiple sensors locating in each direction of the surveillance systems (e.g., loop detectors). Consequently, video-based vehicle counting becomes a key technique for traffic analysis in complex areas [4,5].

In this paper, we present a practical approach for traffic flow analysis based on data from CCTV systems by proposing a comprehensive vehicle counting framework. Specifically, analyzing traffic flow

based on CCTV systems by using computer vision techniques has recently attracted more attention; however, this research field faces many challenges such as the following:

- Tracking moving vehicles is difficult because of the high similarity of vehicle features, heavy occlusion, a large variation of viewing perspectives, and the low resolution of input videos [6].
- Determining more detail of traffic patterns such as the type of vehicles and turning volume still comprises open research issues, especially in the case of scenarios that include multiple movements (e.g., intersections or roundabouts) [7].
- The scalability of monitoring vehicle movements is a critical problem for turning volume analysis; therefore, it requires a common method that can be applied in various scenarios [8].

In order to address the aforementioned problems, in the proposed vehicle counting framework, we present a tracking-by-detection paradigm for vehicle tracking by adopting state-of-the-art methods. Then, a distinguished region approach is proposed for tracking vehicles to improve counting performance. Specifically, instead of focusing on long-time range tracking of vehicles, we are able to divide the considered scenarios into sub-distinguished regions for vehicle trajectories. Generally, the contributions of this paper are summarized as follows:

- An effective vehicle tracking method to avoid the switch ID and occlusion problems of vehicle tracking, especially in case of heavy occlusion, different lighting and weather conditions. Specifically, state-of-the-art detection and tracking methods are integrated into our framework.
- A comprehensive vehicle counting framework with Multi-Class Multi-Movement (MCMM) counting is presented for analyzing traffic flow in urban areas.
- We collect, pre-process, implement and establish CCTV data at a certain urban area in order to evaluate the proposed framework. Specifically, we focus on complex scenarios in which each intersection covers around 12 movements with a single camera angle that can make the scenarios difficult for monitoring vehicles as shown in Figure 1.



**Figure 1.** An example of a complex intersection with 12 movements with a single view of the CCTV.

The remainder of this paper is structured as follows: The literature review of traffic analysis using DL is presented in Section 2. Moreover, recent object detection and tracking methods and vehicle counting systems are also reviewed in this section. In Section 3, we propose a video-based MCMM vehicle counting for large scale traffic analysis. The experiment results of our proposed approach are presented in Section 4 which is evaluated on the CCTV data that have collected and preprocessed at a complex area. The conclusions of this study are given in Section 5.

## 2. Literature Review

### 2.1. Traffic Analysis Using Deep Learning

The rapid growth of traffic data becomes an emergent challenge in ITS, for which traditional processing systems are not able to deal with the data analytics requirements. Recently, DL has introduced as a promising approach to deal with the various characteristics of traffic data (e.g., highly nonlinear, time-varying, and randomness) [9]. Specifically, different DL models enable different data representations to be learned for different applications. In particular, Figure 2 depicts the applications of DL models for different fundamental tasks in ITS [10]. Specifically, there are four well-known DL models have been widely applied for various applications in ITS such as Deep Neural Networks (DNN), Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN), and Deep Q-Network (DQN).

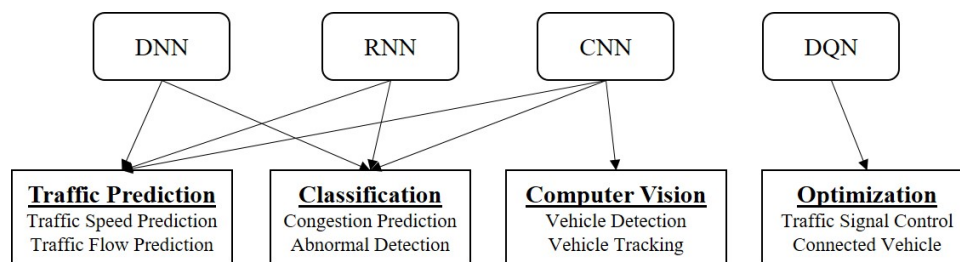


Figure 2. The applications of DL models in ITS.

Consequently, with the recent successful development of CNN-based methods, people detecting and tracking processes have been executed with significant achievements [11]. Recently, vehicle detection and tracking have attracted more attention for the development of ITS. Figure 3 depicts the flowchart of the vehicle monitoring using video data for the smart traffic control system [12]. Specifically, there are three main steps which are: (i) Vehicle detection and tracking are executed for extracting the vehicle information (e.g., type, movement, and speed of vehicles); (ii) Then, traffic analysis tasks (e.g., counting, predictions, and abnormal detection) are performed to understand the traffic condition at a certain time [13,14]; (iii) Finally, the dynamic traffic control algorithms based on traffic condition are proposed to optimize traffic flows [15,16].

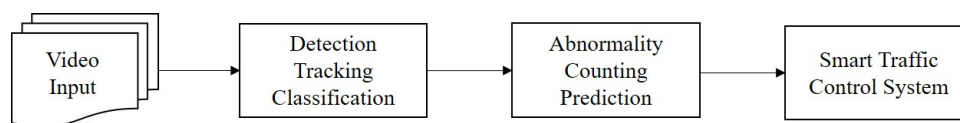


Figure 3. Classic steps in video monitoring.

### 2.2. Moving Object Detection and Tracking Methods

Detection and tracking of moving objects (e.g., people, vehicles, and birds) have been widely applied in many applications (e.g., action recognition, smart traffic control, industrial inspection) and currently represent a major challenge in computer vision [17]. Currently, the standard approach for tracking moving objects from video sequence follows the tracking-by-detection paradigm in which the set of detected bounding boxes extracting in each frame is the input of the tracking process to perform data association for monitoring the object trajectories as shown in Figure 4 [18].

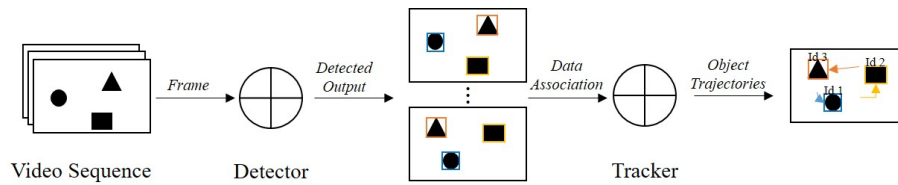


Figure 4. Tracking-by-detection paradigm.

Recently, the rapid development of DL models has achieved great success for object detection in terms of extracting features and classify the type of object [19]. Technically, there are two categories of the object detection methods which are: (i) *Single-stage* methods perform the detection directly over a dense sampling of possible locations, which achieve a high speed of the detection (e.g., SSD [20], YOLO [21], and RetinaNet [22]); (ii) *Two-stage* method first use a region proposal network to generate regions of interests. Then, optimizing the regressions process on the region candidates. Consequently, comparing with Single-stage, this approach achieves a higher accuracy but slower speed for the detection process (e.g., Fast R-CNN [23], Mask R-CNN [24], and R-FCN [25]).

Object tracking is defined as the generation of the path and trajectory process of moving objects across subsequent frames. Specifically, depending on the target of the tracking process, there are two categories of tracking methods which are *Single object tracking* (SOT) and *Multiple object tracking* (MOT) [26]. In the case of SOT, the tracking process do not base on the detection since the methods track a certain object from the beginning. Two well-known methods for SOT are Kalman Filter [27] and Particle Filter [28]. On the other hand, MOT follows the tracking-by-detection paradigm in which the tracking methods reply to the resulting output in each frame of the detection process. Currently, there are two state-of-the-art methods for MOT which are DeepSORT [29], an extension of the SORT algorithm [30] and TC [31], a method using semantic features (e.g., trajectory smoothness and temporal information) for data association in each single view.

### 2.3. Vehicle Counting System

Vehicle counting is one of the main applications of computer vision for traffic management [32]. Figure 5 depicts the general pipeline of a video-based vehicle counting system [5]. Accordingly, detection and tracking processes are executed to detect and monitor vehicles. Then, virtual lines are set for counting vehicles when the centroids of the vehicle pass the lines. Specifically, this concept has been widely applied for both people and vehicle counting [4,33,34].

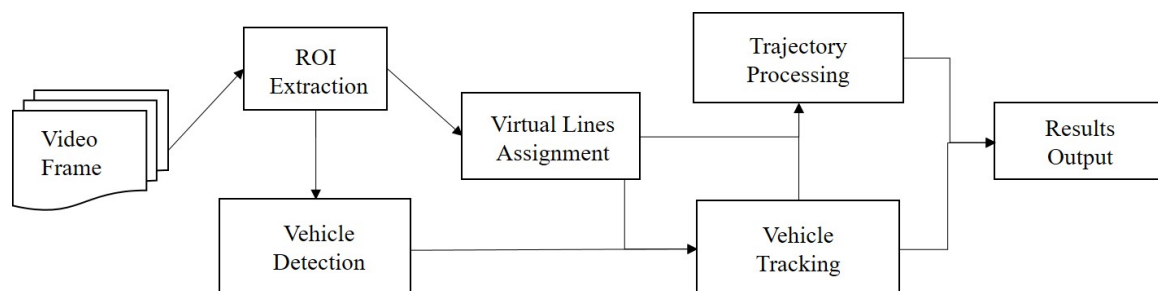


Figure 5. The pipeline of a video-based vehicle counting system.

Recently, many studies have proposed the video-based vehicle counting framework based on this concept. For instance, Xiang et al. [35] presented a novel framework for vehicle counting using aerial video. Specifically, the object can be detected following two cases such as a static background for detecting and moving background to estimate the movement of vehicles. In the case of highway scenarios, Song et al. introduced a counting system by using YOLOv3 for detecting the type and location of vehicles. Then, the ORB algorithm [36] was adopted for the vehicle trajectories [37]. In the case of analyzing traffic flow in complex areas (e.g., intersections) with different types of vehicles,

the authors in [38] proposed a vehicle counting framework by using three-component processes such as object detection, object tracking, and trajectory processing. In particular, the YOLOv3 algorithm was adopted for vehicle detection. Then, a matching method, based on the detection outputs, was proposed for tracking vehicles. Finally, a trajectory counting algorithm based on encoding was proposed for counting vehicles. In order to improve the counting performance, Liu et al. [39] proposed an adaptive pattern based on the virtual loop and detection line methods.

However, referring to previous works, there are several research issues that need to be taken into account to improve the video-based vehicle counting system, which includes:

- Integrating an effective vehicle tracking method in order to deal with the switch ID problem of vehicle tracking in the case of heavy occlusion and different lighting and weather conditions.
- Proposing a counting method by generating semantic regions to deal with the occlusion problem for monitoring and counting vehicles in complex areas that involve complicated directions (e.g., an intersection with 12 conflicting directions/movements).

In this regard, this study proposes a comprehensive framework with multi-class and multi-movement vehicle counting in which we focus on short-term vehicle tracking based on semantic regions in order to improve the tracking process; therefore, the accuracy of counting will be improved. Additionally, we adopt DeepSORT [29], a tracking approach that has proven effectiveness for people tracking, to deal with the varying lighting problem. Furthermore, since our method tracks and counts vehicles by determining the distinguished regions as the input data, the proposed framework is able to work well with different types of scenarios such as intersections, roundabouts, and highways. More detail of our proposed framework is described in the following section.

### 3. Video-Based Multi-Class Multi-Movement Vehicle Counting Framework

#### 3.1. System Architecture

Let  $\mathcal{O}$  represent the set of the result output in which the format of the result is as follows:

$$\mathcal{O} = \langle Vdo_{ID}, V_{ID}, Class_{ID}, Mov_{ID} \rangle \quad (1)$$

where  $Vdo_{ID}$  is the video numeric identifier and  $Mov_{ID}$  denotes the movement identification of  $Vdo_{ID}$ .  $V_{ID}$  and  $Class_{ID}$  represent the identification and type of vehicle, respectively. Figure 6 demonstrates the pipeline of our proposed framework. Specifically, we first follow the tracking-by-detection paradigm for monitoring vehicles, then an appropriate distinguished region approach for reducing long-time tracking is executed to improve the performance of the counting.

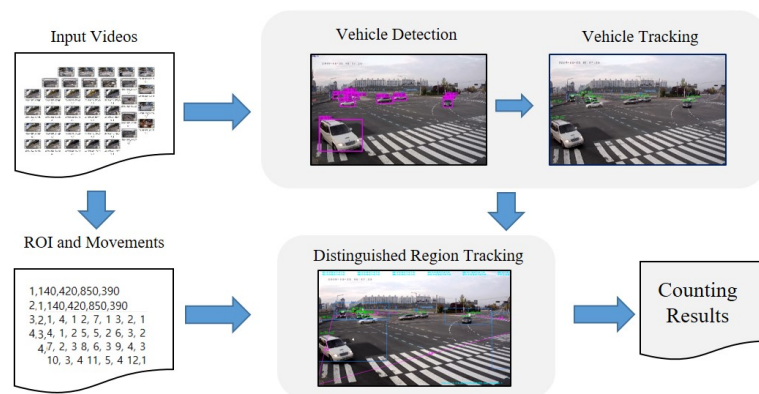


Figure 6. Pipeline of the proposed framework.



### 3.2. Methodology

#### 3.2.1. Vehicle Detection

This process is an essential step for MOT in terms of extracting object information (i.e., location and class) [40]. For the proposed framework, we adopt Yolo which trained on MS-COCO dataset for the vehicle detection process for the following reasons:

- The method belongs to the single-stage which is able to perform the detection process much faster than two-stage methods.
- Recently, the new version of Yolo (Yolov3) is able to perform the high accuracy of the detection for MOT by conducting 53 convolutional layers, which is able to work well with various dimensions in each frame [41].
- MS-COCO dataset provides the labeling and segmentation of over 80 different classes of objects. In this regard, we are able to detect and track with different types of vehicles such as Car, Bus, Truck, and Bike [42].

Specifically, Figure 7 illustrates the vehicle detection process by adopting Yolov3 of the input video. Consequently, the output of this process is a list of the bounding boxes in each frame as follows:

$$\langle class, x, y, w, h, confidence \rangle \quad (2)$$

where *class* and *confidence* denote the type (e.g., car, bus, truck, and bike) and the score of the detected vehicle, respectively. The parameters (*x, y, w, h*) indicate the position of the bounding box. Consequently, to optimize the performance of the detection, the output of the detection process are filtered as follows:

- The *class* of the detected object belongs to the type of vehicles such as a car, bus, truck, and bike.
- The *confidence* of the detected object is larger than a certain threshold.

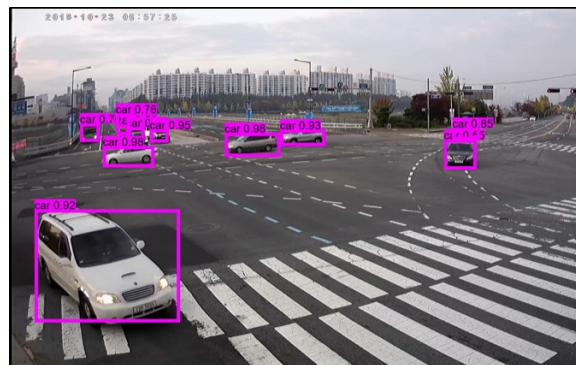


Figure 7. An example of the detection process using Yolov3.

#### 3.2.2. Vehicle Tracking

In this study, since the main focus is to extract traffic flow information of independent scenarios, we adopt the DeepSORT method for the vehicle tracking process. Specifically, there are two main processes in DeepSORT, which include the following: (i) Hungarian algorithm is applied to identify the vehicle appearance across frames; and (ii) Kalman Filter algorithm is used to predict future positions to update the target state. Therefore, the output of this process for a target vehicle is formatted as follows:

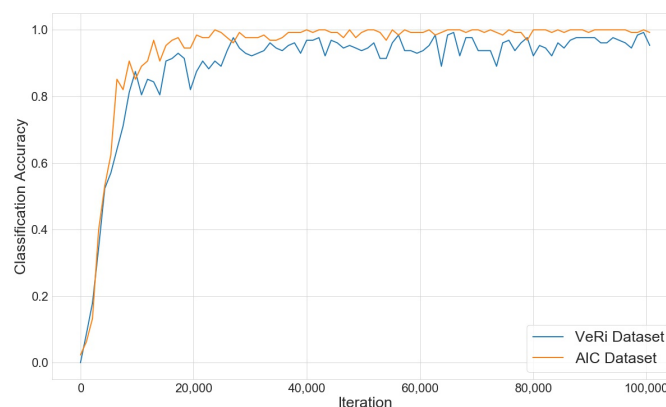
$$\langle x, y, w, h, \dot{x}, \dot{y}, \dot{w}, \dot{h} \rangle \quad (3)$$

where (*x, y, w, h*) and ( $\dot{x}, \dot{y}, \dot{w}, \dot{h}$ ) represent the current and velocity of the target vehicles, respectively. Specifically, Figure 8 illustrates the vehicle tracking process using DeepSORT.



**Figure 8.** Vehicle tracking using DeepSORT.

Furthermore, in order to get the feature extractor of vehicles, we adopted two popular vehicle datasets, which are the VeRi [43] and CityFlow [31] datasets, for training the appearance features instead of using the original feature extraction of DeepSORT, which focuses on a people dataset (i.e., MARS [44]). Consequently, Figure 9 depicts the classification accuracy by training the appearance descriptor using the deep cosine metric [45] of both aforementioned datasets.



**Figure 9.** Classification accuracy comparison between AIC and the VeRi dataset using the deep cosine metric.

### 3.2.3. Distinguished Region Tracking-Based Vehicle Counting

As we mentioned above, for counting vehicles, virtual lines are set in each direction to record the traffic volume as shown in Figure 10. Specifically, in each frame, the current centroid position ( $p_{cur}^v$ ) of the detected vehicle is computed. Then, the vehicle will be counted at a certain virtual line if the centroid of the vehicle passes the line, and the movement of the vehicle will be calculated based on the corresponding line that the vehicle enters the area. For more detail, the Algorithm 1 demonstrates vehicle counting using virtual lines.

However, the main challenge for monitoring the turning movement of vehicles is that we have to track the vehicle in a long-time range. Therefore, the switch ID problem, especially in the scenarios of heavy occlusion and varying lighting conditions, will significantly affect the performance of the counting. In this regard, we define a set of distinguished regions for reducing the range of tracking. Consequently, instead of using the virtual line, we used distinguished regions for counting the vehicles.

**Algorithm 1:** MCMM vehicle counting using virtual lines.

---

**Data:** Current and Previous Bounding Box  $T_{cur}$  and  $T_{pre}$ ;  
Set  $\mathcal{L}$  of virtual lines; Set  $\mathcal{R}$  of regions; Set  $\mathcal{M}$  of movements

**Result:** Counting Result

- 1 **Set**  $(x, y, w, h, cl) = (T_{cur}[0]), T_{cur}[1], T_{cur}[2], T_{cur}[3], T_{cur}[4]$
- 2 **Set**  $(\dot{x}, \dot{y}, \dot{w}, \dot{h}) = (T_{pre}[0]), T_{pre}[1], T_{pre}[2], T_{pre}[3]$
- 3  $p_{cur}^v = (x + \frac{w-x}{2}; y + \frac{h-y}{2})$
- 4  $p_{pre}^v = (\dot{x} + \frac{\dot{w}-\dot{x}}{2}; \dot{y} + \frac{\dot{h}-\dot{y}}{2})$
- 5  $class^v = cl$
- 6 **while**  $l \in L$  **do**
- 7 **if**  $Intersect(p_{cur}^v, p_{pre}^v, l[x], l[y])$  and  $v \notin \mathbb{T}$  **then**
- 8  $\mathbb{T} \leftarrow v, l$
- 9 **end**
- 10 **if**  $Intersect(p_{cur}^v, p_{pre}^v, l[x], l[y])$  **then**
- 11  $l_s = \text{CheckSource}(v)$
- 12  $mo_{temp} = \text{CheckMovement}(l_s, l)$
- 13 **return**  $\text{Count}(mo_{temp}, class^v)$
- 14 **end**
- 15 **end**
- 16 **Function**  $\text{CheckSource}(v)$
- 17 **for**  $i \in \mathbb{T}$  **do**
- 18 **if**  $\mathbb{T}[i][0] = v$  **then**
- 19 **return**  $\mathbb{T}[i][1]$
- 20 **end**
- 21 **end**
- 22 **Function**  $\text{CheckMovement}(l_s, l_d)$
- 23 **while**  $mo \in \mathcal{M}$  **do**
- 24 **if**  $mo[1] = l_s$  and  $mo[2] = l_d$  **then**
- 25 **return**  $mo[0]$
- 26 **end**
- 27 **end**
- 28 **Function**  $\text{Count}(mo, class)$
- 29 **while**  $i \in \mathcal{M}$  **do**
- 30 **if**  $i = mo$  **then**
- 31 **return**  $mo_i^{class} += 1$
- 32 **end**
- 33 **end**

---

**Distinguished Regions' Definition:** We define a set of distinguished regions  $\mathcal{R}$  in each scenario/camera, which is able to cover all the movements for monitoring vehicles. Specifically, the number of regions depends on the number of turning movements.

For instance, Figure 11 depicts an example of the set of regions for monitoring the movement of vehicles. In particular, instead of using virtual lines, we used distinguished regions for tracking and counting vehicles, which were able to improve the performance of the vehicle counting for the two following reasons:

- Reducing the range of tracking.
- Avoiding occlusion in the case of multiple vehicle passing at the same time.



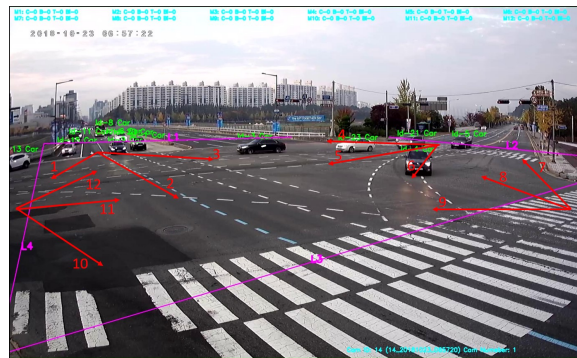


Figure 10. Vehicle counting using multiple virtual lines corresponding to the number of directions.

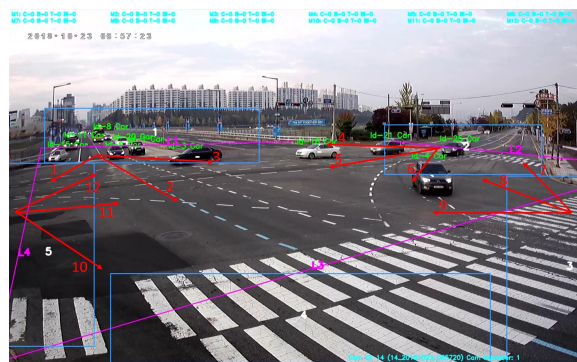


Figure 11. Set of distinguished regions for vehicle monitoring.

**Conflicted Regions Tracking:** Since different scenarios have different geographies and viewing angles of the camera, the vehicles might move across multiple regions. In particular, in some specific cases, multiple movement counting in one vehicle will occur. In order to deal with this issue, we define a list  $\mathbb{T}$ , which is the tracking list of vehicles. Specifically, when a vehicle belongs to the list  $\mathbb{T}$ , it will not be tracked if moving to other regions, which is demonstrated in Figure 12.

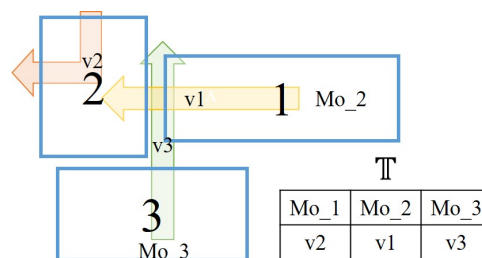
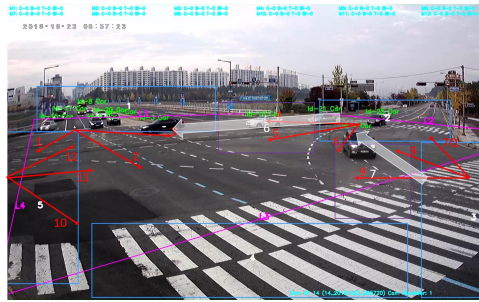


Figure 12. The description of vehicles across multiple regions. Accordingly, the vehicle is only tracked in the first (original) region in which the vehicle moves.

Moreover, another issue of the overlapped regions in a given movement is that vehicles might not be tracked in the original region, but in the others because of the detection problem (e.g., heavy occlusion). Consequently, the movement counting of the vehicle will be wrong. For example, in the scenario of Figure 10, the vehicle in Movement 5 will be counted in Movement 1 in the case it cannot be detected in Region 2. In this regard, we define a set of blank regions  $\mathcal{R}_0$  between tracking regions to reduce the wrong counting in this situation. Specifically, since the blank regions will include the vehicle in the case the vehicle is not detected in the original region, the rate of wrong movement counting will be reduced, which is demonstrated in Figure 13.



**Figure 13.** An example of blank regions for reducing the wrong movement counting. Specifically, blank regions (purple rectangles) are set between tracking regions in order to obtain the vehicles that are not detected in the original regions.

**Multi-Movement Counting Formulation:** The vehicles will be counted if they move to the exit area (region) of the scenario. Accordingly, when the vehicle passes the exit region, the corresponded movement will be counted following the information of source and destination regions. Supposing  $\mathcal{R}^i = r_1^i, r_2^i, \dots, r_n^i$  is the set of regions in scenario/camera  $i$ , the targeted vehicle  $v$  will be tracked in region  $r$  if  $v$  moves into  $r$ , which can be formulated as follows:

$$r = \{v \in \mathcal{V} \mid \forall v \in 1 \dots m : p_0^r \leq p_0^v \leq p_1^r\} \quad (4)$$

where  $p_0^v$  represents the centroid point of vehicle  $v$  and  $p_0^r$  and  $p_1^r$  are the top left corner and bottom right corner of region  $r$ , respectively. Consequently, the set of movements in scenario  $i$ ,  $\mathcal{M}^i = mo_1^i, mo_2^i, \dots, mo_k^i$ , will be counted based on the information of the exit region and the tracking region of the vehicles. In particular, Algorithm 2 demonstrates the modification of our proposed method compared with using multiple virtual lines in Algorithm 1.

---

**Algorithm 2:** MCM Vehicle Counting using Distinguished Regions.

---

**Data:** Current and Previous Bounding Box  $T_{cur}$  and  $T_{pre}$ ;

Set  $\mathcal{L}$  of virtual lines; Set  $\mathcal{R}$  of regions; Set  $\mathcal{M}$  of movements

**Result:** Counting Result

```

1 Set  $(x, y, w, h, cl) = (T_{cur}[0]), T_{cur}[1], T_{cur}[2], T_{cur}[3], T_{cur}[4])$ 
2 Set  $(\hat{x}, \hat{y}, \hat{w}, \hat{h}) = (T_{pre}[0]), T_{pre}[1], T_{pre}[2], T_{pre}[3])$ 
3  $p_{cur}^v = (x + \frac{w-x}{2}; y + \frac{h-y}{2})$ 
4  $p_{pre}^v = (\hat{x} + \frac{\hat{w}-\hat{x}}{2}; \hat{y} + \frac{\hat{h}-\hat{y}}{2})$ 
5  $class^v = cl$ 
6 while  $r \in \mathcal{R}$  do
7   if  $CheckZone(p_{cur}^v, p_0^r, p_1^r)$  and  $v \notin \mathbb{T}$  then
8      $\mathbb{T} \leftarrow v$ 
9     return  $r$ 
10  end
11 end
12 while  $r \in \mathcal{R}_{exit}$  do
13   if  $CheckZone(p_{cur}^v, p_0^r, p_1^r)$  then
14      $r_s = CheckSource(v)$ 
15      $mo_{temp} = CheckMovement(r_s, r)$ 
16     return  $Count(mo_{temp}, class^v)$ 
17   end
18 end
19 Function  $CheckZone(a,b,c)$ 
20 return  $(a[0] > b[0] \text{ and } a[0] < a[1] \text{ and } a[1] > b[1] \text{ and } a[1] < c[1])$ 

```

---

## 4. Experiment

### 4.1. Data Description and Experiment Setup

The dataset contained 39 video clips captured from the same area in which we pre-processed each video for around 10 min. Specifically, the input videos were classified into three time periods such as in the morning (7 a.m.–9 a.m.), afternoon (12 p.m.–2 p.m.), and night (5 p.m.–7 p.m.), which covered different conditions (e.g., lighting). The resolution of each video was around 1080p at 30 Frames Per Second (FPS). The Regions and Movements Of Interest (ROI and MOI) were also determined as the input information. In particular, Particularly, Figure 14 demonstrates the considered area for the implementation.



**Figure 14.** Considered area.

For more detail, Table 1 shows the parameters that we used for the implementation. Consequently, there were four classes of vehicles that we took into account: Car, Bus, Truck, and Bike.

**Table 1.** Parameter Setting.

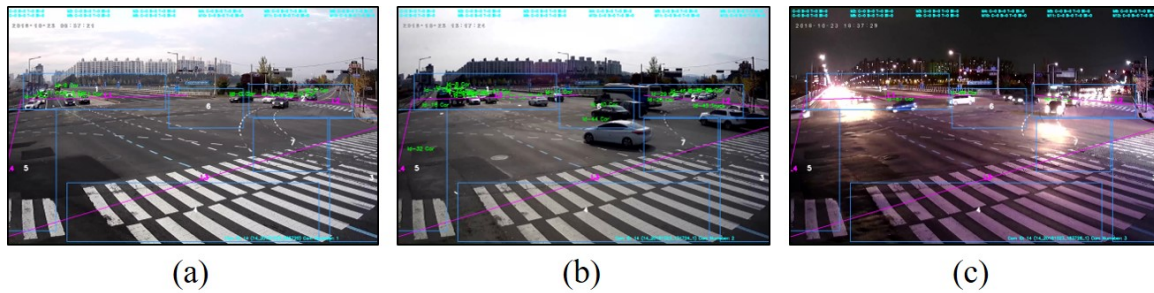
Parameter	Values
Resolution	1920 × 1080
Video Duration	10 min
Frame Rate	30 FPS
Confidence Score Threshold	0.6
Number of Movements	12
Number of Classes	04
Image Reshape Training	128 × 64

### 4.2. Experiment Results

Regarding the experiment, we first compared the counting results with the ground truth in several representative videos in each time period. Then, we compared our proposed method with vehicle counting using virtual lines, which is presented in Algorithm 1. Finally, we adopted the counting results to analyze traffic flow during a certain time period.

#### 4.2.1. Counting Performance

Figure 15 depicts the screenshots of the multi-class multi-movement vehicle counting framework for three considered videos that belonged to different time periods. Accordingly, the experiments worked well with a PC with Core i7 16-GB CPU and 32GB GPU memory in which the GPU was used for the acceleration. As for the observation from the experiments, the FPS is around 12 to 14 which depends on the traffic flow densities.



**Figure 15.** The screenshots of the input videos with different time periods with different condition of lighting: (a) 7 a.m.–9 a.m.; (b) 12 p.m.–2 p.m.; and (c) 5 p.m.–7 p.m.

Table 2 shows the counting results compared with the ground truth. Specifically, with the high condition of lighting and low traffic density, our proposed method was able to achieve more than 90% of the detected vehicles. Moreover, it achieved around 88% and 84% in the case of high traffic density and in the night condition, respectively.

**Table 2.** Counting results compared with the ground truth with test case videos (Vdo).

Video ID	Time Duration	Condition	Ground Truth	Vehicle Counting	Accuracy
Vdo 1	06:57:21–07:07:21	Morning	565	516	92.47%
Vdo 2	13:17:21–13:27:21	Afternoon	814	721	88.57%
Vdo 3	18:37:21–18:47:21	Night	952	804	84.45%

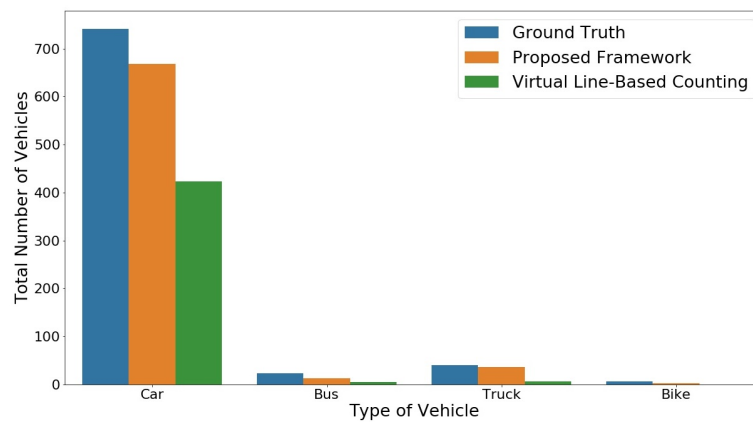
For more detail, Table 3 shows the accuracy in each movement of the first video. In particular, movements 1, 3, 4, and 8 (Figure 10) belong to the opposite side of the camera angle cause lower vehicle monitoring performance than other movements.

**Table 3.** Detailed results in each movement (Mov) of vehicle counting at Vdo 1.

Movement ID	Count Car/ Ground Truth	Count Bus/ Ground Truth	Count Truck/ Ground Truth	Count Bike/ Ground Truth	Accuracy
Mov 1	52/64	2/3	2/1	0/0	80.88%
Mov 2	157/165	2/6	14/8	0/1	92.77%
Mov 3	11/8	0/3	4/4	0/0	80%
Mov 4	2/3	2/2	0/0	0/0	80%
Mov 5	34/30	0/2	1/1	0/2	88.57%
Mov 6	26/25	0/1	0/1	0/0	92.59%
Mov 7	8/9	0/0	1/0	0/0	88.89%
Mov 8	43/52	4/6	4/5	1/1	81.25%
Mov 9	21/21	1/2	5/4	0/0	96.29%
Mov 10	61/61	0/0	9/8	1/2	98.59%
Mov 11	23/28	0/1	9/9	0/0	84.21%
Mov 12	13/16	0/0	3/3	0/0	84.21%

Specifically, as for the observation from the results, another issue of the decreasing accuracy was traffic densities since occlusion frequently occurred. Moreover, the similar appearance features between *Bus* and *Truck* sometimes made the wrong detection, especially when the view of the detected vehicles changed rapidly (e.g., Movement 2). However, the counting system was able to achieve good

results by applying our proposed method. In particular, Figure 16 demonstrates our results compared with vehicle counting using virtual lines (Algorithm 1).

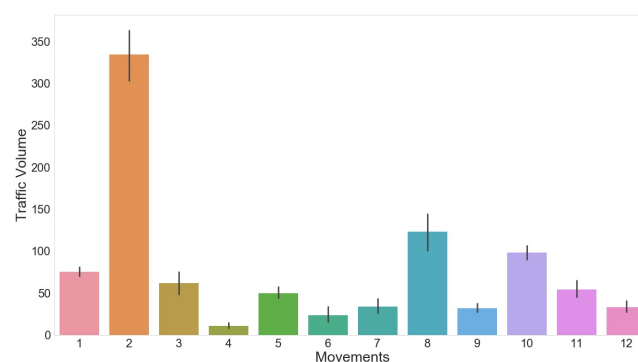


**Figure 16.** The counting comparison between the proposed framework and the counting using the virtual line method in Vdo 2.

#### 4.2.2. Traffic Analysis Based on Counting Results

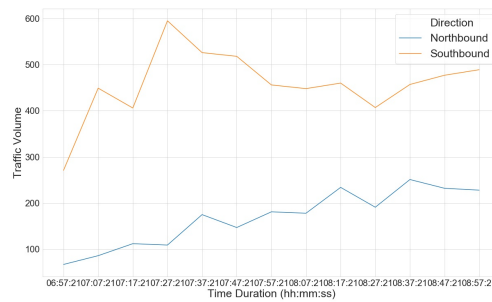
As we mentioned above, one of the main applications of vehicle counting is to determine the traffic pattern and analyze the traffic turning volume for smart traffic control applications in complex areas. In this regard, we executed several implementations for analyzing traffic flow based on the counting results from the proposed framework. Specifically, Figure 17 demonstrates the average traffic turning volume in the morning time (from 7 a.m. to 9 a.m.). As shown in the figure, there was a big difference in the traffic volume for different movements, which could be an important result for applying dynamic traffic light control in order to improve the traffic flow [15].

Furthermore, Figure 18 shows the asymmetric traffic volumes in the considered area with different time intervals. Specifically, the traffic patterns of the implementation made sense since the time intervals were the time for going (Figure 18a) to work and coming home (Figure 18b), respectively.

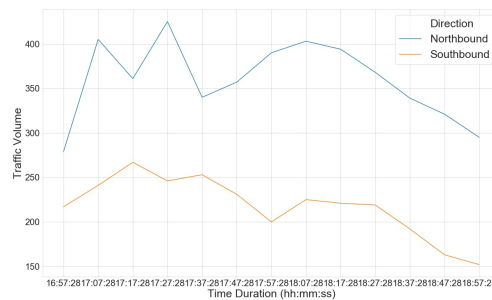


**Figure 17.** Traffic volume turning in the morning time (7:00 a.m.–9:00 a.m.).





(a)



(b)

**Figure 18.** Asymmetric traffic volumes at the main bounds. (a) In the morning; (b) At night.

## 5. Conclusions and Future Work

Recently, with the successful development of DL for computer vision, video-based vehicle counting has become a promising solution for analyzing traffic flow. However, this issue is still a challenging task due to the high similarity of vehicle appearances, heavy occlusion with high traffic density, and large variation in different viewing perspectives. In this study, we presented a comprehensive framework for multi-class multi-movement vehicle counting. Specifically, we first adopted state-of-the-art methods of object detection and tracking such as YOLO and DeepSORT for monitoring vehicles. Furthermore, a distinguished region tracking approach was proposed in order to improve vehicle tracking. As shown in the experiment, our proposed method was able to achieve a high performance for the counting results in which we evaluated the proposed method on the real data that we collected and pre-processed in a certain area.

From our point of view, there were several issues that could improve the proposed framework for vehicle counting as follows: (i) training an appropriate dataset for the detection process that is able to distinguish different types of vehicles (e.g., sedan, SUV, van, pickup truck, main truck, tractor-trailer, and 18-wheeler trucks); specifically, the COCO datasets only classify four types of vehicles, which are car, bus, truck, and bike; (ii) determining distinguished regions for tracking vehicles is still a manual process; in this regard, an optimal approach for generating the regions was able to improve the performance of counting. Moreover, tracking and counting vehicles across multiple cameras (multi-cameras tracking problems) using the proposed framework enabled the large-scale traffic flow analysis. The aforementioned problems are interesting issues that we will take into account in future work regarding this study.

**Author Contributions:** Conceptualization, K.-H.N.B., J.C., and H.Y.; methodology, K.-H.N.B. and H.Y.; investigation, K.-H.N.B.; data curation, J.C.; writing, original draft preparation, K.-H.N.B.; writing, review and editing, J.C. and H.Y.; project administration, H.Y.; funding acquisition, H.Y. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was partly supported by the Institute for Information & communications Technology Promotion (IITP) grant funded by the Korean government (MSIT) (No. 2018-0-00494, Development of deep learning-based urban traffic congestion prediction and signal control solution system) and the Korea Institute of Science and Technology Information (KISTI) grant funded by the Korean government (MSIT) (K-19-L02-C07-S01).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Bui, K.H.N.; Lee, O.; Jung, J.J.; Camacho, D. Dynamic Traffic Light Control System Based on Process Synchronization Among Connected Vehicles. In Proceedings of the 7th International Symposium on Ambient Intelligence (ISAmI), Seville, Spain, 1–3 June 2016; pp. 77–85. [\[CrossRef\]](#)
- Fedorov, A.; Nikolskaia, K.; Ivanov, S.; Shepelev, V.; Minbaleev, A. Traffic flow estimation with data from a video surveillance camera. *J. Big Data* **2019**, *6*, 73. [\[CrossRef\]](#)
- Bui, K.H.N.; Cho, S.; Jung, J.J.; Kim, J.; Lee, O.; Na, W. A novel network virtualization based on data analytics in connected environment. *J. Ambient Intell. Humaniz. Comput.* **2020**, *11*, 75–86. [\[CrossRef\]](#)
- Xia, Y.; Shi, X.; Song, G.; Geng, Q.; Liu, Y. Towards improving quality of video-based vehicle counting method for traffic flow estimation. *Signal Process.* **2016**, *120*, 672–681. [\[CrossRef\]](#)
- Bui, K.H.N.; Yi, H.; Jung, H.; Cho, J. Video-Based Traffic Flow Analysis for Turning Volume Estimation at Signalized Intersections. In Proceedings of the 12th Asian Conference on Intelligent Information and Database Systems (ACIIDS), Phuket, Thailand, 23–26 March 2020; pp. 152–162. [\[CrossRef\]](#)
- Tang, Z.; Wang, G.; Xiao, H.; Zheng, A.; Hwang, J. Single-Camera and Inter-Camera Vehicle Tracking and 3D Speed Estimation Based on Fusion of Visual and Semantic Features. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, 18–22 June 2018; pp. 108–115.
- Zhao, R.; Wang, X. Counting Vehicles from Semantic Regions. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 1016–1022. [\[CrossRef\]](#)
- Shirazi, M.S.; Morris, B.T. Vision-Based Turning Movement Monitoring: Count, Speed & Waiting Time Estimation. *IEEE Intell. Transport. Syst. Mag.* **2016**, *8*, 23–34. [\[CrossRef\]](#)
- Zhu, L.; Yu, F.R.; Wang, Y.; Ning, B.; Tang, T. Big Data Analytics in Intelligent Transportation Systems: A Survey. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 383–398. [\[CrossRef\]](#)
- Wang, Y.; Zhang, D.; Liu, Y.; Dai, B.; Lee, L.H. Enhancing transportation systems via deep learning: A survey. *Transp. Res. Part C Emerg. Technol.* **2019**, *99*, 144–163. [\[CrossRef\]](#)
- Brunetti, A.; Buongiorno, D.; Trotta, G.F.; Bevilacqua, V. Computer vision and deep learning techniques for pedestrian detection and tracking: A survey. *Neurocomputing* **2018**, *300*, 17–33. [\[CrossRef\]](#)
- Datondji, S.R.E.; Dupuis, Y.; Subirats, P.; Vasseur, P. A Survey of Vision-Based Traffic Monitoring of Road Intersections. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 2681–2698. [\[CrossRef\]](#)
- Yi, H.; Bui, K.H.N. VDS Data-Based Deep Learning Approach for Traffic Forecasting Using LSTM Network. In Proceedings of the 19th EPIA Conference on Artificial Intelligence (EPIA), Yogyakarta, Indonesia, 8–11 April 2019; pp. 547–558. [\[CrossRef\]](#)
- Yi, H.; Bui, K.H.N.; Jung, H. Implementing A Deep Learning Framework for Short Term Traffic Flow Prediction. In Proceedings of the 9th International Conference on Web Intelligence, Mining and Semantics (WIMS), Seoul, Korea, 26–28 June 2019; pp. 7:1–7:8. [\[CrossRef\]](#)
- Bui, K.H.N.; Jung, J.E.; Camacho, D. Game theoretic approach on Real-time decision making for IoT-based traffic light control. *Concurr. Comput. Pract. Exp.* **2017**, *29*, e4077. [\[CrossRef\]](#)
- Bui, K.H.N.; Jung, J.J. Cooperative game-theoretic approach to traffic flow optimization for multiple intersections. *Comput. Electr. Eng.* **2018**, *71*, 1012–1024. [\[CrossRef\]](#)
- Naphade, M.; Tang, Z.; Chang, M.; Anastasiu, D.C.; Sharma, A.; Chellappa, R.; Wang, S.; Chakraborty, P.; Huang, T.; Hwang, J.; et al. The 2019 AI City Challenge. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Long Beach, CA, USA, 15–21 June 2019; pp. 452–460.
- Ciaparrone, G.; Sánchez, F.L.; Tabik, S.; Troiano, L.; Tagliaferri, R.; Herrera, F. Deep learning in video multi-object tracking: A survey. *Neurocomputing* **2020**, *381*, 61–88. [\[CrossRef\]](#)
- Jiao, L.; Zhang, F.; Liu, F.; Yang, S.; Li, L.; Feng, Z.; Qu, R. A Survey of Deep Learning-Based Object Detection. *IEEE Access* **2019**, *7*, 128837–128868. [\[CrossRef\]](#)

20. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.E.; Fu, C.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the 14th European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37. [[CrossRef](#)]
21. Redmon, J.; Divvala, S.K.; Girshick, R.B.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 779–788. [[CrossRef](#)]
22. Lin, T.; Goyal, P.; Girshick, R.B.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2999–3007. [[CrossRef](#)]
23. Girshick, R.B. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448. [[CrossRef](#)]
24. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R.B. Mask R-CNN. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988. [[CrossRef](#)]
25. Dai, J.; Li, Y.; He, K.; Sun, J. R-FCN: Object Detection via Region-based Fully Convolutional Networks. In Proceedings of the 30th Annual Conference on Neural Information Processing Systems (NIPS), Barcelona, Spain, 5–10 December 2016; pp. 379–387.
26. Zhong, Z.; Yang, Z.; Feng, W.; Wu, W.; Hu, Y.; Liu, C. Decision Controller for Object Tracking With Deep Reinforcement Learning. *IEEE Access* **2019**, *7*, 28069–28079. [[CrossRef](#)]
27. Weng, S.; Kuo, C.M.; Tu, S. Video object tracking using adaptive Kalman filter. *J. Vis. Commun. Image Represent.* **2006**, *17*, 1190–1208. [[CrossRef](#)]
28. Chang, C.; Ansari, R. Kernel particle filter for visual tracking. *IEEE Signal Process. Lett.* **2005**, *12*, 242–245. [[CrossRef](#)]
29. Wojke, N.; Bewley, A.; Paulus, D. Simple online and realtime tracking with a deep association metric. In Proceedings of the 24th International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 3645–3649. [[CrossRef](#)]
30. Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.T.; Upcroft, B. Simple online and realtime tracking. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 3464–3468. [[CrossRef](#)]
31. Tang, Z.; Naphade, M.; Liu, M.; Yang, X.; Birchfield, S.; Wang, S.; Kumar, R.; Anastasiu, D.C.; Hwang, J. CityFlow: A City-Scale Benchmark for Multi-Target Multi-Camera Vehicle Tracking and Re-Identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Long Beach, CA, USA, 16–20 June 2019; IEEE Computer Society: Washington, DC, USA, 2019; pp. 8797–8806.
32. Zhang, S.; Wu, G.; Costeira, J.P.; Moura, J.M.F. FCN-rLSTM: Deep Spatio-Temporal Neural Networks for Vehicle Counting in City Cameras. In Proceedings of the International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 3687–3696. [[CrossRef](#)]
33. Pizzo, L.D.; Foggia, P.; Greco, A.; Percannella, G.; Vento, M. Counting people by RGB or depth overhead cameras. *Pattern Recognit. Lett.* **2016**, *81*, 41–50. [[CrossRef](#)]
34. Kocak, Y.P.; Sevgen, S. Detecting and counting people using real-time directional algorithms implemented by compute unified device architecture. *Neurocomputing* **2017**, *248*, 105–111. [[CrossRef](#)]
35. Xiang, X.; Zhai, M.; Lv, N.; El-Saddik, A. Vehicle Counting Based on Vehicle Detection and Tracking from Aerial Videos. *Sensors* **2018**, *18*, 2560. [[CrossRef](#)]
36. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G.R. ORB: An efficient alternative to SIFT or SURF. In Proceedings of the International Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011; pp. 2564–2571. [[CrossRef](#)]
37. Song, H.; Liang, H.; Li, H.; Dai, Z.; Yun, X. Vision-based vehicle detection and counting system using deep learning in highway scenes. *Eur. Trans. Res. Rev.* **2019**, *11*, 51. [[CrossRef](#)]
38. Dai, Z.; Song, H.; Wang, X.; Fang, Y.; Yun, X.; Zhang, Z.; Li, H. Video-Based Vehicle Counting Framework. *IEEE Access* **2019**, *7*, 64460–64470. [[CrossRef](#)]
39. Fei, L.; Zhiyuan, Z.; Rong, J. A video-based real-time adaptive vehicle-counting system for urban roads. *PLoS ONE* **2017**, *12*, e0186098. [[CrossRef](#)]
40. Zhao, Z.; Zheng, P.; Xu, S.; Wu, X. Object Detection with Deep Learning: A Review. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3212–3232. [[CrossRef](#)] [[PubMed](#)]
41. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.

42. Lin, T.; Maire, M.; Belongie, S.J.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In Proceedings of the 13th European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
43. Liu, X.; Liu, W.; Ma, H.; Fu, H. Large-scale vehicle re-identification in urban surveillance videos. In Proceedings of the International Conference on Multimedia and Expo (ICME), Seattle, WA, USA, 11–15 July 2016; pp. 1–6. [[CrossRef](#)]
44. Zheng, L.; Bie, Z.; Sun, Y.; Wang, J.; Su, C.; Wang, S.; Tian, Q. MARS: A Video Benchmark for Large-Scale Person Re-Identification. In Proceedings of the 14th European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2016; pp. 868–884. [[CrossRef](#)]
45. Wojke, N.; Bewley, A. Deep Cosine Metric Learning for Person Re-identification. In Proceeding of the IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018; p. 7. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).