Journal of Theoretical and Applied Information Technology
31st October 2019. Vol.97. No 20

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

EFFICIENT APPROACH FOR VEHICLE COUNTING BASED ON DEEP CONVOLUTIONAL NEURAL NETWORKS

HOANH NGUYEN
Faculty of Electrical Engineering Technology, Industrial University of Ho Chi Minh City, Ho Chi Minh City, Vietnam
E-mail: nguyenhoanh@iuh.edu.vn

ABSTRACT

Vehicle counting plays an important role in intelligent transport systems. A vehicle counting system should be fast to be implemented in real-time circumstances. Recent methods for vehicle counting usually include two stages, vehicle detection and vehicle tracking. Vehicle tracking stage requires more computational cost, which makes the system less efficient. In this paper, a new approach for vehicle counting based on deep convolutional neural networks (CNN) is proposed. First, an improved single shot multibox detector (SSD) is proposed for fast vehicle detection. The base network and detection network in the original SSD are replaced and modified to reduce the computational cost. To eliminate vehicle tracking step, a region of interest (ROI) is set in each image frame. The number of vehicles is increased when a vehicle is passing the ROI. Furthermore, an improved algorithm for counting exactly the number of vehicles is introduced. Experimental results on public datasets show that the proposed method is the fastest system for counting vehicle among current systems and achieves comparable accuracy compared with state-of-the-art methods.

Keywords: *Vehicle Counting, Convolutional Neural Networks, Intelligent Transportation Systems, Object Detection, Deep Learning*

1. INTRODUCTION

Vehicle counting in traffic video sequences is an important task in intelligent transportation systems. Exactly counting the number of vehicles offers reliable information for traffic management and control. The number of vehicles on-road reflects the traffic status, such as road-traffic intensity, lane occupancy and congestion level. This kind of information can be used for early incident detection, road congestion prevention and automated route planning. In early intelligent transportation systems, vehicle counting methods are usually based on special sensors such as magnetic loop, microwave or ultrasound detectors to determine whether there is a vehicle. However, the total cost of these methods is high, the detection range is small, and the accuracy is limited. With the fast development of digital video processing, a large number of methods for counting vehicle based on vision have been developed. The vision-based vehicle counting system has the advantages over traditional sensor methods in terms of flexibility, detection range, low implementation cost, easy installation and maintenance [18]. Video-based vehicle counting methods can be categorized into three categories: frame difference methods, optical flow methods and background subtraction methods. The frame difference methods compare the difference between moving objects and the background in successive frames. Thus, these methods are fast and simple but only detects parts of the moving objects. Since the optical flow methods need to calculate the optical information of the whole image, these methods are often difficult to satisfy the real-time circumstances. In background subtraction methods, a background model is built, then the background image is subtracted from every frame to obtain the foreground objects. Due to its efficiency, background subtraction is a very popular technique in vehicle counting. However, in most of these methods, the background model is created for the whole frame or for a large area in it. Also, a vehicle tracking step is usually used after vehicle detection. Therefore, these methods almost have a high computational complexity.

In recent years, the rapid development of deep learning has a great impact on the field of object detection and classification as well as vehicle detection and counting [19]. Unlike traditional methods, deep learning methods can overcome the difficulties of changing the appearance of vehicles

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

and reduce the problem of occlusion. Although object detection based on deep learning has several advantages, it is still difficult to balance between the speed and accuracy. Furthermore, achieving real-time on a limited computing platform still remains a difficult problem.

In this paper, a new approach for vehicle counting based on deep convolutional neural networks (CNN) is proposed. The proposed method can achieve real-time processing on a limited computing platform. First, single shot multibox detector (SSD) [1] is modified for fast vehicle detection. The base network and detection network in the original SSD are replaced and modified to reduce the computational cost. Thus, the vehicle detection stage is faster with a comparable accuracy compared to original SSD framework. To eliminate vehicle tracking step, a region of interest (ROI) is set in each image frame. The number of vehicles is increased when a vehicle is passing this ROI. Furthermore, an improved algorithm for counting exactly the number of vehicles in the ROI is developed. Experimental results on public traffic scene videos show that the proposed method is the fastest system for counting vehicle among current systems and achieves comparable accuracy compared with state-of-the-art methods.

This paper is organized as follows: an overview of previous methods is presented in Section 2. Section 3 describes detail the proposed method. Section 4 demonstrates experimental results. Finally, the conclusion is made in Section 5.

2. RELATED WORKS

Vehicle detection is the first step in a vehicle counting system. Over the past few decades, a large number of vehicle detection methods have been developed. Traditional methods include motion-based methods and statistical learning-based methods. Motion-based methods use the motion to detect the vehicles. Adaptive background models such as Gaussian Mixture Model (GMM) [20], Sigma-Delta Model [21] are widely used in vehicle detection by modeling the distribution of the background as it appears more frequently than moving objects. Optical flow [22] is a common technique to aggregate the temporal information for vehicle detection by simulating the pattern of object motion over time. However, this kind of approach is unable to distinguish the fine-grained categories of the moving objects such as car, bus, van or person. In addition, these methods need lots of complex post-processing algorithms like shadow detection and occluded vehicle recognition to refine the detection results. Statistical learning-based methods are based on the handcrafted features to detect the vehicles from the images directly. These methods first describe the regions of the image by some feature descriptors and then classify the image regions into different classes such as vehicle and non-vehicle. Features like HOG [23], SURF [24], and Haar-like [25] are commonly used for vehicle detection followed by classifiers like SVM [24] and Adaboost [25]. These features, however, have limited ability of feature representation, which is difficult to handle complex scenarios.

Recently, deep CNN-based methods have become the leading method for high quality general object detection [19]. Faster region-based convolutional neural network (Faster R-CNN) [12] defined a region proposal network (RPN) for generating region proposals and a network using these proposals to detect objects. RPN shares full-image convolutional features with the detection network, thus enabling nearly cost-free region proposals. This method has achieved state-of-the-art detection performance and become a commonly employed paradigm for general object detection. SSD framework [1] predicted category scores and box offsets for a fixed set of default bounding boxes using small convolutional filters applied to different scales from feature maps of different scales, and explicitly separate predictions by aspect ratio. This framework showed much faster and comparably performance with other methods.

Vehicle detection provides useful results for video-based vehicle counting. The task of vehicle counting is to estimate the number of vehicles presented in a region or image. Recent vehicle counting methods are usually based on detection [28], clustering [26] and regression [27]. Clustering-based and regression-based methods need to explicitly extract the object feature in order to build an accurate appearance model. The detection-based methods need to explicitly segment the objects from the background. Recently, Zhang et al. [29] proposed a vehicle counting method based on foreground time-spatial image. Their approach includes three stages; background initialization, foreground detection and background updating. They use a self-adaptive sample consensus background subtraction method to model background, aiming to resolve the deficiencies in traditional counting method, such as computationally expensive and failure-prone in realistic traffic scenarios. In [30], Barcellos et al. presented a novel video-based detection and vehicle counting algorithm which combines background

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

subtraction, particle filter tracking and clustering algorithm. They exploit the motion coherence of objects to achieve foreground targets. In addition, Gaussian mixture models are used to perform background modelling. Vehicle tracking is implemented by using particle filter with spatial adjacency of the targets. Then, vehicles are counted by detecting the intersections of the tracked targets with user-defined virtual loops. Quesada et al. [16] proposed an automatic vehicle counting method by using principal component pursuit (PCP) to implement background modelling. These methods need two stages for vehicle counting, including vehicle detection and vehicle tracking. Therefore, these methods almost have a high computational complexity. In this paper, the base network and detection network in the original SSD are replaced and modified to reduce the computational cost. Thus, the vehicle detection stage is faster with a comparable accuracy compared to original SSD framework. Furthermore, a ROI is only created for a narrow region in each frame, and an improved algorithm for counting exactly the number of vehicles is introduced. Thus, there is no need for a tracking step. The algorithm complexity is decreased, and the counting process becomes very fast.

3. APPROACH

3.1 Improving SSD Framework for Fast Vehicle Detection

The original SSD framework [1] is based on a feed-forward deep convolutional neural network, which produces a fixed-size collection of bounding boxes and classifies the bounding box, followed by a non-maximum suppression step to produce the final detection. In this paper, the base network in the original SSD is replaced and the number of convolution kernels from extra layers is adjusted. Furthermore, the detection network after the base network is modified and soft Non-Maximum Suppression (NMS) [2] algorithm is then used to solve the issue of duplicate proposals. Figure 1 shows the architecture of the improved SSD framework. The details about each step are described in the following sections.

3.1.1 The Base Network

In this paper, ResNet [3] is used as the base network instead of the VGG-16 in the original SSD framework. ResNet is an efficient network which adopted residual learning to every few stacked layers such that the training of networks can be eased and substantially deeper than others. ResNet-50 and ResNet-101 have high precision, but they are slower than VGG-16. Thus, they are not suitable for real-time vehicle detection and classification. ResNet-34 is not only more accurate than VGG-16 but also faster than VGG-16. Table 1 shows the comparison of computational cost of ResNet-34 and VGG-16. As shown, ResNet-34 significantly reduces the amount of computation compare with VGG-16. To improve precision, the layer after res4f of the ResNet-34 is removed to retain the original FC6 layer and the FC7 layer. Then extra layers after the ResNet-34, predict scores and box offsets from res3d and the additional layers are added as in [1]. After replacing the base network, the number of convolution kernels from FC6 layer to conv9_2 layer is reduced by the half to adapt to the new network. Table 2 shows details number of convolution kernels of each layer from FC6 layer to conv9_2 layer.

3.1.2 The Detection Network

In [4], a properly scaled uniform distribution for initialization, which is called 'Xavier' initialization, was proposed. Its derivation is based on the assumption that the activations are linear, but it is not suitable for the rectified linear unit [6]. Kaiming He et.al [5] proposed 'Msra' initialization which is better than 'Xavier' when the network is deepened. Since ResNet-34 architecture is deeper than VGG-16 architecture, this paper initializes the parameters for all the convolutional layers with the 'Msra' method. Furthermore, batch normalization is added on the entire convolutional layer to improve the training speed and the accuracy of the model. Since bounding boxes usually overlap over the same object, soft Non-Maximum Suppression (NMS) [2] is used to solve the issue of duplicate bounding boxes. Due to heavy vehicle occlusion in traffic scene images, NMS may remove positive bounding boxes unexpectedly as shown in Figure 2. Thus, with soft-NMS, the neighbor bounding boxes of a winning bounding box (which has the biggest probability) are not completely suppressed. Instead they are suppressed according to updated scores of the neighbor bounding boxes, which are computed according to the overlap level of the neighbor bounding boxes and the winning bounding box.

3.2 Efficient Vehicle Counting Algorithm

A vehicle tracking step is usually used after vehicle detection for vehicle counting in recent methods. Thus, these methods have a high computational complexity. In this paper, vehicle tracking step is eliminated to reduce the computational cost. The flow chart of vehicle counting algorithm used in this paper is shown in Figure 3. First, each image frame of the video sequences is extracted for processing. The region of

interest (ROI) is then set on current image frame. The amount of calculation is reduced, and the accuracy is improved by using ROI. Next, improved SSD framework is used to get the location and the bounding box of each vehicle in current image frame. When any part of the bounding box of any detected vehicle passes the ROI, the common area (CMA) between current inside part of the detected vehicle and all previous inside part is check. If CMA is larger than threshold, the current detected vehicle takes the same vehicle label of the matched detected vehicle in the previous frame, and the vehicle number will not increase. If CMA is not lager than threshold, the current detected vehicle takes a new label and the vehicle number is increased by one.

Figure 4 shows an example for counting vehicles using proposed algorithm at consequent frames on M-30 HD video dataset [14]. This figure represents the moving of a car at three consequent frames from top to bottom. At the second frame (second row), the number of vehicles is increased by one due to detection of a new vehicle in ROI, which does not match any detected vehicle at the previous frame (first row). The number of vehicles is not changed in the third frame because the CMA is larger than threshold, which means that no new vehicle is detected.

4. EXPERIMENTAL RESULTS

The proposed approach is implemented on a Window system machine with Core i5 6400 processor, NVIDIA GTX 1050Ti gpu and 8 Gb of RAM. TensorFlow is adopted for implementing deep CNN frameworks, and OPENCV library is used for real time processing.

4.1 Training

The base network used in this paper is based on ResNet-34 [3], which is pre-trained on the ILSVRC CLS-LOC dataset [7]. In the base network, batch normalization is added after each convolution and before activation as in [8]. This study uses stochastic gradient descent (SGD) with a mini-batch size of 32. The learning rate is set at 0.1 and the models are trained for 100,000 iterations. A weight decay and momentum are set at 0.0001 and 0.9 respectively. With improved SSD model, the model is fine-tuned using SGD with the initial learning rate at 0.001, 0.9 momentums, 0.0001 weight decay, and the batch size is 16. Random horizontal flip and random crop are used for data augmentation. Other parameters are the same as setup in [1].

4.2 Evaluation Vehicle Detection

This paper uses KITTI dataset [9] to evaluate and compare the performance of the proposed method with other state-of-the-art methods for vehicle detection. This dataset consists of 7481 images for training with available ground-truth and 7518 images for testing with no available ground-truth. Images in this dataset include various scales of vehicles in different scenes and conditions and were divided into three difficulty-level groups: easy, moderate, and hard. Since the ground truth of the KITTI test set are not publicly available, this paper splits the KITTI training images into a train set and a test set to conduct experiments as in [11], which results in 3682 images for training and 3799 images for testing.

For evaluation metrics, the average precision (AP) and intersection over union (IoU) metrics [11] are used to evaluate the performance of the proposed method in all three difficulty level groups of the KITTI dataset. The IoU is set to 0.7 in this paper, which means only the overlap between the detected bounding box and the ground truth bounding box greater than or equal to 70% is considered as a correct detection.

Figure 5 shows examples of vehicle detection results of the proposed method on the KITTI test dataset. As shown in this figure, the proposed approach can detect vehicle in difficult environment. Table 3 shows the detection results and processing time on the KITTI test dataset of the proposed method and other state-of-the-art deep CNN-based object detectors, including YOLOv2 [13], Faster R-CNN [12] and SSD [1]. As shown in Table 3, the proposed framework is faster than all the other models on a limited computing platform with high accuracy. Since vehicle counting requires a fast framework, the proposed method is more suitable for this purpose.

4.3 Evaluation Vehicle Counting

To evaluate the efficiency of the proposed method and compare with other state-of-the-art methods on vehicle counting, precision is used as an accuracy measurement. Precision is defined as the following equation:

$$\text{Precision (\%)} = 100 - \text{Error (\%)} \quad (1)$$

where

$$Error\ (\%) = \frac{|count\ number - true\ number|}{true\ number} \times 100 \quad (2)$$

To create the training set, this paper collected 20 videos of vehicle in different scenes and

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

conditions from the internet and converted them to 5000 images. These images are used as the training dataset. The ResNet-34 is employed as the base network which has been pre-trained on ImageNet dataset. To evaluate the efficiency of the proposed approach, two videos from the GRAM road-traffic monitoring dataset [14] are used, including M-30 (800 × 480) and M-30 HD (1200 × 720). Both are colour videos of a highway under different conditions (sunny or cloudy). Examples of vehicle counting results for both videos are shown in Figure 6. Table 4 shows the comparison results of proposed method with other state-of-the-art methods on both videos. Other results are taken from [15]. As shown from Table 4, the proposed method achieved the highest precision. More specific, the proposed method achieves precision at 98.70% with M-30 video and 100% with M-30 HD video. In addition, the proposed method did not need a training stage as in [16]. Table 5 shows the average execution time per frame for tested videos. The proposed execution time is very short on low-end hardware system, with an average of 35 ms on M-30 video and 68 ms on M-30 HD video.

5. CONCLUSIONS

An efficient approach for fast vehicle counting based on deep CNN is presented in this paper. The proposed method can achieve real-time processing on a limited computing platform. SSD framework is modified for fast vehicle detection at first stage. In the improved SSD framework, the base network and detection network in the original SSD are replaced and modified to reduce the computational cost. Thus, the vehicle detection stage is faster with a comparable accuracy compared to original SSD framework. To eliminate vehicle tracking step, a region of interest is set in each image frame. The number of vehicles is increased when a vehicle is passing this region. Furthermore, an improved algorithm for counting exactly the number of vehicles in the ROI is presented. Experimental results on KITTI dataset and GRAM dataset demonstrate the good performance of the proposed method in both vehicle detection and counting.

REFRENCES:

[1] Liu W., Anguelov D., Erhan D., et al., "SSD: Single Shot MultiBox Detector", *European Conf. on Computer Vision*, Amsterdam, Holland, 2016, pp. 21–37.

[2] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Improving object detection with one line of code", arXiv preprint arXiv:1704.04503, 2017.

[3] He K., Zhang X., Ren S., et al., "Deep residual learning for image recognition", *IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, USA, 2016, pp. 770–778.

[4] Glorot X., Bengio Y., "Understanding the difficulty of training deep feed forward neural networks", *J. Mach. Learn. Res.*, 2010, pp. 249–256.

[5] He K., Zhang X., Ren S., et al., "Delving deep into rectifiers: surpassing human-level performance on ImageNet classification", *IEEE Int. Conf. on Computer Vision*, Santiago, Chile, 2015, pp. 1026–1034.

[6] Glorot X., Bordes A., Bengio Y., "Deep sparse rectifier neural networks", *Int. Conf. Artif. Intell. Stat.*, Ft. Lauderdale, USA, 2011, pp. 315–323.

[7] Russakovsky O., Deng J., Su H., et al., "Imagenet large scale visual recognition challenge", *Int. J. Comput. Vis.*, 2015, pp. 211–252.

[8] Ioffe S., Szegedy C., "Batch normalization: accelerating deep network training by reducing internal covariate shift", *Int. Conf. on Machine Learning*, Lille, France, 2015, pp. 448–456.

[9] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite", *Proc. CVPR*, Jun. 2012, pp. 3354–3361.

[10] Y. Xiang, W. Choi, Y. Lin, and S. Savarese, "Subcategory-aware convolutional neural networks for object proposals and detection", *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2017, pp. 924–933.

[11] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge", *Int. J. Comput. Vis.*, vol. 88, no. 2, Sep. 2009, pp. 303–338.

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

[13] Redmon J., Farhadi A., "YOLO9000: better, faster, stronger", *IEEE Conf. on Computer Vision and Pattern Recognition*, Hawaii, USA, 2017, pp. 6517–6525.

[14] Guerrero-Gómez-Olmedo R., López-Sastre R., Maldonado-Bascón S., et al., "Vehicle

tracking by simultaneous detection and viewpoint estimation", *Int. Work-Conf. on the Interplay Between Natural and Artificial Computation*, Mallorca, Spain, June 2013, pp. 306–316.

[15] Yang H., and Qu S., "Real-time vehicle detection and counting in complex traffic scenes using background subtraction model with low-rank decomposition", *IET Intell. Transp. Syst.*, 2017, pp. 75–85.

[16] Quesada J., and Rodriguez P., "Automatic vehicle counting method based on principal component pursuit background modeling", *IEEE Int. Conf. on. Image Processing (ICIP)*, Phoenix, AZ, USA, September 2016, pp. 3822–3826.

[17] Bouvie C., Scharcanski J., Barcellos P., et al., "Tracking and counting vehicles in traffic video sequences using particle filtering", *IEEE Int. Instrumentation and Measurement Technology Conf. (I2MTC)*, Minneapolis, MN, USA, March 2013, pp. 812–815.

[18] Liu Y., Tian B., Chen S., et al., "A survey of vision-based vehicle detection and tracking techniques in ITS", *IEEE Int. Conf. Vehicular Electronics and Safety (ICVES)*, Dongguan, China, July 2013, pp. 72–77.

[19] Hanafi, Nanna Suryana, Abd Samad Bin Hasan Basarideep, "Learning for recommender system based on application domain classification perspective: a review", Journal of Theoretical and Applied Information Technology, Vol. 96, No. 14, 2018, pp. 4513-4529.

[20] Z. Chen, T. Ellis, and S. A. Velastin, "Vehicle detection, tracking and classification in urban traffic", *Proc. ITSC*, Sep. 2012, pp. 951–956.

[21] M. Vargas, J. M. Milla, S. L. Toral, and F. Barrero, "An enhanced background estimation algorithm for vehicle detection in urban traffic scenes", *IEEE Trans. Veh. Technol.*, vol. 59, no. 8, Oct. 2010, pp. 3694–3709.

[22] Abdagic A., Tanovic O., Aksamovic A., et al., "Counting traffic using optical flow algorithm on video footage of a complex crossroad", *Conf. Record of IEEE Int. Conf. Electronics in Marine (ELMAR)*, Zadar, Croatia, October 2010, pp. 41–45.

[23] Q. Yuan, A. Thangali, V. Ablavsky, and S. Sclaroff, "Learning a family of detectors via multiplicative kernels", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, Mar. 2011, pp. 514–530.

[24] J.-W. Hsieh, L.-C. Chen, and D.-Y. Chen, "Symmetrical SURF and its applications to vehicle detection and vehicle make and model recognition", *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 1, Feb. 2014, pp. 6–20.

[25] 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, Jun. 2010, pp. 267–276.

[26] Rabaud V., Belongie S., "Counting crowded moving objects", *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, New York, USA, June 2006, pp. 705–711.

[27] Zhang C., Li H., Wang X., et al., "Cross-scene crowd counting via deep convolutional neural networks", *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Boston, USA, June 2015, pp. 833–841.

[28] Dollar P., Wojek C., Schiele B., et al., "Pedestrian detection: an evaluation of the state of the art", *IEEE Trans. Pattern Anal. Mach. Intell.*, 2012, pp. 743–761.

[29] Zhang Y., Zhao C., Zhang Q., "Counting vehicles in urban traffic scenes using foreground time-spatial images", *IET Intell. Transp. Syst.*, 2017, pp. 61–67.

[30] Barcellos P., Bouvié C., Escouto F., et al., "A novel video based system for detecting and counting vehicles at user-defined virtual loops", *Expert Syst. Appl.*, 2015, pp. 1845–1856.

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

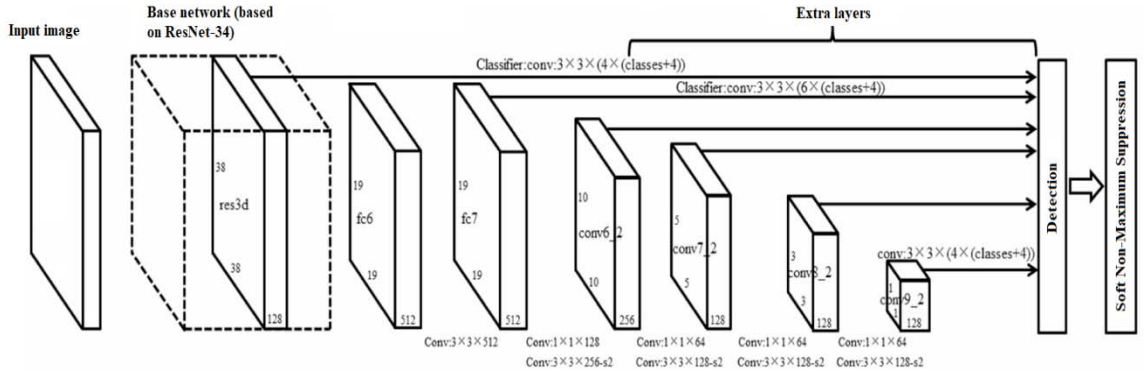

Figure 1: The Architecture of The Improved SSD Framework for Fast Vehicle Detection

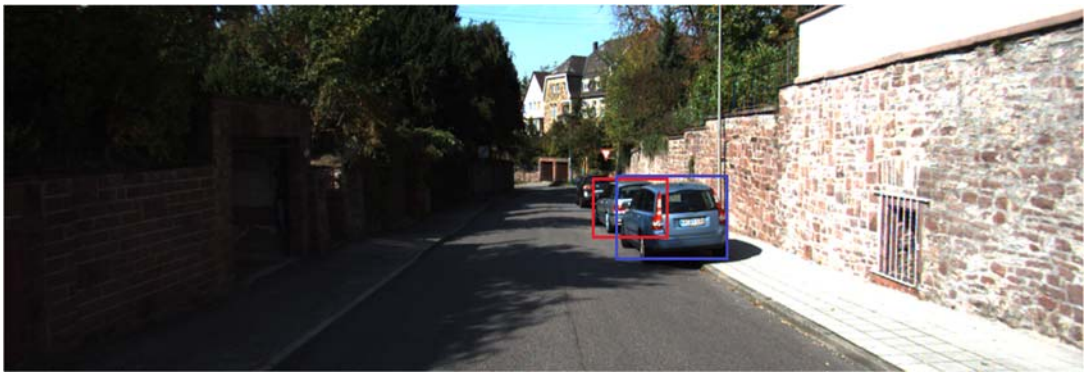

Figure 2: Example of Traditional NMS Algorithm. Front Vehicle May Be Removed with High Probability by The Traditional NMS Method

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

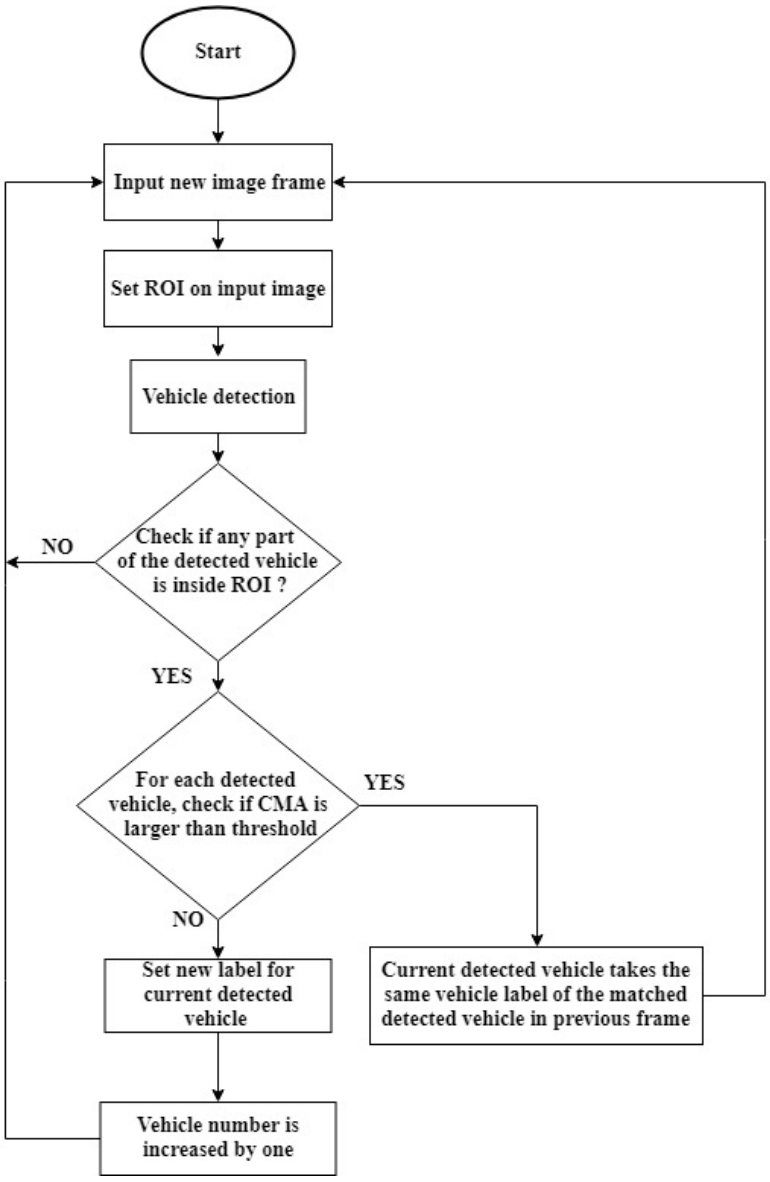

Figure 3: The Flow Chart of Vehicle Counting Algorithm

Journal of Theoretical and Applied Information Technology
31st October 2019. Vol.97. No 20

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

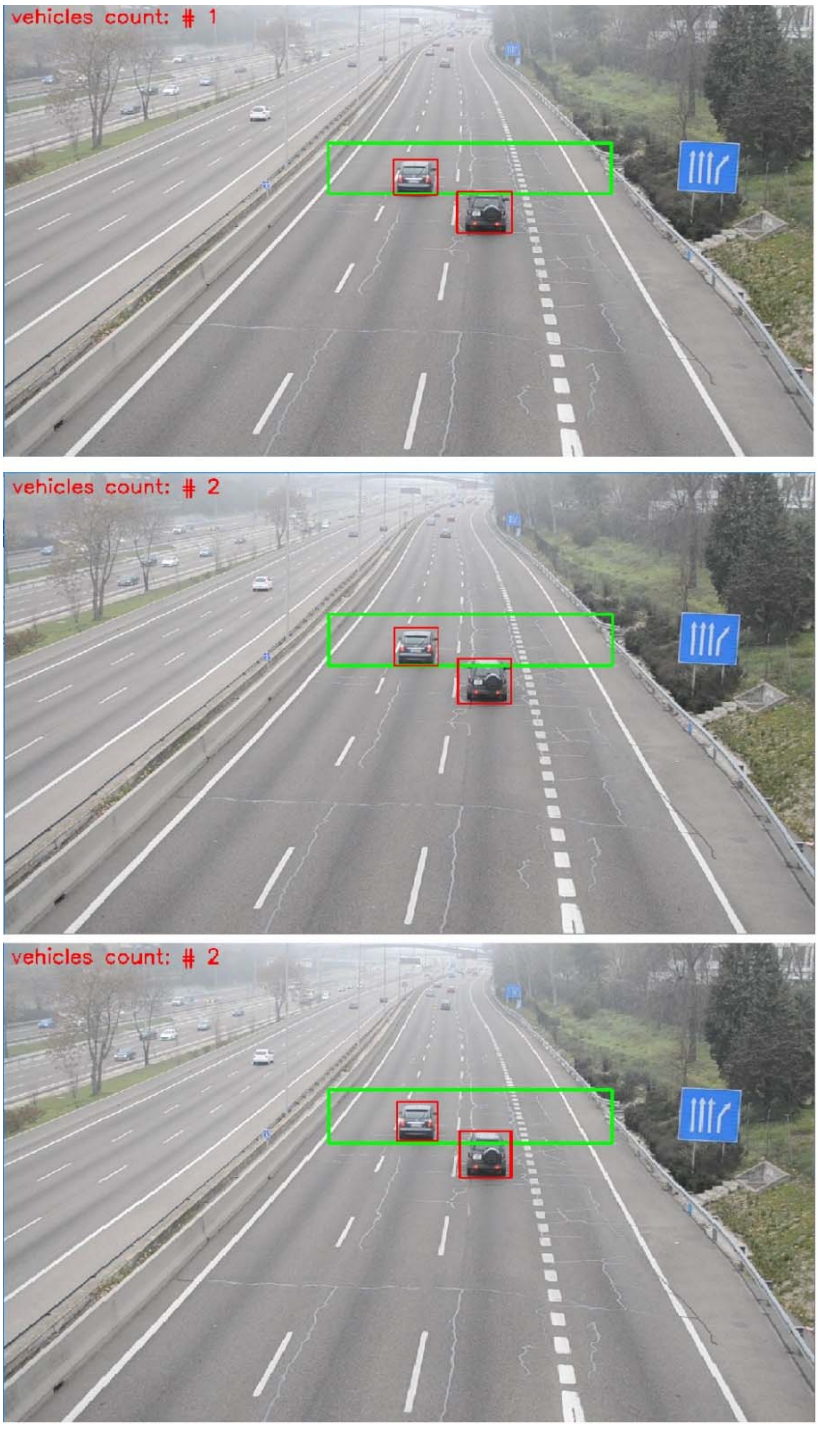

Figure 4: Example for Counting Vehicles Using Proposed Algorithm at Three Consequent Frames on M-30 HD Video

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

Figure 5: Examples of Vehicle Detection Results of The Proposed Method on The KITTI Test Dataset

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

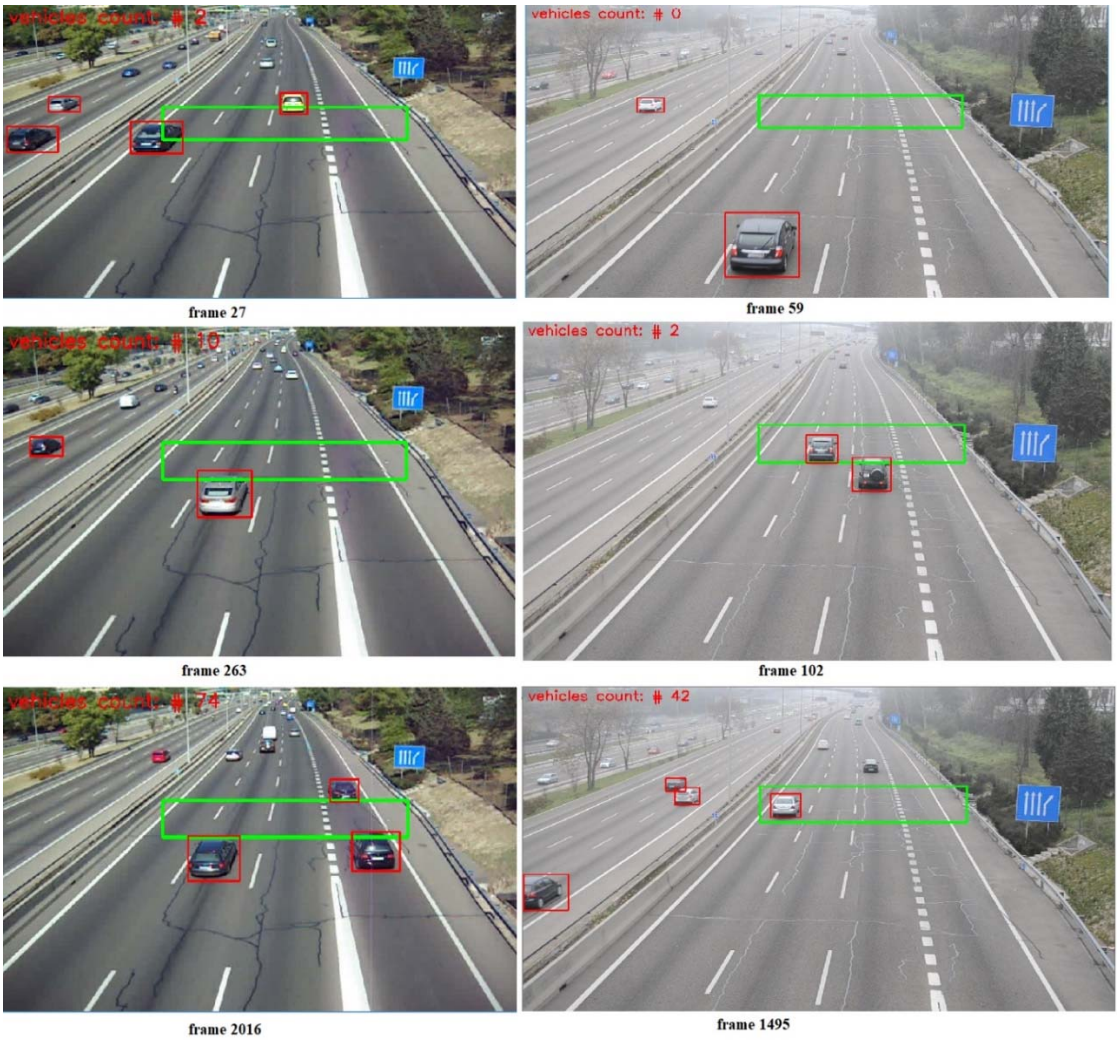

Figure 6: Examples of Vehicle Counting Results on M-30 Video (Left) and M-30 HD Video (Right)

Table 1: The Comparison of Computational Cost of ResNet-34 and VGG-16

Architecture	Computational Cost
VGG-16	15.3 billion FLOPs
ResNet-34	3.6 billion FLOPs

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

Table 2: Details Number of Convolution Kernels from FC6 Layer to conv9_2 Layer

Layer	Input Size	Kernel Size	Output Size
FC6	38 x 38 x 256	3 x 3	19 x 19 x 512
FC7	19 x 19 x 512	1 x 1	19 x 19 x 512
Conv6_1	19 x 19 x 512	1 x 1	19 x 19 x 128
Conv6_2	19 x 19 x 128	3 x 3	10 x 10 x 256
Conv7_1	10 x 10 x 256	1 x 1	10 x 10 x 64
Conv7_2	10 x 10 x 64	3 x 3	5 x 5 x 128
Conv8_1	5 x 5 x 128	1 x 1	5 x 5 x 64
Conv8_2	5 x 5 x 64	3 x 3	3 x 3 x 128
Conv9_1	3 x 3 x 128	1 x 1	3 x 3 x 64
Conv9_2	3 x 3 x 64	3 x 3	1 x 1 x 128

Table 3: Comparison of Different Vehicle Detection Approaches on The KITTI Test Dataset.

Method	Difficulty-level groups			Processing time (s)
	Easy (%)	Moderate (%)	Hard (%)	
Faster R-CNN [12]	87.90	79.11	79.19	2
SSD [1]	83.89	67.17	59.09	0.06
YOLOv2 [13]	86.40	69.01	59.57	0.12
Proposed method	84.50	66.52	60.62	0.04

Table 4: Comparison Results of Proposed Method with Other Methods on Both Videos.

Approach	M-30 video (True Number = 77)		M-30 HD video (True Number = 42)	
	Count Number	Precision (%)	Count Number	Precision (%)
Yang et al. [15]	71	92.20	37	88.10
Quesada et al. [16]	75	97.41	39	92.86
Bouvie et al. [17]	69	89.62	33	78.57
Proposed approach	76	98.70	42	100

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

Table 5: Average Execution Time per Frame for Tested Videos.

Video	Processing time (ms)
M-30	35
M-30 HD	68