A Fast Vehicle Counting and Traffic Volume Estimation Method Based on Convolutional Neural Network

HENGLONG YANG, YOUMEI ZHANG, YU ZHANG, HAILONG MENG, SHUANG LI, AND XIANGLIN DAI

School of Mathematics and Statistics, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, China
School of Electronic and Information Engineering, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, China
Yantai Yinghui Intelligent Technology Company Ltd., Yantai 264003, China

Corresponding author: Youmei Zhang (zym5289@gmail.com)

This work was supported by the National Natural Science Foundation of China under Grant 62001261.

ABSTRACT Vehicle counting and traffic volume estimation on traffic videos has gained extensive attention from multimedia and computer vision communities. Recent vehicle counting and volume estimation methods, including detection based and time-spatial image (TSI) based methods have achieved significant improvements. However, how to balance the accuracy and speed is still a challenge to this task. In this paper, we design a fast and accurate vehicle counting and traffic volume estimation method. Firstly, traffic videos are converted to TSIs and we annotate the vehicle locations in TSIs manually. Then, we design a simple TSI density map estimation network which utilizes attention mechanism to strengthen the features in the traffic locations for vehicle counting. Finally, we use the parameters obtained from the vehicle counting network to further estimate the traffic volume. Experiments on UA-DETRAC dataset demonstrate that the vehicle counting network not only takes a balance between counting accuracy and speed, but also well estimates the traffic volume when the video data is insufficient.

INDEX TERMS Vehicle counting, traffic volume estimation, time-spatial image, density map, attention mechanism.

I. INTRODUCTION

As one of the tasks of intelligent video surveillance, vehicle counting and traffic volume estimation plays an important role in intelligent transportation, including vehicle management and regulation. Due to its broad application prospect, vehicle counting and traffic volume estimation has attracted more and more attention from scholars in the fields of multimedia and computer vision. In recent years, the machine learning based methods have brought great improvement to vehicle counting and traffic volume estimation. However, this task still faces some challenges, such as occlusion, vehicle appearance differences, etc.

Many methods have been proposed to address these challenges in order to improve the accuracy of vehicle counting and traffic volume estimation. These methods can be divided into detection-tracking based methods and TSI based methods. The former one uses detection algorithm to detect the vehicles in each frame, and then utilizes tracking algorithm to match vehicle trajectories and obtains the number of vehicles in the video. This kind of methods usually contain detection and tracking process. The number of vehicles can be calculated according to the tracking trajectories. Popular detection methods such as background subtraction deep neural network (DNN) [1], [2] are usually utilized to detect the vehicles. The tracking methods such as Deep Simple Online Realtime Tracking (Deep Sort) [3] and Kanad-Lucas-Tomasi (KLT) Tacker [4] are often used to track the traffics. Detection-tracking based methods are usually highly complex, thus requiring more time to realize vehicle counting and traffic volume estimation.

TSI based methods first convert the video into time-spatial image (TSI), which represents the vehicle numbers passing in a certain region within a period of time and then obtain the vehicle number in the TSI. Background subtraction is the mostly used method to estimate the vehicle
numbers in TSIs. Background subtraction algorithms include Block-sparse Robust Principal Component Analysis (RPCA) [5], self-adaptive sample consensus back ground model [6], and morphology model [7], etc. These methods do not require cumbersome process, but may encounter problems such as vehicle mutual occlusion, vehicle deformation, and vehicle not fully displayed on TSI, which will lead to the reduction of vehicle counting accuracy.

Inspired by crowd counting, a TSI based vehicle counting method [8] is proposed and has improved the accuracy of vehicle counting evidently. This method first generates TSI and then utilizes a neural network to generate a vehicle density map to count the number of vehicles passing through the region in a certain period of time. This method not only avoids the cumbersome processing process caused by detection and tracking, but also ensures the accuracy of vehicle counting. However, the design of several multi-scale modules complicates the vehicle counting network, and thus leading to a drastic reduction in vehicle counting speed.

In this paper, we design a vehicle counting and traffic volume estimation method based on TSI regression. This method first converts videos to TSIs, and manually labels the training data to obtain accurate TSI density map labels. Then we design a simple neural network to balance the accuracy and speed of vehicle counting. Finally, we use the trained network to carry out vehicle counting and traffic volume estimation. Experimental results demonstrate the effectiveness of the proposed method.

The main contributions of this work are summarized as follows:

1) We propose a fire-new vehicle counting and traffic volume estimation method to avoid the complex detection and tracking operations and balance the accuracy and speed. The counting method converts a video into a TSI and generates the density map which represents the vehicle numbers in the TSI. As a result, vehicle detection and tracking are not required in the proposed method.

2) We design a simple base network for vehicle counting, which achieves accurate vehicle counting and traffic volume estimation with fewer parameters. In the counting process, we design a 6-layer base network and the maximum channels of these layers are 60. The counting accuracy of the base network exceeds most of counting methods.

3) We introduce spatial and channel attention mechanisms into the vehicle counting network, which effectively improves the accuracy of vehicle counting without increasing the parameter size evidently. The spatial attention mechanism aims at guiding the network to focus on vehicle locations and the channel attention mechanisms can strengthen the important feature channels. The two attention modules bring very slight parameter increase to the counting network.

The remainder of the paper is organized as follows. Section II presents some related works about vehicle counting. In section III, the proposed fast vehicle counting and traffic volume estimation method is introduced. Section IV displays and discusses the experimental results. Finally, section V concludes the paper.

II. RELATED WORKS

In this section, we make a brief summary of existing vehicle counting methods, which are methods based on detection and tracking, methods based on TSI and methods by regression.

A. VEHICLE COUNTING BY DETECTION AND TRACKING

Detection-tracking based vehicle counting methods usually contain detection and tracking procedures. The number of vehicles that cross over an area can be calculated according to the tracking trajectory. Detection-tracking based vehicle counting methods can be further divided into background subtraction based methods [9], [10] [11]and DNN based methods [1], [2] [12].

The background subtraction technology is utilized to design the background model and extract the moving vehicles in the video. In this process, some morphological operations are usually applied to segment the vehicles, and then the cross vehicles are counted. When using background subtraction technology, Gaussian mixture model (GMM) [9] is often used to model the background. Abdelwahab et al. [11] applied the background model to only a narrow area in the video frame. Morphological processing is carried out on the extracted target to enhance the target and reduce the influence of vehicle occlusion. El-Khereby et al. [10] modeled the background through an adaptive threshold algorithm, and then applied morphological operations to detect the vehicle. Since it needs to extract enough features for background modeling in BS algorithm, it may not work when dealing with complex background.

The other dominant detection method is neural networks such as Faster Region CNN (Faster-RCNN) [13], You Only Look at Once (YOLO) [14] and Single Shot MultiBox Detector (SSD) [15]. High accuracy can be obtained by using DNN framework. However, these DNN detection methods can only count the vehicles in one frame of image. Tracking methods still need to calculate the vehicles from the video. These methods use tracking algorithms [3], [4] to find the information of the detected vehicles from continuous frames. Liu et al. [16] adopted the latest object detector to detect vehicles from monocular video, and then used Kalman filter based tracking method to track and count vehicles. Kanade-Lucas Tomasi (KLT) tracers are used to extract trajectories and count vehicles [4]. Hoai et al. [3] proposed a comprehensive vehicle counting framework by integrating the latest target detection and tracking technologies, such as YOLO and depth sorting. The tracking is done using a tracing method based on Deep Simple Online Realtime Tracking (Deep Sort). The tracking method in vehicle counting is a complex multi-target training process, and the tracking accuracy is often reduced when there is occlusion or clustering.
B. VEHICLE COUNTING BASED ON TSI

In order to avoid the cumbersome process caused by detection and tracking, some TSI-based methods have been proposed in recent years. This method first fixes a line in each frame of the video to extract the pixels on this line, and then combines the pixels extracted from successive $N$ frames to obtain TSI. Then, background cutting method is designed to detect vehicles in TSI, so as to obtain the traffic flow during this period.

Aiming at the low-rank constraint of spatio-temporal images generated by superposition of pixels on the virtual frame over time, Gao et al. [5] introduced the block sparse robust principal component analysis algorithm to highlight the foreground by using motion tips to achieve high-precision. On city roads, however, downward-facing cameras are rare. Zhang et al. [6] used the adaptive sample consistent background model to form the foreground TSI to count and detect the vehicles on the virtual line. Chen et al. [7] used support vector machine (SVM) and deterministic non-model methods to remove shadows in TSI. They detected the Region of Interest (ROI) through a simple morphology process and used the ROI accumulative curve method and Fuzzy Constraints Satisfaction Propagation (FCSP) to process occlusion problems.

C. VEHICLE COUNTING BY REGRESSION

Inspired by the methods in the field of crowd counting, vehicle counting methods based on regression have also appeared in recent years. The method regresses an input image or every frame in the video to obtain vehicle number. Regression based approaches can be further divided into two categories: approaches based on traditional machine learning and approaches based on deep neural networks (DNN). The former [17], [18] one design feature extraction methods to extract useful features in the image and then carries out regression based on these features to get the vehicle number in the image.

Different from machine learning based methods, DNN based methods train an end-to-end network based on the images and directly output the vehicle numbers. Zhang et al. [19] designed a new FCN-RLSTM network. A Full Convolutional Neural network (FCN) is combined with Long and Short Time Memory network (LSTM) through residual-learning method. To estimate vehicle density and the number of maps, they take advantage of FCN in pixel level prediction and LSTM in complex time dynamics learning. FCN-RLSTM implements a refined feature representation and a novel end-to-end trainable mapping from pixels to the number of vehicles. Different counting tasks are evaluated on three datasets, and the experimental results demonstrate the effectiveness and robustness of the proposed method. Tayara et al. [20] introduced an automatic detection and counting system for aerial images of vehicles. This system uses convolutional neural network to extract the spatial density map of vehicles from aerial images. It evaluates the system on two public datasets, namely the Munich and overflow image study datasets. Experimental results show that this system produces higher accuracy and recall rates than the comparison method. The first contribution of [21] is that they propose a new convolutional neural network solution named Counting CNN (CCNN). It is proved that the object density map can be estimated accurately and effectively, and the network can learn to map and transform the appearance of the image patch into the object density map. The second contribution of [21] is a size perceived counting network-Hydra CNN. The network estimates the object’s nest in a very crowded scene without providing the geometric information about the scene. Hydra CNN further provides a proportional perceived solution, which is intended to learn nonlinear regressors and generate object density maps from the pyramid of a multi-scale image block.

The combination of TSI and regression based methods cannot only avoid the cumbersome process of detection and tracking, but also ensure the accuracy of vehicle counting. Li et al. [8] made the first attempt to regress the vehicle numbers based on TSIs. They designed a TSI density estimation network based on VGG16 [22] and the number of vehicles can be obtained from the density map. In the design of the vehicle counting network, the author firstly extracted the basic features by VGG16, and then stacked several multi-scale modules (MSMs) to further extract features. This network can extract more stable features and get high accuracy. However, the MSMs bring a large amount of parameters and limit the speed of the counting network.

Inspired by [8], we design a fast vehicle counting and traffic volume estimation method. The detail of the proposed method is introduced in section III.

III. OUR METHOD

The fast vehicle counting and traffic volume estimation method proposed in this paper includes three modules, which are TSI and its corresponding density map generation (TSI-DM) module, vehicle counting (VC) module and traffic volume estimation (TVE) module. Among them, the TSI-DM module mainly pre-processes the dataset. In the TSI-DM module, we manually mark the position of traffic flow in TSI and then generate the density map label according to Eq. 1. In the VC module, we design a simple and effective density map estimation network and introduce the attention mechanism to guide the network to focus on the traffic locations. The TVE module estimates the traffic volume by utilizing the trained vehicle counting module. Figure 1 shows the framework of the approach in this article.

A. TSI-DM MODULE

We refer to TM-Net [8] to generate TSIs from traffic videos. Firstly, we select a virtual line perpendicular to the road that all vehicles in the screen will pass through. Then, all the pixels in this line are captured from each frame in the video sequence. Finally, TSI is obtained by stacking these pixels in chronological order, as shown in figure 2(a).
FIGURE 1. Framework of the vehicle counting and traffic volume estimation method. The proposed method first generates TSI from training video and test video respectively. The TSI-DM module is used to generate TSI and its corresponding density map. The outputs of TSI-DM module are feed to the vehicle counting module. Then, the vehicle counting module trains a fast and effective vehicle flow density estimation network based on the data of the TSI-DM module, and adds its output density map to obtain the vehicle flow. The TVE module estimates the traffic volume according to the number of video frames adopted by the vehicle counting module and its outputs.

FIGURE 2. Pipeline of the TSI-DM module. Firstly, the pixels on a straight line are selected and combined in each frame of the video to generate TSI. Then we manually mark the vehicle positions by remarking the top left and bottom right of the traffic block. Finally, the TSI density map is generated according to the labeled traffic locations.

In order to conduct supervised training, the coordinates of vehicles in generated TSI should also be marked. Since the video is converted to TSI, we no longer need to mark each vehicle in a single frame. In this article, we manually mark the traffic flows in the TSI. Specifically, we label the top left coordinate \((x_1, y_1)\) and bottom right coordinate \((x_2, y_2)\) of the traffic block which is generated by each volume of vehicles passing by in the TSI. Based on this, we can calculate the center point coordinate of the traffic block as well as its length and width. Then, we generate the TSI density map label according to formula 1.

\[
D^T = \sum_{j \in J} \delta(o - o_j) * G_{\sigma_j}(o_j), \text{ with } \sigma_j = (w_j, h_j)
\]  

In the formula, \(o_j\) represents the center point of the \(j\)th traffic block, and \(w_j, h_j\) respectively represent the width and height of \(j\)th traffic block. \(\delta(\cdot)\) denotes impulse function and \(G(\cdot)\) is Gaussian function. The labeled TSI and the corresponding density map generated according to this formula is shown in figure 2(b).

B. VEHICLE COUNTING MODULE

Deep neural network method has been widely used for traffic [19], [20] [21] and crowd [23], [24] [25] density estimation of static images, but rarely for TSI based vehicle density estimation. In this paper, a simple and effective convolutional neural network is designed by comprehensively considering...
the accuracy and speed of vehicle counting. The network is shown in figure 1 in the vehicle counting module.

The vehicle counting network consists of 3 units, which are basic feature extraction unit, TSI density map generation unit and attention mechanism unit. Inspired by MCNN [24], the design of the basic feature extraction unit divides the convolution kernel of different sizes into three branches for multi-scale feature extraction, so as to deal with the problem of the scale difference of traffic blocks caused by the different size and speed of vehicles. The sizes of the three convolution kernels are \(7 \times 7\), \(5 \times 5\) and \(3 \times 3\) respectively. The obtained features are stacked to get \(F_{cat}\). On the basis of \(F_{cat}\), channel attention unit is introduced to guide network adaptively enhance the effective feature channels and generate feature \(F_{ch-am}\). Further, we use three-layer convolution for further feature extraction, and use spatial attention mechanism to guide the network to the features where traffic blocks are located, and get \(F_{s-am}\). On the basis of \(F_{s-am}\), we further use 2-layer convolution to generate density map. Since containing 2 max pooling layers in this architecture, it finally outputs a density map with \(1/16\) size of the original image. Details of network parameters, including the size of the convolution kernel, the number of characteristic channels and the number of convolution and pooling layers, can be found in figure 1.

CAM and SAM in figure 3 represent channel attention mechanism and spatial attention mechanism respectively. Attention mechanisms have been proven to be effective in some tasks in natural language processing [26], [27] [28], [29] and computer vision [30], [31] [32], [33]. The channel attention mechanism aims to strengthen the effective feature channels, and the spatial attention mechanism can effectively strengthen the effective features in the feature map. We will introduce these two attention mechanisms in details in the following part.

Among different convolution branches, the one with large kernel pays more attention to large receptive fields and the one with small kernel is more conducive to small-size traffic flow. In order to strengthen the feature channel more efficiently for the different inputs adaptively, we introduce the Squeeze-and-Excitation Networks (channel-wise attention mechanism) [34] after multi-scale feature fusion. The specific setting of the channel attention mechanism is shown in figure 3. Firstly, global average pooling is adopted for the merged features, then the weight of each channel is obtained by using two full connection layers, and the weight is added to the corresponding channel. Finally, \(F_{ch-am}\) is obtained by adding the original features. In addition, there are often green belts on the road where vehicles are driving, resulting in complex backgrounds in the generated TSI. In order to reduce the influence of these backgrounds on vehicle counting, we introduce a spatial attention mechanism. Figure 3 shows the setup of the spatial attention mechanism. Because the number of feature channels is small, we directly use a convolution of \(3 \times 3\) to obtain the probability map. Then, the probability map and the original features are used for element-wise product. After training, this probability map can strengthen the eigenvalues of effective spatial locations. It is noteworthy that neither channel-attention nor spatial attention mechanism significantly increases the number of network parameters.

The network training process is supervised training, and we use Euclidean distance to measure the difference between output and label in the training process. As formula 2 shows:

\[
L_{ED} = \frac{1}{N} \sum_{i=1}^{N} (F(X_i, \Theta) - D_i)^2
\]

where \(N\) is the number of the training samples, \(D\) is the ground-truth density map and \(F\) is the function that mapping the input \(X_i\) to the estimated density map with parameters \(\Theta\).

TSI is generated by continuous \(m\) frames in the period of \(t\), so the number of vehicles passing in the period of \(t\) can be obtained by counting the number of traffic blocks in TSI. The estimated number of traffic flows is finally the sum of pixel values in the estimated density map. As shown in formula 3:

\[
C^e = \sum_{i \in TSI} D^e(i)
\]

where \(C^e\) indicates the estimated number of the TSI and \(D^e(i)\) is the \(i\)th pixel in the estimated density map \(D^e\).

C. TRAFFIC VOLUME ESTIMATION MODULE

Besides the vehicle counting statistics, traffic volume estimation is also an important indicator to evaluate traffic conditions for traffic scheduling. Traffic volume is the number of vehicles passing a certain road in an hour. In this paper, traffic volume is estimated based on the number of traffic flows in time period \(T\) output by the vehicle counting module. Given the data acquisition frequency \(f\) fps, the corresponding time of \(m\) frames is \(mf\) seconds. Vehicle counting module estimates the number of vehicles that cross over a line in \(m\) frames, which is denoted as \(c_m\). Based on this, traffic volume is further estimated as:

\[
V = \frac{f \times c_m}{m} \times 3600
\]

It can be seen from formula 4 that the video sampling frequency and the number of continuous images \(m\) generated by TSI both affect the results of traffic volume estimation. In the experiment part, we will discuss the influence of \(m\) on traffic volume estimation.

TSIs are generated by stacking pixels from lines that perpendicular to traffics’ forward direction. If the vehicles change lanes when crossing the vertical line, the vehicle flow in the TSI will be deformed. For online application, we can set the video camera above the road where there are solid lines on the ground. The vehicles always move straight forward without changing lanes on this part of the road. Then we can convert existing video over a period of time to a TSI by extracting the pixels on the vertical line in every frame. Finally, the vehicle counting module will generate the density map to get the vehicle number in the TSI.
IV. EXPERIMENTS
The proposed method is conducted on UA-DETRAC dataset [35]. This section first introduces the public dataset, experimental setting and evaluation metrics. Then the experimental results are presented to verify the effectiveness of the proposed method.

A. UA-DETRAC DATASET
The UA-DETRAC dataset consists of 10 hours of videos captured with a Canon EOS 550D camera at 24 different locations in Beijing and Tianjin in 2015. These videos are recorded at 25 frames per seconds (fps), with a resolution of 960 × 540 pixels. When using this database to carry out vehicle counting and traffic volume estimation, we refer to [8] to select some data in this database. There are some videos captured in tilted view. We select same data with the compared algorithms. In this database, 11 videos named 20011, 20033, 20035, 20051, 20061, 20062, 20064, 40131, 40141, 63521 and 63525 are selected as training data. Six videos named 20012, 20032, 20052, 20065, 40191 and 40192 are used as testing data.

In the process of training data generation, we first generate the corresponding TSI for each video sequence and marked the position of each vehicle in the TSI. Then, the density maps are generated according to the annotations. For training set, we adopt image cropping and flipping to expand the training sets. Patches with 200 × 200 pixels are cropped in random locations of the image to train the vehicle counting network. Finally, 6300 patches are obtained, of which 450 are randomly selected as validation set, and the rest compose training set. For testing data, we directly input the whole image to the network. It is worth noting that since there are two pooling layers in the counting network, the width and height of the density map are 1/4 of the original one, we zoom the width and height of the test image to an integral multiple of 4.

B. EXPERIMENTAL SETTING
In the training procedure, the learning rate and momentum are set as $10^{-5}$ and 0.9 for Adam optimization. The batch size is set as 1 for training. All of the experiments are conducted on GeForce GTX TITAN-X. In the experiment, TSI generation, data annotation and density map generation are realized by matlab programming. The traffic estimation network is implemented on the GeForce GTX TITAN-X using the Pytorch framework.

C. EVALUATION METRICS
In order to quantify the effectiveness of vehicle counting module, we adopt mean absolute error (MAE) and accuracy...
as the measurement standards. MAE is defined as formula 5:

$$\text{MAE} = \frac{1}{N} \sum_{n=1}^{N} | C_n^e - C_n^g |$$  \hspace{1cm} (5)

where $N$ is the number of TSI to be measured, $C_n^e$ and $C_n^g$ are the estimated number and marked number of traffic flows respectively. The calculation method of accuracy is shown in formula 6:

$$\text{Accuracy} = 1 - \frac{| C_n^e - C_n^g |}{C_n^g}$$  \hspace{1cm} (6)

In order to further measure the accuracy of traffic volume estimation by the algorithm, we introduce mean absolute percentage error (MAPE) to quantify the effectiveness of the traffic volume module. MAPE is formulated as formula 7:

$$\text{MAPE} = \frac{| GT - ES |}{GT}$$  \hspace{1cm} (7)

where $GT$ is the actual traffic volume and $ES$ denotes the estimated traffic volume.

## D. VEHICLE COUNTING RESULTS

### 1)ABLATION STUDY

This section demonstrates the effectiveness of the proposed network and the attention units. We mainly compare the results in literature [8], because this method is also based on TSI and uses DNN regression to count the traffic flow. Li et al. [8] takes VGG16 as the basic network, so we also compare VGG16 and ResNet50, which are two typical basic networks. We make a comparison based on the estimated results of video 40192. The ground truth vehicle number is 258 and the experimental results are shown in table 1.

From the accuracy point of view, the proposed basic network performs better than VGG16, and the MAE of the proposed basic network is 2.63 lower than that of VGG16. With the introduction of channel attention mechanism, MAE is reduced by 0.59. With further introduction of spatial attention mechanism, the vehicle counting module proposed in this paper is formed. The MAE is significantly reduced by 4.31, but it is still 0.79 higher than TM-Net. The parameter size of our BaseNet is $0.56 \times 10^5$, which is $1/78$ of VGG16. Li et al. [8] takes VGG16 as the basic network, but does not give the specific parameters of the subsequent structure, so the number of network parameters exceeds $44 \times 10^5$.

### 2)COMPARISON WITH OTHER METHODS

In order to verify the effectiveness of the proposed vehicle counting method, we compare it with several typical algorithms, including tracking-detection based methods, TSI based background subtraction method and DNN based method. We conduct different methods on video 40192 and the comparison results are shown in table 2.

In table 2, the first three methods carry out vehicle flow statistics based on detection and tracking. These methods firstly detect vehicles from every frame of the video and then use tracking algorithm to get the trajectories. We then count the trajectories that across the virtual line as the vehicle numbers. However, the results of these methods are more dependent on the detection algorithm, and the processing of images slows down the speed of vehicle counting. In paper [11], Abdelwahab et al. designed a method of background subtraction on TSI to realize vehicle counting. The authors modeled the background of a rectangular area which was perpendicular to the traffic line and then extracted the foreground objects of these areas. A whole vehicle is counted by measuring the common area of foreground objects between nth frame and the previous one. We applied this algorithm on UA-DETRAC dataset and founded that it is slightly better than the detection-tracking based methods both in accuracy and speed. Recently, the DNN regression based vehicle counting algorithm [20], [21] for TSI is better than that of FastBS [11]. We converted videos to TSIs and applied these methods to regress the TSI density map. The CCNN [21] has an advantage in speed but is low in accuracy. The FCRN [20] used encoder-decoder to estimate the density map and the counting accuracy was significantly improved over CCNN [21]. Both the proposed method and TM-NET made attempt to conduct vehicle counting based on density map estimation of TSI. Compared with the TM-NET, the proposed method have two advantages in the vehicle counting process. Firstly, the TM-NET labels the vehicles in training TSIs based on the detection and tracking algorithm. However,

## TABLE 1. Vehicle counting results with different network structures.

| Method              | MAE   | ES       | Accuracy | Parameter Size ($\times 10^5$) | Testing times(s) |
|---------------------|-------|----------|----------|--------------------------------|------------------|
| VGG16               | 12.8  | 245.20   | 98.04%   | 44                            | 0.526            |
| TM-Net [8]          | 4.48  | 253.52   | 98.26%   | 44+                           | 0.783            |
| BaseNet             | 10.17 | 268.17   | 96.11%   | 0.56                          | 0.282            |
| BaseNet+CAM         | 9.58  | 267.58   | 96.29%   | 0.57                          | 0.289            |
| BaseNet+CAM+SAM     | 5.27  | 263.27   | 97.96%   | 0.57                          | 0.302            |

## TABLE 2. Comparisons with representative vehicle counting methods.

| Method                        | C     | Accuracy | MAE   | Testing times(s) |
|-------------------------------|-------|----------|-------|------------------|
| Yolov3 + CMOT [36]           | 243   | 94.19%   | -     | -                |
| Yolov3 + HT [37]             | 241   | 93.41%   | -     | -                |
| ESCNN + KLT [11]             | 238   | 92.24%   | -     | -                |
| FastBS [11]                  | 240   | 93.02%   | -     | -                |
| CCNN [21]                    | 241.54| 93.62%   | 16.46 | 0.287            |
| FCRN [20]                    | 246.72| 95.62%   | 11.28 | 0.594            |
| TM-Net [8]                   | 253.52| 98.26%   | 4.48  | 0.783            |
| The proposed method          | 263.27| 97.96%   | 5.27  | 0.302            |
TABLE 3. Vehicle counting results on UA-DETRAC dataset.

| Video   | 20012 | 20032 | 20052 | 20065 | 40191 | 40192 |
|---------|-------|-------|-------|-------|-------|-------|
| Cmaj   | 25    | 5     | 30    | 36    | 288   | 258   |
| Cm    | 23.19 | 4.7   | 33.43 | 35.91 | 294.35| 263.27|
| MAE    | 1.81  | 0.3   | 3.43  | 0.09  | 6.35  | 5.27  |
| Accuracy | 92.75% | 94% | 88.57% | 99.75% | 97.80% | 97.96% |
| GT    | 2404  | 1030  | 3890  | 2700  | 10389 | 10578 |
| ES    | 2307  | 1030  | 4279  | 2700  | 10605 | 10783 |
| MAPE  | 4.0%  | 0%    | 10%   | 0%    | 2.07% | 1.07% |

In order to show the effectiveness of the proposed method more intuitively, we present the vehicle counting and traffic volume estimation results of all test videos in table 3. The evaluation index is the evolution metric introduced in section IV. The evaluation indexes MAE and accuracy of vehicle counting are obtained according to formula 5 and formula 6. MAPE is calculated based on formula 7.

Figure 4 shows the TSI generated from the test videos and the results of density estimation based on the TSI. At the same time, ground-truth density map is also shown to visually compare the estimated density map. Due to the large difference in TSI size, we scaled it to a certain proportion to show the results more clearly. As can be seen from figure 4, the proposed vehicle counting method can effectively focus on the vehicle flows while filtering the backgrounds.

E. VOLUME ESTIMATION RESULTS

In practical application, the traffic volume will be estimated at any time. Firstly, TSI is generated from consecutive $m$ frames, and then its corresponding density map is estimated by the trained vehicle counting network. Then, the values of pixels in the density map are summed up to obtain the traffic flow within the time represented by $m$ frames. Finally, the traffic volume which represents the traffic number that pass in an hour is predicted based on the output of the vehicle counting network and $m$ according to formula 4.

The method in [8] estimates the traffic volume by dividing the whole video into different parts with stride $m$. Then they estimate the vehicle numbers of every part and add them to get the traffic volume. In other words, they always use the whole video to predict the traffic volume even with different $m$. Differently, we use the former $m$ frames to estimate the traffic volume and this method is more reasonable to measure the prediction ability of the volume estimation module. We take $m$ from 200 to 1000 to predict traffic volume, and conduct experiments on video 40191 and 40192 to discuss the influence of parameter $m$. The experimental results are shown in table 4.

TABLE 4. Traffic volume estimation results on UA-DETRAC dataset.

| Num. of frames | Video 40191 | Video 40192 |
|---------------|-------------|-------------|
| $m = 200$     | 9450        | 9000        |
| $m = 300$     | 10200       | 9300        |
| $m = 400$     | 10350       | 9225        |
| $m = 600$     | 10350       | 10050       |
| $m = 700$     | 10542       | 10414       |
| $m = 800$     | 10800       | 10350       |
| $m = 900$     | 10600       | 10200       |
| $m = 1000$    | 10530       | 10400       |
| ES            | MAPE        | ES          | MAPE        |
| 10.66%        | 9.00        | 13.37%      |
| 3.57%         | 10.48%      | 4.71%       |
| 2.16%         | 11.2%       | 3.26%       |
| 0.34%         | 0.24%       | 0.38%       |
| 2.09%         | 0.58%       | 0.05%       |

As can be seen from table 4, with the increase of $m$, the accuracy of traffic volume estimation is steadily improved. The data acquisition frequency in UA-DETRAC dataset is 25fps. When $m = 200$, the MAPE of the two test videos is greater than 10%, indicating that too few continuous images are not enough to achieve accurate traffic volume estimation. When $m = 600$, the MAPE has been reduced to between 2% and 4%. This indicates that the proposed traffic volume estimation method still performs well when the data is insufficient.

V. CONCLUSION

In this paper, we proposed a fast and effective vehicle counting and traffic volume estimation method. The proposed method firstly generates TSIs from traffic videos and labels the traffic locations in the TSI. Then the traffic numbers are estimated based on TSIs, which avoids the complicated operation caused by tracking detection method. In addition, a simple and effective neural network for vehicle flow density map estimation is designed to achieve accurate vehicle counting without using too many parameters. The proposed vehicle density estimation network effectively balances the speed and accuracy of vehicle counting. Based on this result, we use continuous videos in a period of time to estimate the traffic volume, which can provide more comprehensive traffic condition information for intelligent traffic scheduling. However, the proposed method can not well handle the situation that traffics change lanes when crossing the lines where pixels are extracted to form the TSI. In the future, we will explore how to handle the serious deformation of vehicle flows in TSIs as well as further improve the accuracy to make the counting algorithm more applicable.

REFERENCES

[1] R. Ke, Z. Li, J. Tang, Z. Pan, and Y. Wang, “Real-time traffic flow parameter estimation from uav video based on ensemble classifier and optical flow,” IEEE Trans. Intell. Transp. Syst., vol. 20, no. 1, pp. 54–64, Jan. 2019.
[2] Z. Dai, H. Song, X. Wang, Y. Fang, X. Yun, Z. Zhang, and H. Li, “Video-based vehicle counting framework,” IEEE Access, vol. 7, pp. 64460–64470, 2019.

[3] K.-H.-N. Bui, H. Yi, and J. Cho, “A vehicle counts by class framework using distinguished regions tracking at multiple intersections,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jun. 2020, pp. 578–579.

[4] R. Ke, Z. Li, S. Kim, J. Ash, Z. Cui, and Y. Wang, “Real-time bidirectional traffic flow parameter estimation from aerial videos,” IEEE Trans. Intell. Transp. Syst., vol. 18, no. 4, pp. 890–901, Apr. 2017.

[5] Z. Gao, R. Zhai, P. Wang, X. Yan, H. Qin, Y. Tang, and B. Ramesh, “Synergizing appearance and motion with low rank representation for vehicle counting and traffic flow analysis,” IEEE Trans. Intell. Transp. Syst., vol. 19, no. 6, pp. 2675–2685, Aug. 2018.

[6] Y. Zhang, C. Zhao, and Q. Zhang, “Counting vehicles in urban traffic scenes using foreground time-spatial images,” IET Intell. Transp. Syst., vol. 11, no. 2, pp. 61–67, Mar. 2017.

[7] C.-Y. Chen, Y.-M. Liang, and S.-W. Chen, “Vehicle classification and counting system,” in Proc. Int. Conf. Audio, Lang. Image Process., Jul. 2014, pp. 485–490.

[8] L. Li, C. Pi, L. Liu, and F. Chang, “Time-spacial multi-scale net for vehicle counting and traffic volume estimation,” IEEE Trans. Cognit. Develop. Syst., early access, Apr. 16, 2021, doi: 10.1109/TCDS.2021.3073694.

[9] B. G. Rajagopal, N. Vishakraj, N. U. Kumar, and P. Jothivenkatesh, “Vision-based system for counting of moving vehicles in different weather conditions,” in Proc. Int. Conf. Electron., Comput. Aerosp. Technol. (ICECA), Apr. 2017, pp. 86–91.

[10] J. Hou, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 7132–7141.

[11] M. A. Abdelwahab, “Fast approach for efficient vehicle counting,” Electron. Lett., vol. 55, no. 1, pp. 20–22, Jan. 2019.

[12] C. Liu and F. Chang, “Hybrid cascade structure for license plate detection in large visual surveillance scenes,” IEEE Trans. Intell. Transp. Syst., vol. 20, no. 6, pp. 2122–2135, Jun. 2019.

[13] X. Qin, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[14] J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” 2018, arXiv:1804.02767.

[15] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “SSD: Single shot multibox detector,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2016, pp. 21–37.

[16] C. Liu, D. Q. Huynh, Y. Sun, M. Reynolds, and S. Atkinson, “A vision-based pipeline for vehicle counting, speed estimation, and classification,” IEEE Trans. Intell. Transp. Syst., early access, Jul. 1, 2020, doi: 10.1109/TITS.2020.3004066.

[17] Z. Wang, X. Liu, J. Feng, J. Yang, and H. Xi, “Compressed-domain highway vehicle counting by spatial and temporal regression,” IEEE Trans. Circuits Syst. Video Technol., vol. 29, no. 1, pp. 263–274, Jan. 2019.

[18] C. Juan, X. Liu, Z. Wang, J. Feng, and H. Xi, “Highway vehicle counting in compressed domain,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 3016–3024.

[19] S. Zhang, G. Wu, J. P. Costeira, and J. M. Moura, “FCN+LSTM: Deep spatio-temporal neural networks for vehicle counting in city cameras,” in Proc. IEEE Int. Conf. Comput. Vis., Oct. 2017, pp. 3667–3676.

[20] H. Tayara, K. G. Soo, and K. T. Chong, “Vehicle detection and counting in high-resolution aerial images using convolutional regression neural network,” IEEE Access, vol. 6, pp. 2220–2230, 2018.

[21] D. Onoro-Rubio and R. J. López-Sastre, “Towards perspective-free object counting with deep learning,” in Proc. Eur. Conf. Comput. Vis. Springer, 2016, pp. 615–629.

[22] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, arXiv:1409.1556.

[23] Y. Zhang, C. Zhou, F. Chang, and A. C. Kot, “Multi-resolution attention convolutional neural network for crowd counting,” Neurocomputing, vol. 329, pp. 144–152, Feb. 2019.

[24] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, “Single-image crowd counting via multi-column convolutional neural network,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 589–597.

[25] X. Shi, X. Li, C. Wu, S. Kong, J. Yang, and L. He, “A real-time deep network for crowd counting,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), May 2020, pp. 2328–2332.

[26] G. Tang, M. Müller, A. Rios, and R. Sennrich, “Why self-attention? A targeted evaluation of neural machine translation architectures,” 2018, arXiv:1808.08946.

[27] S. Xu, H. Li, P. Yuan, Y. Wu, X. He, and B. Zhou, “Self-attention guided copy mechanism for abstractive summarization,” in Proc. 58th Ann. Meeting Assoc. Comput. Linguistics, 2020, pp. 1355–1362.

[28] D. Kiela, C. Wang, and K. Cho, “Dynamic meta-embeddings for improved sentence representations,” 2018, arXiv:1804.07983.

[29] K. Wang, W. Shen, Y. Yang, X. Quan, and R. Wang, “Relational graph attention network for aspect-based sentiment analysis,” 2020, arXiv:2004.12362.

[30] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16×16 words: Transformers for image recognition at scale,” 2020, arXiv:2010.11929.

[31] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran, “Image transformer,” in Proc. Int. Conf. Mach. Learn., 2018, pp. 4055–4064.

[32] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, “Deformable DETER: Deformable transformers for end-to-end object detection,” 2020, arXiv:2010.04159.

[33] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, “Self-attention generative adversarial networks,” in Proc. Int. Conf. Mach. Learn., 2019, pp. 7354–7363.

[34] J. Xu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 7132–7141.

[35] L. Wen, D. Du, Z. Cai, Z. Lei, M.-C. Chang, H. Qi, J. Lim, M.-H. Yang, and S. Lyu, “UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking,” 2015, arXiv:1511.04136.

[36] S.-H. Bae and K.-J. Yoon, “Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 1218–1225.

[37] L. Wen, W. Li, J. Yan, Z. Lei, D. Yi, and S. Z. Li, “Multiple target tracking based on undirected hierarchical relation hypergraph,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 1282–1289.

---

HENG LONG YANG is currently pursuing the B.E. degree in intelligent science and technology with the School of Mathematics and Statistics, Qilu University of Technology (Shandong Academy of Sciences), Jinan, China. His research interests include solving complex real-world problems arising from intelligent transportation domain by use of deep learning methods.

YOUMEI ZHANG received the Ph.D. degree in pattern recognition and intelligent system from Shandong University, China, in 2019. She is currently working with the School of Mathematics and Statistics, Qilu University of Technology (Shandong Academy of Sciences), as a Lecturer, China. Her research interests include crowd counting, facial expression recognition, and abnormal event detection.
YU ZHANG received the B.E. degree in control science and engineering from Shandong University, Jinan, China, in 2014, and the Ph.D. degree from the Department of Electronic Engineering, The Chinese University of Hong Kong, Hong Kong, in 2019. She is currently a Lecturer with the Qilu University of Technology (Shandong Academy of Sciences), Jinan. Her research interests include hyperspectral image segmentation and classification.

HAILONG MENG received the M.S. degree in control theory and control engineering and the Ph.D. degree in control science and engineering from Hangzhou Dianzi University, in 2013 and 2019, respectively. His current research interests include optimal design of digital filters and optimization theory and methods.

SHUANG LI received the Ph.D. degree in pattern recognition and intelligent systems from Shandong University, Jinan, China, in 2021. Her current research interests include automatic target detection and recognition, machine learning, deep learning, and traffic flow parameter estimation.

XIANGLIN DAI received the B.S. degree in computer science and technology from Jilin University and the M.S. degree in computer engineering from the Ocean University of China. His current research interests include pattern recognition and image processing.

* * *

VOLUME 9, 2021 150531