Received February 8, 2020, accepted February 16, 2020, date of publication February 27, 2020, date of current version March 12, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2976890

Mixed Road User Trajectory Extraction From Moving Aerial Videos Based on Convolution Neural Network Detection

RUYI FENG[1,2,3], **CHANGYAN FAN**[1,2,3], **ZHIBIN LI**[1,2,3], **AND XINQIANG CHEN**[4]

[1]Jiangsu Key Laboratory of Urban ITS, Southeast University, Nanjing 210096, China
[2]Jiangsu Province Collaborative Innovation Center of Modern Urban Traffic Technologies, Nanjing 210096, China
[3]School of Transportation, Southeast University, Nanjing 210096, China
[4]Institute of Logistics Science and Engineering, Shanghai Maritime University, Shanghai 201306, China

Corresponding author: Changyan Fan (fchyan@seu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 71871057, in part by the Fundamental Research Funds for the Central Universities under Grant 2242019R40060, and in part by the Southeast University Zhongying Young Scholars Project.

ABSTRACT Vehicle trajectory data under mixed traffic conditions provides critical information for urban traffic flow modeling and analysis. Recently, the application of unmanned aerial vehicles (UAV) creates a potential of reducing traffic video collection cost and enhances flexibility at the spatial-temporal coverage, supporting trajectory extraction in diverse environments. However, accurate vehicle detection is a challenge due to facts such as small vehicle size and inconspicuous object features in UAV videos. In addition, camera motion in UAV videos hardens the trajectory construction procedure. This research aims at proposing a novel framework for accurate vehicle trajectory construction from UAV videos under mixed traffic conditions. Firstly, a Convolution Neural Network (CNN)-based detection algorithm, named You Only Look Once (YOLO) v3, is applied to detect vehicles globally. Then an image registration method based on Shi-Tomasi corner detection is applied for camera motion compensation. Trajectory construction methods are proposed to obtain accurate vehicle trajectories based on data correlation and trajectory compensation. At last, the ensemble empirical mode decomposition (EEMD) is applied for trajectory data denoising. Our framework is tested on three aerial videos taken by an UAV on urban roads with one including intersection. The extracted vehicle trajectories are compared with manual counts. The results show that the proposed framework achieves an average Recall of 91.91% for motor vehicles, 81.98% for non-motorized vehicles and 78.13% for pedestrians in three videos.

INDEX TERMS Mixed traffic, trajectory construction, unmanned aerial vehicles, vehicle detection, vehicle trajectory, YOLOv3.

I. INTRODUCTION

Mixed traffic flow refers to traffic flow including motor vehicles, non-motorized vehicles (bicycles, motorcycles, etc., simplified as NMV) and pedestrians (simplified as Pts in tables). The research on mixed traffic flow offers theoretical foundations for urban road planning [1], [2], road design optimization [3], [4], traffic organization design [5], [6], traffic safety diagnosis [7], [8], traffic flow prediction [9], [10], and traffic environment improvement [11]. The mixed vehicle trajectories can provide crucial data support on mixed traffic flow research and traffic safety evaluation in the microscopic scope [12], [13]. Despite well-used traffic parameters such as average speed, density and volume, micro traffic parameters such as speed, acceleration, space headway, time headway and gap of individual road users are also available from the trajectory data. These microscopic traffic parameters are essential in data-driven research such as conflict point determination in urban intersections and driving strategy design for unmanned vehicles. Therefore, the importance of mixed road user trajectory data is obvious for traffic-flow-related studies.

The most famous trajectory dataset is the Next Generation Simulation (NGSIM) dataset launched by the Federal

The associate editor coordinating the review of this manuscript and approving it for publication was Cong Pu.

Highway Administration, United States [14]. The project installs multiple fixed cameras on the top of a nearby building to collect traffic videos. It publishes motor vehicle trajectory data on four road segments, containing information such as instantaneous vehicle velocity, acceleration, position coordinates, vehicle length, and vehicle type. This dataset has been widely used since its release. However, the NGSIM dataset does not include trajectories of NMV and pedestrians, which limits its use on mixed traffic situation. Besides, the dataset has some technical limitations such as fixed road segment, insufficient coverage range, limited traffic flow condition, limited vehicle component, as well as erroneous speed and acceleration information. Such limitations are associated with the fixed locations of cameras at inclined shooting angle as well as the trajectory extraction methods applied on those videos [15]–[17].

Recently, unmanned aerial vehicles (UAV) technology brings a new trend of traffic data collection [18], [19]. A researcher can fly an UAV carrying a high-definition camera for traffic flow video capture in different road segments at flexible timetable under permitted conditions. Compared with cameras installed at fixed locations used by NGSIM, the UAV for video collection has the advantage of high acquisition flexibility, excellent continuity of traffic flow and prolonged length of road segment acquisition. However, in the meantime, due to the high shooting altitude and the camera drifting, vehicle targets in aerial videos often have some unique characteristics such as small vehicle size, large target quantity and inconspicuous object features. Moreover, in moving aerial videos, camera motion and object motion are mixed in the view. Thus, they bring significant challenges in vehicle detection and trajectory extraction from UAV videos.

II. RELATED WORKS

In existing studies, the trajectory extraction on mixed road users is mainly divided into trajectory extraction with only motor vehicles and trajectory extraction with pedestrians. The two types of trajectory extraction methods are reviewed below. Xiang, X.et.al uses AdaBoosting classifier and optical flow for detection and tracks vehicles [20]. But optical flow is not stable with the environment change. It lowers the precision of the trajectory extraction. J. Apeltauer.et.al proposed a method via Boosted classifier and sequential particle filter to detect and track [21]. And background subtraction in aerial videos is adopted in vehicle detection by Azevedo et al. They correlate these position result by k-shortest disjoints paths algorithm [22]. However, the position accuracy of their result is not high enough to track a smaller object when their algorithm is applied in trajectory extraction of mixed traffic.

As for pedestrian trajectory extraction, Yang.et al. extracts the trajectory of motor vehicles and pedestrians in the aerial video using the k-nearest-neighbor algorithm to match the scale-invariant feature and generate trajectories [23]. Bian, C.et al. tracks pedestrians in low altitude UAV videos via Histogram of Oriented Gradients and the Support Vector Machine [24]. But these methods are not suitable for massive and multi-classes trajectory extraction because they have strong dependence on color feature and can only work on single-class objects.

The emerging deep learning algorithms, especially those applying Convolution Neural Networks (CNN), such as You Only Look Once (YOLO) v3, and Region-based CNN (R-CNN), have shown great potentials in high accurate target detection tasks. Compared with traditional algorithms, deep learning does not rely solely on single image information such as gray scale or color. It is less affected by light change and image scaling as well as processing strong adaptability and excellent portability to the scene. At present, some researches have used deep learning algorithms in ship recognition [25], [26]. However, only a few studies have tested the performances of those algorithms for vehicle detection on UAV videos. Besides, the correlation algorithms for compensating lost detection need to be further enhanced for more accurate vehicle trajectory construction.

In our study, a trajectory extraction framework on mixed road users is proposed to acquire trajectories in certain road segments fast and accurately. Our framework is able to extract trajectories of majority road users (motor vehicles, non-motorized vehicles like motorcycle and bicycle, and pedestrians) at urban intersections. Also, it is able to deal with aerial videos with moving backgrounds and larger sampling area. The summary of related work is shown in Table 1.

The proposed framework is tested on three aerial videos, and it proposes an effective solution for the high-precision trajectory extraction of mixed traffic in aerial video with moving background. The three videos above are under different traffic conditions at urban intersections, including free-flow and congested conditions.

III. METHODS

A. OVERALL FRAMEWORK

The primary objective of the study is to propose a methodological framework for trajectory extraction in aerial video with moving background on mixed road users. The framework consists of four modules, as shown in Figure 1, namely vehicle detection, background registration, trajectory compensation, and trajectory denoising. In the first module, the YOLOv3 algorithm based on the convolutional neural network is applied for accurate vehicle detection. Bounding boxes of targets are obtained in this step. Then the Shi-Tomasi corner feature is applied on background registration to obtain rule of image motion. Vehicle detection and background registration work at the same time. After knowing the motion of background image, existing coordinates are transformed into a uniform fixed coordinate system. The data correlation algorithm and compensation method based on the judgment of speed limit is proposed to form rough vehicle trajectories. The position points in the fixed coordinate system are associated

TABLE 1. Summary of related work.

Author	Model	Advantage	Disadvantage
Xiao et al. (2007)	Optical flow + Sniff algorithm	Low false alarm and missing detection rates in vehicle tracking.	Insufficient accuracy and not suitable for extracting trajectories of small objects.
J. Apeltauer.et al. (2015)	Boosted classifier + sequential particle filter	Able to tackle the problem of automatic traffic analysis at an intersection from visual data.	
Azevedo et al. (2014)	Background subtraction + k-shortest disjoints paths	Minimize the resources needed for trajectory data collection.	
Yang.et al. (2019)	K-nearest-neighbor algorithm + scale-invariant feature	Develop two datasets for pedestrian motion models.	Needs initialization and not suitable for massive trajectory extraction.
Plaue.et al. (2011)	Lucas-Kanade tracking method	Can work on data obtained from arbitrary observation angle and don't require extra information.	
Our Research	YOLOv3+shi-tomasi+compensation+EEMD	High detection accuracy; able to track trajectories of small objects; support trajectory extraction in moving background.	Requires a vertical shooting angle; Trajectory accuracy of NMV and Pts can be further improved.

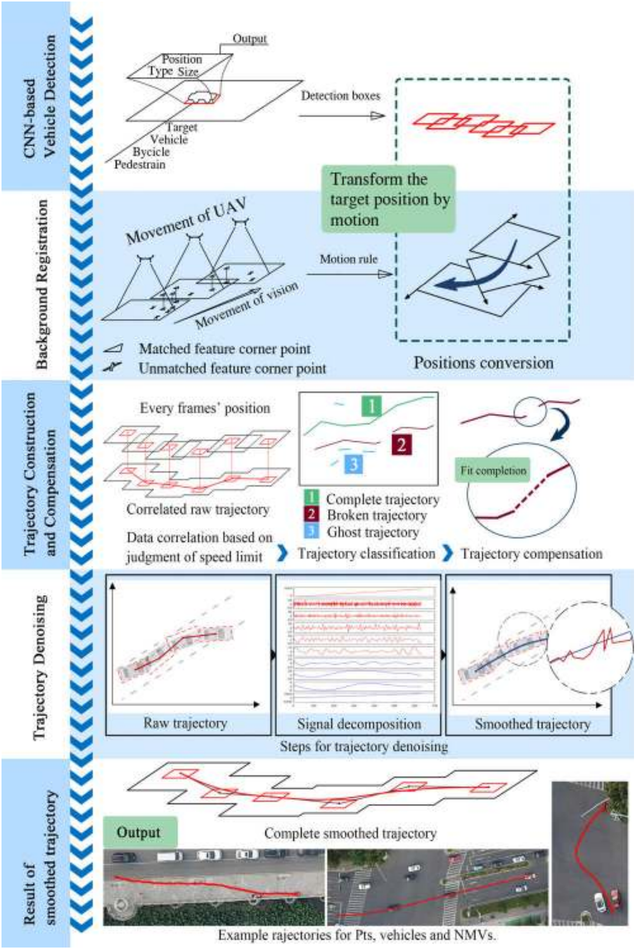

CNN-based Vehicle Detection
Output
Position
Type Size
Target
Vehicle
Bycicle
Pedestrain
Detection boxes
Transform the target position by motion
Background Registration
Movement of UAV
Motion rule
Movement of vision
Matched feature corner point
Unmatched feature corner point
Positions conversion
Trajectory Construction and Compensation
Every frames' position
Correlated raw trajectory
Data correlation based on judgment of speed limit
1
2
3
1 Complete trajectory
2 Broken trajectory
3 Ghost trajectory
Trajectory classification
Fit completion
Trajectory compensation
Trajectory Denoising
Raw trajectory
Signal decomposition
Smoothed trajectory
Steps for trajectory denoising
Result of smoothed trajectory
Output
Complete smoothed trajectory
Example rajectories for Pts, vehicles and NMVs.

FIGURE 1. Framework of our algorithm.

to form a trajectory, and then the trajectories are classified and composed. In the end, the EEMD-based denoising algorithm is applied to eliminate errors to improve trajectory accuracy. The details are presented in the following sections.

B. CNN-BASED VEHICLE DETECTION

The YOLO series [27], [28] and R-CNN [29] series are two main architectures in the field of CNN [30]. After careful literature review, the YOLOv3 proposed initially by Joseph Redmon et al. [28] is considered for vehicle detections in UAV videos. It is reported that YOLOv3 (the best version in YOLO series) is more suitable for vehicle detection in aerial videos than Faster R-CNN (the best version in R-CNN series), and YOLOv3 achieves higher detection accuracy and faster processing speed than R-CNN [30]. In our study, we also test the Faster R-CNN on our videos. The results show that the Recall by the YOLOv3 is 94.26% while by the Faster R-CNN is 89.53% for motor vehicles. As a result, YOLOv3 is chosen for the vehicle detection.

The YOLOv3 algorithm requires large training samples to achieve excellent performance in vehicle detection. In our study, the proposed procedure, which implements the mixed Gaussian background-modeling algorithm [31] is applied for rough vehicle detection and creates the training vehicle samples. The algorithm models each pixel as a mixture of Gaussians and uses an online approximation to update the model. Pixel values that do not fit the background distributions are considered as foreground until there is a Gaussian that includes them with sufficient and consistent supporting evidence. Then the foreground is processed with the opening operation and closing operation to complete holes caused by background noise. At last, a Kalman filter is used to recognize vehicles in the foreground. We used a higher level of constraint to ensure a low false detection rate. A few non-vehicle samples were manually excluded from the training dataset.

During the training process, the training pictures are first scaled to a uniform size and then sent to the CNN in batches (a group of images each time) for logistic regression prediction. The CNN used in YOLOv3 mainly implements the darknet-53 neural network [28], which combines the design of YOLO-v2, darknet-19 and residual network. The input image first goes to convolutional layers to get samples of features. The size of the layer is set to 1 × 1 or 3 × 3 to reduce the floating-point operations and increase the producing speed. The shortcut layers are arranged among convolutional layers at specific intervals to divide the CNN into dozens of pieces. The dividing operation is necessary for controlling the propagation of gradients as well as avoiding the problems of gradient diffusion and gradient exploding. At last, the yolo layers perform classification and position prediction of the targets. Three different scales are provided in the yolo layers for the detection of small, medium-sized, and large targets. The prediction result will be saved and updated in the weights file. The training effect of each batch is represented by the intersection-over-union (IoU).

Area of overlap indicates the overlap range of the prediction box and the true value box. Area of union indicates the range of the prediction box plus the truth-value box. It can be seen that the IoU indicates how accurate the model detects the target.

The training performance after iteration is expressed in terms of loss as:

$$loss = loss_{xy} + loss_{wh} + loss_{confidence} \\ + loss_{class} + loss_0 \quad (1)$$

where $loss_{xy}$ is the error of the center point of the detection box, $loss_{wh}$ is the error of the detection boxes' length and width, $loss_{confidence}$ is the error of confidence of the detection box, and $loss_{class}$ is the error of the classification of the detection box. $loss_0$ indicates the loss value of the last iteration, and the final image detection performance is a superposition of the loss values after all iterations. The training model usually has a better performance with a smaller loss, while *loss* keeps changing when the training procedure is conducting. As a result, the training process is terminated when the minimum loss is found.

The training result is then sent back to the YOLOv3 network for detection and classification to form the raw detection pool. The raw data may contain duplicated detection boxes of the same vehicle, which need to be reduced. We propose the duplication reduction method, in which the duplicated detection boxes are identified by the following Equations 2,3:

$$|x_1 - x_2| < \frac{\frac{l_1}{2} + \frac{l_2}{2}}{2} \quad (2)$$

$$|y_1 - y_2| < \frac{\frac{w_1}{2} + \frac{w_2}{2}}{2} \quad (3)$$

where (x_1, y_1, l_1, w_1) and (x_2, y_2, l_2, w_2) refer to the position of the duplicate boxes respectively, x is the X-axis coordinate of the image top left corner, y is the Y-axis coordinate of the image top left corner, l is the side length of a detection box in X-axis, and w is the side length of a detection box in Y-axis. In the reduction of the duplicate boxes, the one with a higher confidence score is kept to be the correct vehicle detection while the other is dropped.

The generated boundaries of detection boxes by YOLOv3 may not fully comply with the real vehicle boundaries [32]. It may result in the erroneous estimations of vehicle size. Such issue will be eliminated during the trajectory construction.

C. BACKGROUND REGISTRATION

Since the mixed users traffic flow is collected in urban situation, the background will contain plenty of 'corner points' such as facades of buildings and edge of the greenbelt. The corner points are stable in the video when the background moves. So, we choose the corner point as the matching point for registration. The registration is used to obtain a conversion relationship between the video coordinate system and the ground coordinate system for compensating the deviation caused by video motion. This part is implemented in the following steps.

1) CORNER DETECTION AND CORRELATION BASED ON SHI-TOMASI ALGORITHM

Shi-Tomasi Corner Detection [33] is proposed by J. Shi and C. Tomasi. It is an improvement of Harris Corner Detection. Shi and Tomasi derive an image motion model for affine motion and pure translation, which they use for tracking and monitoring target features. In this model, the matrix involved is equivalent to **M**. (Equation 4, 5) Given an image *I*, the algorithm first computes the following matrix for every pixel *(x, y)*, which is an approximation to the local auto-correlation function of image *I*:

$$\mathbf{M}(\mathrm{x, y}) = \begin{bmatrix} \sum_{\mathrm{u,v}} \mathrm{w_{u,v}} \cdot [\mathrm{I_x}(\mathrm{x_r, y_r})]^2 & \sum_{\mathrm{u,v}} \mathrm{w_{u,v}} \cdot \mathrm{I_x}(\mathrm{x_r, y_r}) \mathrm{I_y}(\mathrm{x_r, y_r}) \\ \sum_{\mathrm{u,v}} \mathrm{w_{u,v}} \cdot \mathrm{I_x}(\mathrm{x_r, y_r}) \mathrm{I_y}(\mathrm{x_r, y_r}) & \sum_{\mathrm{u,v}} \mathrm{w_{u,v}} \cdot [\mathrm{I_y}(\mathrm{x_r, y_r})]^2 \end{bmatrix} \quad (4)$$

where $\mathrm{I_x}$ and $\mathrm{I_y}$ denote the derivatives of image I, $(x_r, y_r) = (x + u, y + v)$, and $w(u, v)$ is a window and weighting function. The eigenvalues λ_1, λ_2 of $\boldsymbol{M}$ are computed and a candidate point is accepted if:

$$\mathrm{c(x, y)} = \min(\lambda_1, \lambda_2) > \theta \cdot \max_{\mathrm{x, y}} \{\mathrm{c(x, y)}\} \quad (5)$$

where $c(x, y)$ is the scoring function of the corner points.

Shi-Tomasi is a very popular corner detector and has been used extensively for various real-time video processing applications because of its fast speed and high accuracy [34], [35]. The affine transformation mentioned in the detector is a description of the criterion function used in the Shi-Tomasi algorithm for registration; it does not directly works on images. It can be found that this function works better than the prior one without affine transformation [33]. We embed the Shi-Tomasi operator in the code of YOLOv3 and detect the corners of buildings globally while detecting the vehicle in every video frame. These corner points obtained are matched among adjacent frames by the k-shortest-path algorithm. If there is not a candidate corner point in two adjacent frames, it will be ignored.

2) OPTIMIZATION FOR BACKGROUND TRANSFORMATION PARAMETER

As mentioned above, with background movement, the motion of the feature points in the picture can be decomposed into pure longitudinal translation a_i, pure lateral translation b_i, and pure rotation of θ_i in the *i* frame to *i+1* frame. The affine transformation is used to find the corner points in the image for screen matching. In the case of high altitude, the image distortion of the road surface can be ignored and regarded as a rigid translation. Therefore, we use rigid transformation for the motion transformation of the road surface to obtain the following rules. The matching corner combination of adjacent frames satisfies the following formulas. (Equation 6-8) Among them, a_i, b_i, and θ_i are the generalized displacements of the drone's pure transverse, longitudinal and rotational.

$$\mathrm{R} = \sqrt{\left(x_i^2 + y_i^2\right)} \tag{6}$$

$$x'_{i+1} = R \cdot \cos\left(\theta_0 + \theta_i\right) + a_i \tag{7}$$

$$y'_{i+1} = R \cdot \sin\left(\theta_0 + \theta_i\right) + b_i \tag{8}$$

where x′, y′, is the calculated position via parameters found by Genetic algorithms(GAs) [36], and x, y is true corner position in the *i* frame.

In order to obtain the movement of UAV, it is necessary to acquire the relationship between adjacent frames in the video. That is, to quantify three parameters a_i, b_i and θ_i. Facing the problem of the insufficient number of equations for one couple of matching points and the problem of many couples of matching points' error allocation, we choose to use genetic algorithms for parameter optimization.

Genetic algorithms (GAs) [36] are randomized searching and optimization techniques guided by the principles of evolution and natural genetics, which processes large amounts of implicit parallelism. In GAs, the parameters of the search space are encoded in the form of an array (called chromosomes). A collection of these arrays is named a population. Initially, if a random population is created, it represents different points in the searching space. The searching space is constrained by the maximum theoretical moving distance. The Threshold R_a, the count number N and MSD value are associated with each array that represents the matching degree of the array.

MSD indicates the deviation between the parameter matching position and the true corner position in all frames. (The matching point combination that has been obtained in step1 is considered as true matching positions.) We use the following (Equation 9, 10) for evaluate the parameters.

$$\delta_i = \sqrt{\left(x_i - x'_i\right)^2 + \left(y_i - y'_i\right)^2} \tag{9}$$

$$\mathrm{MSD} = \frac{\sum_{\mathrm{i}=1}^{\mathrm{n}} \delta_{\mathrm{i}}^2}{\mathrm{n}} \tag{10}$$

In which δ_i indicates the distance between the calculated point and the real match point. When δ_i is smaller than R_a, the point *i* can be considered as a correct match. A few of the arrays are selected by *N* and each is assigned a number of copies that go into the mating pool. The chromosome who equipped larger *N* and smaller MSD is the optimized matching result. After enough iteration, we pick the best chromosome as matching parameters. Thus, the background movement is acquired.

D. TRAJECTORY CONSTRUCTION AND COMPENSATION

The trajectory construction and compensation aim at correlating detection boxes in consecutive image frames to form complete and accurate vehicle trajectories. The background movement is known in this step because the detection and background registration functions at the same time. This step contains the following three steps which are data correlation, trajectory classification, and trajectory compensation.

1) DATA CORRELATION

This part deals with the association of the target vehicles in corresponding frames and forms rough trajectories of the vehicles. Because the targets have little overlap in the vertical view, the search range is limited to the range where other vehicle may not break in, by referring bounding box and motion characteristics of the vehicle. The data correlation procedure is described as following. For any detection box that first appears, its position is represented as ($\mathrm{x_0}$, $\mathrm{y_0}$, $\mathrm{l_0}$, $\mathrm{w_0}$). The box may be a true one which means it contains a valid vehicle (also the start of a valid trajectory track), or a ghost one which does not contain a vehicle. The trajectory is constructed by searching the detection box in the following frames that meet the following conditions (Equation 11, 12):

$$|\mathrm{x_{i-1} + v_{i-1} - x_i}| < \mathrm{v_{max}^x} \quad (0 < \mathrm{i} \le \mathrm{TH}) \tag{11}$$

$$|\mathrm{y_{i-1} - y_i}| < \mathrm{v_{max}^y} \quad (0 < \mathrm{i} \le \mathrm{TH}) \tag{12}$$

where $\mathrm{v_{max}^x}$ and $\mathrm{v_{max}^y}$ are the most pixels a vehicle can move in a frame in X-axis and Y-axis respectively, which should be estimated in different test videos;*i* represents the number of frames being searched; and *TH* is the threshold to *i*, indicating the maximum searching area.*TH* significantly affects the trajectory Precision and needs to be carefully determined. A larger*TH* will increase the possibility of finding the following detection boxes and produce trajectories. However, some false trajectories may be included. On the other hand, a smaller *TH* will reduce the false trajectories but may lose correct trajectories. The raw trajectory data is then produced.

We carefully check the raw trajectory data and find the detection box size in a trajectory may fluctuate around the true vehicle size. Thus, the mean size of all detection boxes was used in the trajectory as the vehicle size. We also find some false trajectories as well as missing trajectories and summarize the possible causes as follows.

- Some invalid trajectories are caused by the intrusion of the ghost detection boxes.
- Some ghost detection boxes are close to each other in consecutive frames, which may be wrongly correlated.

- Some broken trajectories are caused by the missing detection boxes.

2) TRAJECTORY CLASSIFICATION

The current trajectories are carefully examined and these trajectories are summarized into different categories according to the difference in validity and completeness:

(1) A complete track can represent vehicle location in every frame where the vehicle appears. We identify the complete trajectories which start when vehicles completely enter the study segment and end when it reaches the downstream boundary.

(2) A ghost track is not a real trajectory of the vehicle. We apply a speed judgment to find the ghost track, which can be identified by the following conditions (Equation 13):

$$\frac{\sqrt{(\mathrm{x_{end}}-\mathrm{x_1})^2+(\mathrm{y_{end}}-\mathrm{y_1})^2}}{\mathrm{p}} \leq 1 \tag{13}$$

where $(\mathrm{x_1}, \mathrm{y_1})$ and $(\mathrm{x_{end}}, \mathrm{y_{end}})$ refers to the beginning and end central point coordinate of a trajectory, p is the number of frames the trajectory lasts. The judgment works because a true trajectory moves along the road lanes while a ghost trajectory usually does not have a fixed moving direction. Therefore, its estimated speed will be much lower than that of surrounding trajectories.

(3) A repeated track has the same position information in some part with another track. The repetition of trajectory is identified if the end positions of two candidate trajectories are the same in the same frame. This judgment works because in the correlation algorithm, a detection box may be correlated with different detection boxes from upstream, but the detection box will be correlated with one and the only one detection box each time from downstream.

After identifying the repeated trajectories, the repeated part is extracted by the following conditions (Equation 14-16):

$$\mathrm{m_i^a} \neq \mathrm{m_i^b} \tag{14}$$

$$\mathrm{m_{i+1}^a} = \mathrm{m_{i+1}^b} \tag{15}$$

$$\mathrm{R_{re}} = \max\left\{\left|\mathrm{y_i}-\mathrm{y_{i+1}}\right|, \left|\mathrm{y_j}-\mathrm{y_{j+1}}\right|\right\} \tag{16}$$

where m represents position information (x, y, l, w) of candidate vehicles. If $\mathrm{m^a} = \mathrm{m^b}$, it means x, y, l, and w are all equal, respectively. If the Equation 14, 15 are true, i is considered as the start frame of the repetition. $\mathrm{R_{re}}$ is used to judge which trajectory candidate possesses the repeated part.

(4) The rests are broken tracks that show vehicle position in partial frames. The other part of the vehicle position is lost due to missing detection boxes.

Separate operations are taken to deal with the four kinds of trajectories separately. The complete trajectories are kept within the trajectory pool. The repeated part in repeated trajectories is invalid and directly deleted. The ghost trajectories are detected and deleted. The left ones are considered as broken trajectories, which will be further processed in the compensation step.

3) TRAJECTORY COMPENSATION

To correlate the broken trajectories and form complete ones, in this step, we propose a compensation algorithm to match broken trajectories and apply the fitting functions to compensate for the missing part. Firstly, the following conditions (Equation 17-19) are used to find broken pieces, which belong to the trajectory of the same vehicle:

$$0 < \mathrm{f} < \mathrm{f_u} \tag{17}$$

$$0 < \Delta x < V_{max} * f \tag{18}$$

$$\Delta y < w_h \tag{19}$$

where f is the number of lost frames between broken pieces, f is restricted by the upper bound $\mathrm{f_u}$. $(\Delta x, \Delta y)$ are the distance between the closer endpoints of broken pieces in X-axis and Y-axis respectively, and $\mathrm{w_h}$ refers to the threshold of Δy. f_u should be selected by the demands of the processing speed of the extraction method because it indicates the searching area. A higher f_u will cost more time in finding pieces and lower the processing speed, but will increase the possibility of finding pieces. A lower f_u will cause opposite affections. w_h indicates the limitation of searching area in the Y-axis, it is usually set to half the distance between two close lanes to avoid interference of trajectories from other lanes.

Once we have identified the broken pieces of a vehicle trajectory, we use the cubic polynomial fitting (assuming the acceleration changing rate is constant during missing part) to reconstruct the trajectory using the consecutive points from both pieces in their closer end. The position of the compensation part can be calculated as follows (Equation 20-23):

$$x_i = \frac{i * (x_1 - x_{end})}{f} + x_{end} \tag{20}$$

$$y_i = f_3(x_i) \tag{21}$$

$$l = \frac{l_1 + l_{end}}{2} \tag{22}$$

$$w = \frac{w_1 + w_{end}}{2} \tag{23}$$

where i is the current compensating frame, and $f_3(x)$ is the cubic polynomial fitting function.

E. TRAJECTORY DENOISING

The trajectories possess some small position deviation due to the camera shaking or background interference. The deviation may result in the erroneous estimation of vehicle speed, acceleration, and other traffic parameters so that a denoising procedure is necessary. The denoising algorithm should be able to eliminate the errors and outliers while remaining the correct trajectory points. After a careful examination, we selected the ensemble empirical mode decomposition (EEMD) method for our research. EEMD is an improvement of Empirical mode decomposition (EMD) which aims at eliminating noise from non-linear and non-stable signals [37]. The main idea of EMD is to decompose the original signal into multiple intrinsic mode functions (IMF) with various frequencies, while the IMFs with low frequencies indicates the contour of the original signal and the ones with high frequencies contains the noise details of the original signal. The EMD is suffered from mode fixing, which is defined as an IMF either consisting of signals of widely disparate scales, or a signal of a similar scale residing in different IMF components. To overcome the problem, Wu et al. [38] proposed the EEMD that defines the true IMF components as the mean of an ensemble of trials, each consisting of the signal plus a white noise of finite amplitude. Specifically, after adding white noise, the current observation can be described as follows:

$$X_i\,(t) = x\,(t) + w_i\,(t) \tag{24}$$

where i is the iteration number, x(t) is the raw trajectory data, $w_i(t)$ is the added white noise, $X_i\,(t)$ is the noise-aided trajectory data.

In the next step, we seek the extrema points of $X_i(t)$ and line maxima points and minima points with cubic spline to form the upper and lower envelope. Then we take local means of the upper and lower envelope as the next observation $x_i\,(t)$. Define the extrema of $x_i\,(t)$ as N_z, the zero-crossing point as N_e, the upper and lower envelope as $f_{max}(t)$ and $f_{min}(t)$, the IMF is found if $x_i\,(t)$ meets the following conditions:

$$N_z - 1 \leq N_e \leq N_z + 1 \tag{25}$$

$$\frac{f_{max}\,(t) + f_{min}\,(t)}{2} = 0 \tag{26}$$

After enough iteration when IMF can no longer be found, original $\mathrm{X_i}$ (t) can be expressed as:

$$X_i\,(t) = \sum_{j=1}^{n} imf_j\,(t) + r_n\,(t) \tag{27}$$

where $imf_j(t)$ is the IMFs separated from $X_i\,(t)$, and $r_n(t)$ is the residual.

The denoised trajectory is the sum of appropriate IMFs, which is determined with an energy-based IMF selection method [39]. In the signal-processing field, energy indicates the amount of stored information in the signal. Therefore, energy of noise signal only accounts for a small proportion while the majority of energy is concentrated in the meaningful signals. Therefore, the energy of noise signal, in the form of log function, should be negative. An IMF is selected and sums to form the smoothed trajectory if the following conditions are met:

$$E_j = \frac{1}{num}\sum_{k=1}^{num}\left[c_j\,(k)\right]^2 \tag{28}$$

$$log_2 E_j > 0 \tag{29}$$

where $\mathrm{E_j}$ is the energy of the jth IMF, $\mathrm{c_j(k)}$ is the point collection of jth IMF, num is the total number of collection points. The sum of the selected IMFs is the denoised trajectory of the vehicle. After the above steps, accurate vehicle trajectories are extracted from our framework.

IV. EXPERIMENT DESIGN

The performance of our proposed framework is evaluated on three aerial videos captured by high definition camera mounted on an UAV (model: DJI Mavic professional) on a city expressway in Nanjing, China. All videos are captured at 24 frame-per-second (fps) and with the resolution of 4096 pixel $*$2160 pixel. Respectively, test video #1 was taken at a 150m altitude at an urban intersection under a free flow traffic condition. Test video #2 was taken at a 170m altitude at an urban main road under a congested condition. Test video #3 was taken at a 150m altitude at an urban intersection under a congested condition. In the course of data acquisition, the UAV was set to be stabilized in the air. The video information is shown in Table 2.

TABLE 2. Test video information.

Video Information	Test Video #1	Test Video #2	Test Video #3
Road Geometry	With intersection	Without intersection	With intersection
Traffic Condition	Free-flow	Congested	Congested
Frame rate	24fps	24fps	24fps
Resolution	4096×2160	4096×2160	4096×2160
Length	207m	234m	220m
Duration	50s	50s	30s
Frame	1200	1200	720
Focal Length	23mm	23mm	23mm
Flying Height	150m	170m	160m

The trajectory extraction algorithm was developed on the platform of Visual Studio 2015 and Matlab 2016a. The experiment was launched on a workstation with a 2.5 GHz E5-2678v3 dual processor and an 11G memory 2080 Ti graphics card.

We calculated the count of detected vehicles and extracted trajectories for validating the performance of our models. The accuracy is evaluated by two goodness of fit measures, which are:

$$Recall = \frac{TP}{GT} \tag{30}$$

$$Precision = \frac{TP}{TP + FP} \tag{31}$$

where *TP* is true positive which refers to the count of valid outputs (vehicle count or trajectory count), trajectories and bounding boxes without serious deviations is counted into *TP*, *GT* is ground truth which refers to the count of truly- existing trajectories (The targets appearing in videos are counted). It's worth mentioning that overlap is not considered between the extracted trajectories and real tracks here because the ground truth position information requires unrealistic labor work. Even if the overlap information is not estimated here, the 'correct' trajectories have been ensured by eye observation. And*FP* is false positive which refers to the count of fake outputs. Recall is the fraction of relevant instances that have been retrieved over total relevant instances, indicating the model performance of obtaining true count of vehicles or trajectories in the test video. Precision is the fraction of relevant instances among the retrieved instances, indicating how badly the algorithm mistook the false vehicle or trajectory as the true one.

V. RESULT

A. MODEL TRAINING AND PARAMETER CALIBRATION

The YOLOv3 model is trained to detect vehicles in the UAV videos. The model performance is profoundly affected by the configuration of training settings, including the number of training samples and training parameter adjustments. Similar to Benjdira et al. [30], in our study, we use 12000 vehicle samples produced by the mixed Gaussian background modeling for the basic training. Besides, 4000 non-motor vehicle samples and 1000 pedestrian samples are applied for enhanced training.

The critical parameters that have significant impacts on results are listed below. Especially, Batch is the number of samples participated in training in each iteration. With a higher Batch, the detection performance is more accurate. Subdivision, which varies according to the capability of the hardware, is set to relieve the usage of GPU by sending samples into training dataset in separate groups. Width and height are the scaled image size of training samples. A larger width and height provides a higher resolution of the scaled image; thus, less information is lost from the samples. Max-batches is the total number of iteration in the training progress. It should be appropriately set to obtain the minimum loss, which indicates the overall performance of the training procedure. Learning rate is the extent of changes in detection model according to the results after iteration. It should be set higher at the beginning of training to get a quick decay in loss and lower when approaching the end of training to avoid missing the optimal solution.

The parameters in the training procedure are carefully determined by conducting multiple preliminary tests with the trial-and-error method. Considering the balance between training performance and capability of the hardware, Batch and Subdivision are set to 64 and 32, while Length and Width are set to 672 and 672. Max-batches is set to 40000 in the training procedure. The minimum loss is found near 35000 batches. The learning rate is set to 0.001 at the beginning of training, and a step decay policy is applied to decrease the learning rate from 0.001 to 0.0001 after 30000 batches.

B. RESULTS OF VEHICLE DETECTIONS AND BACKGROUND REGISTRATION

Vehicles are detected on the test videos using the calibrated model. The detection performances are seen in Figure 2.

To validate the model performance, the ground truth data of vehicle counts are manually counted by the research team members from the UAV videos. The vehicle detection performances are shown in Table 2.

The results show that the calibrated YOLOv3 model performs reasonably well in detecting the motor vehicles in the UAV videos. Notably, the Recall of motor vehicles in test video #1 is 94.26%, which is similar with that in test video #2 (96.23%) and in test video #3 (95.03%). However, the Recall of NMV and pedestrians in three videos are both less than 90% (83.85% and 80.00% in test video #1, 84.33%, 81.07% in test video #2 and 82.57% and 81.28% in test video #3). The main reason of the difference is that NMV and pedestrians possess much smaller target size and spacing than that of motor vehicles which lowers the confidence score in detection section. Besides, Precision of NMV and pedestrians is about ten percent lower than that of motor vehicles. It's because the features of NMV and pedestrians are alike, and they may be recognized wrongly as the other ones.

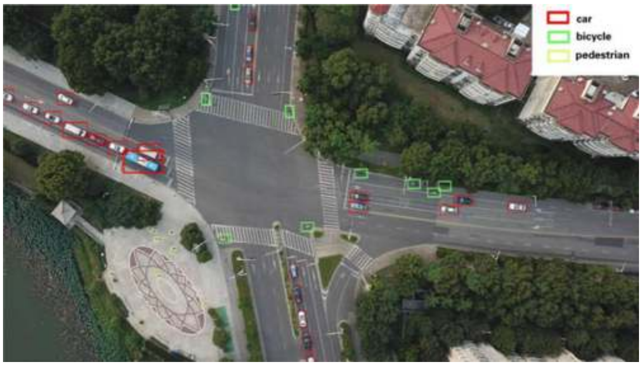


FIGURE 2. Detection Result of three types of vehicles.

Two types of wrong detections identified that are false negative and false positive. False negative indicates the missed detections of true targets. It usually occurs when a lost target has a different shape or color or is covered by road markings, resulting in distinct features from the training samples. The detector may not treat them as targets. Such an issue may be reduced when more diverse training samples are available. False positive indicates fake detections, which occurs when the detector mixes up road surface or road markings with targets because they have similar features.

The background registration successfully links the background-moving videos by a ten-frame-update frequency. The found corner points (the colorful round points) are shown in Figure 3.

As seen in the Figure 3, the upper part of the image is the searching process for global corners, and the bottom is enlarged corner points. The corner points are used to link adjacent images. It is easy to find that most of the corner points are linked to nearby buildings or the stationary vehicles parking on the side of the road. If a stationary vehicle starts moving, it can hardly be judged as corner points, which proves the precision of our method.

C. RESULTS OF TRAJECTORY CONSTRUCTION AND DENOISING

After trajectory construction, the vehicle trajectories are extracted from the three UAV videos. The results are shown in Table 3. The research team manually counts the true count of trajectories, which is 102 (motor vehicles), 55 (NMV),

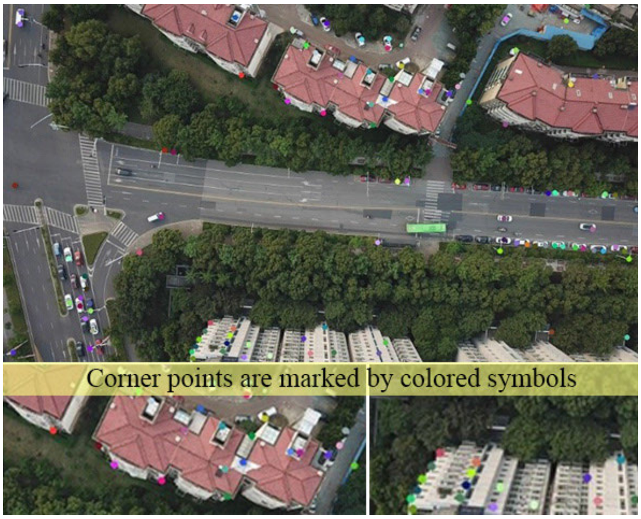
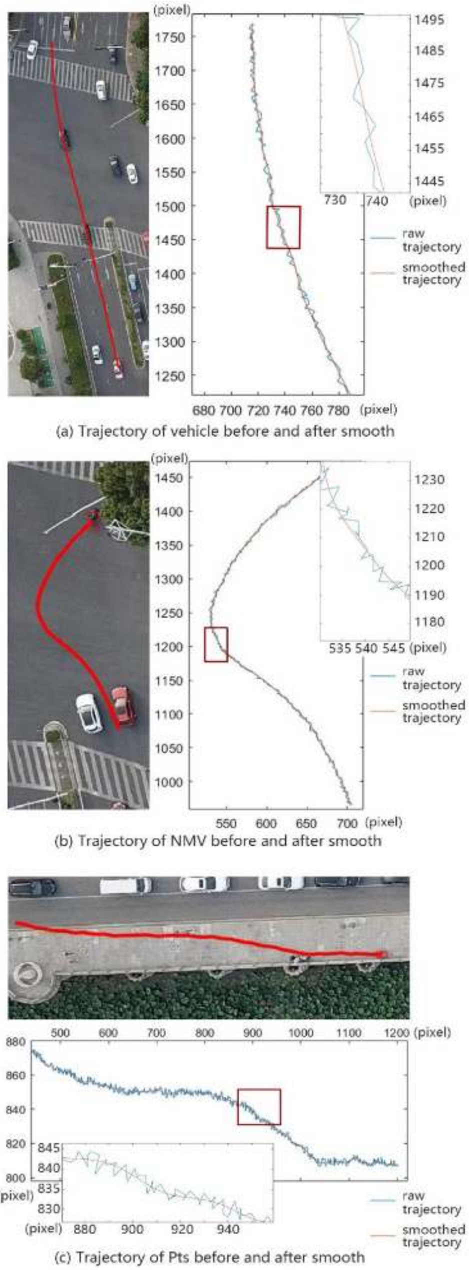


FIGURE 3. Corner detection and background registration.

and 78 (pedestrians) in test video #1, 112 (motor vehicles), 61 (NMV), and 81 (pedestrians) in test video #2 and 131 (motor vehicles), 77 (NMV), and 92 (pedestrians) in test video #3. The trajectory construction result on trajectory count is evaluated in Table 3. Results show that Recall of motor vehicles is nearly equivalent in three videos (92.16%, 91.96% and 91.60%). The Recall of NMV and pedestrians (81.82% and 78.21% in test video #1, 83.61% and 79.01% in test video #2 and 80.52% and 77.17% in test video #3) is lower than that of motor vehicles. This is because the Recall of motor vehicles in detection step is already higher than that of NMV and pedestrians. If comparing the Recall of detection and trajectory construction step in test video #1 for example (94.26%, 92.16% of motor vehicles, 83.85% versus 81.82% of NMV and 80.00% versus 78.21% of pedestrians), all types of vehicles have approximately two percent decrease. It indicates that the performance of the correlation algorithm is less influenced by the target size or spacing in the task of aerial video trajectory extraction.

There are 118 trajectories lost or not correctly correlated in three videos of all vehicle types. We found that most of the missing trajectories are caused by detection failure. The continuum of detection boxes in the trajectories is insufficient, so our methods are not able to form the complete trajectories. Only a few missing trajectories are caused by a detection box in one trajectory correlating wrongly to the trajectory of another vehicle. The mismatch occurs when the to-be-correlated detection box has too much position derivation so that the true vehicle position is not in the searching area.

In the denoising section, the examples of trajectory results for motor vehicles, NMVs and pedestrians using EEMD are shown in Figure 4(a)-(c). The red lines show the smoothed object trajectory in real scenes. The blue lines and orange lines in the rectangular coordinates show the original and smoothed trajectory separately. As can be seen in the figures, the original trajectory data has large fluctuation, while the denoised data is much smoother.



(a) Trajectory of vehicle before and after smooth

(b) Trajectory of NMV before and after smooth

(c) Trajectory of Pts before and after smooth

FIGURE 4. Examples of trajectories before and after denoising.

VI. DISCUSSIONS

Our study provides a new way for mixed road users trajectory extraction which reduces the amount of labor work. Thus, it can help enrich the trajectory data for traffic flow studies. Based on our video data, more detailed mixed traffic flow research can be launched.

But, there are still some deficiencies in this study waiting for further research by further researchers, including but not limited to the following.

First, our test videos are limited in number and contain a narrow range for traffic volume. Researchers will add different traffic conditions videos to test the algorithm framework in further research.

TABLE 3. YOLOv3 detection and trajectory construction performances.

YOLOv3 Detection									
Performances	Test video #1			Test video #2			Test video #3		
	MV	NMV	Pt	MV	NMV	Pt	MV	NMV	Pt
Ground truth	54512	16105	24871	84737	27576	39979	50293	14076	19438
True positive	51383	13504	19897	81542	23255	32411	47792	11623	15799
False negative	3129	2601	4974	3195	4321	7568	2501	2453	3639
False positive	984	2009	3217	840	3084	4989	783	2396	4115
Recall	94.26% (sd=0.0007)	83.85% (sd=0.0026)	80.00% (sd=0.0022)	96.23% (sd=0.0010)	84.33% (sd=0.0021)	81.07% (sd=0.0019)	95.03% (sd=0.0009)	82.57% (sd=0.0018)	81.28% (sd=0.0019)
Precision	98.12%	87.05%	86.08%	98.98%	88.29%	86.66%	98.34%	82.91%	79.34%
Trajectory Construction									
Performances	Test video #1			Test video #2			Test video #3		
	MV	NMV	Pt	MV	NMV	Pt	MV	NMV	Pt
Ground truth	102	55	78	112	61	81	131	77	92
True positive	94	45	61	103	51	64	120	62	71
False negative	8	10	17	9	10	17	11	15	21
False positive	1	4	4	3	6	6	1	10	14
Recall	92.16%	81.82%	78.21%	91.96%	83.61%	79.01%	91.60%	80.52%	77.17%
Precision	98.95%	91.84%	93.85%	97.17%	89.47%	91.43%	99.17%	86.11%	83.53%

Our trajectory results are affected by weather. As a result, the performance results will fluctuate greatly in lower weather conditions. The proposed framework has not been tested in different weather scenarios. Researcher will further expand the trajectory extraction under wind and fog conditions. In subsequent research, it is necessary to add strengthening algorithms such as defogging to the framework.

Our framework is only suitable for trajectory extraction of fixed-angle video, that is, only when the drone is orthographic to the ground. We do not have an effective method to deal with the angle change and road surface distortion.

When large object overlap occurs in video, the searching range of data correlation model is difficult to avoid mismatching. It results in a large possibility for trajectory being misled. This problem also limits the ability of our framework to handle overlapping in inclined shooting angles.

In the background registration, we use genetic algorithm to find the optimal result. The result will be affected by the number of children of the genetic algorithm. There will be some deviations. These deviations are eliminated in the demoising part. Depending on the demoising algorithm, it has a certain impact on the fidelity of the trajectory. Regarding registration, it is very worth trying some state-of-art deep-learning based approach in subsequent research.

Our framework needs to obtain each entire trajectory before it can be reconstructed and smoothed, so even if the operations are performed at the same time, the framework will have operational lags and cannot meet the real-time requirements. The average extraction time of each trajectory of this algorithm is about 0.06s / vehicle / fps. In the subsequent improvement of the algorithm, we will try to improve the content of the framework and conduct attempts on real-time trajectory processing.

Moreover, though the UAV provides the tool for obtaining high-resolution trajectories of mixed traffic users, it does contain several challenge issues. For example, flying a UAV above may draw road users' attentions out of the traffic conditions which may cause traffic safety problems. In addition, since a typical UAV can only operate for 20 to 30 minutes, there is risks of dead batteries and aircraft may drop down causing damages to vehicles and persons. Furthermore, security of UAV should be enhanced to avoid any cyber-attack or threat. Those issues need particular attentions of the researchers.

VII. CONCLUSION

This research proposed a novel trajectory extraction framework aiming at trajectory extraction of mixed traffic flow. The framework integrated high-precision vehicle detection by YOLOv3, the Shi-Tomasi corner feature applying for background registration, trajectory construction with correlation and compensation and trajectory denoising by EEMD. Our framework is tested on three aerial videos taken by an UAV on urban roads including intersections. The extracted vehicle trajectories are compared with manual counts.

The experimental results show that the framework achieves a fine accuracy on detection and trajectory extraction. The trajectory results of three road users are affected significantly by respective detection results. The recall and precision of motor vehicles is about ten percent higher than that of NMVs and pedestrians due to the influence of target size. The accuracy of NMVs is slight higher than the accuracy of pedestrians. The precisions of NMVs and pedestrians are lowered due to the similar image features in detection section. The lost trajectories in the experiments are mainly caused by detection failure. The denoising algorithm performed effectively in eliminating outliners and keeping the traffic parameters within a reasonable range. The average recall of trajectory construction is 91.91% for motor vehicles, 81.98% for non-motorized vehicles and 78.13% for pedestrians in three videos. The framework is proved successful in reducing the amount of labor work on trajectory extraction.

ACKNOWLEDGMENT

The authors would like to thank Ms. Y. Chen and Mr. D. Zhu for their efforts on the aerial video collection, and making useful advises on this paper. P. Li would also like to thank the generous offering on Hardware facility. Also, other crew members should be thanked for the manual labeling work.

AUTHOR CONTRIBUTUIONS

The authors confirm contribution to the paper as follows: study conception and design: Z. Li, X. Chen; data collection: C. Fan; analysis and interpretation of results: Z. Li, C. Fan, R. Feng, X. Chen; draft manuscript preparation: C. Fan, Z. Li, R. Feng. All authors reviewed the results and approved the final version of the manuscript.

REFERENCES

[1] Y. Ji, Y. Fan, A. Ermagun, X. Cao, W. Wang, and K. Das, "Public bicycle as a feeder mode to rail transit in China: The role of gender, age, income, trip purpose, and bicycle theft experience," *Int. J. Sustain. Transp.*, vol. 11, no. 4, pp. 308–317, Nov. 2016.

[2] Y. Guo, Z. Li, Y. Wu, and C. Xu, "Evaluating factors affecting electric bike users' registration of license plate in China using Bayesian approach," *Transp. Res. F, Traffic Psychol. Behav.*, vol. 59, pp. 212–221, Nov. 2018.

[3] Y. Ji, X. Ma, M. Yang, Y. Jin, and L. Gao, "Exploring spatially varying influences on metro-bikeshare transfer: A geographically weighted Poisson regression approach," *Sustainability*, vol. 10, no. 5, p. 1526, May 2018.

[4] P. Liu, J. Wu, H. Zhou, J. Bao, and Z. Yang, "Estimating queue length for contraflow left-turn lane design at signalized intersections," *Transp. Eng. A, Syst.*, vol. 145, no. 6, 2019, Art. no. 04019020.

[5] D. Mingyang and C. Lin, "Better understanding the characteristics and influential factors of different travel patterns in free-floating bike sharing: Evidence from Nanjing, China," *Sustainability*, vol. 10, no. 4, p. 1244, 2018.

[6] Y. Yuan, M. Yang, J. Wu, S. Rasouli, and D. Lei, "Assessing bus transit service from the perspective of elderly passengers in harbin, china," *Int. J. Sustain. Transp.*, vol. 13, no. 10, pp. 761–776, Jan. 2019.

[7] C. Wang, C. Xu, J. Xia, Z. Qian, and L. Lu, "A combined use of microscopic traffic simulation and extreme value methods for traffic safety evaluation," *Transp. Res. C, Emerg. Technol.*, vol. 90, pp. 281–291, May 2018.

[8] C. Wang, C. Xu, J. Xia, and Z. Qian, "Modeling faults among e-bike-related fatal crashes in China," *Traffic Injury Prevention*, vol. 18, no. 2, pp. 175–181, Oct. 2016.

[9] D. Chen, "Research on traffic flow prediction in the big data environment based on the improved RBF neural network," *IEEE Trans Ind. Informat.*, vol. 13, no. 4, pp. 2000–2008, Aug. 2017.

[10] L. Li, J. Zhang, Y. Wang, and B. Ran, "Missing value imputation for traffic-related time series data based on a multi-view learning method," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 8, pp. 2933–2943, Aug. 2019.

[11] Y. Pan, S. Chen, F. Qiao, S. V. Ukkusuri, and K. Tang, "Estimation of real-driving emissions for buses fueled with liquefied natural gas based on gradient boosted regression trees," *Sci. Total Environ.*, vol. 660, pp. 741–750, Apr. 2019.

[12] C. Xu, Y. Wang, P. Liu, W. Wang, and J. Bao, "Quantitative risk assessment of freeway crash casualty using high-resolution traffic data," *Rel. Eng. Syst. Saf.*, vol. 169, pp. 299–311, Jan. 2018.

[13] X. Gu, M. Abdel-Aty, Q. Xiang, Q. Cai, and J. Yuan, "Utilizing UAV video data for in-depth analysis of drivers' crash risk at interchange merging areas," *Accident Anal. Prevention*, vol. 123, pp. 159–169, Feb. 2019.

[14] U.S. Federal Highway Administration. (2006). *Next Generation Simulation Program (NGSIM)*. [Online]. Available: https://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm

[15] M. Montanino and V. Punzo, "Making NGSIM data usable for studies on traffic flow theory: Multistep method for vehicle trajectory reconstruction," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2390, no. 1, pp. 99–111, Jan. 2013.

[16] V. Punzo, M. T. Borzacchiello, and B. Ciuffo, "On the assessment of vehicle trajectory data accuracy and application to the next generation SIMulation (NGSIM) program data," *Transp. Res. C, Emerg. Technol.*, vol. 19, no. 6, pp. 1243–1262, Dec. 2011.

[17] Z. Zheng and S. Washington, "On selecting an optimal wavelet for detecting singularities in traffic and vehicular data," *Transp. Res. C, Emerg. Technol.*, vol. 25, pp. 18–33, Dec. 2012.

[18] E. N. Barmpounakis, E. I. Vlahogianni, and J. C. Golias, "Unmanned aerial aircraft systems for transportation engineering: Current practice and future challenges," *Int. J. Transp. Sci. Technol.*, vol. 5, no. 3, pp. 111–122, Oct. 2016.

[19] T. Nawaz, A. Cavallaro, and B. Rinner, "Trajectory clustering for motion pattern extraction in aerial videos," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Paris, France, Oct. 2014, pp. 1016–1020.

[20] X. Xiang, W. Bao, H. Tang, J. Li, and Y. Wei, "Vehicle detection and tracking for gas station surveillance based on AdaBoosting and optical flow," in *Proc. 12th World Congr. Intell. Control Autom. (WCICA)*, Jun. 2016, pp. 818–821.

[21] J. Apeltauer, A. Babinec, D. Herman, and T. Apeltauer, "Automatic vehicle trajectory extraction for traffic analysis from aerial video data," *ISPRS-Int. Arch. Photogram., Remote Sens. Spatial Inf. Sci.*, vol. XL-3/W2, pp. 9–15, Mar. 2015.

[22] C. L. Azevedo, J. L. Cardoso, M. Ben-Akiva, J. P. Costeira, and M. Marques, "Automatic vehicle trajectory extraction by aerial remote sensing," *Procedia-Social Behav. Sci.*, vol. 111, pp. 849–858, Feb. 2014.

[23] D. Yang, L. Li, K. Redmill, and U. Ozguner, "Top-view trajectories: A pedestrian dataset of vehicle-crowd interaction from controlled experiments and crowded campus," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2019, pp. 899–904.

[24] C. Bian, Z. Yang, T. Zhang, and H. Xiong, "Pedestrian tracking from an unmanned aerial vehicle," in *Proc. IEEE 13th Int. Conf. Signal Process. (ICSP)*, Nov. 2016, pp. 1067–1071.

[25] X. Chen, S. Wang, C. Shi, H. Wu, J. Zhao, and J. Fu, "Robust ship tracking via multi-view learning and sparse representation," *J. Navigat.*, vol. 2019, no. 72, pp. 176–192, Sep. 2018, doi: 10.1017/S0373463318000504.

[26] X. Chen, Y. Yang, S. Wang, H. Wu, J. Tang, J. Zhao, and Z. Wang, "Ship type recognition via a coarse-to-fine cascaded convolution neural network," *J. Navigat.*, to be published. [Online]. Available: https://www.cambridge.org/core/journals/journal-of-navigation/article/ship-type-recognition-via-a-coarsetofine-cascaded-convolution-neural-network/270E2532C239492571B21EED9034BED1, doi: 10.1017/S0373463319000900.

[27] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[28] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: https://arxiv.org/abs/1804.02767

[29] Q. Fan, L. Brown, and J. Smith, "A closer look at faster R-CNN for vehicle detection," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2016, pp. 124–129.

[30] B. Benjdira, T. Khursheed, A. Koubaa, A. Ammar, and K. Ouni, "Car detection using unmanned aerial vehicles: Comparison between faster R-CNN and YOLOv3," in *Proc. 1st Int. Conf. Unmanned Vehicle Systems-Oman (UVS)*, Feb. 2019, pp. 1–6.

[31] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, Jun. 1999, pp. 246–252.

[32] M. B. Jensen, K. Nasrollahi, and T. B. Moeslund, "Evaluating state-of-the-art object detector on challenging traffic light data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 882–888.

[33] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 1994, pp. 593–600.

[34] N. Ramakrishnan, M. Wu, S.-K. Lam, and T. Srikanthan, "Automated thresholding for low-complexity corner detection," in *Proc. NASA/ESA Conf. Adapt. Hardw. Syst. (AHS)*, Jul. 2014, pp. 97–103.

[35] L. Yile, L. Yanping, and W. Yan, "Traffic flow parameter estimation from satellite video data based on optical flow," in *Proc. Comput. Eng. Appl.*, Jul. 2014, pp. 97–103.

[36] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Boston, MA, USA: Addison-Wesley, 1989.

[37] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu, "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," *Proc. Roy. Soc. London. A, Math., Phys. Eng. Sci.*, vol. 454, no. 1971, pp. 903–995, Mar. 1998.

[38] Z. Wu and N. E. Huang, "Ensemble empirical mode decomposition: A noise-assisted data analysis method," *Adv. Adapt. Data Anal.*, vol. 1, no. 1, pp. 1–41, Nov. 2011.

[39] X. Chen, Z. Li, Y. Wang, J. Tang, W. Zhu, C. Shi, and H. Wu, "Anomaly detection and cleaning of highway elevation data from Google earth using ensemble empirical mode decomposition," *J. Transp. Eng. A, Syst.*, vol. 144, no. 5, May 2018, Art. no. 04018015.

RUYI FENG is currently pursuing the bachelor's degree with the School of Transportation, Southeast University. She is also a Research Assistant in Prof. Zhibin Li's group at Southeast University. She was invited by Transportation Research Broad 2020. Her research interests mainly include intelligent transportation, image processing, vehicle tracking, and trajectory extraction. Future research will progress toward intelligent traffic monitoring and traffic intelligent control.

CHANGYAN FAN received the bachelor's degree from Central South University. He used to work on railway traffic control and equipment engineering. He is currently a Research Assistant in Prof. Zhibin Li's group at Southeast University. His research interests include image processing, transportation video analysis, object detection, data correlation, route planning, data quality control, multidimensional traffic data sensing and processing, and so on. He is also a member of the Pilotless Driving Transportation Team.

ZHIBIN LI received the Ph.D. degree from the School of Transportation, Southeast University, Nanjing, China, in 2014. From 2015 to 2017, he was a Postdoctoral Researcher with the University of Washington, USA, and The Hong Kong Polytechnic University. From 2010 to 2012, he was a Visiting Student with the University of California, Berkeley. He is currently a Professor with Southeast University. He has authored or coauthored over 60 articles in international journals. His research interests include intelligent transportation, traffic safety, data mining, and traffic control. He received China National Scholarship, in 2012 and 2013, and the Best Doctoral Dissertation Award by China Intelligent Transportation Systems Association, in 2015.

XINQIANG CHEN received the Ph.D. degree in traffic information engineering and controlling from Shanghai Maritime University, China, in 2018. From September 2015 to September 2016, he was a Visiting Student with the Smart transportation Applications and Research Laboratory (STAR Lab), University of Washington, Seattle, WA, USA. He is currently a Lecturer with the Institute f Logistics Science and Engineering, Shanghai Maritime University. He also serves as an Editorial Board Member for five international journals and technical program committee for six international conferences. His research interests include traffic data analysis, transportation image processing, intelligent transportation systems, computer-vision-based transportation detection and application, smart ship, smart port, and machine learning.