View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Edge Hill University Research Information Repository

Real-time Traffic Analysis Using Deep Learning Techniques And UAV Based Video

Huaizhong Zhang, Mark Liptrott, Nik Bessis
Edge Hill University
St Helens Road, Ormskirk, UK L39 4QP
zhangh@edgehill.ac.uk

Jianquan Cheng
Manchester Metropolitan University
Manchester, UK M15 6BH

Abstract

In urban environments there are daily issues of traffic congestion which city authorities need to address. Real-time analysis of traffic flow information is crucial for efficiently managing urban traffic. This paper aims to conduct traffic analysis using UAV-based videos and deep learning techniques. The road traffic video is collected by using a position-fixed UAV. The most recent deep learning methods are applied to identify the moving objects in videos. The relevant mobility metrics are calculated to conduct traffic analysis and measure the consequences of traffic congestion. The proposed approach is validated with the manual analysis results and the visualization results. The traffic analysis process is real-time in terms of the pre-trained model used.

1. Introduction

The 2018 UN Urbanization Report [1] predicts that smart cities worldwide are developing rapidly and more than 2.5 billion people are going to live in cities by 2050. Thus, many transport problems are emerging simultaneously, especially traffic congestion in urban areas. The study [2] points out that people on average spend more than 75% extra travel time in traffic congestion. Due to the rapidly increasing demand for transport infrastructures, the motorization and the diffusion of the car are the major cause of traffic congestion [1]. Traditionally, camera based video surveillance is used to assess traffic congestion [3]. To address the increasing number of installed cameras, unmanned aerial vehicles (UAV) have become routinely used to deliver autonomous, informative surveillance data. In this study, traffic surveillance and congestion monitoring are being conducted through the on-going collection of traffic information with UAV at a given urban area in China.

Automatic vehicle tracking is crucial to the analysis of real-time traffic video data. Over the years, there have been numerous attempts to tackle this problem with a broad variety of machine learning based algorithms. Two categories can be described. One is to construct a virtual detector with a series of defined bounding boxes [4]. This approach can detect changes in virtual detectors indicating the vehicle's presence. The second is the blob tracking method [5, 6] that is to track vehicles through extracting foreground blobs w.r.t. the vehicles on the scene. However, these conventional approaches depend on handcrafted features to track vehicles and therefore suffer issues of poor accuracy and low robustness. Recently, with the great success of deep convolutional neural networks (CNN) in object detection/recognition [7], the CNN based approaches have the advantage of deep architectures to create the non-handcrafted features accurately representing objects. This approach can outperform conventional approaches both in accuracy and robustness. The CNN based semantic segmentation techniques can be the natural way to detect and track an object [7, 8], especially assist people to track and recognize human activity and action via overcoming physical and cognitive barriers [7, 9]. Using CNN based techniques it is possible to analyze real-time traffic congestion from data produced by UAV's. The most recent approach, Mask RCNN [8], is capable of identifying the instances with a binary mask classifier after performing semantic segmentation on the acquired bounding boxes. In this study, we apply Mask RCNN to track vehicles and pedestrians in the UAV based video. With the efficient and accurate vehicle tracking outcomes, the relevant analysis algorithms are developed to calculate the mobility metrics and visualize traffic scenes so that traffic congestion is assessed and traffic surveillance is conducted with the associated evidence, for example vehicle interactions.

The structure of the paper is as follows, Section 2 outlines the proposed study. Section 3 presents the experimental results and Section 4 discusses those results and presents conclusions.

2. Method

The proposed method is a developed traffic surveillance and congestion analysis framework that consists of a UAV based video collection system, a vehicle tracking system and a traffic analysis system. Fig. 1 illustrates the main blocks of our approach.

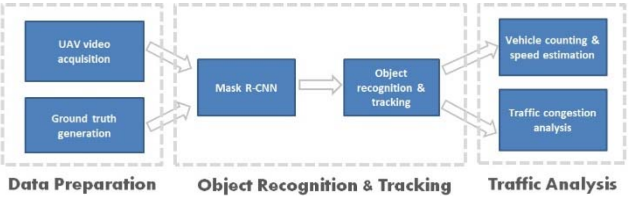
UAV video acquisition
Ground truth generation
Mask R-CNN
Object recognition & tracking
Vehicle counting & speed estimation
Traffic congestion analysis
Data Preparation
Object Recognition & Tracking
Traffic Analysis

Figure 1: Block diagram of our proposed approach.

2.1. UAV and data collection

This initial study uses credible surveillance video data from a fixed UVA (Mavic Pro Platinum Model, Dji Company). The venue is located at Changbo Road, Nanning City, Guangxi Province. The length of the road scene is approximately 112m and its width is about 10m. The video recording lasts 22mins with the MOV format, which the field of view (FOV) is 1080(h)x1920(w) and the frame rate is 24 fps.

2.2. Vehicle recognition and tracking using Mask RCNN

In the tracking system, we consider that the motorbike is a very popular transport tool for ordinary people in China and is actually becoming one of the major factors of causing traffic congestion in China's urban areas. Thus, the object classes defined in this study are named Motorbike, Vehicle and Pedestrian. Here, Vehicle represents all other vehicles except for Motorbike.

2.2.1. Generation of the training set

Forty eight images of the scene selected from the beginning 3 minutes of the video form the basis of the set. The ratios, 5:2:1, are used to divide the images for the training set, the test set and the validate set respectively. The LabelMe [10] annotation tool is employed to create the polygonal annotations of objects in each image. The annotation files are generated with the JSON format in the COCO style [11].

2.2.2. Recognition and tracking using Mask RCNN

Mask RCNN [8] is a most recent successful work for instance segmentation that is able to identify the pixel based location of objects in images. It is an appropriate choice for conducting traffic surveillance in our study with the merits of both accuracy and speed. The Facebook official implementation, Detectron [12], using the Caffe2 framework and the ResNet [13] backbone architecture, is employed to realize the model training and perform vehicle tracking with the obtained model.

In the computing environment of Intel i7-7700, GeForce GTX 1080 Ti GPU and 48 GB DRAM, the processing time for our video sequence (1920x1080) is 0.34s per frame, which demonstrates that the performance of our proposed approach is real-time because one key frame is used for each second's video segment. For improved accuracy, the pre-trained ResNet-50 model weights of ImageNet is used for transfer learning. Fig. 2 illustrates the tracking results for the 4-second video segment where the vehicle is tracked as shown in the red box. The vehicle is identified with a given detection line, coordinates of bounding box, class number and object ID obtained from the instance segmentation system. The same vehicle has different colors in different scenes because it is recognized based on vehicle instance.

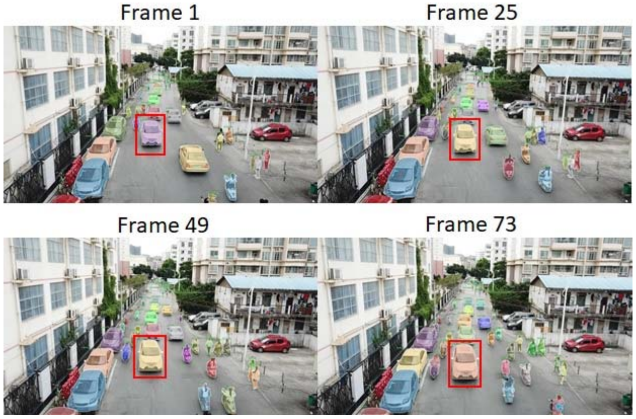

Figure 2: Example for tracking a vehicle (red box) within the 4-second time period.

2.3. Vehicle counting and speed estimation

The algorithm, designed in terms of the detection line, introduced in [6] is used to count vehicles and pedestrians. The main strategy depends on the calculation of the overlap ratio R as:

$$R=\begin{cases}(p_2-q_1)/(p_2-p_1), & if \quad p_1\le q_1\le p_2\le q_2\\ (q_2-p_1)/(p_2-p_1), if & q_1\le p_1\le q_2\le p_2\\ 1, \quad if & q_1\le p_1\le p_2\le q_2\\ (q_2-q_1)/(p_2-p_1), & if \quad p_1\le q_1\le q_2\le p_2\\ 0, \quad otherwise & \end{cases}$$

where p_1, p_2, q_1, q_2 denote the X-coordinates of the end points of the segments that the vehicle intersects the detection line. We set the threshold being 0.75 for *R*. Thus, the same vehicle is detected if *R* exceeds 0.75. Otherwise, the vehicle count is updated.

However, the study in [6] recognizes the significance of lane lines in the ability to recognize neighboring vehicles.

In our case, no lane lines are presented on the road resulting in a more complex traffic situation. Vehicles parked on the roadside were excluded. To address these traffic circumstances, we proposed the following steps to improve the above algorithm in accordance with our UAV based data.

Step 1: The detected line is selected to be near the road end close to the UAV, which is able to clearly differentiate between successive vehicles. In this study, it is the line with the Y-coordinate being 900.

Step 2: A threshold is used to measure the vehicle moving with the center position of the bounding box between two consecutive frames in order to judge whether the vehicle is parked on the roadside. We assess this movement through 3 consecutive frames.

For measuring vehicle speed, the moving distance of the object center from one frame to next frame is estimated according to the acquired bounding box. We use four consecutive frames to calculate the average moving distance that is as the estimated vehicle speed, which is then converted from image pixel based measurement to space meter based measurement.

2.4. Estimation of geometric space for pixels in the image

The distance discussed above is pixel-based. In this initial study, the affine transformation is employed for estimating the geometric space for each pixel in the image. We used the road geometrical information, video resolution and UAV location etc. for the following calculation scheme.

Due to the UAV being fixed, one pixel in the middle of the image is estimated to equal $G_s = 0.148m$ in the geometric space, which is calculated by averaging the street length over the total pixels in the image (*pix_no*=758pixels for the street in this study). For the pixel from the middle point to the far end of the street, the resolution becomes lower and the geometric space for one pixel is proportionally increased in accordance to the scale (length/height for half of *pix_no* pixels so *scale*=0.00985) and then the geometric space for each pixel with $Y_{Coordinate}$ is calculated as follows:

$$G_s + (pix\_no - Y_{Coordinate}) * scale$$

For the pixel from the middle point to the first half of the street, the image resolution is becoming higher and the geometric space for each pixel with $Y_{Coordinate}$ is proportionally decreased when close to the UAV so the calculation of geometric space for a pixel is as follows:

$$G_s - (Y_{Coordinate} - pix\_no) * scale$$

As mentioned above, this calculation scheme used is just for this preliminary study. In further studies, for a better estimation in geometric space, we will apply the orthorectification techniques and image georeferencing [13].

3. Experimental results

In order to examine the performance of the proposed approach, the experimental results include vehicle tracking, traffic measurement metric, and traffic congestion analysis.

3.1. Vehicle recognition and tracking

As described in Section 2, the tracking system is instance based, providing identification information for each instance of a given class. Accordingly, we can calculate the relevant mobility metrics for analyzing the real-time traffic situation. Fig. 3 illustrates that each object on the road is recognized with the colored mask, bounding box, class ID and probability. Some roadside parked vehicles are also detected.

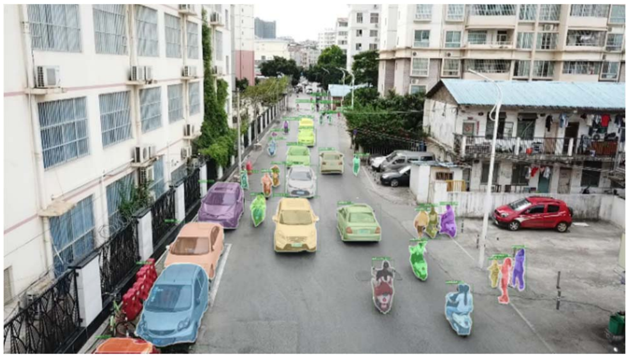

Figure 3: Example for the instance segmentation result

3.2. Calculation of traffic measurement metrics

For the purpose of validation the estimated result of the mobility metrics is compared with the manual detection result. Some basic metrics are employed to do the evaluation such as the maximum and minimum speed of the object, and the passing number of the objects within a time unit. In this initial study, the relevant metrics are calculated within a given region close to the UAV position, which is with the Y-coordinate between 800 and 900. The max/min speed is obtained by comparing the object moving speeds in this region. The passing object count is estimated using the number of the objects across the detection line in the given time period. For the calculation, we selected 3 time periods of the video when there was no significant traffic congestion. Each time period is 1 min.

Table 1 presents the object speed range that we have obtained as described above. The overall speed estimation of our approach is similar to the results using the manual detection. Compared to Vehicle, the estimated speed for Motorbike and Pedestrian falls within a greater range, which may be due to the less accurate center obtained from the bounding box of Vehicle. In addition, Motorbike sometimes drives faster than Vehicle because it is more flexible.

	Motorbike	Vehicle	Pedestrian
Our method	3.54 – 7.18	3.76 – 8.03	0.75 – 1.92
Manual method	3.73 – 6.22	3.58 – 7.56	0.89 – 1.51

Table 1: The object speed detected with our method and the manual detection (unit: m/s), which ranges from min to max.

The counting results for all classes are presented in Table 2. Each number corresponds to each 1 min time period. The numbers of Pedestrians and Vehicles are considered to be accurate. A few vehicles are not counted as their vehicle type is unusual and the tracking system does not recognize them. The motorbikes are sometimes crowded together therefore cannot be accurately counted.

	Motorbike	Vehicle	Pedestrian
Our method	70 – 72 – 89	5 – 8 – 12	6 – 6 – 8
Manual detection	75 – 75 – 92	7 – 8 – 12	6 – 6 – 8

Table 2: Each object count calculated with our method and manual detection (unit: p/min) is the amount of objects passing the detection line in one minute.

3.3. Traffic congestion analysis

Traffic congestion can be detected via the relevant vehicle moving speed on the road. The congestion will happen if the speed of a tracked vehicle slows owing to traffic conditions. This study has established that Motorbikes appear to disregard rules regarding road use by travelling too closely to other vehicles. These traffic incidents are the main cause of traffic congestion.

3.3.1. Vehicle moving affected by motorbike

The analysis of a segment of the video showed the interaction between car and motorbike and the result is displayed in Fig 4. Initially, the car in the yellow box drives at 4.2m/s in Frame 1. The car then slows from Frame 4 to Frame 21 as shown in the graph when two motorbikes approach (see Frame 16). As a result, the car driver believes there may be a collision so slows to a speed of 0.76m/s at Frame 24. This causes traffic congestion. Four scene images are presented to identify the interaction between the vehicle and the motorbikes.

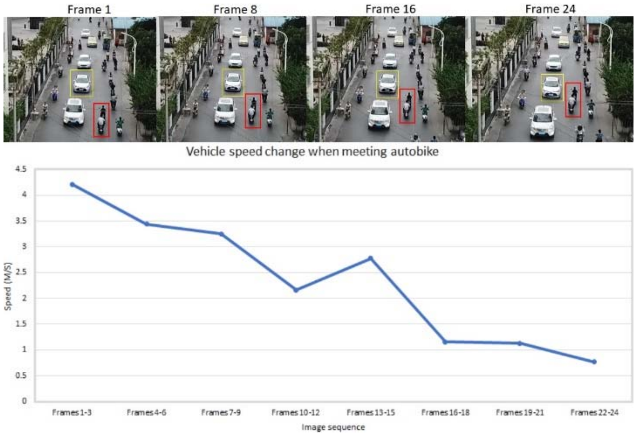

Figure 4: Example for vehicle speed change due to motorbike approaching.

3.3.2. Pedestrian walking affected by motorbike

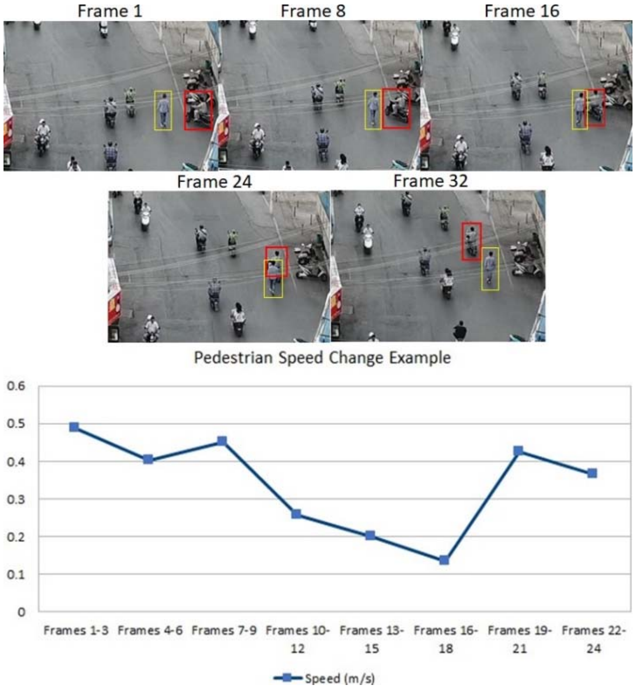

Figure 5: Example for pedestrian walk speed change due to motorbike approaching. The images presented here is zoom-in applied.

Fig. 5 shows another video segment to demonstrate how motorbikes affect pedestrians on the road. Frame 1 shows that a pedestrian in the yellow box walks at 1.4m/s while a motorbike in the red box comes from the roadside. In Frame 16, the motorbike is close to the pedestrian who then adjusts the walking speed to a slow 0.34m/s to avoid a possible collision. When the motorbike leaves, the pedestrian's walk speed becomes normal at 1.2m/s. Figure 5 illustrates the change in the pedestrian's walk speed during the interaction with the motorbike.

4. Discussion and Conclusion

This paper presents a preliminary study on traffic surveillance and congestion using UAV based video and deep learning techniques. This initial research has raised issues surrounding traffic congestion which can be explored in future research offering the opportunity for improved traffic measurement metrics.

This study has established that complex traffic situations can result in inaccurate calculations. In Fig. 6, the traffic deadlock happens because a vehicle in the red box blocks the road due to its moving direction being orthogonal to the road direction. In this case, our approach can detect and recognize the vehicles in both the red box and the yellow box, however the proposed algorithm is unable to correctly count them. It is a challenging issue for future work.

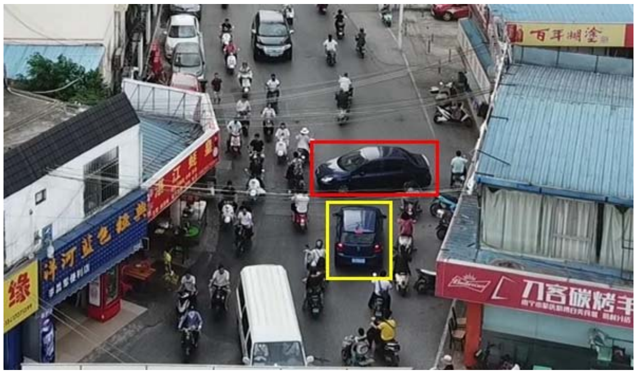

Fig. 6: Wrongly counting example due to serious traffic congestion

In this study, the UAV is fixed to collect the video data so its role is similar to a camera. We apply the geometric configuration and UAV's location to estimate the spatial length pairing to a pixel in the image. In a future study, we will investigate traffic congestion in a large region with a mobile UAV. The pixel measurement could be more accurately calculated using the orthorectification with an appropriate mathematical model and the image georeferencing [14].

As shown in the paper, we focus on seeing the role of motorbike in the formation of traffic congestion in a given city of China. In different areas and with different geometric information, there will be different issues causing traffic congestion. Future studies will expand this initial approach to encompass a range of metropolitan areas to explore reasons for traffic congestion.

Due to the limitation of the acquired video data, this initial study only considers some basic traffic measurement metrics such as Max/Min speed in a given region, vehicle count etc. An improvement in the quality of data could offer sophisticated metrics to investigate a range of traffic congestion issues such as pedestrian/vehicle density, etc.

References

[1] https://population.un.org/wup/, World Urbanization Prospects 2018, United Nations.

[2] S. Çolak, et al. "Understanding congested travel in urban areas," Nat. Commun. 7:10793 doi: 10.1038/ncomms10793, 2016.

[3] G. Haan, H. Piguillet, and H. Post, "Spatial navigation for Context-aware Video Surveillance," IEEE Computer Graphics and Applications, 30(5), pp. 20-31, 2010.

[4] G. Sullivan, K. Baker, A. Worrall, C. Attwood, and P. Remagnino, "Model-based Vehicle Detection and Classification Using Orthographic Approximations", J. Image and Vision Computing, Elsevier, pp. 649-54, 1997.

[5] D. Li, B. Liang, and W. Zhang, "Real-time Moving Vehicle Detection, Tracking, and Counting System Implemented with OpenCV," Proceedings of IEEE ICIST, pp: 631-634, 2014.

[6] F. Liu, Z. Zeng, and R. Jiang, "A Video-based Real-time Adaptive Vehicle-counting System for Urban Roads," PLOS ONE 12(11): e0186098, 2017.

[7] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez, "A Review on Deep Learning Techniques Applied to Semantic Segmentation," arXiv:1704.06857 [cs.CV], 2017

[8] K. He, G. Gkioxari, P. Dollar and R. Girshick, "Mask R-CNN," IEEE TPAMI. doi:10.1109/TPAMI.2018.2844175, 2018.

[9] M. Leo, A. Furnari, G. Medioni, "Deep Learning for Assistive Computer Vision," LNCS, vol. 11134, pp. 3-14, ECCV 2018 Workshops, Munich, Germany.

[10] LabelMe, https://github.com/wkentaro/labelme.

[11] T Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. Zitnick, "Microsoft COCO: Common Objects in Context," Proceedings of ECCV 2014, pp: 740-755, 2014.

[12] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollar, and K. He, "Detectron," https://github.com/facebookresearch/Detectron, 2018.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," arXiv:1512.03385 [cs.CV], 2015.

[14] B. Olawale, C. Chatwin, R. Young, P. Birch, F. Faithpraise, and A. Olukiran, "A Four-Step Ortho-Rectification Procedure for GeoReferencing Video Streams from a Low-Cost UAV," International Journal of Computer and Information Engineering, Vol:9, No:8, 2015.