IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 11, No. 4, December 2022, pp. 1487~1494
ISSN: 2252-8938, DOI: 10.11591/ijai.v11.i4.pp1487-1494

Proposing a route recommendation algorithm for vehicles based on receiving video

Phat Nguyen Huu[1], Phuong Tong Thi Quynh[1], Thien Pham Ngoc[1], Quang Tran Minh[2,3]
[1]School of Electrical and Electronics Engineering, Hanoi University of Science and Technology (HUST), Hanoi City, Vietnam
[2]Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology (HCMUT), Ho Chi Minh City, Vietnam
[3]Vietnam National University Ho Chi Minh City (VNU-HCM), Ho Chi Minh City, Vietnam

Article Info

Article history:

Received Dec 18, 2021
Revised Jul 8, 2022
Accepted Aug 3, 2022

Keywords:

Algorithm A*
Recommendation system
Region of interest
Vehicle detection
YOLOv5

ABSTRACT

In this paper, we propose a method to classify traffic status for the route recommendation system based on received videos. The system will determine the number of vehicles in the region of interest (RoI) to determine and calculate the coefficient of variation (CV) based on the videos extracted from cameras at intersections. It then predicts the congested traffic junctions in the city. The data then goes through the routing module and is transmitted to the website to find the best path between the source and destination requested by users. In this system, we use you only look once (YOLOv5) for vehicle detection and the A* algorithm for routing. The results show that the proposed system achieves 91.67% accuracy in detecting traffic status comparing with YOLOv1, deep convolutional neural network (DCNN), convolutional neural network (CNN), and support vector machine (SVM) models as 91.2%, 90.2%, 89.5%, and 85.0%, respectively.

Corresponding Author:

Phat Nguyen Huu
School of Electrical and Electronics Engineering, Hanoi University of Science and Technology (HUST)
Hanoi, Vietnam
Email: phat.nguyenhuu@hust.edu.vn

1. INTRODUCTION

The population and traffic and transportation demands are increasing, especially in big cities [1], [2]. This causes serious regional traffic jams in urban areas of our countries. Traffic congestion is still a problem not only in Vietnam but also in major cities around the world. This situation leads to many unfortunate consequences such as economic development, environmental pollution, and especially social and security problems. Therefore, this is an issue that needs to be solved with a high priority in our sustainable development plans.

Currently, many systems that detect traffic status and navigate users to avoid congestion are being widely applied around the world such as Google Map, Map, and Waze. In Vietnam, the research and development of similar systems have also received much attention. The most recent can be mentioned as Utraffic-An urban traffic congestion warning system based on data from the community based on analysis of historical data of traffic conditions [3], community data sources [4], and urban traffic conditions from crowdsourced data [1]. Currently, there is a system being deployed for the user community in Ho Chi Minh city. The system collects traffic data from multiple sources and communities through a mobile application. It analyzes the data and applies machine learning techniques to estimate and predict traffic conditions.

We have found that collecting data from the community is a pretty cool and useful solution. Its disadvantage is to take a lot of time to aggregate and analyze data from many different sources. Therefore, we propose a system to detect traffic status in the urban transport network and suggest routes to avoid congestion, and find the shortest path for road users with extracted data from the camera without accessing user data. To solve the problem, we design a system to detect congestion points in the urban traffic network and propose the shortest and most convenient way to avoid congestion for traffic participants. The proposed system has two new features. Firstly, we use the you only look once (YOLOv5) model based on [5], which is a new model for vehicle detection and traffic status determination based on videos extracted from cameras at intersections. Second, we apply a vehicle dataset collected in Vietnam to retrain the YOLOv5 model to improve detection performance in real-time applications. The paper builds a real-time algorithm for displaying and detecting traffic conditions at intersections accurately and to propose optimal routes to help avoid traffic jams for users.

The rest of the paper includes five parts. In section 2, we present several related works. The section 3 proposes the route recommendation algorithm. In the section 4, we will perform the algorithm to evaluate and analyze the results. The final section gives conclusions and future work.

2. RELATED WORK

Currently, there are many methods to determine the traffic condition at a point such as counting the number of vehicles, classifying vehicles, calculating vehicle speed, and vehicle density, calculating the area occupied by vehicles on the road, classifying images from surveillance cameras. Supporting technologies in this process include convolutional neural network (CNN) models such as region - convolutional network (R-CNN) [5], deep convolutional neural network (DCNN) [6], Fast R-CNN [7], and Faster R-CNN [8]. The models have been proposed and achieved many positive results when applied in traffic congestion detection. In [9], the authors use a selective search method to select the candidate regions among possible regions. In [5], they use the R-CNN model because of its candidate regions. In [7], the Fast R-CNN model suggested a less number of candidate regions. However, the using algorithm is not able to learn from the context. In [8], the authors use Faster R-CNN. However, it is difficult to detect objects for real-time applications.

In [10], an intelligent traffic congestion system (CNN model) is introduced by leveraging image classification methods. It uses 1000 images to train for road traffic conditions. The authors just resized and converted the 100-100 grayscale images. This model is proposed to be deployed in a future congestion detection system using closed circuit television (CCTV) cameras that record images on specific locations in real-time.

In [11], the authors use a support vector machine (SVM) and two different deep learning techniques (YOLO and DCNN) to compare the accuracy in classifying congestion images from surveillance cameras. The entire image extracted from the camera. To avoid overfitting, they use DCNN models and millions of images to train. To solve the problem, the authors used SVM model for both the data augmentation method and dropping out. They use oriented fast and rotated brief (ORB) detection tools to detect key points of each image. It then determines the top N points based on the angular distance Harris. Currently, you only look once (YOLO) model [12] is being used to detect traffic that predicts based on the bounding boxes. In [13], the author uses the YOLOv3 model [14] in combination with the Lucas-Kanade method (LK) [15] to identify the vehicles in the region of interest (RoI) and calculate the speed of vehicles. Therefore, it is possible to determine the traffic status at urban intersections as illustrated in Figure 1.

In the Figure 1, RoI is selected to crop the entire image to improve processing speed and accuracy when recognizing images. The obtained RoI mask is detect based on a binary of original image. The vehicles in the RoI were detected using the YOLOv3 model. The four peaks of the bounding boxes obtained by YOLOv3 are optical stream inputs for vehicle speed tracking and calculation. Traffic status will be determined based on the travel speed of the vehicle. The algorithm indicates that if the rate is less than a specified threshold, it will be considered congested. However, the vehicle speed will be very low during the red-light waiting period, and thus it is difficult to distinguish the traffic jam. Therefore, the authors have chosen the signal light period to distinguish the continuous speed and determine the final traffic state. This method also achieves positive results when compared with kernel based fuzzy c-means clustering algorithm (KFCM) [16] and Bayes [17] algorithms. In the context of traffic in Vietnam, the method is not suitable in several cases such as passing a red light or moving vehicles earlier than the time to change the signal and it takes time to wait for one signal cycle to measure vehicle speed. Our recommendation system uses the YOLOv5 model to detect and calculate the number of vehicles on the RoI for higher accuracy than the YOLOv3 model. The problem of congestion identification is also made simpler by analyzing the variability of the obtained data after using YOLOv5.

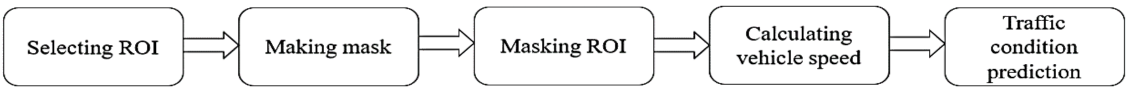

Figure 1. Schematic diagram of the method used by [13]

3. PROPOSAL SYSTEM

3.1. Overview

Currently, many traffic congestion avoidance routing systems have been deployed and shown good results such as Google Map [18], congestion prediction and navigation models based on dynamic traffic networks and balanced Markov chains [19], or a dynamic vehicle navigation system using positioning for mobile phones [20]. Instead of using GPS user positioning to collect data for congestion detection like the systems, our proposed system has the following points. In congestion prediction, we utilize live data from surveillance cameras at intersections. We then apply the YOLOv5 model to analyze the videos to detect and calculate and determine its status. In the routing part, we apply the A* algorithm to find the optimal path after removing the congestion points on the map. Figure 2 is the proposed system.

The overview of the proposed system will include two modules with four main functions. In the module 1 (***Traffic condition detection***) includes three parts, namely detecting and counting vehicles, and predicting traffic condition. Detecting vehicle will detect and classify vehicles. Counting vehicles will calculate the number of vehicles collected at the predefined RoI. Predicting traffic condition will identify traffic congestion based on the average number and the fluctuation of vehicles in the RoI. In the module 2 (***Routing***), the analyzed traffic status data at the intersections are then updated on the urban traffic map. It will then perform the algorithm to find the most optimal path and avoid going through congested nodes. The input to the system is videos extracted from cameras at traffic intersections and the system output is one or more suitable paths.

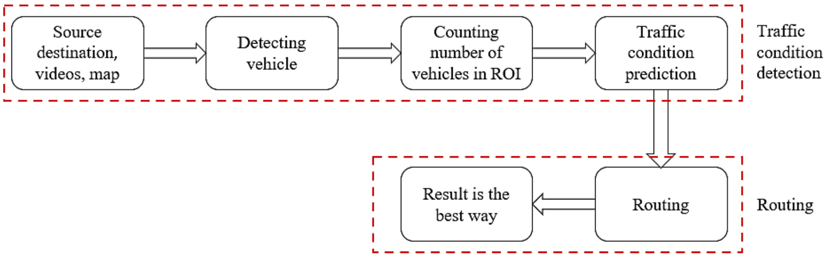

Figure 2. Diagram of the proposed system

3.2. Module 1: Traffic condition detection

3.2.1. Detecting vehicle

For the collecting data input, data for vehicle detection are long videos (20 seconds) extracted from cameras at intersections in the city with frame rate FPS = 30 frames/s and resolution 1280×720 pixels. The videos are divided into 3 main groups corresponding to three common traffic conditions: clear, slow, and congested to ensure the accuracy of the system. For the selecting model, the first goal of the algorithm is to detect and classify traffic from cameras on the streets. Therefore, real-time speed is the most important. We do not use R-CNN, Fast R-CNN, or Faster R-CNN models since they are not as good as YOLO models in term performance and real-time processing. We choose YOLOv5 due to its fast speed and better performance. YOLOv5 is developed from YOLOv4 [21] and SPP-NET for object detection.

YOLOv5 has four versions, namely YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x [22]. All four versions consider the detection speed and real-time performance. In detecting city traffic, performance is the most important issue. Therefore, we choose YOLOv5x [22]. It consists of 607 classes along with 88,568,234 parameters. The model uses the common object in context (COCO) dataset [23] and 80 classes for pre-training. Figure 3 shows parameter values for evaluation among YOLOv5 models on Github [24]. It can be seen that YOLOv5x balances the performance and the speed with an average accuracy (mAP) of 50.4 and a speed of 6.1 ms/image on the V100 GPU. The model perfectly fits the real-time traffic congestion detection problem.

For the counting vehicle, instead of counting the number of vehicles that appear in the entire video frame, we count the number of vehicles in a defined region called the RoI. Due to the influence of camera angle and distance, the number of vehicles obtained will vary greatly. When the camera is far and high, it will capture more cars than the camera with a close angle. Counting vehicles in the RoI both reduces the execution time and helps to define a threshold for the number of countable vehicles. In step 1, we create RoI area using rectangle function of OpenCV library with input coordinates. In step 2, vehicle counting is performed by checking the center of bounding box of object in the RoI area.

For the predicting traffic condition, the average number of vehicles is low in the normal state. Its average is high and the variability of the number of vehicles is very low in a congested state. When congestion occurs, vehicles move at a very slow speed, and thus the number of vehicles entering and leaving the RoI area in a short period is very little. Besides, the variation is almost zero. When complete congestion occurs, cars mostly do not move. The average volume of vehicles in the common traffic state will be between smooth and congested volumes with higher variability due to the inter-vehicle movement in the RoI area with slow traffic. Traffic condition is determined by two factors, namely the average number of vehicles per frame and variability (CV) of vehicles entering the RoI. The thresholds for the mean number of vehicles and the variability are set as Mε and CVε, respectively. These values will be determined as shown in Figure 4.

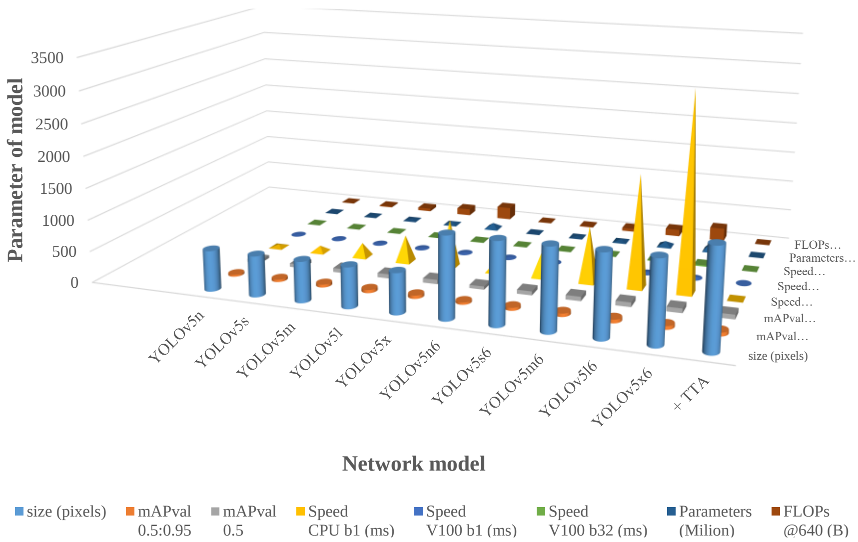

Figure 3. Training test scores of models on the COCO val2017 dataset

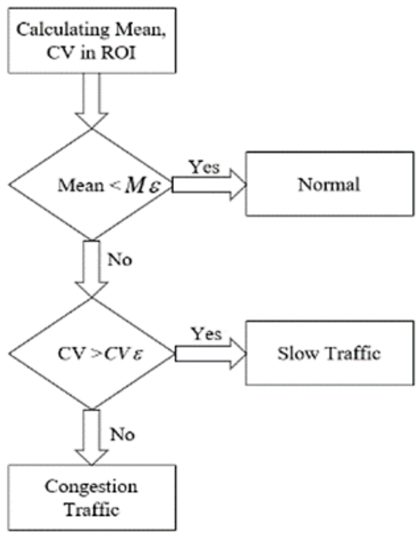

Figure 4. Flowchart of proposed traffic condition classification

For the average number of vehicles (mean), Video is a collection of many frames that appear consecutively, one after another. Assuming the input video of the system has n frames equivalent to n samples. We can count xi cars for each frame. The average number of cars per frame ($\bar{X}$) is calculated by (1),

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{i=n} x_i. \tag{1}$$

For the coefficient of variation (CV), the CV is used to determine the dispersion of data points to compare the volatility of datasets with different mean values. The CV is calculated as,

$$CV = \frac{\sigma}{\mu}, \tag{2}$$

where the standard deviation (σ) is calculated as,

$$\sigma = \sqrt{\frac{\sum_{i=1}^{i=m}(x_i-\bar{X})^2}{m-1}}, \tag{3}$$

where *m* is the points in a dataset. The average value (μ) has been calculated in (1).

4. SIMULATION AND RESULTS

4.1. Setup

The model is tested on three input datasets corresponding to three types of traffic conditions including clear, slow, and congested to determine the threshold values mean (average number of vehicles) and CV. A device used for simulation is Google Colab 12GB NVIDIA Tesla K80 GPU. The data used for network training was recorded at the intersections of Hanoi city, Vietnam (Xa Dan - Pham Ngoc Thach, Pho Hue - Nguyen Du, Le Thanh Nghi - Tran Dai Nghia streets) with resolution 1280×720 resolution and 30 FPS frame rate in both day and night conditions. The experimental parameters used in the training phase of the network are shown in Table 1. We get the vehicle dataset by intercepting each frame of the video captured and dividing them into rates 7:3 including 6926 images (4896 for training and 2030 for testing).

4.2. Collect data

Each dataset consists of two representative videos with the parameters as shown in Table 2. During the testing process, we found that executing the program with 500 ~ 600 frames will take a long time due to using the YOLOv5x model. Therefore, the program performs detection and counts the number of vehicles with 10 new frames. This reduces execution time without greatly affecting efficiency since traffic status is nothing to change for 10 frames (0.33 seconds).

Table 1. Input data parameters

No.	Parameter	Value
1	Batch size	16
2	Resizing input image	640×640
3	Weights	YOLOx.pt
4	Epoch	300

Table 2. Evaluating parameters

No.	Parameter	Value
1	Time	20 seconds
2	Frame rate	25~30 frames/s
3	Resolution	1280 ×720
4	Total frames	500~600

4.3. Results

After running the test of the traffic detection module, we achieved several results. Calculation results on average vehicle amounts, variability coefficients, and execution time of the traffic counting process in the RoI area are given in Table 3. The result of the accuracy of the YOLOv5 model in detecting objects is relatively high in two types of normal and slow traffic. The accuracy of the model is relatively low with congestion traffic. YOLOv5 ignores several objects when they are adjacent or are partially obscured. We suggest to change the higher camera rotation angle and pre-train the YOLOV5 model with datasets of vehicles in Vietnam to solve this issue. Figure 5 shows the number of cars in the RoI.

In Figure 5, the diagram shows the vehicle traffic in the RoI area over time. The number of vehicles remains low as shown in Figure 5(a) for normal traffic (Video1). The number of vehicles has a large variation and the number of vehicles reached over 18 vehicles in the middle range. It has less than 10 cars at the first and end period as shown in Figure 5(b) for slow traffic (Video3). It has high vehicles and maintains quite uniformly between 13 and 15 vehicles as shown in Figure 5(c) for congestion traffic (Video5).

Table 3. Evaluate the parameters for testing with three types of traffic

Video		Mean	CV	Processing time (second)
Normal	Video 1	6.867	0.302	26.294
	Video 2	2.521	0.511	18.506
Slow traffic	Video 3	11.410	0.292	24.029
	Video 4	15.951	0.350	25.762
Traffic congestion	Video 5	13.738	0.133	25.741
	Video 6	17.60	0.126	24.966

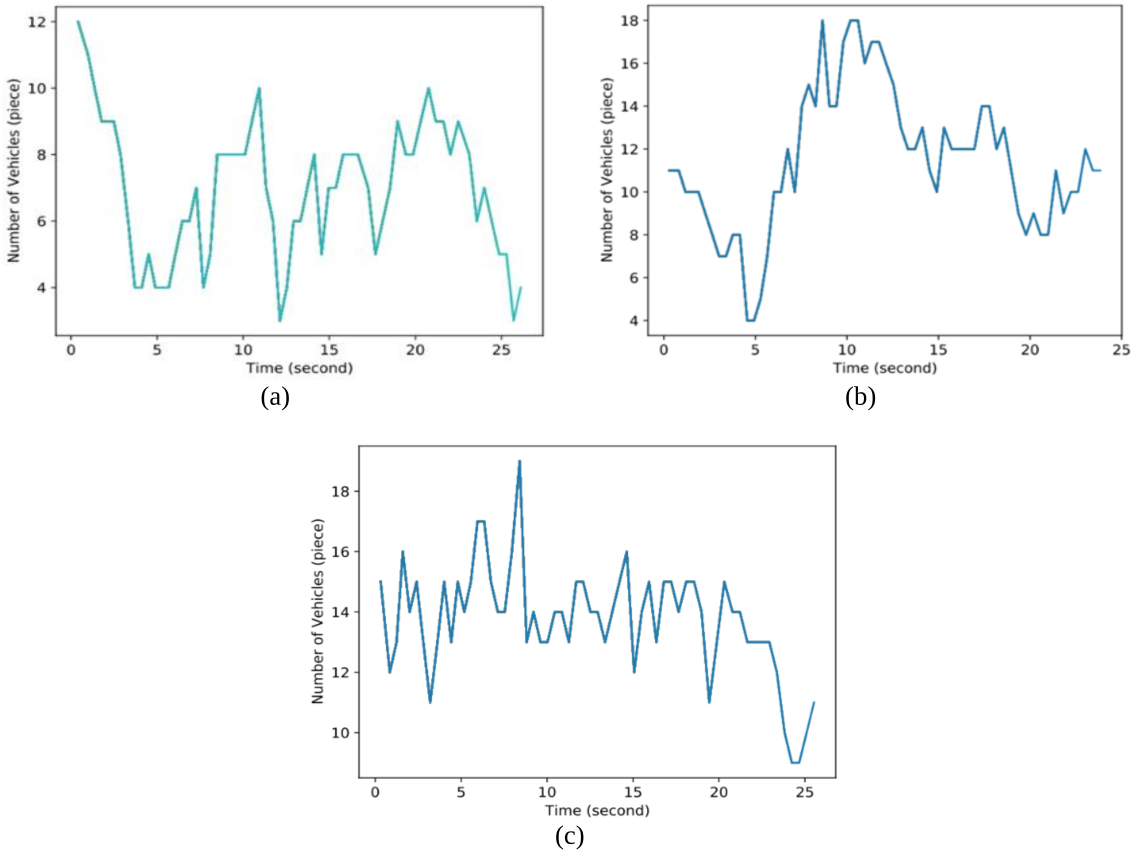

Figure 5. Result of vehicle traffic through the RoI area for; (a) video 1, (b) video 3, and (c) video 5

4.4. Select threshold values

Based on the calculation results, we choose the threshold value Mε = 10 (average number of vehicles/frame) and CVε = 0.2. The process of determining this threshold value to be most accurate one needs to be performed on many input videos with different camera angles and the way to choose a reasonable RoI area. The results shown in Table 4 reveal that the accuracy level for the input data is relative and there are still errors. The error occurs in videos whose parameters are close to the threshold value. It is also important to improve the accuracy of the YOLOv5 model in object detection since this directly affects the selection of threshold values. Table 5 compares between our proposed model and the CNN, PredNet, DCNN, and SVM models in term of the accuracy that have been given in detecting traffic congestions from videos and images. In Table 5, we find that the image classification method using the PredNet model [25] gives the lowest accuracy (88.3%), followed by SVM, CNN, and DCNN. Our proposed model uses YOLO for the highest accuracy in traffic state detection, but there is a trade-off in speed as frame-by-frame processing time is higher than previous models used with YOLOv5.

Table 4. Evaluate the parameters for testing with three types of traffic

Video		Mean	CV	Processing time (second)	Traffic status	Results	Average accuracy (%)
Type 1	Video 1	2.590	0.515	22.790	Normal	True	91.67%
	Video 2	2.583	0.829	22.872	Normal	True	
	Video 3	0.885	0.925	24.751	Normal	True	
	Video 4	1.393	0.684	25.893	Normal	True	
Type 2	Video 1	20.129	0.260	24.405	Slow traffic	True	
	Video 2	40.393	0.088	24.333	Traffic congestion	False	
	Video 3	14.295	0.223	25.778	Slow traffic	True	
	Video 4	13.647	0.256	21.636	Slow traffic	True	
Type 3	Video 1	16.450	0.180	25.383	Traffic congestion	True	
	Video 2	33.355	0.179	27.016	Traffic congestion	True	
	Video 3	33.295	0.127	24.707	Traffic congestion	True	
	Video 4	37.672	0.085	23.545	Traffic congestion	True	

Table 5. Comparing the accuracy among models

Model	Accuracy (%)	Processing speed (fps)
CNN [5]	89.50	-
DCNN [11]	90.20	100
SVM [11]	85.20	300
PredNet (LTSM & CNN) [25]	88.30	-
YOLOv1 [12]	91.20	100
Our proposal (using YOLOv5)	91.67	25

5. CONCLUSION

The main purpose of this work is to build an application that suggests appropriate routes/ways in urban traffic. It is worth noticed that this paper mainly focuses on traffic situation awareness for the routing. A new model, namely YOLOv5, is utilized to detect vehicles and then determine traffic conditions based on videos extracted from traffic cameras. Besides, we use the vehicle dataset collected in Vietnam to retrain the YOLOv5 model to improve the detection performance in real applications. In the future, we will take the steps to improve accuracy of the YOLOv5 model which can be deployed on Web/App platforms for real world applications.

ACKNOWLEDGEMENTS

This research is funded by Vietnam National University Ho Chi Minh City (VNU-HCM) under grant number NCM2021-20-02.

REFERENCES

[1] H. Mai-Tan, H. N. Pham-Nguyen, N. X. Long, and Q. T. Minh, “Mining Urban Traffic Condition from Crowd-Sourced Data,” *SN Computer Science*, vol. 1, no. 4, 2020, doi: 10.1007/s42979-020-00244-6.

[2] Q. T. Minh, E. Kamioka, and S. Yamada, “CFC-ITS: Context-Aware Fog Computing for Intelligent Transportation Systems,” *IT Professional*, vol. 20, no. 6, pp. 35–44, 2018, doi: 10.1109/MITP.2018.2876978.

[3] H. M. Tan, H. N. Pham-Nguyen, Q. T. Minh, and P. Nguyen Huu, “Traffic Condition Estimation Based on Historical Data Analysis,” *ICCE 2020 - 2020 IEEE 8th International Conference on Communications and Electronics*, pp. 256–261, 2021, doi: 10.1109/ICCE48956.2021.9352107.

[4] Q. Tran Minh, H. N. Pham-Nguyen, H. Mai Tan, and N. Xuan Long, “Traffic Congestion Estimation Based on Crowd-Sourced Data,” *Proceedings - 2019 International Conference on Advanced Computing and Applications, ACOMP 2019*, pp. 119–126, 2019, doi: 10.1109/ACOMP.2019.00026.

[5] K. Li and L. Cao, “A review of object detection techniques,” *Proceedings - 2020 5th International Conference on Electromechanical Control Technology and Transportation, ICECTT 2020*, pp. 385–390, 2020, doi: 10.1109/ICECTT50890.2020.00091.

[6] N. Aburaed, A. Panthakkan, M. Al-Saad, S. A. Amin, and W. Mansoor, “Deep Convolutional Neural Network (DCNN) for Skin Cancer Classification,” *ICECS 2020 - 27th IEEE International Conference on Electronics, Circuits and Systems, Proceedings*, 2020, doi: 10.1109/ICECS49266.2020.9294814.

[7] R. Girshick, “Fast R-CNN,” *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2015 International Conference on Computer Vision, ICCV 2015, pp. 1440–1448, 2015, doi: 10.1109/ICCV.2015.169.

[8] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017, doi: 10.1109/TPAMI.2016.2577031.

[9] J. R. R. Uijlings, K. E. A. Van De Sande, T. Gevers, and A. W. M. Smeulders, “Selective search for object recognition,” *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013, doi: 10.1007/s11263-013-0620-5.

[10] J. Kurniawan, C. K. Dewa, and Afiahayati, “Traffic Congestion Detection: Learning from CCTV Monitoring Images using Convolutional Neural Network,” *Procedia Computer Science*, vol. 144, pp. 291–297, 2018, doi: 10.1016/j.procs.2018.10.530.

[11] P. Chakraborty, Y. O. Adu-Gyamfi, S. Poddar, V. Ahsani, A. Sharma, and S. Sarkar, “Traffic Congestion Detection from Camera Images using Deep Convolution Neural Networks,” *Transportation Research Record*, vol. 2672, no. 45, pp. 222–231, 2018, doi: 10.1177/0361198118777631.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December, pp. 779–788, 2016, doi: 10.1109/CVPR.2016.91.

[13] X. Yang, F. Wang, Z. Bai, F. Xun, Y. Zhang, and X. Zhao, “Deep learning-based congestion detection at urban intersections,” *Sensors*, vol. 21, no. 6, pp. 1–14, 2021, doi: 10.3390/s21062052.

[14] J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement,” 2018, [Online]. Available: http://arxiv.org/abs/1804.02767.

[15] A. Ranjan and M. J. Black, “Optical flow estimation using a spatial pyramid network,” *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 2720–2729, 2017, doi: 10.1109/CVPR.2017.291.

[16] Z. Z. Y. Liu, F. Liu, T. Hou, “Fuzzy C-means clustering algorithm to optimize kernel parameters,” *J. Jilin Univ*, vol. 46, pp. 246–251, 2016, doi: 10.1109/ICCIMA.2003.1238099.

[17] S. Wang, W. Huang, and H. K. Lo, “Traffic parameters estimation for signalized intersections based on combined shockwave analysis and Bayesian Network,” *Transportation Research Part C: Emerging Technologies*, vol. 104, pp. 22–37, 2019, doi: 10.1016/j.trc.2019.04.023.

[18] J. Cui and X. Wang, “Research on Google map algorithm and implementation,” *Journal of Information and Computational Science*, vol. 5, no. 3, pp. 1191–1200, 2008.

[19] Y. Zheng, Y. Li, C. M. Own, Z. Meng, and M. Gao, “Real-time predication and navigation on traffic congestion model with equilibrium Markov chain,” *International Journal of Distributed Sensor Networks*, vol. 14, no. 4, 2018, doi: 10.1177/1550147718769784.

[20] A. Shahzada and K. Askar, "Dynamic vehicle navigation: An A* algorithm based approach using traffic and road information," *ICCAIE 2011 - 2011 IEEE Conference on Computer Applications and Industrial Electronics*, pp. 514–518, 2011, doi: 10.1109/ICCAIE.2011.6162189.

[21] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," 2020, [Online]. Available: http://arxiv.org/abs/2004.10934.

[22] G. Jocher, A. Stoken, J. Borovec, and *et al*., "Ultralytics/YOLOv5: v5.0 - YOLOv5-P6 1280 models, AWS, supervisely and youtube integrations," 2021, doi: 10.5281/zenodo.4679653.

[23] T. Y. Lin *et al.*, "Microsoft COCO: Common objects in context," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8693 LNCS, no. PART 5, pp. 740–755, 2014, doi: 10.1007/978-3-319-10602-1_48.

[24] G. Jocher, "ultralytics / YOLOv5," 2021, [Online]. Available: https://github.com/ultralytics/YOLOv5.

[25] R. P. N. Rao and D. H. Ballard, "Predictive coding in the visual cortex: A functional interpretation of some extra-classical receptive-field effects," *Nature Neuroscience*, vol. 2, no. 1, pp. 79–87, 1999, doi: 10.1038/4580.

BIOGRAPHIES OF AUTHORS

Phat Nguyen Huu received his B.E. (2003), and M.S. (2005) degrees in Electronics and Telecommunications at Hanoi University of Science and Technology (HUST), Vietnam, and a Ph.D. degree (2012) in Computer Science at Shibaura Institute of Technology, Japan. Currently, he lecturer at the HUST Vietnam. His research interests include digital image and video processing, wireless networks, ad hoc, and sensor networks, intelligent traffic systems (ITS), and the internet of things (IoT). He received the best conference paper award in SoftCOM (2011), the best student grant award in APNOMS (2011), and the hisayoshi yanai honorary award by Shibaura Institute of Technology, Japan in 2012. He can be contacted at email: phat.nguyenhuu@hust.edu.vn.

Phuong Tong Thi Quynh is a student of Electronics and Telecommunications at Hanoi University of Science and Technology (HUST), Vietnam. Currently, she is working in Sanslab at HUST. Her main is to develop smart products which relate to digital images, video processing, and machine learning. She can be contacted at email: phuong.ttq172756@sis.hust.edu.vn.

Thien Pham Ngoc is a student of Electronics and Telecommunications at Hanoi University of Science and Technology (HUST), Vietnam. Currently, he is working in Sanslab at HUST. His main is to develop smart products which relate to digital images, video processing, machine learning, embedded system, and the internet of things (IoT). He can be contacted at email: thien.pn172833@sis.hust.edu.vn.

Quang Tran Minh is an associate professor at the Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology, Vietnam, and a visiting researcher at Shibaura Institute of Technology, Tokyo, Japan. He has been a researcher at the Network Design Department, KDDI Research Inc., Japan (2014-2015), and a researcher at the Principles of Informatics Research Division, National Institute of Informatics (NII), Japan (2012-2014). His research interests include mobile and ubiquitous computing, IoT, network design and traffic analysis, disaster recovery systems, data mining, and ITS systems. Prof. Quang received his Ph.D. in Functional Control Systems from the Shibaura Institute of Technology. He is a member of IEEE and ACM. He can be contacted at email: quangtran@hcmut.edu.vn.