Proposing a route recommendation system for vehicles based on receiving video

Phat Nguyen Huu[*], Phuong Tong Thi Quynh[*], Thien Pham Ngoc[*], and Quang Tran Minh[+, ++]
[*]School of Electrical and Electronics Engineering, Hanoi University of Science and Technology (HUST), Vietnam
[+]Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology (HCMUT),
268 Ly Thuong Kiet, Dist.10, Ho Chi Minh City, Vietnam
[++]Vietnam National University Ho Chi Minh City (VNU-HCM), Linh Trung Ward,
Thu Duc District, Ho Chi Minh City, Vietnam

Article Info

Article history:

Received Nov. 25, 2021
Revised
Accepted

Keywords:

Region of interest
Vehicle detection
YOLOv5
Algorithm A*
Recommendation system

ABSTRACT

This paper proposes a method to classify traffic status for the route recommendation system based on received videos. The system will count 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 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 at intersections. This result outperforms other state-of-the-art methods using YOLOv1, DCNN, CNN, and SVM models whose accuracies are 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 demands for development in developing countries such as Vietnam and ASEAN countries lead to the increase of population and traffic and transportation demands, 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, etc. 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 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 HoChiMinh 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. However, it will take a lot of time to aggregate data 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 above 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 system that we propose has the following new features:

- Firstly, we use the YOLOv5 model based on [5], [6], [7], which is a new model for vehicle detection and traffic status determination based on videos extracted from cameras at intersections.
- Second, we use a vehicle dataset collected in Vietnam to retrain the YOLOv5 model to improve detection performance in real-world applications.

The goal of the paper is to build a real-time system 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 and is organized as follows. Section II presents several related works. Section III presents the proposed model. Section IV will evaluate the proposed model and analyze the results. In the final section, we give conclusions and future research directions.

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, vehicle density, calculating the area occupied by vehicles on the road, classifying images from surveillance cameras, etc. Supporting technologies in this process include CNN (Convolutional Neural Network) models such as DCNN [8], R-CNN [5], Fast R-CNN [10], Faster R-CNN [11]. The above models have been proposed and achieved many positive results when applied in traffic congestion detection. All of these models use a region-based convolutional neural network (CNN), which uses a technique called selective search [12] to select a small number of best-matching regions (candidate regions) among possible regions. In [5], the R-CNN model is relatively slower among the three models since it generates many candidate regions. In [10], the Fast R-CNN model suggested a less number of candidate regions. However, the selective search algorithm that this model uses is not a machine learning algorithm and thus it cannot learn from the context and often suggest a bad candidate region. In [11], Faster R-CNN has the fastest running time compared to R-CNN and Fast R-CNN. However, it is still not fast enough to detect objects in real-time.

In [13], an intelligent traffic congestion detection method is introduced by leveraging image classification methods. The CNN model was trained for binary classification of road traffic conditions using 1000 CCTV surveillance images with the balanced distribution. The authors just resized and converted the 100×100 grayscale images and did not use any manual features in the preprocessing step. This model is proposed to be deployed in a future congestion detection system using CCTV cameras that record images on specific locations in real-time.

In [14], the authors use a support vector machine (SVM) and two different deep learning techniques such as you only look once (YOLO), DCNN to compare the accuracy in classifying congestion images from surveillance cameras. The entire image extracted from the camera can be classified as either congested or non-congested. DCNN models are computationally expensive and often require millions of images to train the model to avoid overfitting. To solve this, the authors used both the data augmentation method and dropping out. The SVM model is one of the widely used shallow algorithms for image classification tasks. The author used oriented features from accelerated segment test (FAST) and Rotated BRIEF (ORB) feature detection tools to detect key points in each image, and thus the FAST algorithm was used to extract the key points and the angular distance Harris is used to determining the top *N* points. Currently, the YOLO model [15] is being used to detect traffic. YOLO uses a convolutional neural network that predicts the bounding boxes as well as the class for them instead of using the region selection method. In [14], YOLO is the other model applied for the general purpose of congestion detection and localization from CCTV video feeds.

In [16], the author uses the YOLOv3 model [17] in combination with the Lucas-Kanade optical flow method (LK) [18] to identify the vehicles in the ROI and calculate the speed of vehicles. Therefore, it is possible to determine the traffic status at urban intersections as illustrated in Fig. 1.

Fig. 1 Schematic diagram of the method used by [16]

In the above system, ROI is selected to crop the entire image to improve processing speed and accuracy when recognizing images. The obtained ROI mask is based on the selected ROI and it is a binary of the same size as the 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 while comparing with KFCM [19] and Bayes [20] algorithms. In the context of traffic in Vietnam, the above 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 vehicles and count the number of vehicles in 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.

3. PROPOSAL SYSTEM

3.1. Overview

Currently, many traffic congestion avoidance routing systems have been deployed and shown good results such as Google Map [21], congestion prediction and navigation models based on dynamic traffic networks and balanced Markov chains [22], or a dynamic vehicle navigation system using positioning for mobile phones [23].

Instead of using GPS user positioning to collect data for congestion detection like the above 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, count the number of vehicles, and determine the traffic status.
- In the routing part, we use the A* algorithm to find the optimal path after removing the congestion points on the map.

Figure 2 is the block diagram of the proposed system.

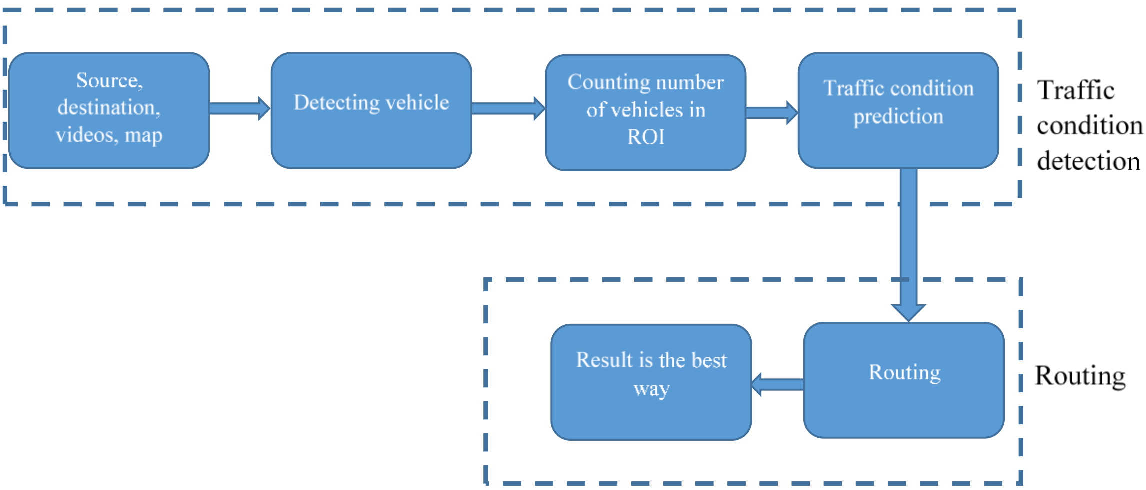

Fig. 2 Diagram of the proposed system

Overview of the proposed system will include two modules with four main functions as follows:

- ***Module 1: Traffic condition detection***
    - Detecting vehicle: It will detect and classify vehicles

- Counting vehicle: It will count the number of vehicles in a frame collected at the predefined ROI
- Predicting traffic condition: It will identify traffic congestion or traffic status based on the average number of vehicles and the fluctuation of vehicle volume in the ROI.

- ***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.

3.2. Module 1: Traffic condition detection

3.2.1. Detecting vehicle

3.2.1.1. *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.

3.2.1.2. *Selecting model*

The first goal of the system is to detect and classify traffic from cameras on streets. Therefore, real-time speed is the most important. We do not use R-CNN, Fast R-CNN, Faster R-CNN models since they cannot compete with YOLO models in both performance and real-time response. We choose YOLOv5 due to its fast speed and better performance. YOLOv5 inherits the advantages of YOLOv4 [24] and adds SPP-NET [25] along with several advanced techniques. It has become a new technology in object detection.

The four versions of YOLOv5 are YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x [26]. YOLOv5s is the lightest model and YOLOv5x is the heaviest one. All four versions have a trade-off between detection speed and real-time performance. The differences between these versions are the number of feature extraction modules and the convolution kernel at the specific locations of the network. The network consists of three networks, namely backbone, neck, and detect networks. Backbone is a convolutional neural network (CNN) for image detail synthesis and feature extraction. The neck is responsible for combining the features of the image and transmitting the feature map to the detect network. The detect network is responsible for object detection and classification based on the use of anchor boxes on the feature map. It contains a softmax class that predicts the probability of the class that the bounding box surrounds the object.

In detecting city traffic, performance is the most important issue. Therefore, we choose YOLOv5x [26]. It contains 607 classes along with 88,568,234 trainable parameters. The model is pre-trained using the common object in context (COCO) dataset [27] to detect 80 classes.

Figure 3 shows parameter values for evaluation among YOLOv5 models in Github [28]. 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.

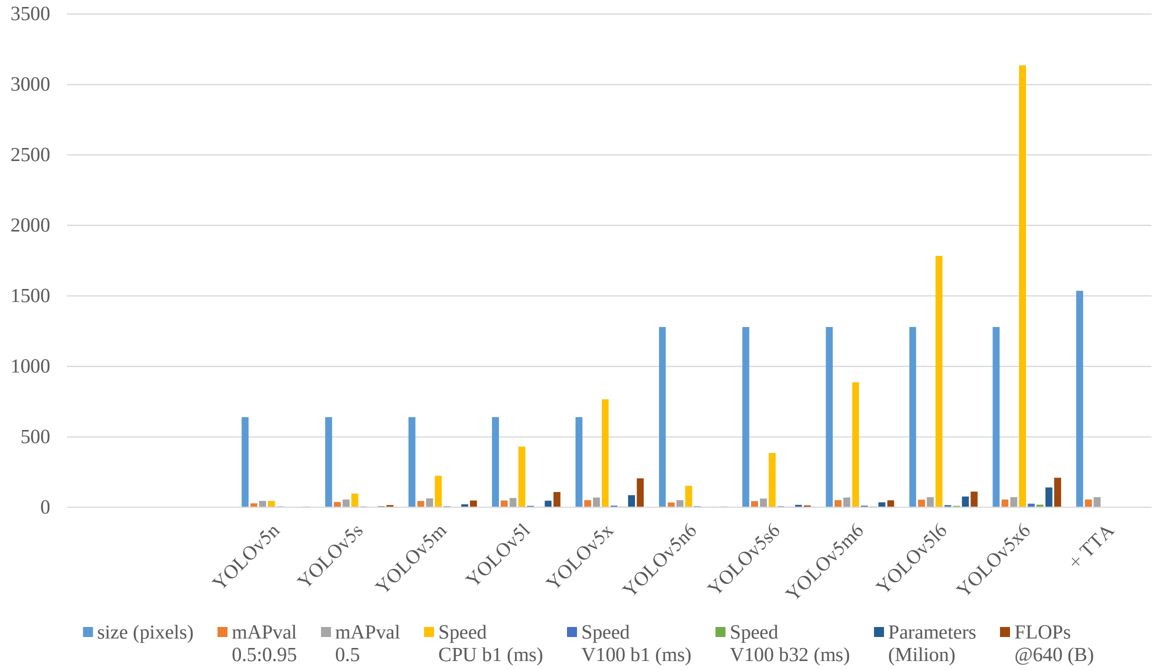

Fig. 3 Training test scores of models on the COCO val2017 dataset

3.2.2. 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 region of interest (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.

The steps are as follows:

- Step 1: Creating ROI area using rectangle function of OpenCV library with input coordinates of ROI area
- Step 2: Vehicle counting is performed by checking whether the center of the bounding box of object coordinates is in the ROI area.

3.2.2.1. *Predicting traffic condition*

Normally, the average number of vehicles is low in the normal state. The average number of vehicles 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:

- ***The average number of vehicles (mean)*** per frame in the ROI area
- ***Variability (CV)*** of vehicles entering the ROI

The thresholds for the mean number of vehicles and the variability, respectively, are set as Mε and CVε. These values will be determined after many tests. The method of showing traffic status is designed in Fig. 4.

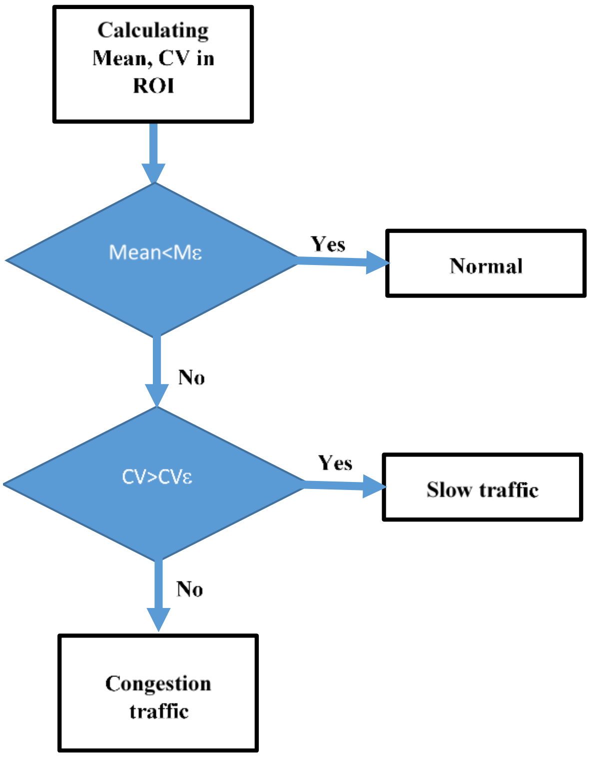

Fig. 4 Flowchart of proposed traffic condition classification

3.2.2.2. *The average number of vehicles (Mean)*

Video is a collection of many still images (frames) that appear consecutively, one after another. Assuming the input video of the system has a frame equivalent to the sample, each frame we can count cars. The average number of cars per frame is calculated by the following formula:

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

where n is the number of cars.

3.2.2.3. *Coefficient of Variation (CV)*

In probability theory and statistics, the coefficient of variation (CV) is a descriptive statistical quantity that is used to measure the dispersion of data points in a dataset of data around the mean. This coefficient is used to compare the volatility of datasets with different mean values.
The coefficient of variation is calculated as the ratio between the standard deviation and the mean as

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

where the standard deviation is calculated as follows:

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

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

4. SIMULATION AND RESULTS

After running the test of the traffic detection module, we achieved several results as follows.

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.

Device used for simulation is computer Intel(R) Core(TM) i3-6100U CPU @ 2.30GHz, RAM 8GB, and operating system Ubuntu 20.04.

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, 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 above and dividing them into rates 7:3 including 6926 images (4896 for training and 2030 for testing).

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

4.1.1. *Collect data*

Each dataset consists of two representative videos with the parameters as shown in Tab. 2.

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

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 counting 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).

4.2. *Results*

Calculation results on average vehicle amounts, variability coefficients, and execution time of the traffic counting process in the ROI area are given in Tab. 3.

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

Video		Mean	CV	Processing time (second)
Normal	Video 1	10.051	0.252	226.939
	Video 2	7.033	0.306	341.878
Slow traffic	Video 3	20.275	0.268	293.155
	Video 4	18.131	0.363	363.789
Traffic congestion	Video 5	23.300	0.106	344.508
	Video 6	20.967	0.126	394.756

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 two solutions to this issue:

- We will change higher camera rotation angle since the objects do not hide.
- We will pre-train the YOLOV5 model with datasets of vehicles in Vietnam.

Charts that show the number of cars counted in the ROI area is shown in Figs. 5, 6, and 7.

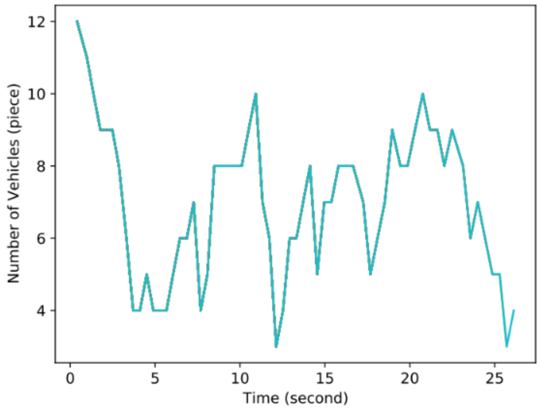

Fig. 5 Result of vehicle traffic through the ROI area of the video 1

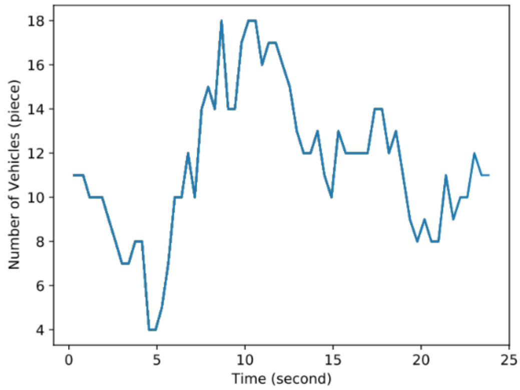

Fig. 6 Result of vehicle traffic through the ROI area of the video 3

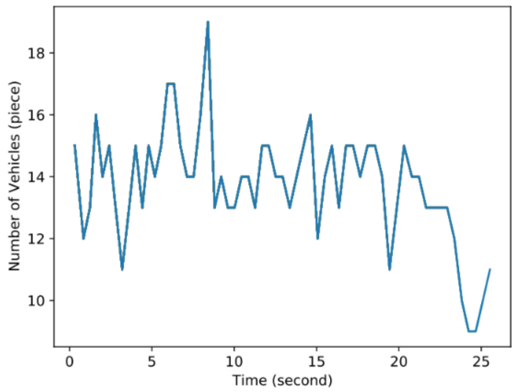

Fig. 7 Result of vehicle traffic through the ROI area of the video 5

The diagram shows the number of vehicles passing through the ROI area over time to show:

- Normal traffic (Video1): The amount of vehicles remains low.
- Slow traffic (Video3): The number of vehicles has a large variation and the number of vehicles reached over 30 vehicles at the middle range. It has less than 15 cars at the first and end period.
- Congestion traffic (video 5): It has high vehicles and maintains quite uniformly between 20 and 27 vehicles.

4.3. Select threshold values

Based on the above calculation results, we choose the threshold value Mε (average number of vehicles/frame) = 15 and CVε = 0.2. The process of determining this threshold value to be most accurate needs to be performed on many input videos with different camera angles and how to choose a reasonable ROI area.

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	

The results shown in Tab. 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 below shows the comparison of the accuracy between our proposed model and the CNN, PredNet, DCNN, and SVM models that have been given in detecting traffic congestions from videos and images.

Table 5. Comparing the accuracy among models

Model	Accuracy (%)	Processing time per image (second)
CNN [5]	89.50	-
DCNN [14]	90.20	0.01
SVM [14]	85.20	0.03
PredNet (LTSM & CNN) [29]	88.30	-
Our proposal (using YOLOv5)	**91.67**	**0.161**
Our proposal (using YOLOv1)	**91.20**	**0.01**

In Tab. 5, we find that the image classification method using the PredNet model [29] gives the lowest accuracy (88.3%), followed by SVM, CNN, 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.

5. CONCLUSION

The purpose of the project is to study the construction of an application that suggests the way in urban traffic. However, this paper only focuses on traffic situation awareness. In the future, we will take the steps based on [29], [31], [32] to:

- Improve accuracy of YOLOv5 model,
- Research the stability and accuracy of the system,
- Implement the routing module,
- Research and development system on Web/App platform.

REFERENCES

[1] M.T Ha, P.N. Hoang-Nam, N.X. Long, T.M. Quang, “Mining Urban Traffic Condition From Crowd-Sourced Data,” SN COMPUT., vol.1, no. 225, 16 pages, 2020.

[2] T.M. Quang, E. Kamioka, S. Yamada, “CFC-ITS: Context-Aware Fog Computing for Intelligent Transportation Systems,” IEEE IT Professional, vol. 20, no. 6, pp. 35-45, 2018.

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

[4] Q.T. Minh, H.N. Pham-Nguyen, H.M. Tan, and N.X. Long, “Traffic Congestion Estimation Based on Crowd-Sourced Data,” 2019 International Conference on Advanced Computing and Applications (ACOMP), 2019, pp. 119-126. DOI: 10.1109/ACOMP.2019.00026.

[5] K. Li and L. Cao, “A Review of Object Detection Techniques,” 2020 5th International Conference on Electromechanical Control Technology and Transportation (ICECTT), 2020, pp. 385-390. DOI: 10.1109/ICECTT50890.2020.00091.

[6] J. Tao, H. Wang, X. Zhang, X. Li, and H. Yang, “An object detection system based on YOLO in traffic scene,” 2017 6th International Conference on Computer Science and Network Technology (ICCSNT), 2017, pp. 315-319. DOI: 10.1109/ICCSNT.2017.8343709.

[7] P. Adarsh, P. Rathi, and M. Kumar, “YOLO v3-Tiny: Object Detection and Recognition using one stage improved model,” 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), 2020, pp. 687-694. DOI: 10.1109/ICACCS48705.2020.9074315.

[8] N. Aburaed, A. Panthakkan, M. Al-Saad, S. A. Amin and W. Mansoor, “Deep Convolutional Neural Network (DCNN) for Skin Cancer Classification,” 2020 27th IEEE International Conference on Electronics, Circuits and Systems (ICECS), 2020, pp. 1-4. DOI: 10.1109/ICECS49266.2020.9294814.

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation,” 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580-587. DOI: 10.1109/CVPR.2014.81.

[10] R. Girshick, “Fast R-CNN,” 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1440-1448. DOI: 10.1109/ICCV.2015.169.

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

[12] J.R.R. Uijlings, K.E.A. van de Sande, T. Gevers, et al., “Selective Search for Object Recognition,” Int J. Comput Vision, vol. 104, pp. 154–171, 2013. DOI: 10.1007/s11263-013-0620-5

[13] J. Kurniawan, S.G.S. Syahra, C.K. Dewa, Afiahayati, “Traffic Congestion Detection: Learning from CCTV Monitoring Images using Convolutional Neural Network,” Procedia Computer Science, vol. 144, 2018, pp. 291-297, 2018. DOI: 10.1016/j.procs.2018.10.530.

[14] P. Chakraborty, Y. 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, pp. 222 – 231, 2018. DOI: 10.1177/0361198118777631.

[15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788. DOI: 10.1109/CVPR.2016.91.

[16] X Yang, F. Wang, Z. Bai, F. Xun, Y. Zhang, X. Zhao, “Deep Learning-Based Congestion Detection at Urban Intersections,” Sensors, vol. 21, no. 6, pp. 20-52, 2021. DOI: 10.3390/s21062052.

[17] J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement,” ArXiv, abs/1804.02767, 2018.

[18] A. Ranjan and M. J. Black, “Optical Flow Estimation Using a Spatial Pyramid Network,” 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2720-2729. DOI: 10.1109/CVPR.2017.291.

[19] Y. Liu, F. Liu, T. Hou, Z. Zhang, “Fuzzy C-means clustering algorithm to optimize kernel parameters,” J. Jilin Univ, vol. 46, pp. 246–251, 2016.

[20] S. Wang, W. Huang, 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.

[21] C. Jin, “Research on Google Map Algorithm and Implementation,” Computer Science, vol. 5, no. 3, pp. 1191-1200, 2007.

[22] Y. Zheng, Y. Li, C-M Own, Z. Meng, 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, pp. 1-13, April 2018. DOI: 10.1177/1550147718769784.

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

[24] A. Bochkovskiy, C. Wang, and H.M. Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection,” Computer Vision and Pattern Recognition, ArXiv, abs/2004.10934, pp. 1-17, 2020.

[25] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 9, pp. 1904-1916, 2015. DOI: 10.1109/TPAMI.2015.2389824.

[26] G. Jocher et al., “Ultralytics/YOLOv5: v5.0 - YOLOv5-P6 1280 models, AWS, supervisely and youtube integrations,” version v5.0, April 2021. DOI: 10.5281/zenodo.4679653.

[27] T.Y. Lin et al., “Microsoft COCO: common objects in context,” in Computer Vision – ECCV, 2014, pp. 740–755. . DOI: 10.1007/978-3-319-10602-1_48.

[28] G. Jocher, “ultralytics / YOLOv5,” 2021. Online: https://github.com/ultralytics/YOLOv5.

[29] R. Rao, D. Ballard, “Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects,” Nat Neurosci, vol. 2, pp. 79–87, 1999. DOI: 10.1038/4580.

[30] C. T. Phan, D. D. Pham, H. V. Tran, T. V. Tran, and P. N. Huu, "Applying the IoT platform and green wave theory to control intelligent traffic lights system for urban areas in Vietnam," KSII Transactions on Internet and Information Systems, vol. 13, no. 1, pp. 34-51, 2019. DOI: 10.3837/tiis.2019.01.003.

[31] N.H. Phat and V.Q. Cuong, “Proposing WPOD-NET combining SVM system for detecting car number plate,” IAES International Journal of Artificial Intelligence (IJ-AI), vol. 10, pp.652-657, 2021. DOI: 10.11591/ijai.v10.i3.pp657-665.

[32] Minh Q.T., Huu P.N., Tsuchiya T., Toulouse M., “Openness in Fog computing for the Internet of Things,” In Future Data and Security Engineering (FDSE), vol 11814, pp. 343-357, 2019. DOI: 10.1007/978-3-030-35653-8_23

BIOGRAPHIES OF AUTHORS (10 PT)

	**Phat Nguyen Huu** received his B.E. (2003), M.S. (2005) degrees in Electronics and Telecommunications at Hanoi University of Science and Technology (HUST), Vietnam, and Ph.D. degree (2012) in Computer Science at Shibaura Institute of Technology, Japan. Currently, he lecturer at the School of Electronics and Telecommunications, HUST Vietnam. His research interests include digital image and video processing, wireless networks, ad hoc, and sensor network, and intelligent traffic system (ITS), and the internet of things (IoT). He received the best conference paper award in SoftCOM (2011), best student grant award in APNOMS (2011), hisayoshi yanai honorary award by Shibaura Institute of Technology, Japan in 2012.
	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 developing smart products which relate to a digital images, video processing, and machine learning.
	**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 developing smart products which relate to the digital image, video processing, machine learning, embedded system, and the internet of things (IoT).
	**Quang Tran Minh** (quangtran@hcmut.edu.vn) is an associate professor at the Faculty of Computer Science and Engineering, Hochiminh 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 Shibaura Institute of Technology. He is a member of IEEE, ACM.