

RESEARCH ON TRAFFIC CONGESTION DETECTION FROM CAMERA IMAGES IN A LOCATION OF DA LAT

Nguyen Thi Luong^{a*}

^aThe Faculty of Information Technology, Dalat University, Lam Dong, Vietnam

**Corresponding author: Email: luongnt@dlu.edu.vn*

Article history

Received: March 28th, 2021

Received in revised form (1st): May 29th, 2021 | Received in revised form (2nd): June 30th, 2021

Accepted: July 12th, 2021

Available online: October 4th, 2021

Abstract

Many researchers are interested in traffic congestion detection and prediction. Traffic congestion occurs increasingly in many cities in Vietnam, including the city of Da Lat. This paper focuses on SVM, CNN, DenseNet, VGG, and ResNet models to detect traffic congestion from camera images collected at Nga 5 Dai Hoc, Da Lat. These images are labeled with the words traffic congestion or no traffic congestion. The experimental results have an accuracy of over 93%. The study is an initial contribution to a future system for predicting traffic congestion in Da Lat when the camera system is fully installed.

Keywords: Da Lat; Deep learning; Detection; Traffic; Traffic congestion.

DOI: [http://dx.doi.org/10.37569/DalatUniversity.11.4.879\(2021\)](http://dx.doi.org/10.37569/DalatUniversity.11.4.879(2021))

Article type: (peer-reviewed) Full-length research article

Copyright © 2021 The author(s).

Licensing: This article is licensed under a CC BY-NC 4.0

1. INTRODUCTION

Nowadays, traffic congestion in Da Lat is not only concentrated in the central areas of the city but also occurs in the suburbs. According to the People's Committee of Da Lat, the number of cars and buses traveling to tourist destinations in the city is the main cause of traffic congestion, especially on weekends and public holidays. The city has already set up traffic control centers to observe and handle traffic jams on the roads.

Many scientists have researched traffic congestion detection and prediction. The results of this study will support users who wish to choose good routes with low traffic and the traffic police to coordinate traffic and limit traffic congestion in Da Lat. However, traffic congestion detection is also a challenging research problem that requires data collection and accurate predictive models. Statistical techniques are commonly applied to solve the problem of predicting traffic congestion. Davis and Nihan (1991), Chang et al. (2012), and Xia et al. (2016) used k-nearest neighbor (KNN) to forecast traffic flow. Some improved models, such as Wu et al. (2004), predicted travel time with a support vector machine (SVM). Hong (2011) used support vector regression (SVR) to predict traffic flow, and Castro-Neto et al. (2009) used online-SVR with typical and atypical traffic conditions for short-term traffic flow forecasting. Ma et al. (2016) forecasted short-term traffic flow with a distributed spatial-temporal weighted model on MapReduce. Asif et al. (2014) and Clark (2003) used multivariate nonparametric regression to predict traffic problems. Haworth and Cheng (2012) also used nonparametric regression to solve the problem of predicting the weather in space and time with missing data.

Recently, Li et al. (2017) predicted short-term highway traffic flow based on a hybrid strategy that considers temporal-spatial information. Zhu et al. (2016) used a linear conditional Gaussian-Bayesian network to predict short-term traffic flow. Li et al. (2016) measured chaotic time for traffic flow prediction based on Bayesian theory. Ma et al. (2017) predicted large-scale transport network speed based on a deep convolutional neural network (CNN). Dao and Zettsu (2018) applied a raster-image-based method to understand urban sensing data. Chakraborty et al. (2018) used deep convolutional neural networks for traffic congestion detection from camera images. Other researchers, such as Mihaita et al. (2020), also used deep learning for traffic congestion detection and prediction. Ke et al. (2020) used a two-stream, multi-channel convolutional neural network for traffic speed prediction, and Bogaerts et al. (2020) used a graphical CNN-LSTM (long-short term memory) neural network for traffic forecasting in the short- and long-term. A year later, Akhtar & Moridpour (2021) summarized all methods that use artificial intelligence for predicting traffic congestion and concluded that deep learning is the best method for solving this problem.

In Vietnam, hundreds of cameras have been installed to monitor traffic in Ho Chi Minh City and Hanoi. The information obtained from the cameras helps the traffic control departments capture and promptly handle traffic incidents to reduce congestion and traffic jams. A number of applications have also been built to help residents in these two big cities keep track of current traffic conditions and choose good routes to avoid traffic jams. There is no system of installed cameras on the roads in Da Lat. Moreover, being the only

city without traffic lights in Vietnam, Da Lat is currently mobilizing many police on routes prone to traffic jams at peak hours. The objective of this article is to explore some machine deep learning models to inform traffic controllers in Da Lat about current traffic flows at certain locations, to choose good routes, and to limit traffic congestion.

2. BACKGROUND

Currently, there are two popular methods to detect traffic congestion from traffic images. The first method consists of identifying objects from traffic images, counting the number of vehicles, and applying traffic congestion labels when the number of vehicles reaches a threshold. The second method consists of manually labeling images collected from traffic cameras as congested or not congested, followed by the use of a layered machine learning model. In this study, the second method was used because of its simplicity and the time-consuming process of labeling each particular vehicle on the image.

SVM is a popular method for traffic congestion prediction because of its idea. Many authors have applied SVM to predict congestion (Tseng et al., 2018; Wang et al., 2015; Feng et al., 2019; Lu & Liu, 2018). Other authors have compared their models with SVM for traffic congestion prediction (Zhang & Qian, 2017; Shen et al., 2017; Chen et al., 2016). These studies also proved the usefulness of SVMs in traffic pattern classification for traffic congestion forecasting. Moreover, CNNs can extract features for image classification and have shown good performance in traffic congestion forecasting in many studies (Ke et al., 2020; Bogaerts et al., 2020; Zhang et al., 2019). The VGG, DenseNet, and ResNet are especially effective deep learning models for CNNs in image classification. Therefore, this paper uses a SVM and deep machine learning methods, namely, CNN, VGG, DenseNet, and ResNet to classify traffic congestion. Next, the author describes the machine learning methods used in this study.

2.1. Support Vector Machine

SVM is a machine learning model widely used in image classification problems. The main idea of the SVM is to create a nonlinear Φ mapping function to convert from the original data space to feature space. Then the SVM automatically detects the optimal separation plane for classifying the object. SVMs are used for both binary and multi-class classification. For binary classification there are two methods with different margins: hard margins and soft margins. For multi-class classification, one vs all (1-n), one vs one (1-1), and directed acyclic graph support vector machines are used.

In our traffic congestion detection study, we used a SVM with a binary classification for a training set of traffic images labeled *traffic congestion* or *no traffic congestion* to get the model results.

2.2. Convolutional Neural Network

The convolutional neural network (LeCun et al., 1989) is a deep learning model widely used with high accuracy in image processing fields for object recognition and

image classification. More specifically, the convolutional neural network combines convolution layers with a number of nonlinear activation functions, such as rectified linear unit (ReLU) to generate input information for the next layer. Convolution was first used by scientists in the field of digital signal processing. Later, convolution techniques based on the process of information change were applied to image and video processing.

The layers in a CNN are connected through the convolution mechanism. Each layer uses as input the convolution result of the previous layer. Other layers used in CNN, such as the pooling layer, are used to filter information to remove noise, and the fully connected layer is used after data are passed through the convolution and pooling layers. The model learns some of the features of the data to combine and give the model results.

The author designed the CNN for detecting traffic jams shown in Figure 1 based on a convolution neural network with convolution and pooling layers. Input is a traffic image resized to 200 x 200 to avoid memory overflow during training. The input data are first processed with two convolutional layers, a max pooling layer, and a dropout layer. The results are then inputted to two convolutional layers and additional max pooling and dropout layers. Finally, the results are processed with two dropout and dense layers to give the output of the model.

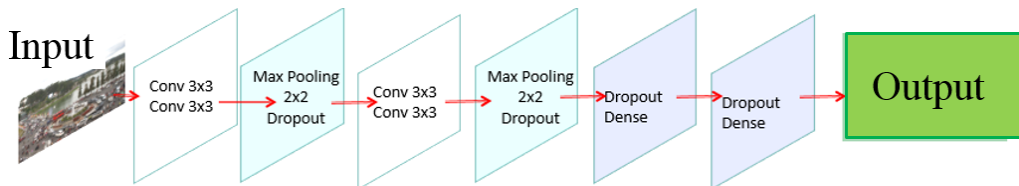


Figure 1. CNN architecture for traffic congestion detection (DCNN)

2.3. VGG

Simonyan and Zisserman of Oxford University proposed VGG16, a convolutional neural network that has been trained for many weeks and uses NVIDIA (Simonyan & Zisserman, 2014). The input of the convolution layer has a fixed image size of 224 x 224 x 3. The total number of model parameters is 138,000,000.

This article uses the VGG16 model (Figure 2) to extract traffic images and the model output to classify traffic images as congested or not congested.

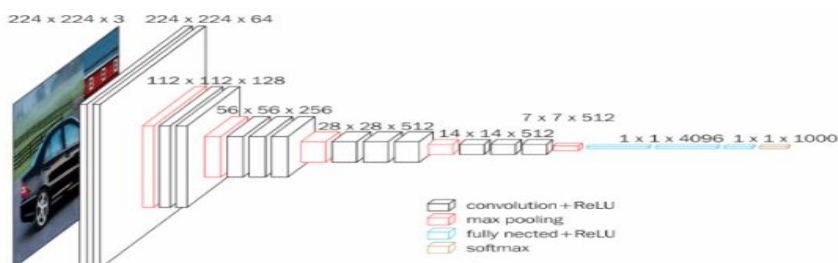


Figure 2. VGG16 architecture

Source: Simonyan & Zisserman (2014).

2.4. Residual Networks

Residual Networks (ResNet) is a convolutional neural network architecture designed with hundreds to thousands of convolution layers. ResNet is a deep learning network that uses redundant networks for optimization and accuracy. A redundant block with the nonlinear mapping class $\mathcal{F}(x) = \mathcal{H}(x) - x$ is shown in Figure 3. Mapping is calculated according to the formula $\mathcal{F}(x) + x$ and is performed by feedforward neural networks.

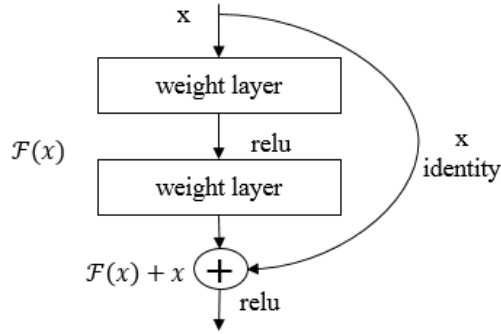


Figure 3. ResNet block

Source: He et al. (2016).

Currently, ResNet has many types of architectures, such as ResNet50, ResNet101, ResNet201, ResNet250, etc. The index number for each architecture type indicates the number of designed classes. This study uses the ResNet50 model because of its simplicity and high efficiency in object recognition.

2.5. DenseNet

Layers	Output size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112×112	7×7 conv, stride 2			
Pooling	56×56	3×3 max pool, stride 2			
Dense Block (1)	56×56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56×56	1×1 conv			
	28×28	2×2 average pool, stride 2			
Dense Block (2)	28×28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28×28	1×1 conv			
	14×14	2×2 average pool, stride 2			
Dense Block (3)	14×14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 64$
Transition Layer (3)	14×14	1×1 conv			
	7×7	2×2 average pool, stride 2			
Dense Block (4)	7×7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$
Classification Layer	1×1	7×7 global average pool			
	7×7	1000D fully-connected, softmax			

Figure 4. DenseNet architecture

Source: Huang et al. (2017).

Dense connected convolutional network (DenseNet) is similar in architecture to ResNet but with some changes: DenseNet contains dense blocks and transfer layers. The advantages of DenseNet are that it can reduce the vanishing-gradient problem, increase the backpropagation feature length, improve feature reusability, and reduce the number of parameters significantly. DenseNet is designed to achieve high efficiency with low memory requirements (Huang et al., 2017). The DenseNet architecture with different numbers of classes is shown in Figure 4.

3. METHODS

Our model to detect traffic congestion includes the following main steps: collecting traffic images, extracting features from the traffic images, using machine learning models to classify the traffic images, and finally, predicting traffic congestion (Figure 5).

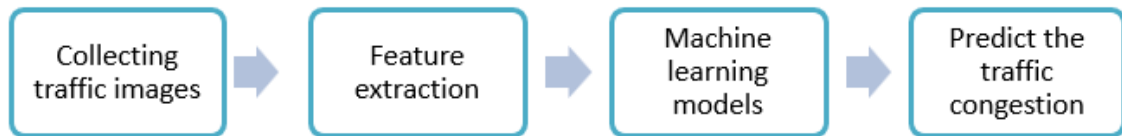


Figure 5. Traffic congestion detection processing

3.1. Collecting traffic images

Traffic images were collected from traffic a traffic camera at predetermined times. Image data extracted from the traffic camera video were stored as *.jpg files. Images were manually labeled to indicate traffic congestion or no traffic congestion.

Some collected and labeled traffic images are shown in Figure 6. The images in Figure 6(a-c) were labeled *no traffic congestion*, and the images in Figure 6(d-f) were labeled *traffic congestion*.

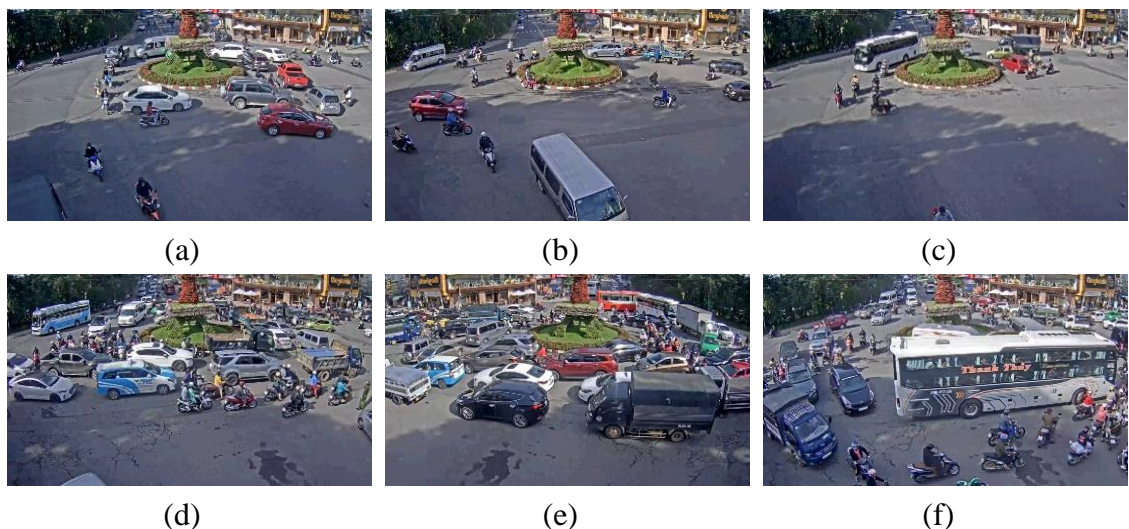


Figure 6. Traffic images labeled as no congestion (a-c) and congestion (d-f)

3.2. Feature extraction

We used the Histogram of Oriented Gradients (HOG) method (Dalal & Triggs, 2005) to extract features of the traffic images for classification with the SVM method (Figure 7).



Figure 7. Histogram of Oriented Gradients feature extraction for SVM

As with other methods, we used available training models to extract features from the stored traffic image data. Then we added the binary layer: traffic congestion or no traffic congestion.

3.3. Machine learning models

We used machine learning for binary classification: traffic congestion or no traffic congestion. In this paper, we focus on SVM, CNN, and architectures developed from CNN for binary classification. The completed training parameters are stored and used in the prediction step.

3.4. Traffic congestion prediction

Traffic congestion prediction is the last step. In this step, we input a new traffic image that is not part of the training dataset to predict either traffic congestion or no traffic congestion. Real-time traffic congestion predictions help users choose a route to avoid traffic congestion. This information also assists the traffic police to control directly or to control traffic lights to reduce traffic congestion.

4. EXPERIMENTS

4.1. Dataset

We collect traffic images from the traffic monitoring camera at Nga 5 Dai Hoc, Da Lat. We extracted images at a frequency of one image per minute. More than 43,000 pictures were extracted.

We manually labeled traffic images as: traffic congestion or no traffic congestion. Our group has three members who labeled all images in the dataset. Each traffic image was labeled three times and finally labeled with the most-approved label. After manual labeling, 820 images were labeled as with traffic congestion and the rest were labeled as without traffic congestion. The number of images for the two classes differed greatly. Therefore, we took 820 images labeled with traffic congestion and randomly selected 820 images labeled with no traffic congestion to have an equal number of images for the two classes. The dataset is presented in Table 1. However, we increased the number of training

images by image rotation and brightness changes to obtain 10 times the number of images than in the original dataset to avoid overfitting in the training step. In this phase, we rotated the images by -10, 10, -20, or 20 degrees and made brightness changes of 0.25, 0.5, 0.75, 1.25, or 1.5, respectively, for all images in the training set. Finally, we obtained 13,120 images for the training set.

Table 1. Dataset details

Dataset	Traffic congestion	No traffic congestion	Size
All Images	820	820	
Training (80%)	656	656	224 x 224
Test (20%)	164	164	

4.2. Evaluation

We used the 10-fold cross-evaluation method. We performed random data shuffling and divided the training dataset into 10 parts with each part having 1312 images. We used one part of the data for validation and the other nine parts for training. Finally, we averaged the prediction results for the test dataset on 10 models. To evaluate the models, we used the following three metrics: precision, recall, and accuracy, calculated according to equations (1), (2), and (3), respectively. Precision is the ratio of the number of correct traffic congestion labels to the total number of correct and incorrect traffic congestion labels. Recall is the ratio of the number of correct traffic congestion labels to the sum of the correct and incorrect traffic congestion labels. Accuracy is the ratio of the number of correct labels to the total number of labels.

$$P(\textit{Precision}) = \frac{TP}{TP + FP} \quad (1)$$

$$R(\textit{Recall}) = \frac{TP}{TP + FN} \quad (2)$$

$$A(\textit{Accuracy}) = \frac{TP + TN}{TP + FP + TN + FN} \quad (3)$$

where TP is the number of correct traffic congestion labels, FP is the number of incorrect traffic congestion labels, TN is the number of correct no traffic congestion labels, and FN is the number of incorrect no traffic congestion labels.

4.3. Results

In this paper, we used the scikit-image, scikit-learn, and keras libraries for image preprocessing and algorithms SVM, DCNN, ResNet50, VGG16, and DenseNet-121.

Table 2 shows the precision, recall, and accuracy indexes with 80% training data and 20% test data for algorithms SVM, DCNN, ResNet50, VGG16, and DenseNet-121.

These results are averaged from the 10-fold cross-evaluation method. The SVM method achieved the lowest classification accuracy of 68.99%, followed by SVM with the HOG method at 84.9%, and the DCNN method at 89.47%. The ResNet50, VGG16, and DenseNet-121 methods have high accuracies, over 93%. Thus, when using architectures based on CNN to classify traffic congestion, the accuracy is over 10% higher than with the SVM method.

Table 2. Precision, recall, and accuracy of the methods

Method	Precision (%)	Recall (%)	Accuracy (%)
SVM	68	70	69
SVM+HOG	82.3	87.5	84.9
DCNN	88	91	89.5
ResNet50	93	94	93.5
VGG16	93.5	93.6	93.6
DenseNet-121	92.5	93.7	93.1

In addition, we also evaluated the effect of the size of the training dataset on the accuracy of the methods. This evaluation proved that the accuracy increases when we increased the size of the training dataset. The larger the training set, the higher the accuracy of the model. The VGG16 method achieved the best results when the training dataset accounted for 90%.

Table 3. Accuracy (%) according to the size of the training dataset

Training set Method	60%	70%	80%	90%
SVM	55.4	65.3	69	70.5
SVM+HOG	69.2	74.6	84.9	85.7
DCNN	70.7	78.0	89.5	90
ResNet50	76.3	85.4	93.5	94.1
VGG16	76.5	85.8	93.6	94.3
DenseNet-121	75.8	85.0	93.1	93.6

Images with correct and incorrect predictions obtained with the VGG16 method are shown in Figure 8. Figures 8(a, d) are predicted correctly; Figures 8(b, c) are incorrectly predicted.

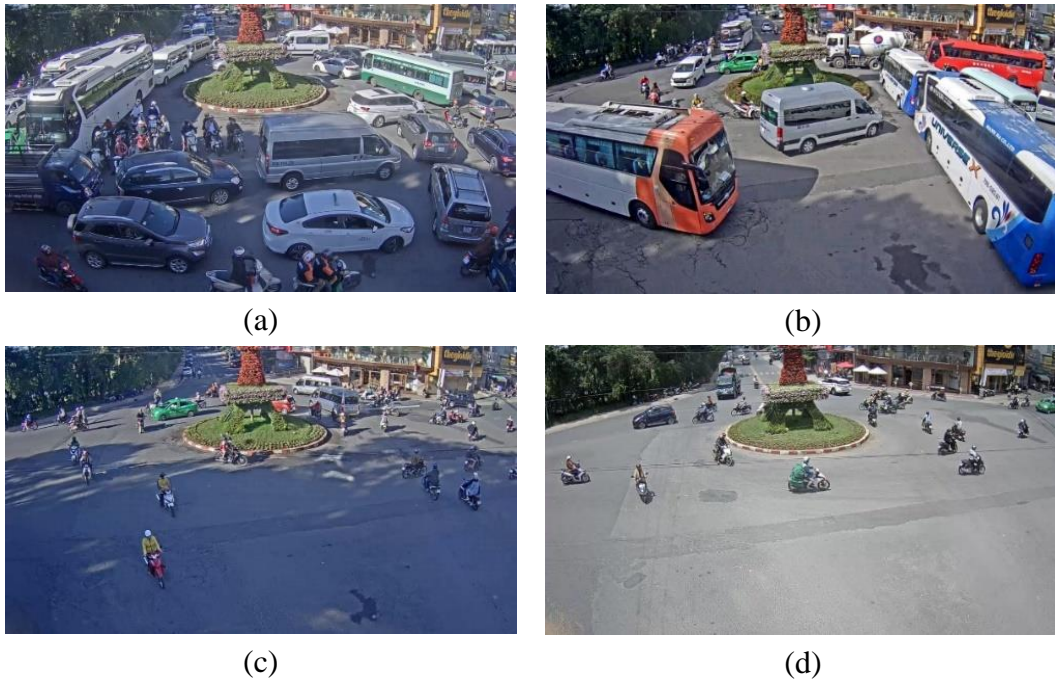


Figure 8. Congestion detection classification examples with the VGG16 model

5. CONCLUSIONS

In this paper, we evaluated the results of traffic congestion classification with traffic image data collected at Nga 5 Dai Hoc, Da Lat by the SVM, DCNN, ResNet50, VGG16, and DenseNet-121 methods. The results show that the accuracy is higher when using CNN-based architectures than with the SVM classification method. The highest accuracy for classifying traffic images with or without traffic congestion is 93.6%.

Based on these results, we plan to collect traffic images at many locations in Da Lat, and for each location, find the most suitable model for predicting traffic congestion. Next, we will build applications that provide users with traffic congestion information at various locations in the city so they can find good routes. Moreover, when enough data are available over a long time at enough traffic locations, we will build a traffic congestion prediction system to advise drivers on suitable routes to reduce future traffic congestion.

REFERENCES

- Akhtar, M., & Moridpour, S. (2021). A review of traffic congestion prediction using artificial intelligence. *Journal of Advanced Transportation*, 2021, 1-18.
- Asif, M. T., Dauwels, J., Goh, C. Y., Oran, A., Fathi, E., Xu, M., Dhanya, M. M., Mitrovic, N., & Jaillet, P. (2014). Spatiotemporal patterns in large-scale traffic speed prediction. *IEEE Transactions on Intelligent Transportation Systems*, 15(2), 794-804.
- Bogaerts, T., Masegosa, A. D., Angarita-Zapata, J. S., Onieva, E., & Hellinckx, P. (2020). A graph CNN-LSTM neural network for short and long-term traffic forecasting

- based on trajectory data. *Transportation Research Part C: Emerging Technologies*, 112, 62-77.
- Castro-Neto, M., Young-Seon, J., Jeong, M. -K., & Han, L. D. (2009). Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions. *Expert Systems with Applications*, 36(3), 6164-6173.
- Chakraborty, P., Adu-Gyamfi, Y. O., Poddatr, S., Ahsani, V., Sharma, A., & Sarkar, S. (2018). Traffic congestion detection from camera images using deep convolution neural networks. *Transportation Research Record*, 2672(45), 222-231.
- Chang, H., Lee, Y., Yoon, B., & Baek, S. (2012). Dynamic near-term traffic flow prediction: System oriented approach based on past experiences. *IET Intelligent Transportation Systems*, 6(3), 292-305.
- Chen, Y., Lv, Y., Li, Z., & Wang, F. (2016). Long short-term memory model for traffic congestion prediction with online open data. *Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 132-137.
- Clark, S. (2003). Traffic prediction using multivariate nonparametric regression. *Journal of Transportation Engineering*, 129(2), 161-168.
- Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 1, 886-893.
- Dao, M. -S., & Zettsu, K. (2018). A raster-image-based approach for understanding associations of urban sensing data. *2018 IEEE First International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, 134-137.
- Davis, G. A., & Nihan, N. L. (1991). Nonparametric regression and short-term freeway traffic forecasting. *Journal of Transportation Engineering*, 117(2), 178-188.
- Feng, X., Ling, X., Zheng, H., Chen, Z., & Xu, Y. (2019). Adaptive multi-kernel SVM with spatial-temporal correlation for short-term traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*, 20(6), 2001-2013.
- Haworth, J., & Cheng, T. (2012). Non-parametric regression for space-time forecasting under missing data. *Computers, Environment and Urban Systems*, 36(6), 538-550.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770-778.
- Hong, W. -C. (2011). Traffic flow forecasting by seasonal SVR with chaotic simulated annealing algorithm. *Neurocomputing*, 74(12-13), 2096-2107.
- Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2261-2269. doi: 10.1109/CVPR.2017.243.
- Ke, R., Li, W., Cui, Z., & Wang, Y. (2020). Two-stream multichannel convolutional neural network for multi-lane traffic speed prediction considering traffic volume

- impact. *Transportation Research Record: Journal of the Transportation Research Board*, 2674(4), 459-470.
- LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4), 541-551.
- Li, L., He, S., Zhang, J., & Ran, B. (2017). Short-term highway traffic flow prediction based on a hybrid strategy considering temporal-spatial information. *Journal of Advanced Transportation*, 50(8), 2029-2040.
- Li, Y., Jiang, X., Zhu, H., He, X., Peeta, S., Zheng, T., & Li, Y. (2016). Multiple measures-based chaotic time series for traffic flow prediction based on Bayesian theory. *Nonlinear Dynamics*, 85(1), 179-194.
- Lu, S., & Liu, Y. (2018). Evaluation system for the sustainable development of urban transportation and ecological environment based on SVM. *Journal of Intelligent and Fuzzy Systems*, 34(2), 831-838.
- Ma, X., Dai, Z., He, Z., Ma, J., Wang, Y., & Wang, Y. (2017). Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction. *Sensors*, 17(4), 818.
- Ma, Z., Luo, G., & Huang, D. (2016). Short term traffic flow prediction based on on-line sequential extreme learning machine. *Proceedings of the 2016 Eighth International Conference on Advanced Computational Intelligence (ICACI)*, 143-149.
- Mihaita, A. -S., Li, H., & Rizoiu, M. -A. (2020). *Traffic congestion anomaly detection and prediction using deep learning*. arXiv:2006.13215v1.
- Shen, Q., Ban, X., & Guo, C. (2017). Urban traffic congestion evaluation based on kernel the semi-supervised extreme learning machine. *Symmetry*, 9(5), 70.
- Simonyan, K. & Zisserman, A. (2014). *Very deep convolutional networks for large-scale image recognition*. arXiv:1409.1556.
- Tseng, F. -H., Hsueh, J. -H., Tseng, C. -W., Yang, Y. -T., Chao, H. -C., & Chou, L. -D. (2018). Congestion prediction with big data for real-time highway traffic. *IEEE Access*, 6, 57311-57323.
- Wang, X., An, K., Tang, L., & Chen, X. (2015). Short term prediction of freeway exiting volume based on SVM and KNN. *International Journal of Transportation Science and Technology*, 4(2), 337-352.
- Wu, C. -H., Ho, J. -M., & Lee, D. T. (2004). Travel-time prediction with support vector regression. *IEEE Transactions on Intelligent Transportation Systems*, 5(4), 276-281.
- Xia, D., Wang, B., Li, H., Li, Y., & Zhang, Z. (2016). A distributed spatial-temporal weighted model on MapReduce for short-term traffic flow forecasting. *Neurocomputing*, 179(C), 246-263.
- Zhang, P., & Qian, Z. (2017). User-centric interdependent urban systems: Using time-of-day electricity usage data to predict morning roadway congestion. *Transportation Research Part C: Emerging Technologies*, 92, 392-411.

- Zhang, W. Yu, Y., Qi, Y., Shu, F., & Wang, Y. (2019). Short-term traffic flow prediction based on spatio-temporal analysis and CNN deep learning. *Transportmetrica A: Transport Science*, 15(2), 1688-1711.
- Zhu, Z., Peng, B., Xiong, C., & Zhang, L. (2016). Short-term traffic flow prediction with linear conditional Gaussian Bayesian network. *Journal of Advanced Transportation*, 50, 1111-1123.