ICIC Express Letters
Volume **17**, Number **11**, November **2023**
ICIC International ©2023 ISSN 1881-803X
pp. 1187–1194

TRAFFIC DENSITY ESTIMATION BASED ON BLOCK OCCUPANCY CLASSIFICATION

Agus Harjoko[1,*], Wahyono[1], Nanang Susyanto[2], Irwan Endrayanto[2]
Muhammad Ardi Putra[1] and Dea Angelia Kamil[1]

[1]Department of Computer Science and Electronics
[2]Department of Mathematics
Faculty of Mathematics and Natural Sciences
Universitas Gadjah Mada
Jl. Geografi, North Sekip, Bulaksumur, Yogyakarta 55281, Indonesia
{ wahyo; nanang_susyanto; endrayanto }@ugm.ac.id
{ muhammadardi2017; dea.angelia.kamil }@mail.ugm.ac.id
*Corresponding author: aharjoko@ugm.ac.id

Received November 2022; accepted February 2023

Abstract. *High levels of traffic density cause negative impacts in society. Due to this problem, the authors of this research proposed a method to estimate traffic density. The proposed method was implemented on Junction 1 dataset. We utilized Local Binary Patterns (LBP), Histogram of Oriented Gradients (HOG), and Gray-Level Cooccurrence Matrix (GLCM) for extracting image features, and Support Vector Machine (SVM) for classification. The results showed that CNN can be considered as the best model when it comes to classification accuracy with a score of up to 99.6%. Meanwhile, the fastest processing speed was obtained by LBP+SVM with an average of 30.2 FPS.*
Keywords: Traffic density estimation, Local binary patterns, Histogram of oriented gradients, Gray-level co-occurrence matrix

1. Introduction.

Traffic lights with no time distribution regulation may cause traffic congestion in parts of a road segment that have a high traffic flow. Traditional traffic lights usually have the same time division for each junction without considering the density level of road segments. High traffic flow in peak hours has bad impacts on the productivity of the society. Traffic lights may be programmed to be adaptive by performing traffic density estimation. Traffic density estimation is a part of ITS (Intelligent Traffic System). There are a number of studies beyond traffic density estimation in this research area, such as vehicle speed estimation [1], illegally parked vehicle detection [2,3], vehicle counting [4], and traffic sign detection [5].

One of the most famous methods is to implement an Inductive Loop Detector (ILD) [6]. However, this approach is expensive in terms of installation and maintenance costs. On the other hand, vision-based methods tend to be more affordable and easier to install. Generally, computer vision-based methods can be divided into two categories: macroscopic and microscopic [6]. The microscopic approach accurately estimates density level, but the computation is expensive. This method uses an approach for detecting and tracking vehicles one by one to count the number of vehicles in the road segment. Besides, the macroscopic technique uses a holistic approach to estimate traffic density. The captured traffic video is analyzed to know whether the road is dense or empty. Such a method has lower computational complexity.

In this research, the authors focus on estimating traffic density using a macroscopic approach to reduce the complexity and still maintain great accuracy. Handcrafted feature

DOI: 10.24507/icicel.17.11.1187

descriptors based on image processing techniques were utilized to extract image features and machine learning models to classify traffic density conditions. The specific methods implemented are LBP, HOG, and GLCM for feature extraction with SVM model. The accuracy score will be compared with the research that uses the CNN model [7]. The proposed method will be implemented on Junction 1 dataset [8], which is publicly available on the Internet. This paper's main contribution is using the mentioned handcrafted features over CNN, which was previously proposed by [7] to perform block classification of size 15×15, 12×12, and 10×10.

The following sections will cover the whole process of the research. Section 2 discusses the previous research related to traffic density estimation. Section 3 explains the methodology of the proposed method utilized in this research. In Section 4, the experiment results are discussed and analyzed. The conclusion of this study is concluded in Section 5.

2. Literature Review.

Based on the parameters processed for estimating traffic density, the approach is divided into microscopic and macroscopic categories.

2.1. Microscopic approach.

The general approach to performing microscopic-based traffic density estimation is to performing object detection. In the research conducted by [9], the authors use the Mixture of Gaussian (MOG) algorithms to implement background subtraction. This method is used to separate the foreground object from the background. A comparison between Gaussian components of two different frames is then made to perform object tracking. The traffic density estimation is calculated by taking the ratio of the number of vehicles in the lane and the length of the lane. However, this approach is considered to be effective only on low-density traffic. Using a similar approach, [10] and [11] utilize Raspberry Pi camera modules to capture traffic video. In [10], contouring is used to identify the moving vehicles and mark the vehicle's centroid. The number of vehicles is calculated by counting the number of vehicle centroids that pass through the artificial entry and exit line. The number of vehicles will give influence to the timing of the traffic light. The accuracy of this research reaches 95.65%. In [11], the accuracy of vehicle detection is 97.39%, and the tracking accuracy reaches about 98.4%. The approaches are accurate for several weather conditions like snowy, dusty, and night. However, these methods are susceptible to the occlusion between vehicles and camera shaking since microscopic approaches depend on the accuracy of vehicle detection.

2.2. Macroscopic approach.

With the improvement of deep learning, research in traffic density estimation was developed using a deep learning-based approach [12,13]. Deep learning approaches have good accuracy yet require higher computational complexity and huge numbers of training data. In the case of handcrafted feature extraction [14-18], an extracted feature is used for classifying the traffic density condition. In [14], the first process is extracting the Region of Interest. The authors divide the frame into 8×9 grids of cells. Support Vector Machine (SVM) model is used to classify the grid fill of a vehicle or not. In this research, Junction 2 dataset is used. For extracting features, Histogram of Oriented Gradients (HOG) combined with Local Binary Patterns (LBP) is implemented to fix the problem of HOG that is only good at capturing edges and corners. The methods have a good accuracy score of about 94.88%. However, the combination of two descriptors causes the computational complexity to be high. In [15], the authors use two different approaches for extracting features such as texture and edge features. Gray-Level Co-occurrence Matrix (GLCM) is an extractor feature based on the texture that is used. Besides, the authors utilize Edge Histogram Descriptor (EHD) for edge-based feature extraction. The approach reaches a good accuracy of about 94.48% but only reaches 13-FPS processing rate and is distracted by static shadow in the ROI. Wassantachat et al. [16] also use GLCM for extracting features and use background modeling with On-line SVM to reduce the training effort. The accuracy reaches about 89.43%. However, it is highly computational in time and resources. Moreover, research based on pixel intensity is common for use in traffic density estimation [6,19]. The technique can significantly reduce the computational complexity. However, defining the rule for calculating the threshold is challenging.

3. Research Methodology.

All the steps required to be done in this research can be drawn as a flowchart which is displayed in Figure 1.

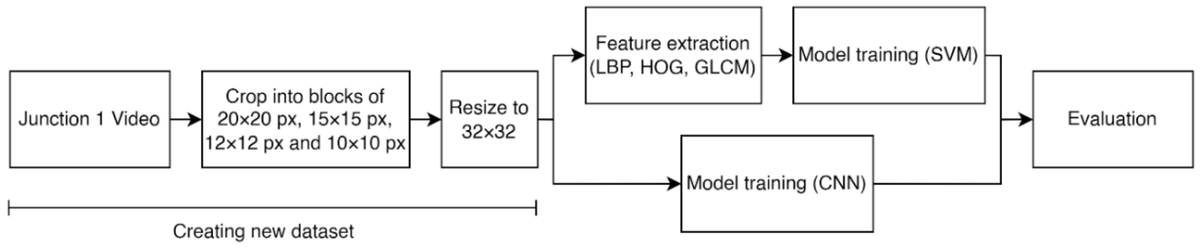

Figure 1. The flowchart of this research

3.1. Creating a new dataset.

The main idea of this research was to take a bunch of blocks from a particular road region and to predict whether every single of those is being occupied by vehicles. The overall traffic density can then be obtained by aggregating the prediction results. The illustration shown in Figure 2 displays how the blocks were arranged on a road area from Junction 1 video such that they will be able to estimate the overall traffic density.

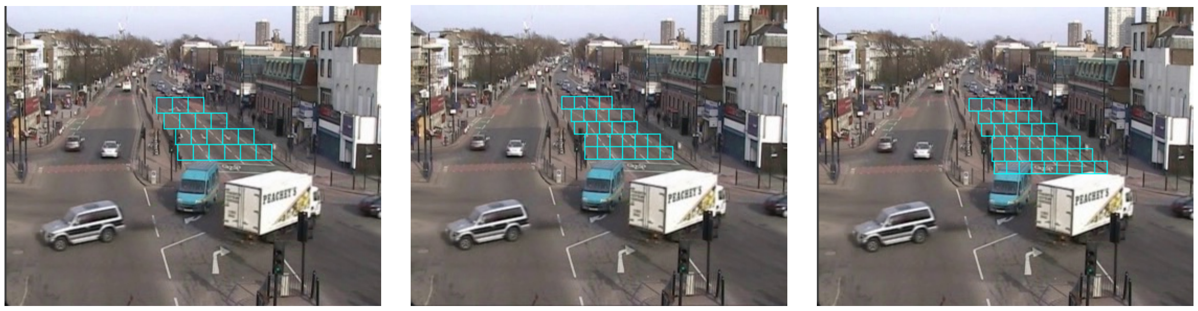

Figure 2. How the blocks were arranged. 15×15 (left), 12×12 (middle), 10×10 (right).

In order to do so, it was necessary to train a machine learning model such that it would be able to predict block occupancy. It implies that a new dataset containing occupied and non-occupied blocks was required to train the model. The new dataset was created based on the Junction 1 video. 495 blocks from the video were taken at a random position, in which it consisted of 254 occupied blocks and 241 unoccupied blocks. There would be four datasets to be generated which were grouped according to the block size, namely 20×20, 15×15, 12×12, and 10×10 pixels. Those blocks were resized to 32×32 prior to entering the next stage.

3.2. Feature extraction: LBP (Local Binary Patterns).

There would be three different feature extraction methods to be employed. This section is going to focus on LBP. The feature extraction algorithm which was first proposed by Ojala et al. [20] is suitable for enhancing texture features. In this case, an unoccupied block generally looks like a flat region. Or in other words, it is considered to be textureless. Meanwhile, there will be rough textures on an occupied block thanks to the presence of vehicles. Due to this, it was expected that LBP features will be able to enhance block textures so that it would be distinguishable by the machine learning model.

3.3. **Feature extraction: HOG (Histogram of Oriented Gradients).** HOG has an ability to extract shape features within the blocks. Since it works by detecting edges, the algorithm should be able to enhance the difference between occupied and unoccupied blocks. The HOG algorithm produces feature vectors by taking account of edge magnitudes as well as edge orientations. All these magnitudes will be grouped into several bins which are determined according to the gradient orientations. Such an operation forms a histogram which then acts as a feature vector.

3.4. **Feature extraction: GLCM (Gray Level Co-occurrence Matrix).** Similar to LBP, GLCM is employed to do extraction on texture features [21]. Even though the objectives of the two are the same, the process is completely different. Two main parameters are required to be defined: angle and distance. The first parameter determines the comparison direction while the second denotes the distance between the current pixel and the pixel to be compared. The comparison results of every pixel in the image will be stored in a co-occurrence matrix. The feature vector is then generated by calculating its second order statistical features such as contrast, correlation, and homogeneity.

3.5. **Model training.** A block is classified as either being occupied or unoccupied using SVM. This machine learning model was selected since it is highly used in image classification tasks thanks to its good performance, especially when it is trained on handcrafted features [5,14-16]. SVM works by separating different classes using a hyperplane. The two hyperparameters required to be set are the kernel and C, where the kernel determines the hyperplane shape while C is a regularization term. For comparison purposes, CNN was also implemented to perform block classification since this deep learning approach can be considered as the current state-of-the-art model for image-related tasks. CNN automatically performs feature extraction on its own at its convolution layers. This concept was first introduced under the term "Neocognitron" [22]. The term CNN itself was finally used in [23]. Additionally, the handcraft feature extractions explained earlier only take the grayscale version of the block. This is essentially because textures are detectable without taking account of color features.

3.6. **Model evaluation.** The performance of a model is evaluated by calculating its accuracy score. Actual testing on the video data will also be done to find out the frame processing rate. This is important to do since it is expected that the proposed model will be actually implemented in real time.

4. Results and Discussions.

4.1. **Experiment setting.** This research consisted of several experiments. Each of those was named LBP+SVM, HOG+SVM, GLCM+SVM, and CNN, in which the names are basically self-explanatory. Initially, the authors used only the 20×20 blocks dataset. This dataset was then split into train and validation sets with a ratio of $80 : 20$. All models were trained 20 times which the average accuracy was taken for the final result. The best model was then brought to the testing phase, and its classification performance is evaluated qualitatively, while at the same time the average time used for processing 1000 frames was also calculated. The overall traffic density was obtained by dividing the number of occupied blocks by the number of all blocks in the frame. All these experiments were done on a machine that uses Intel Core i5-8250U processor with 8 GB of RAM and Nvidia Geforce MX150 GPU. In the next experiments, the authors also attempted to do the same for the 15×15, 12×12, and 10×10 blocks.

4.2. **Parameter selection.** The P and R parameters of LBP were set to 8 and 1, respectively. The block of LBP features was then divided into 4 regions, in which every single of those produced a histogram of size 15. To the HOG features, an area of 8×8 pixels was grouped into a single cell, while a group of 2×2 formed a block. Every single of these blocks then produced a feature vector size of 36. Thus, the final feature vector dimension was 324. Prior to extracting GLCM features, the pixel levels were quantized to 64-levels first. Six statistical features (mean, standard deviation, contrast, correlation, energy, and homogeneity) were then extracted from GLCM. Since the angles parameter was set to 4 directions ($0°$, $45°$, $90°$ and $135°$), the final feature vector size was 24. All these extracted features were then used to train an SVM with RBF kernel. To the CNN, the deep learning model employed a single convolution layer with 32 kernels of size 3×3 and stride of 1×1. ReLU (Rectified Linear Unit) activation function was applied to the convolved image. Further processing was done by connecting a maximum-pooling layer of size 2×2 which directly flattened the result. Three consecutive dense layers with 32, 16 and 1 neurons were then connected in which all of those employed sigmoid activation function.

4.3. **Evaluation and discussion.** As mentioned in Chapter 4.1, the first experiment was related to the block classification of size 20×20 pixels. The results are shown in Figure 3.

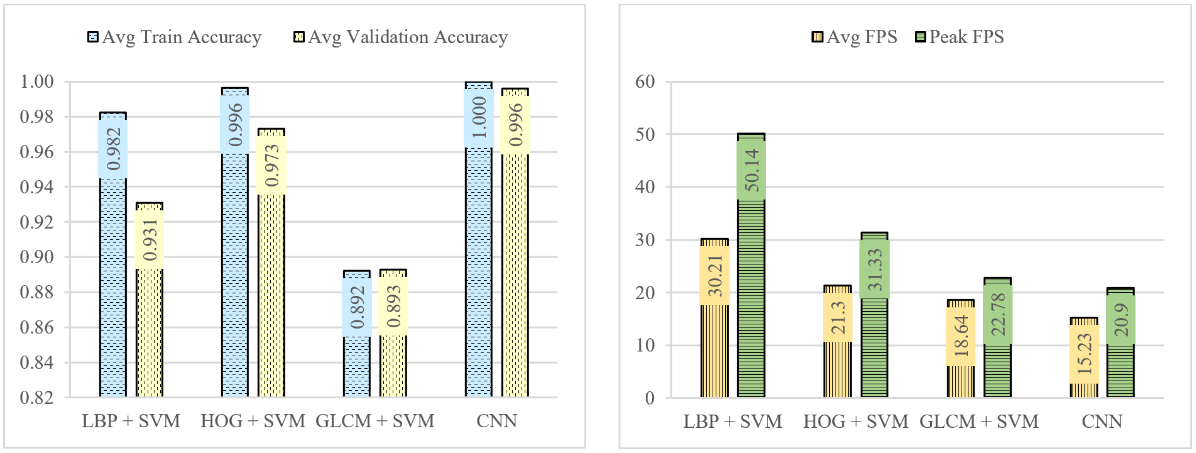

FIGURE 3. Classification accuracy scores of 20×20 blocks (left); frame processing speed comparison in FPS (Frames per Second) (right)

It is clearly seen in Figure 3 (left) that CNN performs best in distinguishing the occupied and unoccupied blocks as it reached the validation accuracy of 99.6%. This accuracy score was then followed by HOG, LBP, and GLCM, respectively. According to this result, it is found that color information might be important in this case due to the fact that only the CNN model took the entire RGB channel. Furthermore, the deep learning approach also allowed its convolution layer to freely extract particular information. This is different from the other SVM-based models which only considered texture (LBP and GLCM) and shape (HOG) features.

The chart in Figure 3 (right) shows that even though the CNN achieves the greatest classification accuracy score, it is quite slow compared to the SVM models. This absolutely makes sense because the CNN model accepts the input size of 3,072 (obtained from $32 \times 32 \times 3$), which is a lot larger as compared to the other three. The high number of weights and biases also give influence on the CNN predicting speed thanks to the depth and width of the network. This is different from the SVM which algorithmically does not use such a multi-layer concept. Furthermore, the feature extraction stage of the LBP+SVM is computationally cheaper, which in return causes the processing speed averages and peaks at 30 and 50 FPS, respectively. The computation time of GLCM can be considered to be slow compared to all other models with handcrafted features, which is actually also proven by [24]. Despite the fact that the CNN model is not the fastest one, it can still be considered good enough since 15 FPS on average is actually decent enough to be implemented in real time.

Other experiments related to different block sizes produce the results shown in Figure 4. As expected earlier, the greatest classification rate was obtained by the largest block (20×20). This is due to the fact that larger blocks contain more information compared to the smaller ones. These results are consistent since all feature extraction variations, including the CNN, behave the same way. According to Figure 4 (left), CNN obtained the best classification rate compared to all others. Even more, the worst CNN performance (12×12 blocks, 0.949 accuracy) could only be outperformed by the CNN models themselves and the HOG+SVM with the block dimension of 20×20 pixels (0.973 accuracy).

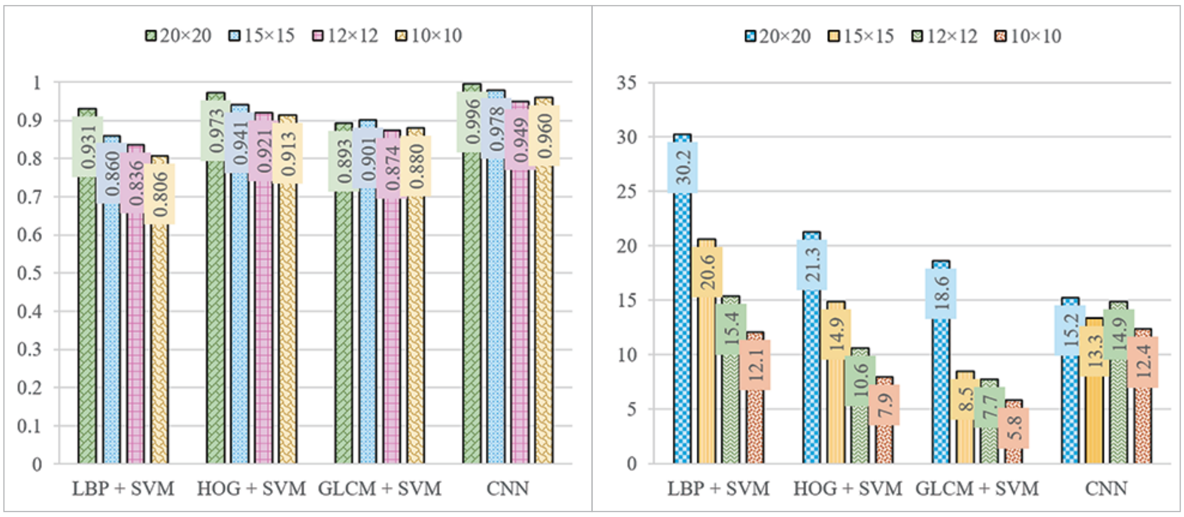

FIGURE 4. Validation accuracy of different block sizes (left); frame processing speed comparison between different block sizes (right)

When it comes to processing speed, it was found that experiments with smaller blocks have a tendency to be slower. This kind of behaviors occurs since more blocks are required to cover the entire road segment. There are 40, 27, 18 and 10 blocks for the size of 10×10, 12×12, 15×15 and 20×20, respectively. Among the models in Figure 4 (right), CNN is the one in which the processing speed and block size seem uncorrelated to each other. It is proven due to the fact that it does not show the similar trend compared to the other machine learning models. Such behavior is probably happening thanks to the difference in model backend implementation. In the earlier part of this chapter, we mentioned that all SVMs were taken from the Scikit-Learn module, while the CNN was constructed using Keras. Since Keras is capable of running its code on GPU, the authors of this paper suspected that it is the one that causes CNN to behave differently.

4.4. **Comparison with other works.** The block classification scores on Table 1 show that the proposed method is comparable with other similar traffic density estimation approaches.

The research paper of [14] performed block occupancy prediction by using the combination of HOG and LBP features. These features were then used to train an SVM model and achieved an accuracy score of 94.88%. On the other hand, [16] reached 89.43% of accuracy. This result was obtained by using the OSVM-BG (On-line SVM Classifier and Background Modeling Technique) model. The model was trained using first and second-order statistical features and another image feature gathered from background subtraction. Instead of

Table 1. Comparison with other research

Author(s)	Accuracy score
Li et al. [15]	94.48%
Wassantachat et al. [16]	89.43%
Garg et al. [6]	99.47%
Prasad et al. [14]	94.88%
Proposed method (LBP+SVM)	93.10%
Proposed method (HOG+SVM)	97.30%
Proposed method (GLCM+SVM)	89.30%
Proposed method (CNN reproduced from Putra et al. [7])	99.60%

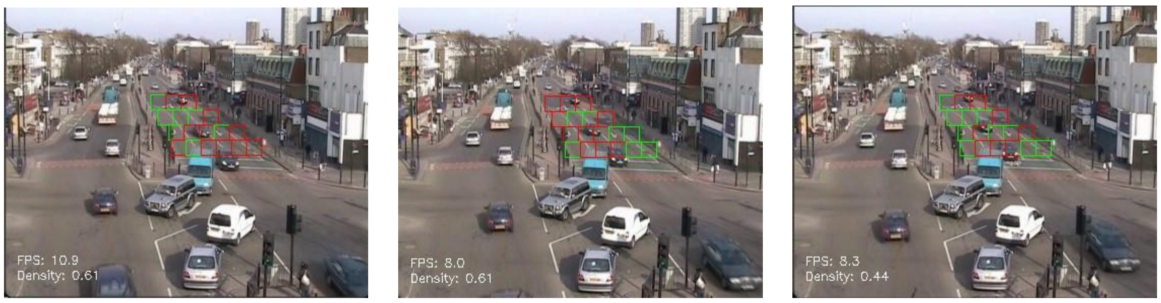

Figure 5. Results obtained on 15 × 15 blocks using LBP+SVM (left), HOG+SVM (middle), and CNN (right)

only using the background reconstruction method, [6] also employed a shadow elimination process. This kind of approachs achieved the best TPR (True Positive Rate) of 99.47%. Lastly, similar to [16], [15] also used first and second-order statistical features. However, the authors also added EHD (Edge Histogram Descriptor) as additional feature. This approach helped SVM to achieve 94.48% of classification accuracy.

5. **Conclusions and Future Works.** CNN could be considered the best model in terms of accuracy since the value remains high even on the smallest block size. On the other hand, LBP+SVM was the one that achieved the fastest frame processing rate, especially on 20×20 block size, yet with a decent accuracy score. Several things can still be experimented to determine whether there is a better approach for the classification task. In the earlier part of this paper, it was mentioned that all the block size variations were resized to 32×32 pixels prior to entering the feature extraction stage. This causes the processing of smaller blocks to be slower since the exact same road segment requires more blocks to get covered entirely, especially when a traditional feature extraction method is used. In order to address this problem, it is possible to conduct further research on the block resizing dimension variations. This implies that the best feature extraction parameters, which can lead to better accuracy scores, also need to be searched.

Acknowledgment. This work was supported by the Ministry of Education and Culture, Research, and Technology with Grant Number 2697/UN1/DITLIT/Dit-Lit/PT.01.03/20 22.

REFERENCES

[1] H. S. Sundoro and A. Harjoko, Vehicle counting and vehicle speed measurement based on video processing, *J. Theor. Appl. Inf. Technol.*, vol.84, no.2, pp.233-241, 2016.
[2] Wahyono and K. H. Jo, Cumulative dual foreground differences for illegally parked vehicles detection, *IEEE Trans. Ind. Informatics*, vol.13, no.5, pp.2464-2473, DOI: 10.1109/TII.2017.2665584, 2017.

[3] Wahyono, A. Filonenko and K. H. Jo, Illegally parked vehicle detection using adaptive dual background model, *IECON 2015 – The 41st Annu. Conf. IEEE Ind. Electron. Soc.*, pp.2225-2228, DOI: 10.1109/IECON.2015.7392432, 2015.

[4] A. Harjoko, I. Candradewi and B. A. A. Aldino, Intelligent traffic monitoring systems: Vehicles detection, tracking, and counting using Haar cascade classifier and optical flow, *ACM Int. Conf. Proceeding Ser.*, pp.49-55, DOI: 10.1145/3177404.3177441, 2017.

[5] A. Sugiharto, A. Harjoko and S. Suharto, Indonesian traffic sign detection based on Haar-PHOG features and SVM classification, *Int. J. Smart Sens. Intell. Syst.*, vol.13, no.1, pp.1-15, DOI: 10.21307/ijssis-2020-026, 2020.

[6] K. Garg, S. K. Lam, T. Srikanthan and V. Agarwal, Real-time road traffic density estimation using block variance, *2016 IEEE Winter Conf. Appl. Comput. Vision (WACV 2016)*, DOI: 10.1109/WACV.2016.7477607, 2016.

[7] M. A. Putra, A. Harjoko and Wahyono, Estimation of traffic density using CNN with simple architecture, *2022 International Workshop on Intelligent Systems (IWIS)*, pp.1-5, DOI: 10.1109/IWIS56333.2022.9920811, 2022.

[8] D. Russell, *Junction Dataset*, www.eecs.qmul.ac.uk.http://www.eecs.qmul.ac.uk/∼sgg/QMUL_Junction_Datasets/Junction/Junction.html, Accessed on Oct. 12, 2022.

[9] M. F. Chowdhury, M. R. A. Biplob and J. Uddin, Real time traffic density measurement using computer vision and dynamic traffic control, *2018 Jt. 7th Int. Conf. Informatics, Electron. Vis. 2nd Int. Conf. Imaging, Vis. Pattern Recognition (ICIEV-IVPR 2018)*, pp.353-356, DOI: 10.1109/ICIEV.2018.8641039, 2019.

[10] A. Gupta, C. Gandhi, V. Katara and S. Brar, Real-time video monitoring of vehicular traffic and adaptive signal change using Raspberry Pi, *2020 IEEE Students' Conf. Eng. Syst. (SCES 2020)*, pp.1-5, DOI: 10.1109/SCES50439.2020.9236731, 2020.

[11] A. P. Kulkarni and V. P. Baligar, Real time vehicle detection, tracking and counting using Raspberry-Pi, *Proc. of the 2nd Int. Conf. Innov. Mech. Ind. Appl. (ICIMIA 2020)*, pp.603-607, DOI: 10.1109/ICIMIA48430.2020.9074944, 2020.

[12] H. Yang, Y. Zhang, Y. Zhang, H. Meng, S. Li and X. Dai, A fast vehicle counting and traffic volume estimation method based on convolutional neural network, *IEEE Access*, vol.9, pp.150522-150531, DOI: 10.1109/ACCESS.2021.3124675, 2021.

[13] Z. Sun, P. Wang, J. Wang, X. Peng and Y. Jin, Exploiting deeply supervised inception networks for automatically detecting traffic congestion on freeway in China using ultra-low frame rate videos, *IEEE Access*, vol.8, pp.21226-21235, DOI: 10.1109/ACCESS.2020.2968597, 2020.

[14] D. Prasad, K. Kapadni, A. Gadpal, M. Visave and K. Sultanpure, HOG, LBP and SVM based traffic density estimation at intersection, *2019 IEEE Pune Sect. Int. Conf. (PuneCon 2019)*, DOI: 10.1109/PuneCon46936.2019.9105731, 2019.

[15] Z. Li, E. Tan and J. Chen, On traffic density estimation with a boosted SVM classifier, *Proc. of Digit. Image Comput. Tech. Appl. (DICTA 2008)*, pp.117-123, DOI: 10.1109/DICTA.2008.30, 2008.

[16] T. Wassantachat, Z. Li, J. Chen, Y. Wang and E. Tan, Traffic density estimation with on-line SVM classifier, *The 6th IEEE Int. Conf. Adv. Video Signal Based Surveillance (AVSS 2009)*, vol.2, no.1, pp.13-18, DOI: 10.1109/AVSS.2009.43, 2009.

[17] P. R. L. De Almeida, L. S. Oliveira, A. S. Britto, E. J. Silva and A. L. Koerich, PKLot-A robust dataset for parking lot classification, *Expert Syst. Appl.*, vol.42, no.11, pp.4937-4949, DOI: 10.1016/j.eswa.2015.02.009, 2015.

[18] F. Kurniawan, H. Sajati and O. Dinaryanto, Image processing technique for traffic density estimation, *Int. J. Eng. Technol.*, vol.9, no.2, pp.1496-1503, DOI: 10.21817/ijet/2017/v9i2/170902117, 2017.

[19] F. Kerouh and D. Ziou, Real-time android application for traffic density estimation, *IEEE Access*, vol.6, pp.49896-49901, DOI: 10.1109/ACCESS.2018.2868610, 2018.

[20] T. Ojala, M. Pietikäinen and D. Harwood, A comparative study of texture measures with classification based on feature distributions, *Pattern Recognit.*, vol.29, no.1, pp.51-59, DOI: 10.1016/0031-3203(95)00067-4, 1996.

[21] R. M. Haralick, I. Dinstein and K. Shanmugam, Textural features for image classification, *IEEE Trans. Syst. Man Cybern.*, vol.SMC-3, no.6, pp.610-621, DOI: 10.1109/TSMC.1973.4309314, 1973.

[22] K. Fukushima, Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, *Biol. Cybern.*, vol.36, no.4, pp.193-202, DOI: 10.1007/BF00344251, 1980.

[23] Y. Lecun, L. Bottou, Y. Bengio and P. Ha, Gradient-based learning applied to document recognition, *Processdings of IEEE*, no.11, pp.1-46, 1998.

[24] Y. Cai, G. Xu, A. Li and X. Wang, A novel improved local binary pattern and its application to the fault diagnosis of diesel engine, *Shock Vib.*, DOI: 10.1155/2020/9830162, 2020.