Received: September 4, 2024. Revised: November 1, 2024.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Confidence-Based Traffic Density Estimation Using Square Binary Patterns

Muhammad Ardi Putra[1] **Agus Harjoko[1]*** **Wahyono[1]** **Kanghyun Jo[2]**

[1]*Department of Computer Science and Electronics, Universitas Gadjah Mada, Indonesia*
[2]*Department of Electrical, Electronic and Computer Engineering, University of Ulsan, South Korea*
* Corresponding author's Email: aharjoko@ugm.ac.id

Abstract: Traffic density estimation is very important in the field of Intelligent Transportation Systems thanks to the growing number of road users. This research proposes a new method for estimating traffic density using road surveillance cameras. The approach involves two main steps: block occupancy prediction and traffic density classification. The aggregation of the block occupancy prediction which results in the so-called density coefficient is the first contribution of this paper. The second contribution involves the use of semi-supervised mechanism for classifying traffic density states. Experimental results show that the proposed approach successfully obtained high accuracy for classifying block occupancies on both the proposed dataset and the existing public datasets with the best score of 99.8%. The traffic state classification result itself is satisfactory as it achieved the classification rate of up to 94%. Furthermore, the use of SBP features allows the system to work up to 3.8 times faster than LBP.

Keywords: Intelligent transportation systems, Traffic density estimation, Machine learning, Feature extraction.

1. Introduction

Vehicles are considered one of the primary needs for many people. Due to this reason, it makes perfect sense to observe the increasing number of vehicle ownership year by year. According to the statistical data published by BPS (Badan Pusat Statistik), a central statistics agency of Indonesia, the number of vehicles in the country increases from approximately 119 million to 127, 134, 136, and 142 million respectively from 2017 to 2021 [1], indicating a continuous rise in vehicle count. This high number of vehicles has a significant impact on traffic, leading to increased instances of congestion especially in large cities. Several attempts were made by the government to address this issue, such as Three-in-One rule and Even-Odd license plate policy. The former essentially obliges a car to have at least 3 people inside, while the latter only allows vehicles with even or odd-numbered license plates to enter specific roads based on the current date. Despite these efforts, traffic congestion problem persists. Thus, a more sophisticated system is necessary to be developed to better address the problem.

This is essentially where ITS (Intelligent Transportation System) plays a crucial role. The field of ITS itself has plenty of branches, in which traffic density estimation is the most relevant to be implemented for this case. With the mechanism, the captured information can further be transferred to a traffic control system, i.e., traffic light, so that it will be able to work adaptively according to the road density conditions. When it comes to estimating traffic density, the traditional approach to do so is to implement a device named ILD (Inductive Loop Detector). Unfortunately, this system is considered to be inefficient due to the expensive installation and maintenance cost [2]. Vision-based systems, on the other hand, can be utilized as an alternative to ILD as they are able to address the cost-related issues.

Talking more specifically about vision-based traffic density estimation, one of the simplest way to do so is to use the so-called macroscopic approach, in which it works by analyzing either the entire road region or smaller road patches, as opposed to detecting individual vehicles. The research conducted by [3] used the former, while some research papers that employed the latter method are

Received: September 4, 2024. Revised: November 1, 2024.

[4] and our previous works of [5, 6]. Despite their satisfactory results, there is one drawback that they have in common, namely the discrete prediction mechanism. It was mentioned in [3] that their approach could only predict 3 traffic density classes. Meanwhile, [4-6] work by just predicting whether a particular small road region is being occupied by vehicles or not. This can basically be seen as a problem since it is unable to precisely approximate the traffic density in the frame, considering that their proposed methods are unable to take into account partially-occupied regions.

The objective of this research is to create a traffic density estimation system using block occupancy prediction method which is similar to the one proposed in [4-6], yet with several improvements. The contributions of this paper are listed below:

1) Overall traffic density is calculated based on the aggregation of the block occupancy prediction confidence scores as an approach to account the partially-occupied blocks to make the traffic density estimation system to be more precise. This new measure is named density coefficient.
2) The block occupancy prediction itself is done using SBP (Square Binary Patterns) features, which is a kind of texture feature proposed in our previous work. This new texture feature, which has never been implemented for this case, has been proven to be computationally fast and capable of producing good quality features even under noisy conditions.
3) Traffic density classification threshold is determined using semi-supervised mechanism based on the density coefficient value.

2. Literature review

Generally speaking, traffic density estimation can be divided into two main categories: microscopic and macroscopic approaches. The former is a method where every individual vehicle is detected, whereas the latter works by analyzing the entire road area or a specific sections of it at once. The microscopic approach is considered more complex due to the requirement of using vehicle detection and tracking algorithms. Such a complexity leads to another disadvantage as it often fails when the road is too congested due to occlusions [7].

To avoid all these issues, macroscopic approaches can be utilized as a replacement for the microscopic ones. The method for estimating traffic density with macroscopic approach itself can be divided into three according to the techniques used, namely traditional image processing, machine learning, and deep learning. Several research papers that employ traditional image processing involve [8, 9], in which both of those utilize background subtraction. Not only that, [10, 11] also belong to this category, where the former do the task by taking into account pixel intensity variance, while the latter uses pixel intensity standard deviation. Unfortunately, the four researches mentioned above highly rely on illumination. Hence, their proposed methods will fail when the illumination is terrible, such as during severe weather conditions or in the nighttime.

Machine learning models are also intensively implemented in this field. For instance, the research conducted by [12] utilizes SVM classifier trained on IFLT (Invariant Features of Local Textures) features to determine whether a road region belongs to either empty, low, high, or full class. Next, [3, 13] also do the similar thing, except that both of those perform 3-class classification on TrafficDB dataset [14]. The methods proposed by [15, 4] are even simpler, as they accomplish the task by classifying whether a small road patch is being occupied by vehicles using SVM. Being inspired by this paper, we also conducted the similar research which we presented in [6]. In our case, we compare the performance of LBP, HOG and GLCM features for block occupancy classification task.

However, all attempts to estimate traffic density with machine learning explained above have a common drawback, in which they are only able to categorize a specific road condition into discrete classes. This causes them not to be able to give precise prediction if the current road condition is in between two density classes. Furthermore, in the case of block occupancy classification, the models were not able to distinguish partially-occupied and fully-occupied blocks. Instead, it will regard both of them as being occupied, in which in larger scale this might cause imprecision in the overall road density estimation results.

Deep learning models can also be utilized to do the exact same thing as the machine learning models. One example for this is our another previous research of [5], where we implemented CNN for block occupancy classification task. Despite obtaining better accuracy than machine learning models, yet neural networks can work in decent speed only when GPU is available.

The traffic density estimation system proposed in this paper attempts to overcome the problems encountered by the existing ones discussed above. First, the proposed model, which is designed to predict block occupancy, is actually not completely discrete as it also takes into account the block prediction confidence scores. This is essentially where the name “Confidence-Based Traffic Density

Estimation" comes from. Such a mechanism allows it to distinguish partially-occupied blocks from fully-occupied blocks. With this idea, the overall traffic density can be estimated more precisely. Next, the utilization of SBP features from our previous research guaranteed the proposed system to be able to work in real-time since the feature extraction algorithm works even faster than the standard LBP.

3. Proposed method

Generally speaking, the main idea of this research is to perform block classification, i.e., predicting whether a particular road region is occupied by vehicles or not. Next, the occupancy prediction results of all blocks in each frame are going to be aggregated, and the resulting value will serve as the basis for overall traffic density classification. The detailed steps are shown in Fig. 1.

3.1 Data acquisition

As shown in Fig. 1, the experimental procedure in this research can be divided into two: training and testing phase. The first step of the training phase is to gather traffic video recordings, which were captured from three different surveillance cameras located in BPK (Badan Pemeriksa Keuangan) Junction, Permata Junction, and Mirota Junction, all of which are in Yogyakarta, Indonesia. Each of these cameras records video from three different times of day: morning, daytime, and night. Thus, there will be 9 traffic videos to experiment with in total. Fig. 2 displays video frames captured by the three different cameras, in which all these traffic videos and time would later undergo the exact same processes. The entire processes to be done on these videos are explained coherently in the subsequent sub-sections. Additionally, all videos used in this research were provided by The Ministry of Transportation of Yogyakarta and were gathered with permission.

3.2 Block occupancy prediction dataset

After collecting the traffic videos, the next step to do was to create block classification datasets from the 9 traffic videos by performing random cropping on the video frames. Since the first objective is to train a model for predicting whether a block is being occupied by vehicles or not, hence there were two classes required to be created, namely "occupied" and "unoccupied" blocks. Each of the two classes comprised of 250 images. The size of the blocks varied depending on their distance from the camera. This was essentially done since farther areas appear smaller than the closer ones. Thanks to this reason, blocks were set to be larger for the region closer to the camera as compared to the farther ones. This kind of approach ensures that the vehicle size appearing in the block to be proportional to the block size. In fact, this is different from [4-6], in which all of those were using fixed block size for the entire road area regardless of its distance from the camera.

3.3 Data augmentation

Having a large dataset is extremely important when it comes to training a machine learning model. In this case, data augmentation was performed to increase the number of blocks to be used for training a classifier. For each video, the blocks count was increased from 500 to 1000 through the application of random affine transformation, brightness change, horizontal flip, and horizontal shear to every block in the original dataset.

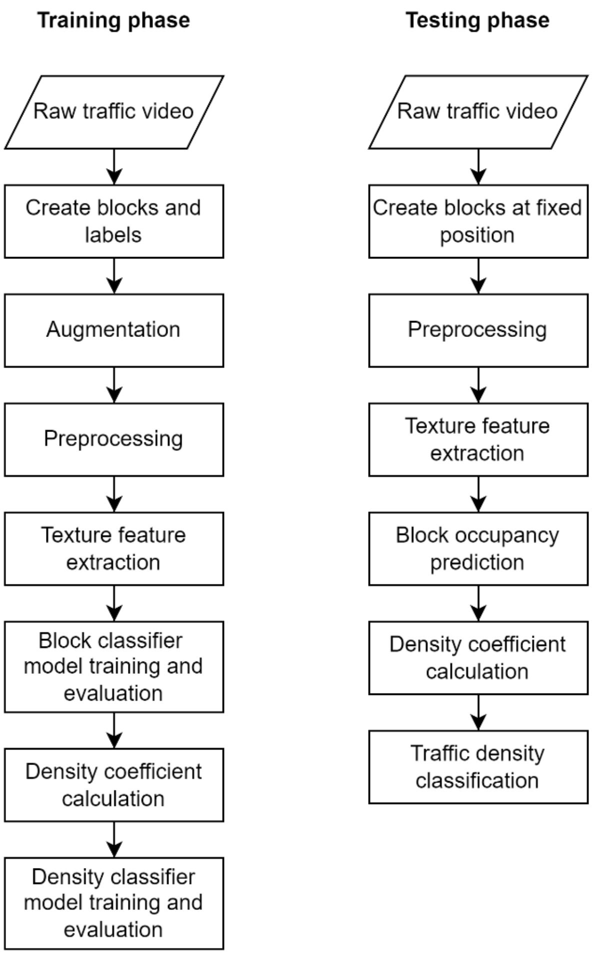

Figure. 1 Training and testing flowchart

Received: September 4, 2024. Revised: November 1, 2024.

3.4 Preprocessing

Every single block image, both the original and the augmentation result, needed to be preprocessed prior to getting into the feature extraction stage. The initial preprocessing step was to convert the image to grayscale. Such a conversion was done since this research focused on the use of textural features which completely ignores color information. Next, all blocks were resized to 32×32 regardless of their original dimensions made in the block cropping stage. Such a uniform block size allowed us to use the same feature extraction parameters for the entire dataset. This process is considered important as it might affect the fairness of the final result. Finally, the last preprocessing step to do was smoothing. In this case, average smoothing was employed with a kernel size of 5×5. This type of smoothing was chosen since it has the lowest computational complexity among the other algorithms such as Gaussian, median, or bilateral smoothing.

3.5 Texture feature extraction

This research compared nine different texture feature extraction algorithms, namely LBP (Local Binary Patterns), LBP^{riu2} (Rotation-Invariant Uniform LBP), LBP^{u2} (Uniform LBP), LBP^{ri} (Rotation-Invariant LBP), MS-LBP (Multi-Scale LBP), LTP (Local Ternary Patterns) [16], FbLBP (Feature-based LBP) [17], ILBP (Improved LBP) [18], and SBP (Square Binary Patterns) which is the new texture feature extraction algorithm proposed in our previous work. All these algorithms would later be used independent of one another, aiming to determine the texture features that perform best in terms of both accuracy and computational complexity. After feature extraction is done, the next step to do was to divide the resulting feature, which has the exact same dimension as the original block size (32×32) into non-overlapping four sub-blocks of size 16×16. The histograms of all sub-blocks would later be concatenated in which it acted as the feature vector of the corresponding block. Such a histogram concatenation method theoretically allows the resulting feature vector to contain a little amount of spatial information. In the subsequent stage, this feature vector would be used as the input of the classifier model.

Talking more specifically about the histogram, the number of bins for each sub-block was set to 40. This can actually be thought of as representing the number of unique colors generated using the color vquantization mechanism. In fact, the original number of different colors should have been 256. However, this number was not chosen considering the potential for overfitting due to the large feature vector size. It is worth noting that the value of 256 is obtained by assuming that the LBP parameter of P, i.e., the number of sampling points, is set to 8.

Figure. 2 Images captured from BPK Junction, Permata Junction, and Mirota Junction (from top to bottom, respectively) as well as the block occupancy prediction results (red: occupied, green: unoccupied)

Figure. 3 Several examples of occupied (first row) and unoccupied (second row) blocks from Permata Junction video in the morning

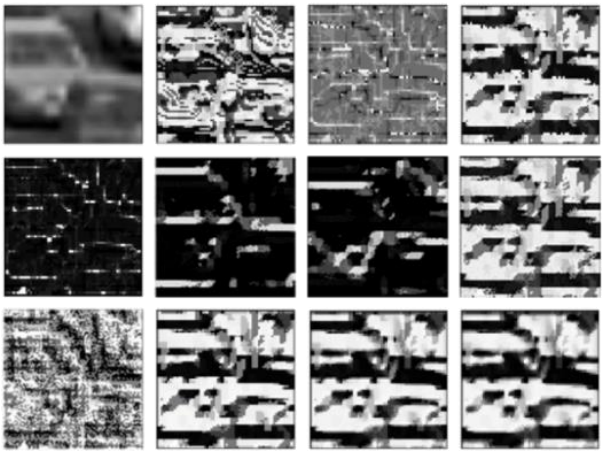

Figure. 4 First row: original image of an occupied block from QMUL Junction 1 dataset [19], LBP, LBP^{riu2}, LBP^{u2}. Second row: LBP^{ri}, LTP lower pattern, LTP upper pattern, FbLBP. Third row: ILBP, SBP1, SBP2, SBP3 (all rows are mentioned from left to right)

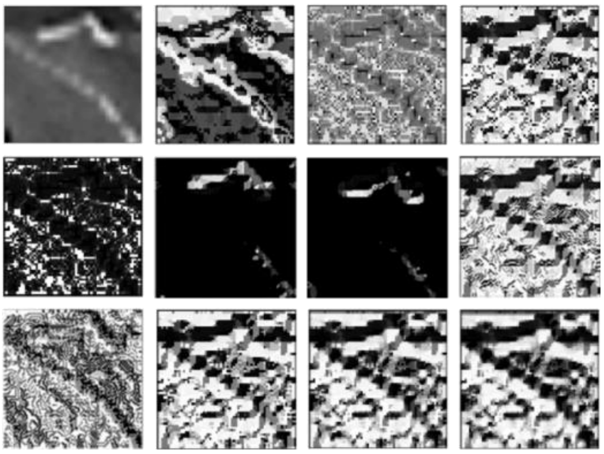

Figure. 5 First row: original image of an unoccupied block from QMUL Junction 1 dataset [19], LBP, LBP^{riu2}, LBP^{u2}. Second row: LBP^{ri}, LTP lower pattern, LTP upper pattern, FbLBP. Third row: ILBP, SBP1, SBP2, SBP3 (all rows are mentioned from left to right)

Even though the parameters of the feature extraction algorithms had been set to be the same, this does not necessarily mean that all of those will produce the exact same feature vector length thanks to the nature of those algorithms. In the case of LBP, FbLBP, ILBP and SBP, including SBP with level 1, 2 and 3 (abbreviated as SBP1, SBP2, and SBP3), it was indeed possible to set it fixed to a specific number. However, the feature vector lengths of LBP^{u2}, LBP^{ri}, and LBP^{riu2} will always have the feature vector length of 59, 36 and 10, respectively. The feature vector of MS-LBP, on the other hand, is generated by concatenating the histogram of each scale. Thus, if the number of bins is set to 256 and the number of scales is set to 3 (MSLBP3), then the feature vector dimension is going to be 768. Next, the feature vector length of LTP will always be twice as large as what were set thanks to the concatenation of the lower and upper patterns.

3.6 Block classifier training

After feature extraction and representation had been done, the next step to do was to train a classifier for predicting block occupancy. SVM was the machine learning model chosen for this task since it is considered to be more advanced than the others while at the same time having considerably fast predicting time, especially when it is compared to the lazy-learner of KNN-based model. The ability of SVM to handle image classification tasks is also proven since there had been plenty of research papers that employed this model, such as [20, 21] to perform face and plant recognition, respectively.

The SVM classifier itself has several adjustable parameters. In the experiment, the SVM kernel was set to RBF (Radial Basis Function), allowing the model to capture nonlinearity in the dataset. Next, the C parameter was set to 1.0 while the gamma was set to be inversely proportional to the number of features, which implies that different LBP variants might be treated by SVM of different gamma.

The block classifier training is going to be done using K-fold cross validation method with K=5 and the train-validation split proportion of 80:20. It is worth noting that this splitting was only applied to the original dataset, leaving the augmented blocks only used for training. This essentially implies that the validation data was always coming from the original set. Later in the testing phase, the model was then employed to predict the occupancy of every block within a single frame.

3.7 Density coefficient calculation

As all block occupancies in a frame had been predicted, now that the density of the traffic could be estimated using the so-called density coefficient D_{coef}, i.e., a new measure proposed in this research. The value of the coefficient itself can be obtained using Eqs. (1) and (2).

In the first equation, B is the total number of blocks within a single frame, while $NZO_{(conf_i)}$ and $ZO_{(conf_i)}$ denote the confidence of the i-th occupied block (Non-Zero Occupancy) and i-th unoccupied block (Zero Occupancy), respectively. The term NZO and ZO themselves were adopted from [15]. Next, the Kronecker delta function δ is employed to determine whether the corresponding confidence value will be taken into account or not. For instance, if a block is predicted as occupied (NZO), then the second term of the Eq. (1) is going to be completely ignored. Otherwise, if the prediction made by the SVM is unoccupied (ZO), the first term of the equation will cancel out. Using this equation, larger number of occupied blocks causes density coefficient to be larger as well. The value range of D_{coef} is basically bounded by the number of blocks. For example, if there are 20 blocks in a single frame, hence the minimum and maximum possible value are -20 to 20 (inclusive). However, it is worth noting that it might be quite impossible to achieve the two values since the block classifier model might not have a perfect confidentiality score when making the predictions.

$$D_{coef} = \left(\sum_{i=1}^{B}\left(NZO_{conf_i}\right) \times \delta(pred_i, NZO)\right) - \left(\sum_{i=1}^{N}(ZO_{conf_i}) \times \delta(pred_i, ZO)\right) \quad (1)$$

$$\delta(pred, expected) = \begin{cases} 1 & if\ pred = expected \\ 0 & otherwise \end{cases} \quad (2)$$

The purpose of using confidence score for calculating density coefficient over the rounded block prediction probability is to accommodate unobvious predictions. For instance, whenever a block is only half-occupied, the confidence score of the prediction will be low. Hence, this method is theoretically able to make road density approximation better without needing to employ smaller-sized blocks as proposed in our previous works of [5, 6].

3.8 Density coefficient calculation

The density coefficient value was then used as the basis for traffic density classification, in which it acted as the feature for a logistic regression model. The idea of this stage was to train the machine learning classifier using semi-supervised method such that it would later be able to predict whether the current traffic is categorized into light, medium, or heavy density. Semi-supervised mechanism was utilized because it was difficult to create the 3-class classification dataset which is labeled based on traffic density coefficients, considering that d_{coef} is something that highly relies on the SVM predictions rather than being able to be examined by visual inspection alone. To illustrate this, a completely occupied block should ideally be predicted with 100% confidence score, but in practice, the score might be less for no specific reason.

The logistic regression training was done by taking several traffic images which the density was manually chosen according to the criteria mentioned in [10]. It is written in the manuscript that a traffic state is categorized into light whenever its occupancy is less than 40%. If the occupancy is greater than 65%, then it belongs to heavy density state. Any occupancy percentage within this range (between 40% and 65%, inclusive) is categorized as medium density. The selected images acted as the initial dataset for the density classification task. Next, the traffic images in training data were then iteratively predicted using the logistic regression model, in which the prediction results which the confidence were higher than 0.9 would become the new training data. Finally, as the iteration completed, the model would be ready for predicting whether a traffic video frame belongs to light, medium, or heavy class.

4. Experimental results and discussions

4.1 Block occupancy classification results

All the processes that were done followed the steps explained in the previous chapter. Nevertheless, some parameter variations were experimented in order to obtain the highest possible accuracy. The experimental configuration discussed in the methodology can be perceived as the baseline.

Initially, the classification task was done without using augmented images. This was basically done to find out whether the resulting accuracy is already decent even when the SVM model is trained only using 250 images from each class in every dataset. The result showed that the accuracies obtained by all features, including the SBP that was first proposed in our previous work, were actually good enough.

However, most of results were still getting better when data augmentation was performed. This definitely makes sense since by using this technique a classifier model can learn better thanks to the wider range of image variations. The resulting accuracy scores after augmentation was used are summarized in Table 1. The cell colored in blue in the table indicates the best accuracy score by the proposed feature extraction method from our previous work (SBP), whereas the cells colored in orange and green denote the results worse and better than the proposed method, respectively.

According to Table 1, it is evident that the majority of the best results, especially in Permata and Mirota Junction, were achieved with the datasets captured during daytime. Subsequently, the ones captured in the morning and at night produced comparatively lower scores, respectively. This kind of behavior occurred since apparently illumination played a very important role for this case, in which the road is well-illuminated during the daytime, allowing the model to obtain high accuracy in this dataset. Furthermore, the classification score obtained in the morning was not as high as that of during the daytime. This was actually because sunlight reflections commonly occurred which resulted in misclassifications on predicting block occupancy

Table 1. Block classification results on different dataset with data augmentation

Features	BPK			Permata			Mirota		
	Morning	Daytime	Night	Morning	Daytime	Night	Morning	Daytime	Night
LTP [16]	.962	.952	.938	.978	.992	.930	.988	.992	.922
FbLBP [17]	.958	.958	.932	.954	.996	.900	.970	.992	.914
ILBP [18]	.912	.944	.902	.950	.974	.924	.954	.970	.890
LBP [22]	.958	.956	.936	.952	.996	.904	.968	.992	.914
LBP^{riu2} [22]	.928	.922	.928	.934	.986	.888	.966	.968	.822
LBP^{u2} [22]	.966	.964	.964	.980	.998	.950	.984	.998	.946
LBP^{ri} [22]	.924	.894	.918	.934	.974	.882	.962	.970	.762
MSLBP2 [22]	.942	.942	.914	.910	.992	.872	.952	.986	.838
MSLBP3 [22]	.938	.942	.914	.928	.994	.872	.948	.990	.846
SBP1 (proposed)	.946	.960	.932	.954	.998	.914	.968	.99	.908
SBP2 (proposed)	.940	.946	.916	.950	.986	.890	.962	.984	.904
SBP3 (proposed)	.948	.942	.890	.944	.982	.884	.942	.968	.890

Table. 2 The amount of time required to process 250 blocks of size 32×32 pixels (in seconds)

Features	Time
LTP [16]	29.3
FbLBP [17]	43.4
ILBP [18]	23.1
LBP [22]	25.9
LBP^{riu2} [22]	21.4
LBP^{u2} [22]	28.3
LBP^{ri} [22]	59.9
MSLBP2 [22]	45.4
MSLBP3 [22]	69.7
SBP1 (proposed)	6.8
SBP2 (proposed)	10.8
SBP3 (proposed)	15.1

There are some other interesting things in Table 1, especially the accuracies from BPK Junction in the morning and during the daytime. Different from Permata and Mirota Junction, in the case of BPK Junction the accuracy of both times appears to be stable. Moreover, there are also several cases where the results in the morning are better than those of during the daytime instead. It was suspected that this happened since the camera position in BPK Junction was placed quite high which allowed it to be more robust against low-angle sunlight and thus causing the images to have more consistent illumination. On the other hand, the low accuracy obtained during the night in all junctions occurred probably due to the model relies on the appearance of vehicle headlights rather than the vehicles themselves.

According to the same table, it can be observed that the accuracy scores of SBPs are comparable to the existing LBP-based features. At the same time, our experiment on computation time shown in Table 2 as well as in our previous research show that the computational complexity of SBP is much lower than the others, in which it is going to be suitable for real-time use. Thanks to this reason, SBP variants are going to be the only features to be taken into account for the next experiments.

The objective of the remaining experiments was to perform parameter tuning so that SBP could obtain the highest possible classification accuracy score. The first parameter to be experimented with is the number of bins per patch. In the previous experiments the value of this parameter was set to 40. This means that since there were 4 patches for a single block, hence each block would have a vector representation of length 160. Here the number of bins of 16, 24, 32, 40, 48, 56, 64 and 72 were used, in which they produced feature vectors of size 64, 96, 128, 160 192, 224, 256, and 288 for each number of bins, respectively.

The results presented in Table 3 indicate that SBP1 performed the best with 48 and 64 bins, while the optimal number of bins for both SBP2 and SBP3 is 56. This suggests that the previously set number, i.e., 40 bins, was insufficient for storing important information to be learned by the SVM. Consequently, in the upcoming experiment 48 number of bins would be used for SBP1 since it is smaller than 64, which potentially reduces computational complexity while maintaining high accuracy. As for SBP2 and SBP3, 56 number of bins will be used in the next trials.The next experiment to be done was related to the number of patches, i.e., sub-blocks. In the previous chapter it was discussed that the default number of patches to divide the block was 4 as shown in Fig. 4, in which it was intended to capture a little amount of spatial

Received: September 4, 2024. Revised: November 1, 2024.

information within the block. With this mechanism, it was expected that the classifier would have more features to learn from. In this particular case, the objective was to find out whether the accuracy score obtained by SBP could still further be improved by changing the number of patches to 1, 4, 9 and 16.

According to the results displayed in Table 4, 4 number of patches almost always obtained the best score, in which it implies that our baseline parameter was already optimal. Taking a closer look at the three tables, it can be seen that a parameter configuration of 16 blocks never achieved the highest accuracy among all other configurations. This was probably because too many patches resulted in a very long feature vector dimension which potentially causing the block classifier model to suffer from overfitting. Another observation from these tables is that SBP3 never exclusively outperformed both SBP1 and SBP2 in terms of accuracy. Therefore, in the next experiment, the parameter search space was reduced by excluding SBP3.

Different SVM kernels were employed for the upcoming experiments which the results are shown in Table 5. In fact, one of the advantages of using SVM is that there are several different kernels possible to be used, in which it can be altered according to the pattern in the data distribution. A dataset is said to be linearly separable whenever it can be classified with high accuracy using linear kernel. Despite this clear definition, it is actually not quite feasible to manually examine the data distribution in terms of its separability. Thus, conducting experiments of different kernels is a good idea to better understand the data distribution as well as to find out which kernel can achieve the highest accuracy score. In our case, it seemed like the blocks had feature vectors which the distribution was somewhat complex. This is probably the reason that an SVM of linear kernel did not work as well as the non-linear kernel of RBF and polynomial. In addition to the polynomial kernel, here its degree was set to 3.

According to these discussions, it can be concluded that the highest accuracy for the block classification dataset is achieved by using either SBP1 or SBP2. Specifically, SBP1 and SBP2 could yield the best result when the number of histogram bins were set to 48 and 56, respectively. When it comes to the number of patches, 4 is the best number for the two SBPs. Next, the resulting features need to be employed to train an SVM model with RBF kernel. These selected parameters would later be used for classifying traffic density which is going to be discussed in the subsequent section.

The final block classification accuracy results after completing parameter tuning are summarized in Table 5 (highlighted in blue). Not only the ones using our own dataset, but further experiments were also conducted independently on TrafficDB [14] and QMUL Junction 1 [19] datasets which the results are shown in Table 6. It is worth mentioning that all these datasets underwent the exact same blocks collection, augmentation and preprocessing procedure as explained earlier in Chapter 3.

According to Table 6, it is clearly seen that SBP is able to outperform existing image features in most cases. The performance of SBP both in BPK Junction in the daytime and in TrafficDB in rainy weather is able to match the performance of LBP^{u2} with the accuracy of 96.4% and 99.8%, exceeding the performance of all other features in these two datasets. The LBP^{u2} itself is essentially a texture feature that has an impressive discriminative power. This is proven by the previous experimental results shown in Table 1 that it consistently outperforms all other features. However, with a thorough parameter tuning, SBP is actually able to perform as well as LBP^{u2}.

The performance of LBP^{u2} and SBP also look similar in TrafficDB dataset in clear weather with the accuracy of 99%. Despite this remarkable result, the performance of HOG is even better in this case. It is interesting to see this because HOG is originally proposed for extracting edge orientation features rather than textures. However, since the superiority of HOG is not validated in other dataset, it can not be concluded as the most proper feature to be used for such a road block occupancy classification task.

Next, the performance of SBP features under overcast weather in TrafficDB dataset is also considered remarkable as it ranked second behind LBP^{u2} with the gap of only 0.2%. It is worth mentioning that in this case the SBP variants that perform best are SBP2 and SBP3 for some reasons. This exact same accuracy score is also obtained by LBP+HOG features proposed by [4]. This essentially implies that SBP is more efficient as it is able to obtain the same accuracy without needing to be combined with other features. Meanwhile, it is necessary to acknowledge that SBP might not work as well as the existing features in other dataset as it somehow only ranked fourth when classifying block occupancy on Junction 1 dataset. Despite this fact, it is still worth to appreciate SBP as the resulting accuracy of 98.2% in Junction 1 dataset is actually not bad either. Moreover, the fast computation speed of SBP shown back in Table 2 can certainly compensate for this slight dip in accuracy.

Received: September 4, 2024. Revised: November 1, 2024.

Table 3. Block classification results obtained by SBP1, SBP2 and SBP3 using different number of bins

	SBP1									SBP2									SBP3								
	BPK			Permata			Mirota			BPK			Permata			Mirota			BPK			Permata			Mirota		
No of Bins	Morning	Daytime	Night	Morning	Daytime	Night	Morning	Daytime	Night	Morning	Daytime	Night	Morning	Daytime	Night	Morning	Daytime	Night	Morning	Daytime	Night	Morning	Daytime	Night	Morning	Daytime	Night
16	.926	.942	.896	.952	.982	.892	.938	.974	.880	.910	.928	.896	.944	.982	.874	.920	.948	.896	.904	.916	.876	.918	.966	.842	.894	.916	.880
24	.902	.944	.914	.952	.988	.914	.938	.986	.902	.902	.930	.910	.950	.984	.888	.940	.976	.902	.904	.922	.882	.948	.976	.866	.906	.962	.904
32	.900	.946	.926	.952	.998	.898	.956	.990	.900	.896	.934	.908	.946	.984	.890	.938	.988	.896	.902	.928	.896	.944	.980	.860	.928	.966	.902
40	.908	.960	.932	.954	.998	.914	.960	.990	.908	.904	.946	.916	.950	.986	.890	.938	.984	.904	.890	.942	.890	.944	.982	.884	.928	.968	.890
48	.904	.964	.942	.956	.994	.918	.960	.990	.904	.900	.942	.944	.944	.990	.898	.950	.992	.90	.90	.948	.898	.938	.982	.882	.930	.988	.900
56	.906	.950	.938	.954	.998	.910	.954	.990	.906	.912	.954	.928	.954	.992	.910	.950	.992	.912	.892	.944	.918	.950	.986	.890	.936	.990	.892
64	.900	.956	.928	.958	.996	.910	.964	.992	.900	.912	.950	.938	.948	.990	.922	.950	.994	.912	.886	.950	.914	.950	.984	.898	.932	.988	.886
72	.908	.952	.938	.950	.998	.916	.962	.988	.908	.912	.952	.940	.950	.992	.914	.952	.990	.912	.890	.944	.910	.950	.986	.892	.926	.986	.890

Table 4. Block classification results obtained by SBP1, SBP2 and SBP3 using different number of patches

	SBP1									SBP2									SBP3								
	BPK			Permata			Mirota			BPK			Permata			Mirota			BPK			Permata			Mirota		
No of Patches	Morning	Daytime	Night	Morning	Daytime	Night	Morning	Daytime	Night	Morning	Daytime	Night	Morning	Daytime	Night	Morning	Daytime	Night	Morning	Daytime	Night	Morning	Daytime	Night	Morning	Daytime	Night
1	.950	.932	.938	.952	.990	.926	.952	.996	.856	.954	.940	.942	.938	.990	.906	.946	.990	.890	.954	.932	.950	.936	.990	.894	.932	.990	.868
4	.904	.964	.942	.956	.994	.918	.960	.990	.904	.912	.954	.928	.954	.992	.910	.950	.992	.912	.892	.944	.918	.95	.986	.890	.936	.990	.892
9	.870	.944	.880	.950	.996	.882	.958	.982	.870	.854	.932	.870	.940	.986	.852	.946	.976	.854	.828	.914	.864	.932	.968	.818	.920	.962	.828
16	.848	.936	.902	.942	.990	.852	.948	.986	.848	.820	.904	.876	.920	.976	.818	.922	.956	.820	.832	.900	.836	.906	.960	.790	.904	.952	.832

Table 5. Block classification results obtained by SBP1 and SBP2 using different SVM kernels

SVM Kernel	SBP1									SBP2								
	BPK			Permata			Mirota			BPK			Permata			Mirota		
	Morning	Daytime	Night	Morning	Daytime	Night	Morning	Daytime	Night	Morning	Daytime	Night	Morning	Daytime	Night	Morning	Daytime	Night
RBF	.946	.964	.942	.956	.994	.918	.960	.990	.904	.944	.954	.928	.954	.992	.910	.950	.992	.912
Linear	.792	.922	.864	.950	.994	.846	.924	.982	.792	.776	.902	.802	.940	.996	.788	.920	.978	.776
Sig-moid	.486	.862	.856	.908	.986	.768	.936	.988	.486	.648	.842	.834	.868	.918	.724	.930	.980	.648
Poly-nomial	.912	.956	.936	.956	.996	.916	.960	.994	.912	.920	.946	.918	.956	.992	.918	.954	.984	.920

Table 6. Accuracy comparison between the proposed and state-of-the-art methods

Features	Junction 1	TrafficDB Clear	TrafficDB Overcast	TrafficDB Rain	BPK Daytime
LBP+HOG (proposed in [4])	.984	.986	.992	.996	.932
CNN (proposed in [5])	.978	.980	.978	.970	.898
HOG [23] (proposed in [6])	.978	.992	.984	.988	.962
GLCM [24] (proposed in [6])	.930	.948	.956	.986	.902
Variance (proposed in [10])	.958	.940	.952	.956	.814
Standard Deviation (proposed in [11])	.954	.948	.954	.956	.812
LTP [16]	.978	.976	.968	.968	.952
FbLBP [17]	.982	.978	.982	.992	.958
ILBP [18]	.954	.988	.980	.982	.944
LBP [22]	.988	.978	.982	.990	.956
LBP^{riu2} [22]	.958	.882	.878	.962	.922
LBP^{u2} [22]	.992	.990	.994	.998	.964
LBP^{ri} [22]	.954	.856	.868	.954	.894
MSLBP2 [22]	.966	.966	.974	.988	.942
MSLBP3 [22]	.974	.974	.974	.992	.942
1st and 2nd order statistics (proposed in [25])	.986	.990	.988	.984	.906
SBP1 (proposed)	.982	.990	.990	.998	.964
SBP2 (proposed)	.976	.984	.992	.992	.954
SBP3 (proposed)	.958	.988	.992	.986	.944

Table 7. Traffic density classification results

Dataset		SBP1 Test Accuracy	SBP2 Test Accuracy
BPK	Morning	.920	.900
	Daytime	.820	.920
	Night	.920	.940
Permata	Morning	.860	.800
	Daytime	.940	.880
	Night	.920	.840
Mirota	Morning	.840	.860
	Daytime	.900	.920
	Night	.920	.900

4.2 Traffic density classification results

It was previously discussed that after block occupancy prediction had been done, the subsequent step to do was to perform traffic density classification based on the value of density coefficient. The equation to obtain d_{coef} itself was mentioned earlier in Eq. (1).

The proposed semi-supervised method for classifying traffic density performed well in our case as summarized in Table 7. It can be inferred from the table that apparently both SBP1 and SBP2 had the similar classification rate due to the fact that in some cases the former obtained higher accuracy and in some other cases SBP2 was superior. It is important to note that the logistic regression models which implemented the semi supervised mechanism were unique for each dataset. This essentially means that the classifier used for Permata Junction in the morning was different to the one used for BPK Junction at night as well as the others even though all of those were using the exact same SBP feature extraction configuration. The reason behind this was that each dataset has its unique density coefficient distribution.

5. Conclusion

The traffic density estimation system based on macroscopic approach proposed in this paper has been proven to perform very well in various circumstances. This notion is based on the fact that it successfully overcome the problems faced by the previous researchers. It was previously mentioned that the systems based on traditional image processing highly rely on the illumination. Our proposed method, on the other hand, is able to work even in low light conditions. Using the best parameter configuration, the block occupancy classification accuracy obtained in the nighttime reached up to 94.2% which occurred in BPK Junction. However, it is necessary to acknowledge that this accuracy score is still not as good as the one achieved during the day, which reached up to 99.6% as demonstrated in Permata Junction.

Regarding the overall traffic density estimation, it was found that the density coefficient calculated from block prediction confidence scores is able to represent the density of the entire road very well. This notion is proven by the fact that the traffic state classification done based solely on the density coefficient successfully obtained excellent accuracy. It was noted that the logistic regression model is able to do the three-class classification with the accuracy of up to 94% both during the day and in the nighttime.

When it comes to computation speed, it was also proven that the SBP feature extraction algorithm from our previous research is able to work up to 3.8 times faster than the conventional LBP. This enables the entire traffic density estimation system to operate in real-time without the need for a GPU. Thus, this proposed method also successfully overcome the problem which commonly occurred in deep learning-based models.

Despite the mentioned advantages, the block occupancy classification mechanism had a limitation where it sometimes made misclassifications, especially in a reflective road condition which usually occurs in the morning, and during the night when there is limited amount of illumination. In order to address this issue, authors recommend future research to combine SBP features with non-texture features, such as color intensity or edge. This kind of approach may be able to improve model accuracy in such conditions since color intensity features might capture a more detailed information regarding the pixel brightness, thus, allowing the machine learning model to recognize the presence of vehicles in different lighting conditions. Meanwhile, the use of edge features might provide additional information regarding soft edges that appear when the light is minimal.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Muhammad Ardi Putra: Conceptualization, Methodology, Software, Formal analysis, Resources, Data curation, Writing – Original draft, Writing – Review & editing, Agus Harjoko: Valiadtion, Supervision, Funding acquisition, Project administration, Writing – Review & Editing, Wahyono: Conceptualization, Methodology, Validation, Supervision, Writing – Review & Editing, Kanghyun Jo: Resources, Validation, Supervision.

Acknowledgments

This research is supported by Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi of Indonesia through PKPI-PMDSU Scholarship Program with Grant Number 2069/UN1/DITLIT/PT.01.03/2024.

References

[1] Badan Pusat Statistik, Statistik Transportasi Darat. BPS-Statistics Indonesia, 2021.

[2] T. Tangkocharoen and A. Srisuphab, “Vehicle detection on a pint-sized computer”, In: *Proc. of 2017 9th International Conference on Knowledge and Smart Technology: Crunching Information of Everything*, pp. 40-44, 2017, doi: 10.1109/KST.2017.7886078.

[3] M. A. Dinani, P. Ahmadi, and I. Gholampour, “Efficient Feature Extraction for Highway Traffic Density Classification”, In: *Proc. of Iranian Conference on Machine Vision and Image Processing, MVIP*, pp. 14-19, 2015, doi: 10.1109/IranianMVIP.2015.7397494.

[4] D. Prasad, K. Kapadni, A. Gadpal, M. Visave, and K. Sultanpure, “HOG, LBP and SVM based Traffic Density Estimation at Intersection”, In: *Proc. of 2019 IEEE Pune Section International Conference*, 2019, doi: 10.1109/PuneCon46936.2019.9105731.

[5] M. A. Putra, A. Harjoko, and Wahyono, “Estimation of Traffic Density Using CNN with Simple Architecture,” In: *Proc. of International Workshop on Intelligent Systems (IWIS) 2022*, 2022, doi: 10.1109/IWIS56333.2022.9920811.

[6] A. Harjoko, Wahyono, N. Susyanto, I. Endrayanto, M. A. Putra, and D. A. Kamil, “Traffic Density Estimation Based on Block Occupancy Classification”, *Innovative Computing, Information and Control Express Letters*, Vol. 17, No. 11, pp. 1187-1194, 2023, doi: 10.24507/icicel.17.11.1187.

[7] O. Asmaa, K. Mokhtar, and O. Abdelaziz, “Road Traffic Density Estimation using Microscopic and Macroscopic Parameters”, *Image Vis Comput*, Vol. 31, No. 11, pp. 887-894, 2013, doi: 10.1016/j.imavis.2013.09.006.

[8] F. Kurniawan, H. Sajati, and O. Dinaryanto, “Image Processing Technique for Traffic Density Estimation”, *International Journal of Engineering and Technology*, Vol. 9, No. 2, pp. 1496-1503, 2017, doi: 10.21817/ijet/2017/v9i2/170902117.

[9] M. S. Uddin, A. K. Das, and Md. A. Taleb, “Real-time Area Based Traffic Density Estimation by Image Processing for Traffic Signal Control System: Bangladesh Perspective”, In: *Proc. of 2nd International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT)*, 2015.

[10] K. Garg, S.-K. Lam, T. Srikanthan, and V. Agarwal, “Real-time Road Traffic Density Estimation using Block Variance”, In: *Proc. of 2016 IEEE Winter Conference on Applications of Computer Vision*, 2016, doi: 10.1109/WACV.2016.7477607.

[11] F. Kurniawan, H. Sajati, and O. Dinaryanto, “Adaptive Traffic Controller Based On Pre-Timed System”, *Telecommunication Computing Electronics and Control*, Vol. 14, No. 1, pp. 56-63, 2016, doi: 10.12928/telkomnika.v14i1.2798.

[12] P. Janney and G. Geers, “Framework for Illumination Invariant Vehicular Traffic Density Estimation”, *Advances in Image and Video Technology*, Vol. 5414, pp. 531-541, 2009, doi: 10.1007/978-3-540-92957-4_46.

[13] H. Hu, Z. Gao, Y. Sheng, C. Zhang, and R. Zheng, “Traffic Density Recognition Based on Image Global Texture Feature”, *International Journal of Intelligent Transportation Systems Research*, Vol. 17, No. 3, pp. 171-180, 2019, doi: 10.1007/s13177-019-00187-0.

[14] A. B. Chan and N. Vasconcelos, “Classification and Retrieval of Traffic Video Using Auto-Regressive Stochastic Processes”, In: *IEEE Intelligent Vehicles Symposium, Proceedings*, pp. 771-776, 2005, doi: 10.1109/IVS.2005.1505198.

[15] Z. Li, E. Tan, J. Chen, and T. Wassantachat, “On Traffic Density Estimation with a Boosted SVM Classifier”, In *Proc. of Digital Image Computing: Techniques and Applications*, pp. 117-123, 2008, doi: 10.1109/DICTA.2008.30.

[16] X. Tan and B. Triggs, “Enhanced Local Texture Feature Sets for Recognition Under Difficult Lighting Conditions”, *IEEE Transactions on Image Processing*, Vol. 19, No. 6, pp. 1635-1650, Jun. 2010, doi: 10.1109/TIP.2010.2042645.

[17] Z. Pan, Z. Li, H. Fan, and X. Wu, “Feature Based Local Binary Pattern for Rotation Invariant Texture Classification”, *Expert Syst Appl*, Vol. 88, pp. 238-248, 2017, doi: 10.1016/j.eswa.2017.07.007.

[18] 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 and Vibration*, 2020, doi: 10.1155/2020/9830162.

[19] D. Russell, “Junction Dataset.”, Accessed: Dec. 13, 2023. [Online]. Available: http://www.eecs.qmul.ac.uk/~sgg/QMUL_Junction_Datasets/Junction/Junction.html.

[20] R. Ravi, S. V. Yadhukrishna, and R. Prithviraj, “A Face Expression Recognition Using CNN LBP”, In: *Proc. of the 4th International Conference on Computing Methodologies and Communication*, pp. 684-689, 2020, doi: 10.1109/ICCMC48092.2020.ICCMC-000127.

[21] M. Turkoglu and D. Hanbay, "Plant Recognition System Based on Deep Features and Color-LBP Method", In: *Proc. of 27th Signal Processing and Communications Applications Conference*, 2019, doi: 10.1109/SIU.2019.8806592.

[22] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution Gray Scale and Rotation Invariant Texture Classification with Local Binary Patterns", *IEEE Trans Pattern Anal Mach Intell*, Vol. 24, No. 7, pp. 971-981, 2002, [Online]. Available: http://www.ee.oulu.fi/research/imag/texture

[23] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection", In: *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 886-893, 2005, doi: 10.1109/CVPR.2005.177.

[24] 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, 1973, doi: 10.1109/TSMC.1973.4309314.

[25] T. Wassantachat, Z. Li, J. Chen, Y. Wang, and E. Tan, "Traffic Density Estimation with On-line SVM Classifier", In: *Proc. of 6th IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 13-18, 2009, doi: 10.1109/AVSS.2009.43.