International Journal Of Artificial Intelegence Research ISSN: 2579-7298
Vol 5, No 1, June 2021, pp. 78-90

Intelligent Traffic Monitoring Systems: Vehicle Type Classification Using Support Vector Machine

Ika Candradewi[1] , Agus Harjoko[2], , and Bakhtiar Alldino Ardi Sumbodo[3]

[1,2,3]*Universitas Gadjah Mada, FMIPA Sekip Utara Bulaksumur BLS 21, Yogyakarta, Indonesia*
[1] *ika.candradewi@ugm.ac.id*, [2]aharjoko@ugm.ac.id,[3] b.alldino.as@ugm.ac.id*
** corresponding author*

ARTICLE INFO

Article history:
Received 19-11-2020
Revised 10-01-2021
Accepted 19-06-2021

Keywords:
Computer Vision
Vehicle Classification
SVM

ABSTRACT

In the automation of vehicle traffic monitoring system, information about the type of vehicle, it is essential because used in the process of further analysis as management of traffic control lights. Currently, calculation of the number of vehicles is still done manually. Computer vision applied to traffic monitoring systems could present data more complete and update. In this study consists of three main stages, namely Classification, Feature Extraction, and Detection. At stage vehicle classification used multi-class SVM method to evaluate characteristics of the object into eight classes (LV-TK, LV-Mobil, LV-Mikrobis, MHV-TS, MHV-BS, HV-LB, HV- LT, MC). Features are obtained from the detection object, processed on the feature extraction stage to get features of geometry, HOG, and LBP in the detection stage of the vehicle used MOG method combined with HOG-SVM to get an object in the form of a moving vehicle and does not move. SVM had the advantage of detail and based statistical computing. Geometry, HOG, and LBP characterize complex and represents an object in the form of the gradient and local histogram. The test results demonstrate the accuracy of the calculation of the number of vehicles at the stage of vehicle detection is 92%, with the parameters HOG cellSize 4x4, 2x2 block size, the son of vehicle classification 9. The test results give the overall mean recognition rate 91,31 %, mean precision rate 77,32 %, and mean recall rate 75,66 %.

I. Introduction

The development of video sensors and video processing hardware opens an excellent opportunity for the development of vision-based monitoring system technology because it can provide more detailed information for further analysis needs such as traffic light management — distribution of types of vehicles that have been set in MKJI (Indonesian Road Capacitance Manual) based on the type of road. So far to obtain data about traffic conditions is still done manually through surveys. The density level of a road is known by analyzing traffic volume. This analysis is used to determine the number of vehicles that move in a particular direction on the part of the road that passes a certain point or place per hour, per day, or every week. Traffic monitoring that has developed a lot has been using a magnetic loop detector and radar as a monitoring tool for the number of vehicles. This tool has limitations in measuring several parameters of traffic conditions such as vehicle type, number of vehicles, average vehicle speed, and congestion that occurs [1]. Currently, in urban areas, a monitoring camera that is integrated with ATCS (Automatic Traffic Control System) has been installed so that traffic light monitoring and control can be done manually remotely, but it is still not optimally utilized.

Vehicle type classification systems based on video processing and digital image processing techniques have been developed [2] [3] [4], one of them by combining vehicle detection methods, and classification of particular features of vehicles obtained through geometric feature extraction methods [5], extraction of the HOG (Histogram of Oriented Gradient) feature [6] [7], and extraction of LBP (Local Binary Pattern) features [8]. Furthermore, the features obtained will be evaluated using SVM (Support Vector Machine) to solve the problem of multiple classifications of vehicle types [9] [10]. SVM has advantages in addition to its high accuracy in classifying vehicle types; it is also easy to optimize so that it is faster and produces the desired minimum error. SVM method is evidenced by several studies on the classification of vehicles using SVM [11] [12] [13].

The accuracy of the results of the classification of types of vehicles depends on the vehicle that was successfully detected (the result of segmentation) traffic video frames. Proper vehicle segmentation will produce good features for SVM inputs and vice versa. Therefore, vehicle detection methods and vehicle recognition methods play a crucial role in the classification system of vehicle-based video processing types to calculate the volume of traffic per vehicle type. Based on the explanation presented and the results of previous studies the Support Vector Machine method is considered appropriate to be implemented in this study.

II. PROPOSED METHOD

A. Architecture

The design of the training process for sample datasets with Support Vector Machine-Multi Class is shown in **Figure 1** and the design process for vehicle classification testing is shown in **Figure** 2.

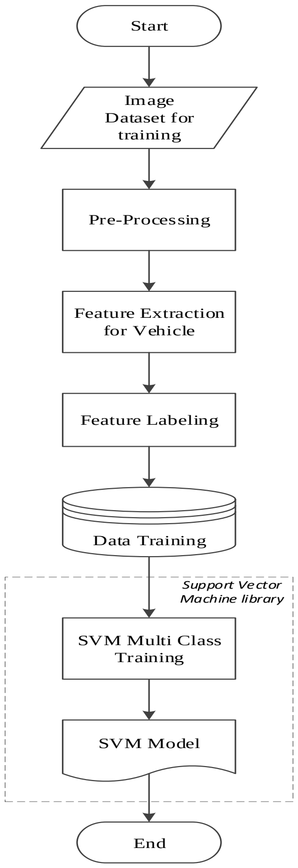

Figure 1. The Training Stage

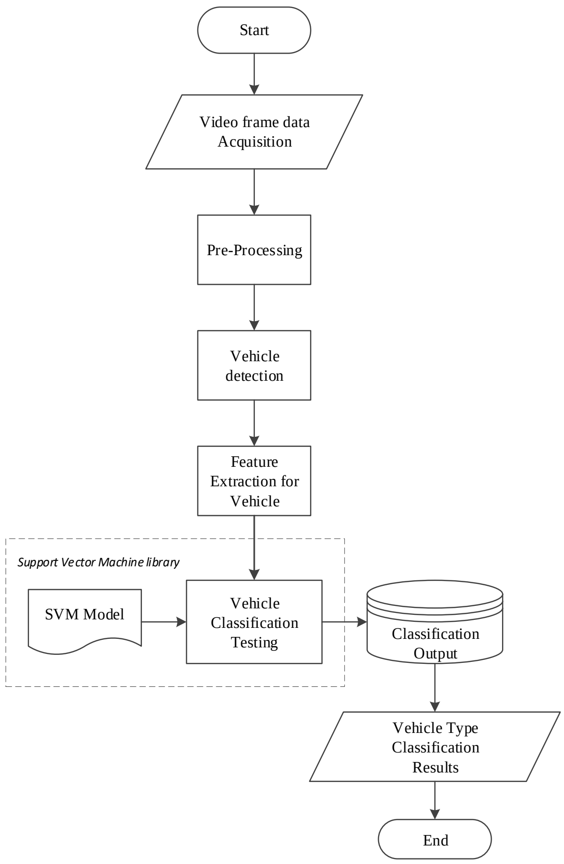

Figure 2. Testing Stage

Preprocessing, feature extraction, training methods used in the training process for training dataset. Next, the features extraction process, the characteristics used in this study are the geometry of objects, HOG, and LBP. After that the feature vector is stored in the XML file which is then read and converted into a training data matrix to be processed by SVM, the output of the training is SVM Model (SV, alpha, rho) which is used for matching features in the testing process. Vehicle type classification architecture includes (1). Image acquisition, (2). Pre-Processing, (3). Vehicle detection, (4) vehicle feature extraction, and (5) classification of vehicle types.

B. Pre-Processing

At this stage grayscaling and histogram equalization are carried out to overcome uneven lighting or sudden lighting changes at the time of data acquisition from the camera, besides grayscaling aims to alleviate system computing

C. Vehicle detection

At the vehicle detection stage, it is used to separate the foreground and background. The foreground segmentation results are then cropped and resized. Two steps are used, namely: Detect vehicles with the MOG method to get the results of segmentation in the form of a binary object vehicle blob, which is used for object geometry feature extraction. Detect vehicles with the HOG-SVM method with the results of segmentation in the form of cropping a vehicle's RGB image used for global feature extraction of HOG.

D. Feature Extraction

At this stage, the cropped image is extracted geometric features, a histogram of oriented gradient features and local binary pattern features.

1) Training

This stage is training vehicle type dataset samples with SVM to get the best SVM model parameters.

2) Vehicle Classification

This stage classifies the type of vehicle in the test video frame based on the results of SVM training.

E. Dataset

Samples of datasets, we made through cropping images (derived from sample video frames) each type of vehicle manually with sample size is 256 x 256 pixels. In the Indonesian Haighway, so the vehicle classification we took based Indonesian vehicle type ordinance (MKJI – Indonesian Highway Capacitance Manual). In this study two types of datasets are needed in the training process, namely:

1) HOG-SVM Training Data for vehicle detection

Positive sample data contains 2, 4 and 6 or more wheels. The number of positive images used is 7614 training images. Negative sample data containing objects, not vehicles, generally in the form of background images (roads, trees, and houses). The number of negative samples is 9360 training images.

2) Training Data for classification of vehicle types

Classification training data consists of image samples (a) Large Bus (HV-LB) = 700 images, (b) Coupled Large-Truck Trucks (HV-LT) = 520 images, (c) Medium Buses (MHV-BS) = 134 images, (d) Medium Truck (MHV-TS) = 1000 images, (e) Microbus (LV-Mik) = 330 images, (f) Cars (LV-Mbl) = 3630 images, (g) Small Trucks (LV) -TK) = 920 images, (h) Motorbikes (MC) = 714 images. Dataset positive image given by **Figure 3,** and negative image given by **Figure 4**.

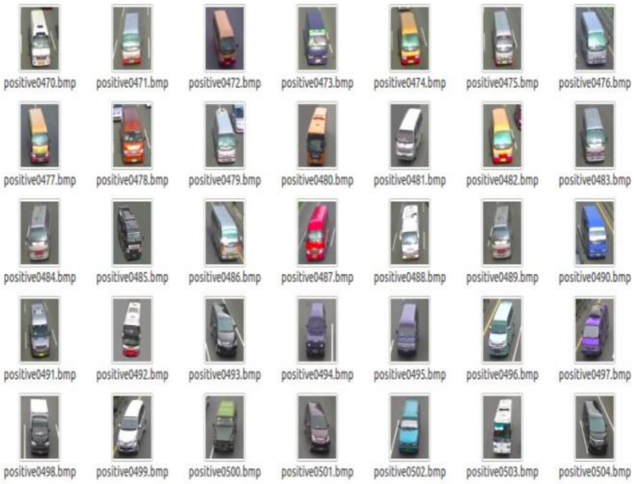

Figure 3 Dataset Vehicle as Positive Image

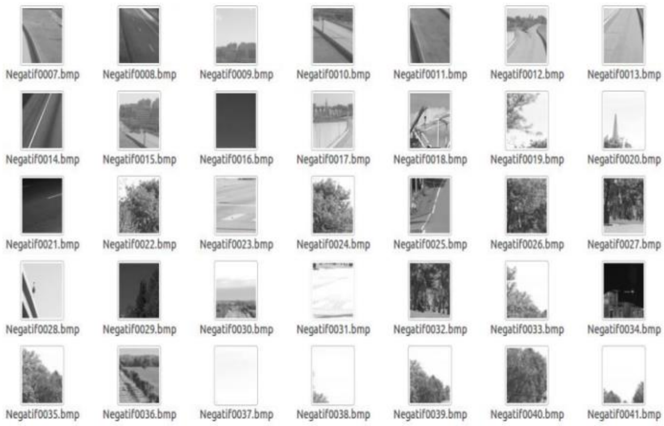

Figure 4 Negative Image

III. VEHICLE DETECTION AND VEHICLE CLASSIFICATION

A. Mixture of Gaussian (MOG)

MOG is used in the vehicle detection stage. In the MOG Method, the motion of an object is detected by comparing the MOG parameters in the previous frame with the current MOG parameter. The MOG algorithm is divided into two parts, parameter initialization, and for every frame video processing consisting of input matching, update parameters, and the estimation of the background model in the final stage, also foreground detection results. The function of the MOG2 operation is an improvement from current MOG used to build adaptive background subtraction detail algorithms found in [14].

B. HOG-SVM

The stages of the HOG-SVM training process for vehicle detection and after we get vehicle object we used HOG-SVM also for vehicle classification but using SVM for multiclass, the process is as follows:

1. Read samples of positive and negative training images in the directory path
2. Image pre-processing: change the image to grayscale and HE
3. Calculate the HOG feature and visualize the HOG features of each training image (for positive images, negative images) with parameters (cv:: HOGDescriptor):
4. winSize (w): window, the size is the same as the training image.

5. blockSize (bh x bw): pixel block size in cell, (2x2 cell, default: 16x16 pixel).
6. block stride (bsh x bsw): the size of multiple cells is used for the sliding window (default: 8x8).
7. cellSize (ch x CW): number of pixels per cell (default: 16x16).
8. nbins: number of histogram bin orientations (default: 9)
9. derivative Securities (default: 1)
10. winSigma = Gaussian Smoothing Window parameter; (default: -1)
11. L2-Hys Threshold: Normalization method (default: 0.2)
12. Gamma correction: true or false
13. The number of features (N) generated by the HOG feature can be calculated from Eq. (1).
14. $$\boldsymbol{N} = \left(\frac{\boldsymbol{w}-\boldsymbol{bw}}{\boldsymbol{bsw}}+\mathbf{1}\right)\left(\frac{\boldsymbol{h}-\boldsymbol{bh}}{\boldsymbol{bsh}}+\mathbf{1}\right)\frac{\boldsymbol{bw}}{\boldsymbol{cw}}\frac{\boldsymbol{bh}}{\boldsymbol{ch}}\boldsymbol{nbin} \qquad (\mathbf{1})$$
15. the Save feature maps (vector) into the file system (in this study used XML files).
16. Read XML features that have been saved, change to row matrix form = amount of training data (post, neg) and col = column HOG feature, then labeling positive and negative features (1 and -1)
17. SVM Training, with specific parameters and SVM kernel selection (LINEAR, RBF, or POLY)
18. Save the training results into the XML file (SVM model and Support Vector (SV, alpha, rho) for the testing process.

The vehicle detection testing stage is as follows:

1. Read n-th video frame
2. Pre-processing (grayscale and HE)
3. Extracting the video frame HOG feature performs a scanning window based on the block stride size. The HOG parameter used must be the same as the training image so that the same number of HOG features is produced (Visualization of the HOG feature).
4. Performs the SVM process for two classes (HOGDescriptor:: setSVMDetector) based on the SVM Model of the training, so that another obtains the SV candidate closest to the hyperplane (class 1) -1
5. The classification results (HOGDescriptor:: DetectMultiScale) are obtained, mark and crop the classification results of the vehicle, display the rectangle detected on the frame.

C. Feature Extraction

Extraction of Geometry Features

Measurement-Based feature [5] is calculated from the shape geometry of the object. In this study, the area and perimeter features are used to obtain equivdiameter characteristics ($\sqrt{4.A/\pi}$) as the equivalence of a circle with an area equal to the object region, Dispersedness (I^2/A) as a ratio of squares perimeter to area. Compactness factor ($I^2/4\Pi A$) describes how natural an object is. The object that has a normal contour is higher than the compactness factor value; then the object is categorized as having an irregular contour.

D. Extraction of the Histogram of Oriented Gradient

The HOG feature consists of many oriented gradient histograms from a localized area so that it can describe the contour of an object and is insensitive to changes in light [15].

The HOG feature extraction calculation consists of:

1. Gradient Calculation

   Calculating the gradient value of each pixel for an image I, calculated the gradient value horizontal and vertical gradients for each pixel, given by Eq. (1) – Eq. (5)

$$Gx(x,y) = [-1\ 0\ 1].I(x,y) \quad (2)$$

$$Gy(x,y) = [-1\ 0\ 1].^{T} I(x,y) \quad (3)$$

$$G(x,y) = \sqrt{Gx(x,y)^2 + Gy(x,y)^2} \quad (4)$$

$$\theta(x,y) = arctan\left(\frac{Gy(x,y)}{Gx(x,y)}\right) \quad (5)$$

Where: $G_x(x,y)$ = horizontal gradient of an object, $G_y(x,y)$= vertical gradient of an object, G (x, y) = object gradient, $\Theta(x,y)$ = object orientation angle, $I(x,y)$= Pixel intensity at coordinates (x, y).

2. Block and Cell

   The HOG operation uses a sliding window, the size of the HOG cell of a sliding window is usually 4 x 4, 6 x 6, and 8 x 8 pixels when the size of a 32 x 32, 48 x 48, and 64 x 64-pixel sliding window — a block that is configured by grouping 2 x 2 cells. Locally grouped larger size cells, through neighboring blocks. The HOG descriptor is a component vector from normalized cell histograms from all block regions. This block usually overlaps, meaning that each cell has a role more than once to produce the final descriptor.

3. Orientation Binning

   Each pixel that plays a role gives weight to the magnitude of the cell histogram with a Gaussian function placed in the middle of the block and weighing the gradient value to select the direction gradient of the middle element. Middle bin position to reduce the aliasing effect. The angle value of the gradient is $0^0 < \theta(x,y) < 360^0$, then behind $0^0 < \theta(x,y) < 180^0$. Therefore the bin index value of the direction gradient value is calculated $0^0 < \theta(x,y) < 180^0$, to increase the recognition value. The orientation of $0^0 < \theta(x,y) < 180^0$ is divided into nine channels (0.20), (20.40), ..., (160-180). When the angle value is θ (x, y) in the range (0-20), the correspondence orientation for pixels (x, y) is given a weight = 1. Other orientations are given a weight of 0.

4. Gradient Direction Histogram

   For each cell after calculating the magnitudes of m (x, y) and θ (x, y) gradients, the histogram of each cell is executed. The histogram is calculated for cells 4 x 4, 6 x 6, 8 x 8 pixels based on the size of the sliding window.

5. Block Normalization

   The final result of HOG feature extraction is calculated on one block, which consists of cells 2 x 2. The histogram of this block is obtained from the weighting, interpolation, and normalization block operations. The histogram of the cell is combined and normalized to the HOG descriptor form of the block. Normalization $v = \frac{v_{i,j}}{\sqrt{\left\|v_{k,l}\right\|^2 + \epsilon^2}}$ for each block will increase reliability. L2-norm is used for the detection of people and vehicles, where $V_{i,j}$ = vector which corresponds to a combination of histograms with blocks *i, j, k, and l* are indices of vector V, ϵ = small constants, which results in close to 0, in general, = 1. *v* = normalized vector, the result of the HOG feature.

LBP Feature Extraction

LBP [16] is a feature extraction technique based on texture analysis transforming an image into an integer statistical label, using computational mean efficiently. Basic LBP operators label Pn pixels for (n = 0.1, ..., 7) from an image with neighbor thresholding 3 x 3 for each pixel. LBP results at the pixel at location (x_c, y_c). LBP results obtained as Eq. (6).

$$LBP_{P,R(x_c,y_c)} = \sum_{n=0}^{n-1} S(P_n - P_c)2^n \qquad (6)$$

Where P is the number of neighbors involved, R is the neighboring distance, P_c is the grey level value of the center pixel and P_n the eight neighbors around c (pixel center) and the value S is the function of pixel reduction P_n-P_c, given by Eq. (7)

$$S(P_n, P_c) = \begin{cases} 1 \; if \; P_n \geq c \\ 0 \; otherwise \end{cases} \qquad (7)$$

LBP is shown in **Figure 5**. The histogram label is calculated entirely, and each region is used as a local descriptor that describes local changes such as flat areas, curves, edges, spots and so on.

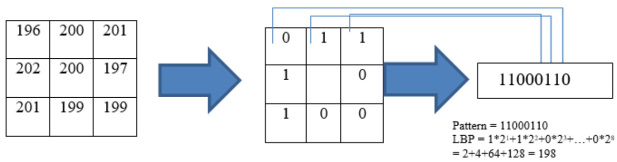

Figure 5 LBP feature extraction

E. Support Vector Machine

1) SVM Training Stage

The training algorithm for each SVM classifier can be written as follows, Input = *B* matrix (training feature extraction matrix) and vector Y = input-target pair and output = *w, x, b* (variable hyperplane equation). The steps are as follows

1. Determine the input Z = B and target (Y) as a training pair from 2 classes for vehicle detection and eight classes for vehicle classification.
2. Calculate Kernel $K(Z, Z_i)$
3. Calculate the Hessian Matrix $H = K(x_i, x) * Y * Y^T$
4. Set the values C and γ (for RBF kernel)
5. Set vector E as a unit vector that has dimensions equal to the dimension Y
6. Calculate quadratic programming solutions. The solution to quadratic programming is the implementation of a solution to the problem of $min \frac{1}{2}|w|^2 + C(\sum_{i=1}^{n} \xi_i)$. If implemented in the form of a matrix it becomes min $min \frac{1}{2} w^T w + C(\sum_{i=1}^{n} \xi_i)$, with $y_i^T(w^T \phi(x_i) + b) \geq 1 - \xi_i$. If the formula in the form of the matrix is changed to a dual problem, then the formula becomes $min_{L(\alpha)} = \frac{1}{2}\alpha^T H \alpha - E^T \alpha$. Where: $y^T \alpha = 0 \; dan \; 0 \leq \alpha \leq C$, $H = y_i y_j K(x_i, x_j)$, $K(x_i, x_j) = \exp\left(-\gamma |x_i - x_j|^2\right)$, $\gamma > 0$ (kernel function), E = unit vector whose dimensions are equal to Y, C> 0 as the upper limit of the value α. In this study, the values C = 10000 and epsilon = 1 x 10-7 are used. The results of the monqp (quadratic programming) function are the values of variables *w, x, and b* which will later be used for the testing process.

2) SVM Testing Stage

After the training process, the values of variables *w, x, and b* are obtained for each class. Then defined as vector *w, x, and b*. The input of classified data is the D feature matrix produced in the test feature extraction process. The D feature matrix is first transformed into a vector to 1 x $(l_1 x l_2)$ named T. The steps are as follows:

1. Input: T vector (test data), vector w, x, b, and k (number of classes).
2. Calculate Kernel $K(T, x_i)$

3. Calculate the value $f_i = K(T, x_i)w_i + b_i$
4. Repeat steps 2 and 3 for i = 2 to k
5. Determine the maximum value of f_i
6. Class I with the largest f_i is the class of vector T.

The T value is the transformation of the D feature matrix into a vector. The next step is to calculate the kernel $K(T, x_i)$, the kernel tested in this study is linear and RBF with T is the input data and x_i is the support vector that results from the SVM training process. For each value I, The decision function $f_i = K(T, x_i)w_i + b_i$ is calculated. Where I = 1 to k (k is the number of classes). The output of this algorithm is the index I with the largest f_i which is the class of vector T. Training data that has been projected by HOG, LBP, and MBF features, then becomes SVM training data. Data from the feature extraction is then transformed into the feature field (space feature) because the number of huge features on SVM is used "Kernel Trick." The function of the kernel tested in this study is the equation of the Kernel Radial Basis Function (RBF) and Linear to find out which kernel is suitable for feature data in classifying.

Vector support numbers for each training data must be sought to obtain the best separator solution. The problem of the best separator field solutions can be formulated in Eq. (8).

$$max\ L_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1,j=1}^{n} \alpha_i \alpha_j\, y_i y_j \vec{x}_i \vec{x}_j \quad (8)$$

where : $\sum_{i=1}^{n} \alpha_i y_i = 0, \alpha_i \geq 0\ (i = 1,2, \ldots, n)$

Data $\vec{x}_i$ which correlates with $\alpha_i > 0$ is what is called support vector. Thus, a value can be obtained which will be used to find the value of w. The solution of the separator is obtained by the formula $w = \sum \alpha_i y_i$ and the value of $b = y_{k_}wx_k$for each x_k, with $\alpha_k \neq 0$. The process of testing or classification is also carried out on each binary SVM using the values $w, b, and\ \vec{x}_i$ which are generated in the binary SVM training process. The functions generated for the testing process are: $f_i = K(x_i, x_d)w_i + b_i$, Where: i = 1 to k (class), $x_i = support\ vector$, x_d = test data. The output is in the form of index I with the largest fi which is the class of the test data.

IV. RESULT AND DISCUSSION

System development is done by using Microsoft Visual Studio 2012 and MATLAB 2012b programs for experiments. The hardware specifications used for the implementation of the system are processor core i7, 16GB RAM, 8GB GPU CUDA, the library used in this study is the OpenCV library and SVM Library. The language used is C ++ with an application console.

A. Vehicle Detection Testing

At the stage of vehicle detection, moving errors often occur because of detection stability. From the results of experiments through the trial and error process, the detection process of moving vehicles based on MOG provided the best output with 127 threshold settings. Value 127 is the threshold value used to eliminate objects on the detection results of MOG2. Blob detection results with MOG2 are used for the geometry feature extraction process, with the ROI (region of interest) of the object being the rectangle detected by vehicle detection with HOG-SVM. Vehicle detection testing using the HOG-SVM method is done to obtain the best parameter settings for HOGs in detecting and classifying types of vehicles by varying cell size data in 3 variations namely 4 x 4, 8 x 8, and 16 x 16. Calculation of accuracy using the Accuration formula = (number of vehicles detected by the system) / (total number of vehicles). The results of testing HOG parameters with 4 test variations with a fixed block size of 2 x 2 and fixed bin orientation 9, winSize 128 x 128 size are presented in

Table 1.

ISSN: 2579-7298

Table 1 Vehicle Detection Test Results

No	Cell size variations (*pixel*)	Accuracy
1	4 x 4	93,76 %
2	8 x 8	92 %
3	16 x 16	90,15 %
4	32 x 32	71 %

From the results of testing by varying the value of cell size in the HOG, it was found that the increase in the accuracy of vehicle detection was not influenced by the size of the cell that was getting bigger. The best cell size for detecting vehicles in this study is 4 x 4 pixels which give an accuracy rate of 93.76%. The visualization of the HOG for positive and negative sample data in cell size 4 x 4 is shown in **Figure 6**.

Figure 6 Vehicle Detection Result with HOG Visualization

B. *Vehicle Classification Testing*

In testing the classification of vehicles used two pieces of video samples. The first video is 25 minutes 02 seconds, 25 fps, so the total frame is 37550 frames. The second video is 58 minutes 1 second, 25 fps, so the total number of frames is 87025 frames.

Table 2 Results of Video Sample Testing 1

System / Real	LV – Small Truck	LV – Car	LV –Micro-bus	MHV – Medium Truck	MHV - Medium Bus	LT - Large Truck	LB - Large Bus	MC – Motor- cycle	Non- Vehi-cle
LV – Small Truck (106)	92	2	4	3	0	0	0	0	7
LV – Car (1756)	0	1706	2	0	0	0	0	0	48
LV – Mikrobus (42)	0	8	40	0	0	0	0	0	4
MHV – Medium Truck (112)	8	0	0	100	0	2	0	0	2
MHV - Medium Bus (14)	0	0	0	0	10	0	2	0	2
LT - Large Truck (45)	0	0	0	6	0	37	0	0	2
LB - Large Bus (103)	0	0	0	0	5	0	90	0	8
MC - Motorcycle	0	0	0	0	0	0	0	0	0
Non- Vehicle									

Each vehicle detected then extracted the features of HOG, LBP, equivdiameter, dispersedness, and compactness then carried out a classification process based on the support vector values generated from the training or training that had been carried out. Vehicle classification test results with SVM Multiclass are presented in **Table 2** for videos 1 and 3 for video 2.

From the confusion table 2, we can obtain a system evaluation component TP, FP, FN, TN through the formula as follow :

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN}$$

TP	=	2075	FN	=	115
TN	=	10115	FP	=	42

$$Accuracy = \frac{2075+10115}{2075+42+115+10115} x\ 100\%$$
$$= 98{,}72\ \%$$

$$Precision = \frac{TP}{TP+FP}$$
$$= \frac{2075}{2075+42} x\ 100\% = 98{,}01\ \%$$

$$Recall = \frac{TP}{TP+FN}$$
$$= \frac{2075}{3809+115} x\ 100\% = 94{,}75\ \%$$

The results of the table confusion Table 2 obtained an accuracy value of 98, 72%; this indicates that the system can classify the type of vehicle correctly. Many errors occur in the detection of cars where many cars cannot be classified; this occurs because of an error in detecting vehicles where miss vehicles are detected so the real thing is the car cannot be classified. In the classification of large bus types, some buses are detected as medium buses; this occurs because of the similarity on the head of the bu so that the HOG and LBP features as the dominant features of the system have similarities to each other. Misclassification also occurs in large trucks where several objects of large trucks are in the category of trucks; this is due to the similarity of shape to the object header.

The results of the confusion **Table 3**, we can be obtained from the system evaluation component TP, FP, FN, TN through the formula, as follow

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN}$$

TP	=	3167	FN	=	2430
TN	=	22144	FP	=	2425

$$Accuracy = \frac{3167+22144}{3167+2425+2430+22144} x\ 100\%$$
$$= 83{,}9\ \%$$

$$Precision = \frac{TP}{TP+FP}$$
$$= \frac{3167}{3167+2425} x\ 100\% = 56{,}63\ \%$$

$$Recall = \frac{TP}{TP+FN}$$
$$= \frac{3167}{3167+2430} x\ 100\% = 56{,}58\ \%$$

Table 3 Video sample testing results in video 2

System / Real	LV – Small Truck	LV – Car	LV – Micro-bus	MHV – Medium Truck	MHV - Medium Bus	LT - Large Truck	LB - Large Bus	MC – Motor-cycle	Non-Vehi-cle
LV – Small Truck (106)	10		2	3	0	0	0	0	3
LV – Car (1756)	0	2002	56	0	0	0	0	0	427
LV –Mikrobus (42)	3	2	5	0	0	0	0	0	0
MHV – Medium Truck (112)	4	0	0	10	0	0	0	0	2
MHV - Medium Bus (14)	0	0	0	0	35	0	5	0	2
LT - Large Truck (45)	0	0	0	1	0	3	0	0	0
LB - Large Bus (103)	0	0	0	0	2	0	40	0	8
MC - Motorcycle	0	0	0	0	0	0	0	1062	1905
Non- Vehicle									

The results of testing the classification of vehicle types in the test video sample 1 show that the system is not good at classifying objects, this is indicated by the low value of precision systems in a collection of objects resulting from the detection of the result object is only 56%.

The parameters used for SVM are linear kernels, HOG winsize parameters 128x128, cellSize 4x4, block stride 4x4, blockSize 8x8. Error classifying the type of vehicle occurs because of the similarity between one type of vehicle and other types of vehicles.

1. CONCLUSIONS

Based on the results of the video processing research for vehicle classification and the discussions that have been conducted, some conclusions can be drawn, as follows. The combination of Histogram of Gradient and LBP techniques to get the characteristics of each type of vehicle successfully classifies the type of vehicle quite well and can detect vehicles that appear to coincide with the best parameters: 8x8 pixel Cell size, 2x2 block size, and bin 9 orientation size for HOG, and LBP is divided into 8 x 8 regions.The MOG2 algorithm is proven to be able to detect vehicle objects and withstand changes in light when taking video with the best parameter setting is the 127f threshold. Support Vector Machine algorithm is proven to apply to a vehicle type classification system properly. The results of classification testing of vehicle types with the support vector machine method as a whole provide an average accuracy value of 91.31%, average precision values of 77, 32%, and recall value of 75.66%. The results of vehicle detection tests with a combination of MOG and HOG-SVM methods have an accuracy rate of 93.76%.

The success rate of the system in classifying the type of vehicle depends on the results of detection and the combination of features that represent the characteristics of the object. Further research is needed to improve the quality of the classification system of vehicle types in this study. Testing uses more sample data variations for each type of vehicle and various parameter variations so that the level of system reliability is known, more diverse. The choice of the use of features that are more contrasting and unique for each type of vehicle so that it can distinguish one type of vehicle with another type more clearly. Improvement from the engineering side removes shadows and occlusion objects in vehicles that appear to coincide so that the process of calculating the number of vehicles can be more accurate. Using Deep Learning Method to increase vehicle classification. Further research is needed for the method of determining the HOG parameters and

SVM parameters for the vehicle datasets used in this study. Based on the results of the study it can be seen that the determination of the right parameters can provide a model with good performance and to get it needs to be tested repeatedly with different combinations.

Acknowledgements

The authors would like to thank the Electronic Research and Instrumentation Laboratory of DIKE FMIPA UGM and the DIY Transportation Office for providing support for this research and funding in Penelitian Dosen Muda from Universitas Gadjah Mada.

REFERENCES:

[1] Chen, Z., Ellis, T., Velastin, S.A. 2012, Vehicle Detection, Tracking and Classification in Urban Traffic, 15th International IEEE Conference on Intelligent Transportation Systems Anchorage, Alaska, USA, September pp. 16-19.

[2] Chiu, C., Ku, M., and Wang, C., 2010, Automatic Traffic Surveillance System for Vison-Based Vehicle Recognition and Tracking, Journal of Information Science and Engineering 26, pp. 611 – 629.

[3] Ghasemi, A., dan Safabakhsh., 2012. A real-time multiple vehicle classification and tracking system with occlusion handling, IEEE 8th International Conference on Intelligent Computer Communication and Processing, pp. 109 – 115.

[4] Mishra, P.K, dan Banerjee, B., 2013, Vehicle Classification using Density-based Multi-feature Approach in Support Vector Machine Classifier, International Journal of Computer Applications (0975 – 8887), Volume 71 - No. 7, June 2013.

[5] Chen, Z., and Ellis, Tim., 2011, Multi-shape Descriptor Vehicle Classification for Urban Traffic, International Conference on Digital Image Computing: Techniques and Applications, IEEE, pp. 456 - 461.

[6] Ling, M., Mei, X., Yi, H., dan Yuefei, Z. 2010.Preceding Vehicle Detection Using Histogram of Oriented Gradients. International Conference on Communications, Circuit, and Systems Proceedings Volume II. pp. 28-30 July, China.

[7] Chen Z, Chen, K., and Chen, J. 2013. Vehicle and Pedestrian Detection Using Support Vector Machine and Histogram of Oriented Gradients Features.International Conference on Computer Sciences and Applications, pp 366 – 268.

[8] Rabiu, H., 2013, Vehicle Detection And Classification For Cluttered Urban Intersection, International Journal of Computer Science, Engineering and Applications (IJCSEA) Vol.3, No.1, February 2013, pp 37 – 47.

[9] Teng Ng., Azmin, S.S., dan Teoh, S.S. 2014. Vehicle Classification Using Visual Background Extractor and Multi-Class Support Vector Machine. The 8th InternationalConference on Robotic, Vision, Signal Processing & Power Applications, pp 221 – 227

[10] Rahim, A.N., MP. P., and Adom, A.H., 2013, Adaptive Boosting with SVM Classifier for Moving Vehicle Classification, Procedia Engineering 53, pp. 411 – 419.

[11] Li, Xing dan Guo Xiasong. 2013.A HOG Feature and SVM Based Method for forwarding Vehicle Detection With Single Camera. Fifth International Conference on Intelligent Human-Machine Systems and Cybernetics

[12] Zhang, G., Gao, F., Cong, L., Liu, W., and Yuan, H. 2010. A Pedestrian Detection Method Based on SVM Classifier and Optimized Histograms of Oriented Gradients Feature, Sixth International Conference on Natural Computation (ICNC 2010)

[13] Sasidharan, S.K., and Kumar, N.K.K., 2013, Vehicle Detection in Image using SVM, International Journal of Advanced Electrical and Electronics Engineering (IJAEEE), Volume-2, Issue-6, ISSN (Print): 2278-8948.

[14] Zivkovic Zoran., 2004. Improved Adaptive Gaussian Mixture Model for Background Subtraction. Intelligent and the the Autonomous Systems Group University of Amsterdam, The Netherland.

[15] Laopracha, N., Thongkrau, T., Sunat, K., Songrum, P., dan Chamchong, R. 2014. Improving Vehicle Detection by Adapting Parameters of HOG and Kernel Functions of SVM. International Computer Science and Engineering Conference (ICSEC)

[16] Wahyudi, E., Kusuma, H., dan Wirawan, 2011, Perbandingan Unjuk Kerja Pengenalan Wajah Berbasis Fitur Local Binary Pattern dengan Algoritma PCA dan Chi. Square.Seminar on Intelligent Technology and its applications, ITS Surabaya.