EPI International Journal of Engineering
Vol. 1, No. 1, February 2018, pp. 69-75
DOI: 10.25042/epi-ije.022018.11

ISSN:2615-5109

Support Vector Machine Method to Reduce The Execution Time of Vehicle Plate Recognition System

Muh Ismail,[a]* Indrabayu,[b] Intan Sari Areni,[c]

[a]Graduated Student of Electrical Engineering Department, Engineering Faculty, Hasanuddin University, Makassar. Email:imuh77@gmail.com
[b]Department of Informatics, Engineering Faculty, Hasanuddin University, Makassar. Email:indrabayu@unhas.ac.id
[c]Department of Electrical Engineering, Engineering Faculty, Hasanuddin University, Makassar. Email:intan@unhas.ac.id

Abstract

This research aims to create a vehicle plate detection and recognition system with Cascade Classifier, Support Vector Machine (SVM) and Optical Character Recognition (OCR). Cascade Classifier with Local Binary Patterns (LBP) descriptor is used to detect the car licence plate (Coarse Location). SVM is used to reduce plate candidate detection error and the execution time. Optical Character Recognition (OCR) is used to recognize the characters in plates. The system test is performed using 19 video data of moving vehicles at night and rain conditions. Each video has a duration of 30 seconds and contains 4-10 cars per video. The testing results show reduction of the execution time of vehicle plate recognition system which reached 60% with the average accuracy of plate recognition 61.94%.

Keywords: Cascade classifier; license plate recognition; optical character recognition; support vector machine; video processing

1. Introduction

Research on plate detection or plate recognition become an interesting topic and has been widely practiced in the last two decades that is part of the field of intelligent transportation systems. Some utilization of vehicle license recognition are in parking applications, traffic monitoring, and vehicle traffic controlling [1, 2].

Although vehicle plates are very easy to be recognized by humans, they leave a problem or challenge if applied in a computer system, especially in rain and night conditions [3]. The ones that are still a challenge in various studies start from plate detection, detection image enhancement, and plate character recognition [4].

The research that has been done by Xiao-Hua in 2014 about vehicle plate recognition combining two methods of Pulse Couple Neural Networks (PCNN) and Template Matching (TM). In this study, PCNN is applied to segment a gray image that contains the plates and character extraction of the detected plates while TM is used for character recognition. However, plate recognition under irregular lighting became an obstacle in this study [5].

In 2015, Bunay et al also study about the recognition of vehicle plates that proposed hybrid techniques with Neural Network (NN) and TM methods. Recognition accuracy resulted from this study reached 100%. The data were taken in normal light conditions [6].

Animesh et al. have also studied plate recognition with complex backgrounds using the Morphological Approach and TM methods. The plate studied consists of two lines, where the first line contains texts and the second line includes numbers. This study has a recognition accuracy of 88.8%. However this research is still focused on the recognition of static images and one vehicle object [7].

The present research focuses on minimizing execution time on the system to detect and recognize moving vehicle plates at night and rain conditions. The Support Vector Machine (SVM) method is added to the classification process to reduce the execution time.

This paper is organized as follows: the following section describes literature review. Section 3 is proposed method which describes the algorithm of the vehicle plate coarse location detection, plate candidate preprocessing, vehicle plate precise location detection, and the plate candidate recognition. Section 4 provides results and discussions. Finally, the conclusions and further research are described in Section 5.

*Corresponding author. Tel.: +62-812-4420-9266
Jalan Poros Malino km. 6, Bontomarannu
Gowa, Sulawesi Selatan, Indonesia, 92171

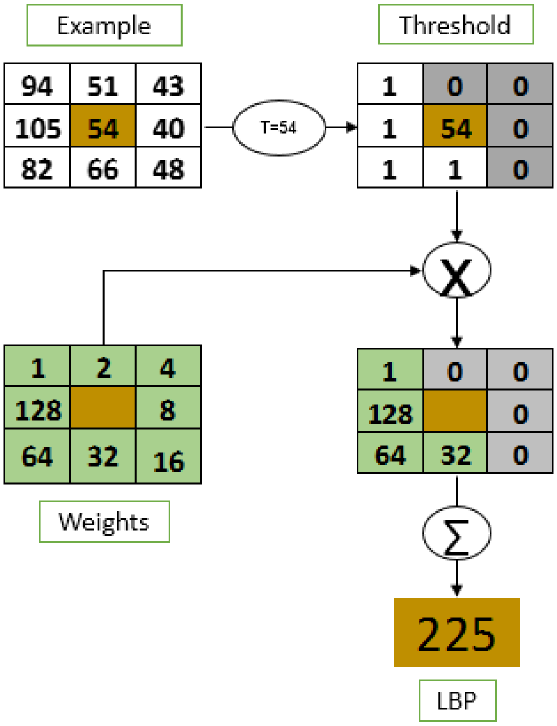
2. Literature Review

2.1. Local Binary Pattern

Local Binary Pattern (LBP) is a binary code describes local texture patterns. It is built with a boundary environment with gray values from its center [8]. An example of LBP computing at 3x3 pixels is shown in Fig. 1. The basic version of LBP uses the center pixel value as a threshold for 3x3 neighbor pixels. The threshold operation creates a binary pattern that represents the texture characteristics. The equation for the basic version of LBP is shown in Eq. 1 [9]:

$$LBP(x_c, y_c) = \sum_{n=0}^{7} 2^n g(I_n - I(x_c, y_c)) \qquad (1)$$

$LBP(x_c, y_c)$ is LBP value in the centre pixel (x_c, y_c). And then, , I_n and $I(x_c, y_c)$ are the neighbor pixel values and the center of each pixel. Index n is an index of neighbour pixels. The function $g(x) = 0$ if $x < 0$ and $g(x) = 1$ if $x \geq 0$.



94	51	43
105	54	40
82	66	48

1	0	0
1	54	0
1	1	0

1	2	4
128		8
64	32	16

1	0	0
128		0
64	32	0

Figure 1. LBP Pixels Counting Process

In Fig. 1, the value 54 as a central pixel is selected to be a threshold. The neighbor pixel value becomes 0 if the value is less than 54. Meanwhile, the neighbor pixel value becomes 1 if the pixel value is greater than or equal to 54. Then the LBP value is obtained through the result of scalar multiplication between binary matrix (threshold) and weight. Therefore, the LBP value of the 3x3 matrix in Fig. 2 assigned 1 + 32 + 64 + 128 or equal to 225.

2.2. Normalization

Normalization is the process of scaling attribute values or variables of a data so that values can be in a certain range. The result of normalization data (y_{ij}) is calculated based on the following Eq. 2.

$$y_{ij} = \frac{x_{ij}}{\sqrt{\Sigma(x_{ij}{}^2)}} \qquad (2)$$

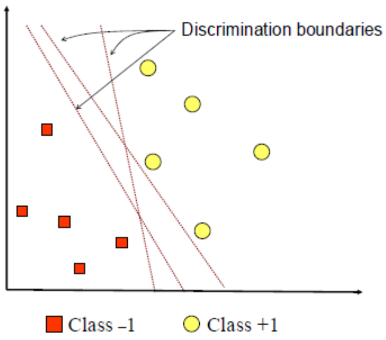
where:
x = data
i = row index
j = colomn index

2.3. Support Vector Machine

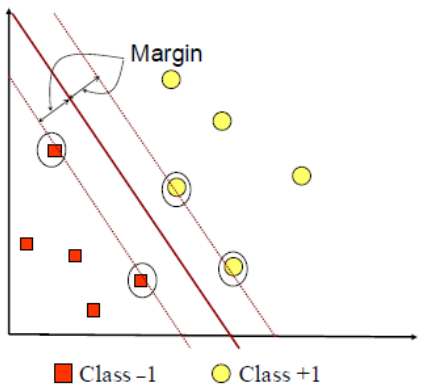
Support Vector Machine (SVM) is a prediction technique that can be used for cases classification and regression. SVM uses linear basic classifier principles, but has been developed to work on nonlinear problems by incorporating kernel concepts in high-dimensional workspaces. The basic idea of this technique is to find the best hyperplane by maximizing the hyperplane margin that serves as a separator of two data classes in the input space. Hyperplane with maximal margin will give better generalization on the classification method, so the result is more accurate [10].

Figure 2 shows an illustration of the SVM method in maximizing hyperplane boundaries [11]. Various alternative lines of discrimination (discrimination boundaries) are shown in Fig. 2(a), in which some patterns are members of two classes namely +1 and -1. Patterns belonging to class -1 are symbolized in red (box), while the pattern in class +1, symbolized by yellow (circle). The classification problem can be solved by finding a hyperplane that separates the two groups.

The best separator hyperplane between the two classes can be found by measuring the hyperplane's margins and searching for the maximum point. Margin is the distance between the hyperplane and the nearest pattern of each class. The closest pattern of the hyperplane is called support vector. The solid line in Fig. 2(b) shows the best hyperplane, which is located right in the middle of the two classes while the red and yellow dots that are in the black circle is the support vector.



(a)



(b)

Figure 2. Finding Best Hyperplane Process
a. Alternative Hyperplane; b. Best Hyperplane

Suppose there are data denoted as x_i:

$$x_i \in \Re^{\mathrm{D}}, i = 1,2,\dots,N \qquad (3)$$

where N is the number of data with the positive class which is denoted as +1 and the negative class which is denoted as -1. Then each data and class label is denoted as [12]:

$$y_i \in \{-1,+1\} \qquad (4)$$

It is further assumed that both classes can be perfectly separated by the hyperplane defined as [13]:

$$f(x) = w.x + b \qquad (5)$$

so the obtained equation is:

$$[(w.x_i) + b] \geq 1\, for\, y_i = +1$$
$$[(w.x_i) + b] \leq -1\, for\, y_i = -1 \qquad (6)$$

with x_i = training dataset $i = 1,2,\dots n$ and y_i = class label of x_i.

Hyperplane is best obtained by searching for a hyperplane located in the middle of two class dividing areas. The best position is equal to maximizing the margin or distance between the hyperplane and the nearest object. The best hyperplane determination can be formulated into Quadratic Programming (QP) Problem that minimizes [13]:

$$\frac{||w||^2}{2} \qquad (7)$$

provided that $y_i(w.x_i + b) \geq 1, \quad i = 1,2,3,\dots n.$

The solution for optimization is solved by using the Lagrange function as follows [14]:

$$L(w,b,\alpha) = \frac{1}{2}||w||^2 - \Sigma_{i=1}^{n}\alpha_i\{y_i[(w.x_i) + b] - 1\} \ (8)$$

with α_i = multiplier Lagrange function and $i = 1,2,\dots n$.

The optimal value can be calculated by maximizing L against α_i, and minimizing L against w and b as the case of dual problems [15]:

$$max_\alpha W(\alpha) = max_\alpha(min_{w,b}L(w,b,\alpha)) \qquad (9)$$

The minimum value of the Lagrange function is given by [16]:

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \Sigma_{i=1}^{n}\alpha_i y_i = 0$$
$$\frac{\partial L}{\partial w} = 0 \Rightarrow w = \Sigma_{i=1}^{n}\alpha_i x_i y_i \qquad (10)$$

To simplify, the Eq. 10 must be transformed into the Lagrange Multiplier function itself, thus becoming [13]:

$$L(w,b,a) = \frac{1}{2}w^T w - \Sigma_{i=1}^{n}\alpha_i y_i(w^T.x_i) - b\Sigma_{i=1}^{n}\alpha_i y_i + \Sigma_{i=1}^{n}\alpha_i \qquad (11)$$

become [17]:

$$L_d = \Sigma_{i=1}^{n}\alpha_i - \frac{1}{2}\Sigma_{i=1}^{n}\Sigma_{j=1}^{n}\alpha_i\alpha_j y_i y_j\, \mathrm{x}_i^T.x_j \qquad (12)$$

and obtained dual problems:

with limits,

$$\alpha \geq 0, \quad i = 1,2,\dots,n \quad dan \ \Sigma_{i=1}^{n}\alpha_i y_i = 0 \qquad (13)$$

The training data with $\alpha \geq 0$ which lies in hyperplane called support vector. The training data which is not located on the hyperplane has $\alpha_i = 0$.

Not all data can be separated linearly. However, SVM can also work on nonlinear data by applying the kernel approach to the initial data set data feature. The kernel function used to map the initial dimension (lower dimension) of the data is set to a new dimension (relatively higher dimension). So the data is mapped to a higher dimension to find the hyperplane. Suppose there is data on the input space denoted as [12]:

$$(\{x_i \in \Re^D\}) \qquad (14)$$

where D is the data dimension. It must be mapped to a new dimension of higher definition space (feature space) defined as:

$$(\{x'_i \in \Re^Q\}) \qquad (15)$$

The method used is kernel method in which a data x is mapped by the function $\phi: x \to \phi(x)$ [19]. This sorting is done while maintaining the data characteristics. For example, if there are two data whose distance is closer to the input space then the feature space space must also be adjacent and vice versa. Thus, the two classes can be separated linearly by a hyperplane [12]. So:

$$\phi : \mathfrak{R}^D \rightarrow \mathfrak{R}^Q, D < Q \tag{16}$$

Furthermore, the same training process as Linear SVM. The learning process in SVM in finding support vector points depends only on the dot product of both vectors (x_i) and (x_j) which have been transformed to feature space, i.e. φ $\phi(x_i).\phi(x_j)$. The dot product value of these two vectors can be calculated indirectly, i.e. without knowing the transformation function ϕ. This technique is called the kernel trick, which is counting the dot product of two vectors in the feature space by using the two vector components in the input space formulated as follows [12]:

$$K(x_i, x_j) = \phi(x_i).\phi(x_j) \tag{17}$$

To perform non-linear classification on SVM against data testing x, then formulated as follows:

$$\begin{aligned} f(\phi(x)) &= w.\phi(x) + b \\ &= \sum_{i=1, x_i \in SV}^{N} \alpha_i y_i \, \phi(x).\phi(x_i) + b \\ &= \textstyle\sum_{i=1, x_i \in SV}^{N} \alpha_i y_i \, K(x, x_i) + b \end{aligned} \tag{18}$$

where Support Vector (SV) is a subset of training data with a non-negative α. So classification of two classes can be done if $f(\phi(x)) < 0$, then class 1 and if $f(\phi(x)) > 0$, then the class is +1.

2.4. Optical Character Recognition

Optical Character Recognition (OCR) is a computer technology that can recognize characters in letters and numbers, whether from a printer (printer or typewriter) or from handwriting.

To use OCR, RGB image has to be converted into a binary image called thresholding process. The threshold value becomes very influential, where pixels that have values below the threshold become black, and pixels that have values above the threshold become white. However, many of the images encountered have rather large contrast ranges. In this case, a more sophisticated method for thresholding is required to get suitable results [18].

To get better character recognition results, an image enhancement process is required before performing OCR. One of the methods that can be used is Top Hat Transform.

3. Proposed Method

The system stages proposed in this study are shown in Fig. 3.

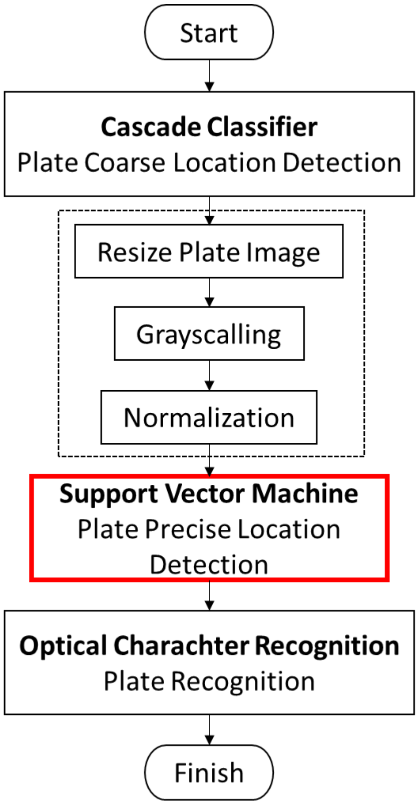


Figure 3. Flowchart of the proposed recognition system

3.1. Vehicle Plate Coarse Location Detection

The formation of the plate detection model proposed in this research uses the cascade classifier method because it has the advantage of being able to detect objects that have different sizes and be able to identify multi-objects. Then the descriptor used is LBP which is useful to describe the features of the image feature. LBP is defined as an invariant grayscale texture size, derived from the general definition of texture in the surrounding area. The plate coarse location detection training process in this study is shown in Fig. 4.

The training process of detection of plate candidates using two data, namely 1700 positive data frame and 3400 negative data as seen in Fig. 5. Parameters used are False Alarm Rate with value 0.2 and Number Stages for LBP set 20 to maximize training.

Based on the results of the analysis on the training process, obtained the size of the bounding box for the vehicle plate that has a wide range between 45 px – 70 px and long range between 150 px – 250 px. So all the detection results of plate candidate sized beyond the bounding box size are removed. Fig. 6 shows the bounding box results of the plate candidates detected by LBP. While, Fig. 7 shows the effect of cutting the bounding box of the plate candidate that has been recognized.

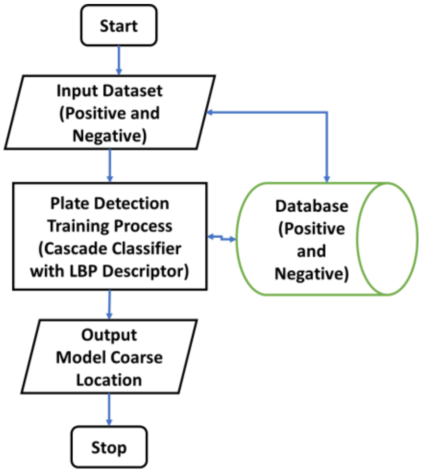
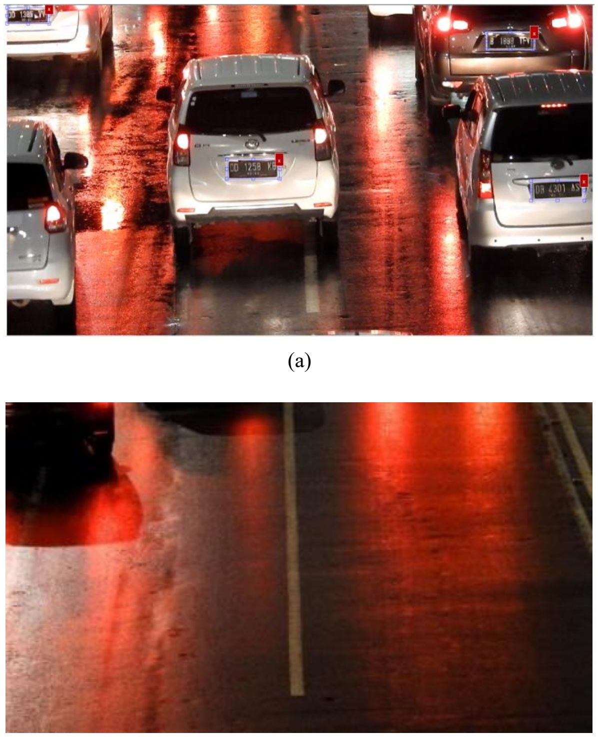

Start
Input Dataset (Positive and Negative)
Plate Detection Training Process (Cascade Classifier with LBP Descriptor)
Database (Positive and Negative)
Output Model Coarse Location
Stop

Figure 4. Plate Coarse Location Detection Training Process with LBP



(b)

Figure 5. LBP Training Data
(a) Positive Sample; (b) Negative Sample

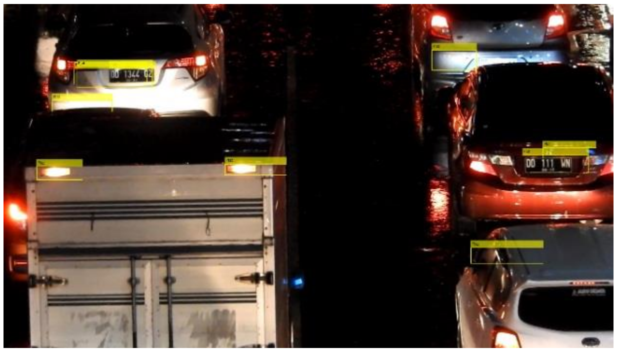


Figure 6. Bounding box Results example of the Plate Candidates Detected by LBP

There are many plate candidates detected incorrectly in Fig. 6. SVM method is added in the classification process to solve the problem. In addition, the plate candidate in Fig. 7 have different size, so it is necessary through the preprocessing process.

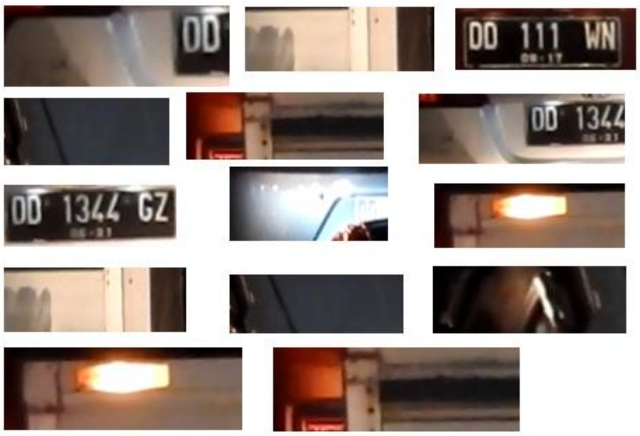


Figure 7. The example of plate candidate detected by LBP

3.2. Plate Candidate Preprocessing

Preprocessing the image of the plate candidate is required before performing the classification process using the SVM method. The first part, image conversion to grayscale. The second part, resizing the image become 14 x 42 px. This process is done to uniform all plate candidate matrices. The third part, normalization for scaling the value of each data. The recommended value of each plate candidate lies within the range of different values. After normalization, all features will be within the same value range [10]. So normalization is done by using Eq. 2 which produces all features within the same range of values.

3.3. Vehicle Plate Precise Location Detection

Detection of vehicle plates in a precise location in the research is a classification process using SVM method shown in Fig. 8.

In this study, the training data used were 300 plate images and 300 non-plate images that shown in Fig. 9.

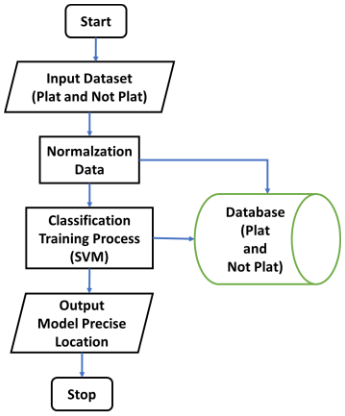
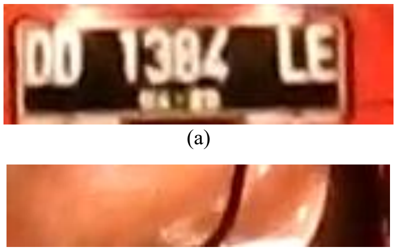

Start
Input Dataset (Plat and Not Plat)
Normalzation Data
Classification Training Process (SVM)
Database (Plat and Not Plat)
Output Model Precise Location
Stop

Figure 8. Plate Precise Location Detection Training Process with SVM



(a)

(b)

Figure 9. SVM Training Data
a. Plate; b. Non-Plate

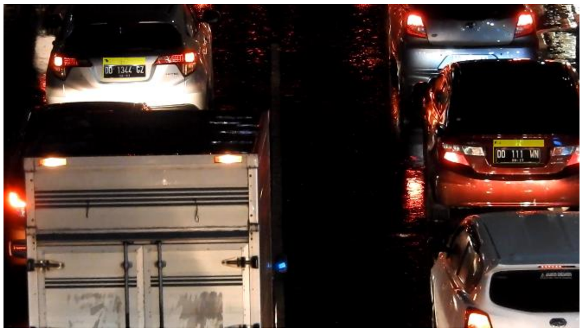


Figure 10. Detection results example using SVM

After SVM training, the training result model is used for a precise location detection testing process using video data whose results are shown in Fig. 10, where SVM succeeded in reducing the error of false plate candidate detection compared to the result of plate candidate detection with LBP shown in Fig. 6.

3.4. Plate Candidate Recognition

After the Precise Location detection process, then the next plate candidate goes into the recognition process using the method of Optical Character Recognition (OCR) with the addition of image enhancement method that is Top-Hat Transform to improve the results of recognition accuracy. The plate recognition training process with OCR is shown in Fig. 11.

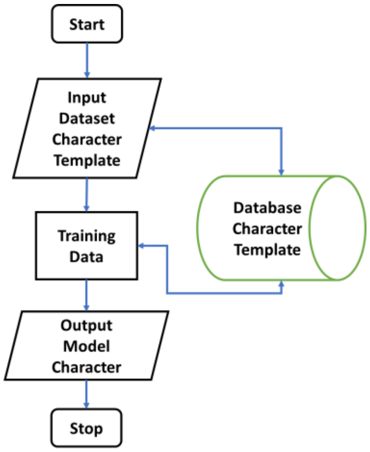


Figure 11. Plate Recognition Training Process with OCR

3.5. System Performance Validation

The performance of the plate recognition system using video data is shown in Fig. 12. The system performance to be measured in this study focuses on system execution time. Because the speed of the system in the process of recognition becomes very important, system validation is done by comparing execution time between recognition system using precise location detection process using SVM and recognition system without precise location detection process.

4. Results and Discussion

Testing the vehicle plate recognition system on moving vehicle at night and rain conditions using 19 videos with a duration of 30 seconds and includes 4-10 vehicles per video. This research has got average the accuracy of plate recognition of 61.94%.

Comparison of execution time with and without SVM in the classification process is shown in Table 1. The table shows that the average of execution time with SVM can reduce the average execution time of system testing process around 60% (=628 seconds).

The large magnitude of the plate candidate detection error in the process with the cascade classifier descriptor LBP method makes the recognition system takes a very long time. Because the detected plate candidates is detected correctly (plates) and those identified incorrectly (non-plates) all go into the plate recognition stage so that it still takes a lot of time for preprocessing before it is recognized. While, SVM be able to reduce the number of plate candidates is processed into recognition system so that the execution time becomes faster.

Table 1. Comparison of Execution Time

Video	**Execution Time (second)**	
	With SVM	**Without SVM**
1	383	923
2	364	1031
3	394	1208
4	360	1197
5	402	902
6	498	1258
7	471	1229
8	452	1067
9	330	1309
10	328	1120
11	402	1194
12	468	1112
13	415	1140
14	365	890

15	568	1244
16	311	927
17	429	664
18	495	798
19	379	528
Average	**411**	**1039**

5. Conclusion

The moving vehicle plate recognition system at night and rain conditions have its challenges because the system should have a fast execution time to ease the development of the system into real time. This research succeeded in reducing the execution time of system testing on video data by adding SVM classification process for plate candidates and not plates from average 1039 seconds to 411 seconds (= 60%).

For further research, the execution time of the system should be reduced again, and the accuracy of plate recognition also needs to be improved. Image enhancement method is one of the alternatives to enhance the image quality of plate candidates before being recognized.

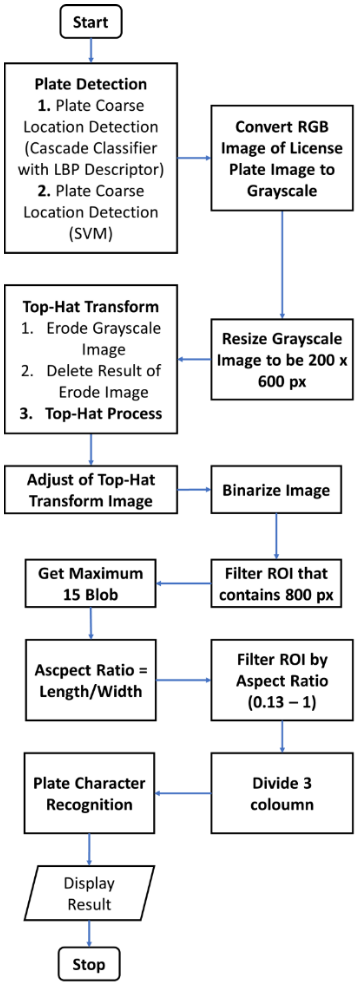

Start
Plate Detection
1. Plate Coarse Location Detection (Cascade Classifier with LBP Descriptor)
2. Plate Coarse Location Detection (SVM)
Convert RGB Image of License Plate Image to Grayscale
Resize Grayscale Image to be 200 x 600 px
Top-Hat Transform
1. Erode Grayscale Image
2. Delete Result of Erode Image
3. Top-Hat Process
Adjust of Top-Hat Transform Image
Binarize Image
Filter ROI that contains 800 px
Get Maximum 15 Blob
Ascpect Ratio = Length/Width
Filter ROI by Aspect Ratio (0.13 – 1)
Divide 3 coloumn
Plate Character Recognition
Display Result
Stop

Figure 12. Plate Recognition System Validation Process in Data Video

References

[1] Y. Yuan, W. Zou, Y. Zhao, X. Wang, X. Hu, and N. Komodakis, “A Robust and Efficient Approach to License Plate Detection,” *IEEE Trans. Image Process.*, vol. 26, no. 3, pp. 1102–1114, Mar 2017.

[2] A. Choudhury and A. Negi, “A new zone based algorithm for detection of license plate from Indian vehicle,” in *2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, 2016, page. 370–374.

[3] N. H. Barnouti, M. A. S. Naser, and S. S. M. Al-Dabbagh, “Automatic Iraqi license plate recognition system using back propagation neural network (BPNN),” in *2017 Annual Conference on New Trends in Information Communications Technology Applications (NTICT)*, 2017, page. 105–110.

[4] A. Elbamby, E. E. Hemayed, D. Helal, and M. Rehan, “Real-time automatic multi-style license plate detection in videos,” in *2016 12th International Computer Engineering Conference (ICENCO)*, 2016, page. 148–153.

[5] X. h Wang, J. j Yu, Z. h Miao, and Y. Song, “License plate recognition based on pulse coupled neural networks and template matching,” in *Proceedings of the 33rd Chinese Control Conference*, 2014, page. 5086–5090.

[6] B. B. Kumar, M. Bansal, and P. Verma, “Designing of Licensed Number Plate Recognition system using hybrid technique from neural network template matching,” in *2015 International Conference on Computing, Communication and Security (ICCCS)*, 2015, page. 1–6.

[7] A. C. Roy, M. K. Hossen, dan D. Nag, “License plate detection and character recognition system for commercial vehicles based on morphological approach and template matching,” in *2016 3rd International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*, 2016, page. 1–6.

[8] T. Ahonen, A. Hadid, and M. Pietikainen, “Face Description with Local Binary Patterns: Application to Face Recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, hal. 2037–2041, Des 2006.

[9] E. Prakasa, “Texture Feature Extraction by Using Local Binary Pattern,” J. INKOM, vol. 9, no. 2, pp. 45, Mei 2016.

[10] EkoPrasetyo, Digital image processing and application using Matlab. Yogyakarta: Andi, 2011. [Indonesian]

[11] A. S. Nugroho, A. B. Witarto, and D. Handoko. Support Vector Machine, Theory and Its Application in Bioinformatics. Public Lecture IlmuKomputer.Com. 2003.[Indonesian]

[12] Anonim, “Support Vector Machine (SVM) Java | Contoh Program,” 2013.

[13] B. Santosa, Data Mining Data Utilization Techniques for Business Purposes. Yogyakarta: Graha Ilmu, 2007. [Indonesian]

[14] C. Cortes dan V. Vapnik, “Support-vector networks,” Mach. Learn., vol. 20, no. 3, pp. 273–297, Sep 1995.

[15] S. R. Gunn, “Support Vector Machines for Classification and Regression,” University of Southampton, Southampton, 1998.

[16] Octaviani and P. Anna, “Application of Support Vector Machine Classification Method (SVM) on Primary School Accreditation Data (SD) In Magelang District,” Faculty of Science and Mathematics, Diponegoro University, Semarang, 2014. [Indonesian]

[17] T. Hastie, R. Tibshirani, and J. H. Friedman, The elements of statistical learning: data mining, inference, and prediction, 2nd ed. New York, NY: Springer, 2009.

[18] L. Eikvil, OCR - Optical Character Recognition. Norsk Regnesentral, 1993.

[19] H. Santoso, “Analysis and Prediction on Student Behavior of Diploma to Continue Study to Undergraduate Degree Using Decision Tree Technique and Support Vector Machine, "University of North Sumatera, Medan, 2012. [Indonesian]