# Comparison of Support Vector Machine Classifier and Naïve Bayes Classifier on Road Surface Type Classification


1st Susi Marianingsih
*Computer Vision Research Group*
*Faculty of Computer Science*
*Brawijaya University*
Malang, Indonesia
ssmarianingsih@student.ub.ac.id

2nd Fitri Utaminingrum
*Computer Vision Research Group*
*Faculty of Computer Science*
*Brawijaya University*
Malang, Indonesia
f3_ningrum@ub.ac.id



***Abstract*—This study describes the comparison of road surface classification results using Support Vector Machine (SVM) classification and Naive Bayes classification. The dataset in this study is a collection of sub-images (750 images) from the road area of the image from Google Street View. From the dataset, 600 images as data training and 150 images as data testing. Texture features are extracted from road surface images in the dataset and then we build SVM and Naive Bayes classifiers to classify road surface images in 3 categories, asphalt, gravel, and paving. Evaluation of performance classification using precision, recall, f-measure, and accuracy. The results show that SVM classifier accuracy better than Naive Bayes classifier.**



***Keywords—Texture analysis, GLCM, SVM, Naïve Bayes.***


## I. INTRODUCTION

Knowledge of the type of road surface that will be passed is information that is important for the driver because the driver can determine the speed of the vehicle and avoid accidents. Research to determine road surface conditions (such as dry, wet, snowy, icy and watery road surface conditions) has been carried out. Yang et al. [1] use various features (global and local) and SVM classifiers to classify road surface conditions. Zhao et al. [2] using 13-dimensional color and texture eigenvectors and SVM classifiers to identify road surface conditions with hybrid conditions (different video scenes). Grid searching algorithms and PSO (Particle Swarm Optimization) algorithms are also used to optimize the kernel function factor and penalty factor of SVM. Sun et al. [3] Extracted road surface texture feature using GLCM.

Feature extraction is a critical stage in the classification of road surfaces. One feature that is often used in identifying road surfaces is texture. Texture can be defined as a repetitive pattern arrangement in a region [4]. Texture provides information on the arrangement of surface structures or changes in color intensity or brightness [Siqueira]. Texture features can be obtained using statistical methods, structural methods, and spectral methods. One statistical method that has proven to be the most powerful texture descriptor in image analysis is GLCM [5]. GLCM was introduced by Haralick et al. [6]. GLCM is a second-order statistical method that describes the spatial relationship of the gray level of an image. GLCM contains elements that are calculated from the number of pairs of pixels, separated by a certain distance and in certain angular directions.

KNN, Naïve Bayes, Artificial Neural Networks, Fuzzy Logic, and SVM are some commonly used learning machine algorithms. Vapnik et al. developed SVM at AT & T Bell Laboratories in the 1990s. [7][8]. The main idea of SVM is to separate classes with surfaces that maximize margins between them. In this study SVM classifier is used because SVM has a high generalization capability without additional requirements even though it has high dimension input [9]. SVM is a very useful approach to data classification and regression [10].

This research aims to compare the SVM and Naïve Bayes classifier in road surface type problem. Road surface image features extraction using GLCM.

The remainder of the paper writing structure is as follows, Section II describes the methodology of this study, Section III is the experiment result, and Section IV contains the conclusion.

## II. METHODOLOGY

### *A. Road Surface Image Dataset*

The source of the dataset from this study was taken from Instant Google Street View [11]. Instant Google Street View is one of the Google Map features that provides $360^0$ street views and frees users to view and take pictures from an area. Sources of datasets consist of 70 paved road images, 70 gravel road pictures, and 90 paving road pictures, taken in good lighting conditions. Fig. 1 is an example of each road image. Referring to the source road images, we manually extract 100×100 sub-images to build our dataset (see Fig. 2).

This research uses GLCM to extract texture features and uses the support vector machine method for training and testing dataset.

The dataset was partitioned into a training set of 600 images (200 images asphalt road, 200 image gravel road, and 200 image pavement road), and a test set of 150 images (50 images asphalt road, 50 image gravel road, and 50 image pavement road). Fig. 3(a), Fig. 3(b) and Fig. 3(c) are some sample image of the dataset.


978-1-5386-7407-9/18/$31.00 ©2018 IEEE

Fig. 1. The sample of road images from (a) Asphalt Road, (b) Gravel Road, and (c) Pavement Road classes.


Fig. 2. Capture sub-images from road area.

(a)

(b)

(c)

Fig. 3. The sample of Dataset (Asphalt Road (a), Gravel Road (b), Paving Road (c)

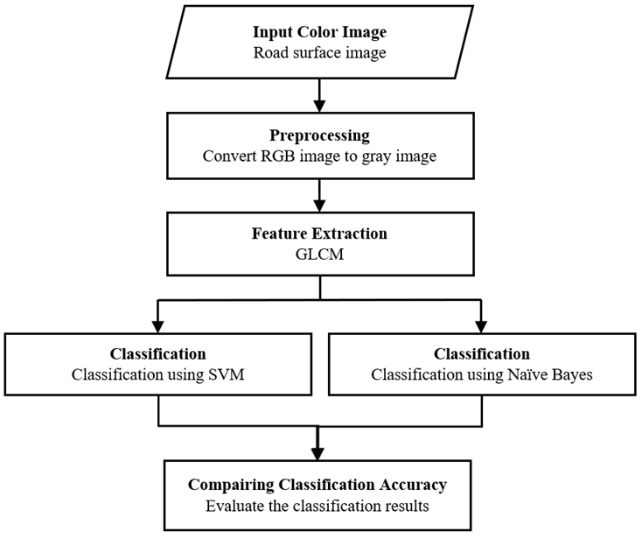


Fig. 4. System Design.

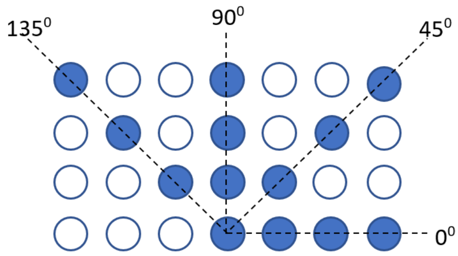


Fig. 5. The direction $\vec{r}$ can be angular ($\theta$) and distance ($d$).

## B. System Design

The system design is made to show research workflows. The process starts with the acquisition of the road surface image and continues with the pre-processing stage (changing the color image to gray image). Next, the feature extraction process uses GLCM [6], [12]. The GLCM results feature is input for SVM Classifiers and Naive Bayes Classifiers. Fig. 4 presented the system design. Then the classification results are compared.

## C. Texture Feature Extraction

GLCM was proposed by Haralick et al. [6] with 14 features describing the spatial pattern of gray images. GLCM is a calculation of how often different combinations of pixel brightness values occur in an image. GLCM uses the second-order statistical calculation on the second order. Second order means considering the relationship between two-pixel groups of gray images [12].

The first step to extracting features with GLCM is to change the color image into an image on a grayscale, then it will be referred to as the original image. The next step is to form a GLCM Matrix framework from the original image. The GLCM matrix $(P_{\vec{r}}(i,j))$ is the number of pixels with $i \in 1..L$ that occurs in the direction of $\vec{r}$ against pixels with a value of $j \in 1..L$. The direction $\vec{r}$ can be angular $(\theta)$ and distance $(d)$, Fig. 5. Fig. 6 is an example of a process to produce a GLCM matrix framework from the original image with 4 gray level (0,1,2,3) for $d$=1 and $\theta$=1.

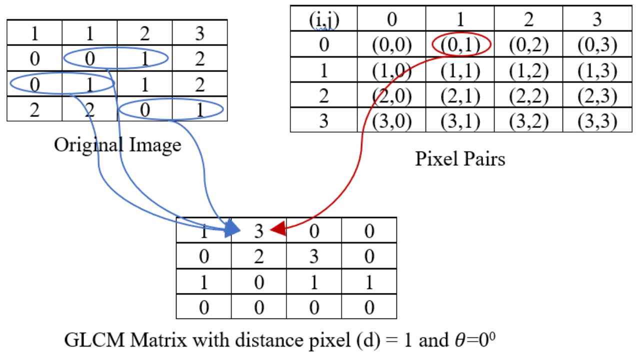


Fig. 6. An example of a process to produce a GLCM matrix framework from the original image with 4 gray level (0,1,2,3) for $d$=1 and $\theta$=1.

The GLCM matrix framework is converted into a symmetrical GLCM matrix by adding up the matrix transpose, equation (1). To eliminate dependence on image size, the symmetrical GLCM matrix [12] is normalized by the formula in the equation (2).

$$\begin{bmatrix}1&3&0&0\\0&2&3&0\\1&0&1&1\\0&0&0&0\end{bmatrix}+\begin{bmatrix}1&0&1&0\\3&2&0&0\\0&3&1&0\\0&0&1&0\end{bmatrix}=\begin{bmatrix}2&3&1&0\\3&4&3&0\\1&3&2&1\\0&0&1&0\end{bmatrix} \quad (1)$$

Denotes $i$ is the row number, $j$ is the column number, $V_{i,j}$ is the value in the cell $i,j$ of symmetrical GLCM matrix, $P_{i,j}$ is the probability value recorded for the cell $i,j$.

$$P_{i,j} = \frac{V_{i,j}}{\sum_{i,j=0}^{N-1} V_{i,j}} \quad (2)$$

Normalization of symmetrical GLCM matrix in the equation (3) (called simply the GLCM from).

$$\begin{bmatrix}\frac{1}{12}&\frac{1}{8}&\frac{1}{24}&0\\\frac{1}{8}&\frac{1}{6}&\frac{1}{8}&0\\\frac{1}{24}&\frac{1}{8}&\frac{1}{12}&\frac{1}{24}\\0&0&\frac{1}{24}&0\end{bmatrix} \quad (3)$$

Based on GLCM form, we calculate the features for each image in the dataset. This study uses 14 features proposed by Haralick [6]: Angular Second Moment (f_1), Contrast (f_2), Correlation (f_3), Sum of Squares (f_4), Inverse Difference Moment (f_5), Sum Average (f_6), Sum Variance (f_7), Sum Entropy (f_8), Entropy (f_9), Difference Variance (f_10), Difference Entropy (f_11), and Information Measure of Correlation (f_12, f_13).

*D. Support Vector Machine.*

The SVM method is one of the states of the art in pattern recognition [13]. SVM was developed by Vapnik et al. [8], [14] and was first presented at the Annual Workshop on Computational Learning Theory. The first development of SVM is used to classify linear data, and in its development can solve non-linear problems by using the kernel trick concept on high-dimensional data.

Fig. 7 shows a pattern that is a member of two classes. The white pattern symbolizes the pattern in class -1 and the pattern in class 1 is symbolized in black. Classifying means finding a line (hyperplane) that separates the patterns based on their class. Black, red and blue lines are some alternative dividing lines. The solid line in Fig. 8 is the best hyperplane. The margin is the distance between the hyperplane and the closest pattern of each class. The pattern closest to the hyperplane is called support vector. The process of finding the best hyperplane pattern from these patterns is the basic concept of SVM [9]. The data is denoted by $x_i \in R^d$ while the class is denoted $y_i \epsilon \{-1,1\}$ for $i = 1, \ldots, l$ where $l$ is any data. Suppose that both classes can be separated by a dimension $d$ hyperplane, which is defined in equation (4).

$$w.x + b = 0 \quad (4)$$

The $x_i$ data which includes class -1 and class 1 are formulated in equations (5) and (6).

$$w.x_i + b \le -1 \quad for\ y_i = -1 \quad (5)$$

$$w.x_i + b \ge 1 \quad for\ y_i = 1 \quad (6)$$

These can be combined into one set of inequalities (7).

$$y_i(w.x_i + b) - 1 \ge 0 \quad \forall i \quad (7)$$

The biggest margin is determined by maximizing the distance value of the hyperplane with its closest point, which is $1/\|w\|$ . This can be formulated as a Quadratic Programming (QP) problem, which is finding the minimum point of equation (8), by considering the constraint of equation (7).

$$\min \tau(w) = \frac{1}{2}\|w\|^2 \quad (8)$$

This problem can be solved with the Lagrange Multiplier Technique in equation (9).

$$L(w,b,\alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{l} \alpha_i y_i((x_i.w + b) - 1) \quad (9)$$

Denotes $i = 1, \ldots, l$ , $l$ is many data, $\alpha_i$ is a Lagrange multiplier, which is zero or positive ($\alpha_i \ge 0$). The optimal value of equation (9) can be calculated by minimizing $L$ against $w$ and $b$ and maximizing $L$ against $\alpha_i$ . At the optimal point of the gradient $L = 0$, equation (9) can be modified as problem maximization, equation (10).

$$L_D \equiv \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i,j=0}^{l} \alpha_i\alpha_j y_i y_j x_i.x_j \quad (10)$$

Subject to equations (11) and (12).

$$\alpha_i \ge 0,\ i = 1, \ldots, l \quad (11)$$

$$\sum_{i=1}^{l} \alpha_i y_i = 0 \quad (12)$$

This data which correlates with positive $\alpha_i$ is called support vector.

If the hyperplane cannot separate the data perfectly, then the constraint in equation (7) cannot be fulfilled. To overcome this problem, SVM was reformulated with soft margin techniques. Equation (7) is modified by entering the slack variable (ξ), (see equation (13)).

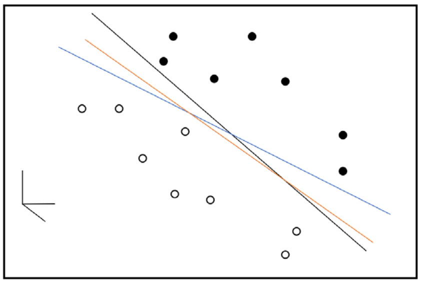

Fig. 7. Black, red and blue lines are some alternative dividing lines (hyperplane).

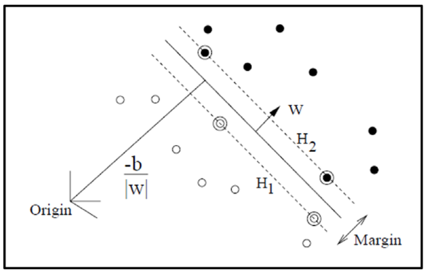


Fig. 8. Linear separating hyperplanes. The support vectors are circled.

$$y_i(w.x_i + b) \geq 1 - \xi, \quad \forall i \qquad (13)$$

Then equation (8) is changed to equation (14).

$$\min \tau(w, \xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l} \xi_i \qquad (14)$$

Parameter $C$ serves to control the tradeoff between margins and classification errors ξ. A large $C$ value means that it will give a greater penalty for classification errors.

For non-linear cases, SVM uses kernel functions. The $x_i$ data is mapped by the function $\Phi(x_i)$ to the vector space with a higher dimension. The mathematical notation of this mapping in the equation (15)

$$\Phi: R^d \rightarrow R^q,\ d < q \qquad (15)$$

The learning process in SVM in finding support vector points only depends on the dot product of the data in the new space with a higher dimension, namely $\Phi(x_i).\Phi(x_j)$. According to Mercer's theory, the dot product calculations can be replaced with kernel functions $K(x_i, x_j)$. This is called the Kernel trick, which is formulated in the equation (16).

$$K(x_i, x_j) = \Phi(x_i).\Phi(x_j) \qquad (16)$$

Table I shows some types of kernel functions. Furthermore, the classification results from data $x$ are obtained from the equation (17).

$$f(\Phi(x)) = \sum_{i=1, x_i \epsilon SV}^{l} \alpha_i y_i K(x, x_j) + b \qquad (17)$$

Denotes $SV$ is the data selected as support vector

TABLE I. SOME KERNELS USED BY SVM CLASSIFIER

| Kernel | Definition |
|---|---|
| Polynomial | $K(x_i,x_j) = (x_i.x_j + 1)^p$ |
| Gaussian RBF | $K(x_i,x_j) = exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})$ |
| Sigmoid | $K(x_i,x_j) = tanh(\alpha x_i.x_j + \beta)$ |

TABLE II. THE CONFUSION MATRIX

| | | Predictive | |
|---|---|---|---|
| | | *Relevant* | *Irrelevant* |
| **Actual** | ***Relevant*** | True Positive *(TP)* | False Negative *(FN)* |
| | ***Irrelevant*** | False Positive *(FP)* | True Negative *(TN)* |

## *E. Naïve Bayes Classifier*

The Naive Bayes classifier is based on Bayes's theorem about the probability [15]. For data $x \epsilon X$ and class $k \in C$, the conditional probability that an event $x$ belongs to a class $k$ can be calculated by using the equation (18).

$$P(c_k|x) = P(c_k)\frac{P(x|c_k)}{P(x)} \qquad (18)$$

$P(c_k|x)$ is the posterior probability of class $c_k$ given feature $x$. $P(c_k)$ is the prior probability of class $c_k$ and $P(x)$ is the prior probability of feature $x$.

Next, we need to estimate $P(x|c_k)$ and assume that any particular value of vector $x$ conditional on class $c_k$ is statistically of each dimension and can be written as the equation (19).

$$(x|c_k) = \prod_{i=0}^{n} P(x|c_k) \qquad (19)$$

Where $x$ is a $n$-dimensional vector data $x = (x_1, x_2, \dots, x_3)$

We assume a certain form of probability distribution if the feature $x$ is the continuous value. Gaussian distributions are used to represent conditional-class probabilities for continuous variables. For each class $c_k$, the conditional probability function for feature $x_i$ is formulated in the equation (20).

$$P(x_i \mid c_k) = \frac{e^{-\frac{(x_i - \mu_{c_k})^2}{2\sigma_{c_k}^2}}}{\sqrt{2\pi\sigma_{c_k}}} \qquad (20)$$

Where the probability density is formed during the training stage of local classifiers with the mean $\mu_{c_k}$, and standard deviation, $\sigma_{c_k}$, of each $i$-th class data for each feature vector $x_k$.

The posterior probability is calculating for each class and the class with the highest posterior probability is the predicted one. The estimate class corresponding to $x_k$ is shown in the equation (21).

$$\hat{C}_k = max\ P(c_k|x(x_1, x_2, \dots, x_n) \quad (21)$$

TABLE III. CONFUSION MATRIX

| *Classifier* | *Class* | Experiment Result | | |
|---|---|---|---|---|
| | | *Asphalt* | Gravel | Pavement |
| ***SVM*** | ***Asphalt*** | **42** | 0 | 8 |
| | ***Gravel*** | 0 | **40** | 10 |
| | ***Pavement*** | 0 | 0 | **50** |
| ***Naïve Bayes*** | ***Asphalt*** | **45** | 2 | 3 |
| | ***Gravel*** | 0 | **48** | 2 |
| | ***Pavement*** | 3 | 14 | **33** |

TABLE IV. PERFORMANCE MEASUREMENTS

| *Classifier* | *Class* | *Recall* | *Precision* | *F-Measure* |
|---|---|---|---|---|
| ***SVM*** | ***Asphalt*** | 1.0 | 0.84 | 0.91 |
| | ***Gravel*** | 1.0 | 0.8 | 0.89 |
| | ***Pavement*** | 0.74 | 1.0 | 0.85 |
| ***Naïve Bayes*** | ***Asphalt*** | 0.94 | 0.90 | 0.92 |
| | ***Gravel*** | 0.75 | 0.96 | 0.84 |
| | ***Pavement*** | 0.87 | 0.66 | 0.75 |

TABLE V. PERFORMANCE MEASURE OF CLASSIFIERS

| *Classifier* | *Recall* | *Precision* | *F-Measure* | *Accuracy* |
|---|---|---|---|---|
| ***SVM*** | 0.91 | 0.88 | 0.91 | 0.88 |
| ***Naïve Bayes*** | 0.85 | 0.84 | 0.84 | 0.84 |

## III. RESULT

### A. Performance Measures

This study uses recall, precision, classifier F1 rating and accuracy [16], [17] to measure the performance of the system. The Confusion Matrix as the predictive classification shown in Table II.

Recall ($R$) is a proportion of the relevant pages that were correctly identified. Precision ($P$) is a proportion of the predicted relevant pages that were correct. F-Measure ($F$) derives from precision and recall value. Accuracy ($A$) is a portion of the total number of prediction that was correct. The measurements equation we consider are :

$$R = \frac{TP}{FN + TP} \quad (22)$$

$$P = \frac{TP}{FP + TP} \quad (23)$$

$$F = \frac{2\ x\ Recall\ x\ Precision}{Recall + Precision} \quad (24)$$

$$A = \frac{TN + TP}{TN + FN + TP + FP} \quad (25)$$

Denotes *TP (True Positive)* is number of correct predictions that an instance is relevant, *TN (True Negative)* is number of correct predictions that an instance is irrelevant, *FP (False Positive)* is number of correct predictions that an instance is relevant, and *FN (False Negative)* is number of incorrect predictions that an instance is irrelevant.

### B. Experiment Result

The road surface dataset is classified as the four types such as asphalt road, gravel road, and paved road. The configured dataset as shown in Fig. 3. Dataset consists of 750 images, 600 images (200 asphalt, 200 gravel, 200 pavement) as data training and 150 images (50 asphalt, 50 gravel, 50 pavement) as data testing.

RBF kernel is applied to SVM. The classification results of the system are compared with the Naive Bayes Classifier. Tables III show confusion matrices and bold values in Table III show the correct test results for each class. Tables IV shows Recall, Precision, and F-Measure of the two methods for each class. Table V shows the average performance values of SVM classifier and Naive Bayes classifier, SVM Classifier is slightly higher than Naïve Bayes classifier.

## IV. CONCLUSION

The dataset in this study is a collection of sub-images of the road area on the road image taken from Google Street View. The dataset consists of 750 images, 600 images as training data and 150 images as testing data. We use 13-feature vectors extracted using the GLCM method. This feature vector is an input for SVM and Naive Bayes classifiers. This experiment has been evaluated by using Precision, Recall, F-measure, and Accuracy. The experimental results show that SVM classifiers have better accuracy than Naive Bayes classifiers

## ACKNOWLEDGMENT

The research activities in this paper were supported by the Computer Vision Research Group, Faculty of Computer Science, Brawijaya University.

## REFERENCES

[1] Hun-Jun Yang, Hyeok Jang, Jong-Wook Kang and Dong-Seok Jeong, "Classification Algorithm for Road Surface Condition," *IJCSNS International Journal of Computer Science and Network Security*, vol. 14, Jan. 2014.

[2] J. Zhao, H. Wu, and L. Chen, "Road Surface State Recognition Based on SVM Optimization and Image Segmentation Processing," *Journal of Advanced Transportation*, vol. 2017, pp. 1–21, 2017.

[3] Z. Sun and K. Jia, "Road Surface Condition Classification Based on Color and Texture Information," 2013, pp. 137–140.

[4] W. R. Schwartz and H. Pedrini, "Textured Image Segmentation Based on Spatial Dependence using a Markov Random Field Model," 2006, pp. 2449–2452.

[5] F. Roberti de Siqueira, W. Robson Schwartz, and H. Pedrini, "Multi-scale gray level co-occurrence matrices for texture description," *Neurocomputing*, vol. 120, pp. 336–345, Nov. 2013.

[6] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural Features for Image Classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, no. 6, pp. 610–621, Nov. 1973.

[7] Corinna CortesVladimir Vapnik, “Support-Vector Networks,” *Kluwer Academic Publishers-Plenum Publishers*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[8] Vladimir N. Vapnik, *Statistical Learning Theory*. Canada: John Wiley & Sons, Inc., 1998.

[9] CHRISTOPHER J.C. BURGES, “A Tutorial on Support Vector Machines for Pattern Recognition,” *Kluwer Academic Publishers-Plenum Publishers*, vol. 2, no. 2, pp. 121–167, Jun. 1998.

[10] R. Salat and S. Osowski, “Accurate Fault Location in the Power Transmission Line Using Support Vector Machine Approach,” *IEEE Transactions on Power Systems*, vol. 19, no. 2, pp. 979–986, May 2004.

[11] “https://www.instantstreetview.com/.”.

[12] Mryka Hall-Beyer, “GLCM Texture: A Tutorial v. 3.0 March 2017,” 2017.

[13] H. Byun and S.-W. Lee, “A SURVEY ON PATTERN RECOGNITION APPLICATIONS OF SUPPORT VECTOR MACHINES,” *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 17, no. 03, pp. 459–486, May 2003.

[14] B. E. Boser, I. M. Guyon, and V. N. Vapnik, “A training algorithm for optimal margin classifiers,” 1992, pp. 144–152.

[15] D. Lowd and P. Domingos, “Naive Bayes models for probability estimation,” in *Proceedings of the 22nd international conference on Machine learning - ICML ’05*, Bonn, Germany, 2005, pp. 529–536.

[16] I. K. Somawirata and F. Utaminingrum, “Road detection based on the color space and cluster connecting,” 2016, pp. 118–122.

[17] POWERS, D.M.W., "EVALUATION: FROM PRECISION, RECALL, AND F-MEASURE TO ROC, INFORMEDNESS, MARKEDNESS & CORRELATION," *Journal of Machine Learning Technologies*, 2011.