JPPIPA 10(11) (2024)

Jurnal Penelitian Pendidikan IPA

*Journal of Research in Science Education*

http://jppipa.unram.ac.id/index.php/jppipa/index

# Support Vector Machine for Classification: A Mathematical and Scientific Approach in Data Analysis

Yulia Restiani[1*], Joko Purwadi[1]

[1] Department of Mathematics, Faculty of applied science and technology, Universitas Ahmad Dahlan, South Ringroad Street, Kragilan, Tamanan, Banguntapan District, Bantul Regency, Yogyakarta, 55191, Indonesia.

Received: June 15, 2024
Revised: August 24, 2024
Accepted: November 25, 2024
Published: November 30, 2024

Corresponding Author:
Yulia Restiani
yuliarestiani2@gmail.com

DOI: 10.29303/jppipa.v10i11.8122



**Abstract:** In this research, SVM will be used to differentiate between plain nail art designs (class 0), 3D nail art designs (class 1), and hand painting nail art designs (class 2). The dataset used consists of images of nail designs that have been collected and analyzed previously. First, the dataset is divided into three different classes based on the type of nail design. The first class (class 0) includes plain nail art designs, then the second class (class 1) is 3D nail art designs, and the third class (class 2) is hand painting nail art designs. This process is carried out to allow SVM to learn the feature differences between the two types of designs. The data used will be divided into training and testing data and divided into three data division schemes, namely 60/40, 70/30, and 80/20. Based on the results of the research discussed, it can be concluded that classification using the Linear SVM model on three data sharing schemes provides the best level of accuracy on the 80/20 scheme, namely 81.25%. Meanwhile, classification using the non-linear SVM model achieved the highest level of accuracy of 95% in the 80/20 scheme with the RBF Kernel. Thus, the SVM model that is suitable for classifying nail art designs is a non-linear SVM model with the 80/20 scheme. The accuracy results obtained from this research also show that SVM provides good performance in classifying nail art designs.

**Keywords:** Classification; Kernel; Support vector machine

## Introduction

Along with technological advances, the availability of information on the internet is also increasingly widespread and significant (Abbas, 2018; Divayana, Suyasa, & Widiartini, 2021). However, without utilization, this information will just be a collection of data that has no value (Ardani, Utomo, & Rahmawati, 2021; Divayana et al., 2021; Jain, Ahirwar, & Pandey, 2019). The existence of this information allows individuals to identify and interpret patterns that can support the decision-making process, such as through the application of data mining techniques. One of the data mining methods used to predict decisions is through the classification process (Agarwal, 2014). Classification is a method of grouping objects based on the characteristics they have (Wibawa et al., 2018). This classification includes several algorithms that use fuzzy logic, Bayes classification, decision trees, support vector machines, artificial neural networks, and k-nearest neighbor methods (Jain et al., 2019).

Support Vector Machine (SVM) shows a higher level of accuracy when compared to k-nearest neighbor, decision trees, and linear regression methods (Charleonnan et al., 2017). Experimental results show that the SVM classifier achieves the highest level of accuracy and shows the highest sensitivity after going through the training and testing process using the proposed method. SVM is used to identify the optimal hyperplane that can well separate the two classes, resulting in superior generalization capabilities. In general, this approach involves using mostly labeled

**How to Cite:**
Restiani, Y., & Purwadi, J. (2024). Support Vector Machine for Classification: A Mathematical and Scientific Approach in Data Analysis. *Jurnal Penelitian Pendidikan IPA*, *10*(11), 9896–9903. https://doi.org/10.29303/jppipa.v10i11.8122

data to find the optimal hyperplane. However, when dealing with large-scale experimental data, the complexity of the computational process can increase (Avci et al., 2023; Elshewey et al., 2023; Niu et al., 2024; Yao et al., 2013).

In situations where the data cannot be separated linearly, the approach is carried out through a kernel method that transforms the input data into a feature space that has higher dimensions (Octaviani et al., 2019). Kernel usage varies depending on the characteristics of the data at hand. Linear kernels are used when the data to be classified can be separated by a hyperplane in the form of a line. In other words, linear kernels are suitable for two-dimensional data. Meanwhile, non-linear kernels are used when the data is separated by a plane-shaped hyperplane in a higher dimensional space (Atik et al., 2023; Jana et al., 2023; Puspitasari et al., 2018).

Nails are a small part of the human body that grow and are connected to the skin folds, consisting of dead and hardened skin epidermal cells. They develop into plates and form as they emerge from the fingertips, aiming to protect the underside of the nail. To maintain their health and increase their attractiveness, nail care is necessary, and they can be decorated with techniques such as nail art (Krisnawati et al., 2022). Nail art is a development of the practice of manicure and pedicure. Recently, nail art has become an important element in completing a fashion style, and many new techniques continue to develop. Women often like nail designs in styles such as French manicure or patterns such as dots or lines. Sometimes, they choose to put solid colors on their nails and add decorative elements on some fingers (Kim & Jeong, 2014). There are many techniques that can be classified in creating nail art, including dangling, stickers, design sculptures, rhinestones, marble, striping tape, airbrush, 3D art, hand painting, and so on. In addition, nail art techniques are often used to create various shapes and designs on nails (Jeong & Kim, 2015).

Classifying nail art involves handling large amounts of visual data, such as images of nails with various designs (Burges, 1998; Chang & Lin, 2011; Gandhi, 2018). SVM can be used to create models that can differentiate between various nail art styles and motifs with a high degree of accuracy. By using this approach, we can speed up the design identification process, provide a better experience for users, and even open up opportunities to develop applications that can provide nail design suggestions that suit personal desires.

This research aims to apply the Support Vector Machine (SVM) algorithm in nail art classification with the main aim of evaluating and measuring the accuracy of the model in correctly recognizing and classifying various nail designs.

## Method

Support Vector Machine (SVM) was first made famous by Vladimir Vapnik in 1992 with two friends, Bernhard Boser and Isabelle Guyon (Nayak et al., 2015). Support Vector Machine (SVM) is a machine learning algorithm and is used for classification and regression problems (Ritonga & Purwaningsih, 2018). Support Vector Machine (SVM) is applied to separating class data or identifying decision boundaries in high-dimensional space. The basic concept of Support Vector Machine (SVM) is to find the largest margin on the hyperplane and be able to separate two data classes as well as possible (Saputra, Puspitasari, & Baidawi, 2022). A hyperplane is a decision boundary that divides two classes or data separation boundaries between classes. In two-class classification problems (binary classification), Support Vector Machine (SVM) looks for hyperplanes by maximizing margins. Margin is the shortest distance between the hyperplane and the closest point or pattern from each class (Rizal, Girsang, & Prasetiyo, 2019). These points or patterns are then called support vectors (Somantri, Wiyono, & Dairoh, 2016). Then Support Vector Machine (SVM) is updated by combining several binary classifiers so that it can be used for multi-class classification problems.

Linear Support Vector Machine can be applied to any data separated linearly. The concept is to look for hyperplanes or lines the best separation between the two classes. Let $\boldsymbol{x_i} = \{\boldsymbol{x_1}, \dots, \boldsymbol{x_n}\}, \boldsymbol{x_i} \in \boldsymbol{R^n}$ is a data set. Positive class is denoted by +1, and negative class denoted by -1. Then the class label is denoted as $\boldsymbol{y_i} \in \{+\boldsymbol{1}, -\boldsymbol{1}\}$, $\boldsymbol{i} = \boldsymbol{1}, \boldsymbol{2}, \dots, \boldsymbol{n}$, where $\boldsymbol{n}$ denotes the amount of data.

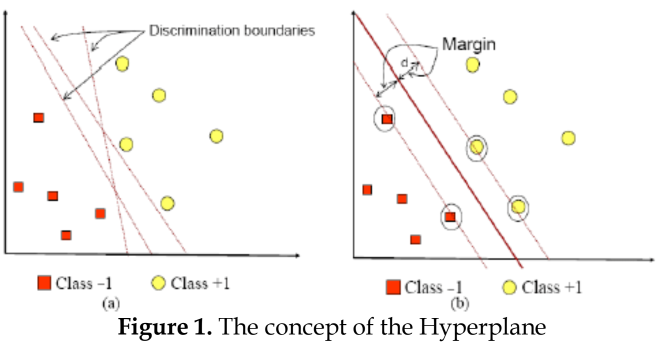


**Figure 1.** The concept of the Hyperplane

In the image above, there are several patterns that imply members of each class, namely the yellow class, class +1 and the red class, namely class -1. Then, in image (a), the existing data set can be separated based on its class by applying several hyperplanes (discrimination boundaries). In figure (b) above, there is a parallel hyperplane that divides the two classes with the pattern in the hyperplane called the support vector. The similarities in Support Vector Machine can be seen in (N. Deng, 2013):

$$w_i x_i + b = 0 \quad (1)$$

Where $\boldsymbol{w_i}$ is a weight parameter and $\boldsymbol{b}$ is a bias parameter and the inequality of 2 hyperplanes applies with each hyperplane limiting each class, namely:

$$w_i x_i + b > 0, y_i = +1 \quad (2)$$
$$w_i x_i + b < 0, y_i = -1 \quad (3)$$

For example, $\boldsymbol{H}: \boldsymbol{w_i x_i + b = 0}$ is the hyperplane to be searched, $\boldsymbol{H_1}: \boldsymbol{w_i x_i + b = 1}$ and $\boldsymbol{H_2}: \boldsymbol{w_i x_i + b = -1}$ are hyperplanes from each class, namely class +1 and class -1. For To find the best $\boldsymbol{H}$, the distance between $\boldsymbol{H_1}$ and $\boldsymbol{H_2}$ to $\boldsymbol{H}$ must be the same, namely for $\boldsymbol{H_1}$ using positive data samples and negative data samples for $\boldsymbol{H_2}$. This data sample is then called a support vector because it functions as a determinant for finding the optimal hyperplane.

Suppose $(\boldsymbol{x_0}, \boldsymbol{y_0}) \in \boldsymbol{R^2}$ is any point then the distance from that point to the line $\boldsymbol{ax + by + c = 0}$ is:

$$\frac{|ax+by+c=0|}{\sqrt{a^2+b^2}} \quad (4)$$

So, the distance between data $\boldsymbol{x}$ samples from $\boldsymbol{H_1}$ to $\boldsymbol{H}$ is:

$$\frac{|wx+b|}{\sqrt{w^T w}} = \frac{1}{||w||} \quad (5)$$

because the distances between $H_1$ and $H_2$ to $H$ are the same, the distance between $H_1$ and $H_2$ is $\frac{2}{||w||}$.

The problem of maximizing $\frac{2}{||w||}$ is the same as minimizing $\frac{||w||^2}{2}$ with the condition that there are no data samples between $H_1$ and $H_2$ as follows:

$$w_i x_i + b \geq 1, y_i = +1 \quad (6)$$
$$w_i x_i + b \leq 1, y_i = -1 \quad (7)$$

If you combine the two, you get $\boldsymbol{y_i(w_i x_i + b) \geq 1}$. The problem of finding the optimal $\boldsymbol{w}$ and $\boldsymbol{b}$ parameters to obtain the best or optimal hyperplane can be formulated in the Quadratic Programming (QP) problem, namely:

$$\min_{w.b} \frac{1}{2} ||w||^2 \quad (8)$$

With constraints

$$y_i(w_i x_i + b) - 1 \geq 0 \ , i = 1, \dots, n \quad (9)$$

Then, the primal form above is converted into its dual form using the Lagrange multiplier.

Suppose $\boldsymbol{\alpha} \in \boldsymbol{R^{n\times 1}}$ is a Lagrange multiplier, the quadratic programming problem above is changed to:

$$L = \frac{1}{2}||w||^2 - \sum_{i=1}^{n} \alpha_i [y_i(w_i x_i + b) - 1] \quad (10)$$

Karush-Kuhn-Tucker (KKT) conditions are met, as follows:

$$\frac{\partial L}{\partial w} = 0 \ \rightarrow w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0 \rightarrow w = \sum_{i=1}^{n} \alpha_i y_i x_i \quad (11)$$
$$\frac{\partial L}{\partial b} = 0 \ \rightarrow 0 - \sum_{i=1}^{n} \alpha_i y_i = 0 \rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0 \quad (12)$$

In addition, optimization can be done by maximizing $\boldsymbol{L}$ against $\boldsymbol{\alpha_i}$, by substituting equations (11) and (12) into equation (10) as follows:

$$L = \frac{1}{2}||w||^2 - \sum_{i=1}^{n} \alpha_i [y_i(w_i x_i + b) - 1]$$
$$L = \frac{1}{2}(w.w) - \sum_{i=1}^{n} \alpha_i [y_i(w_i x_i + b) - 1]$$
$$L = \frac{1}{2}\left(\sum_{i=1}^{n} \alpha_i y_i x_i \, . \sum_{j=1}^{n} \alpha_j \, y_j x_j\right) - \sum_{i=1}^{n} \alpha_i \left[y_i\left(\left(\sum_{j=1}^{n} \alpha_j y_j x_j\right) x_i + b\right) - 1\right]$$
$$L = \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i x_j - \sum_{i=1}^{n} \alpha_i y_i \left(\left(\sum_{j=1}^{n} \alpha_j y_j x_j\right) x_i + b\right) + \sum_{i=1}^{n} \alpha_i$$
$$L = \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i x_j - \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i x_j - b\sum_{i=1}^{n} \alpha_i y_i + \sum_{i=1}^{n} \alpha_i$$
$$L = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i x_j - b\sum_{i=1}^{n} \alpha_i y_i \quad (13)$$

Remember that $\frac{\partial L}{\partial b} = 0 \ \rightarrow 0 - \sum_{i=1}^{n} \alpha_i y_i = 0 \ \rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0$

So, it is obtained

$$L = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i x_j \quad (14)$$

Where $\boldsymbol{\alpha_i \geq 0}$, $\boldsymbol{i = 1, 2, \dots, n}$ and $\boldsymbol{\sum_{i=1}^{n} \alpha_i y_i = 0}$

The weight parameters are calculated using the equation:

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i \quad (15)$$

And the bias parameters are obtained through:

$y_i(w_i x_i + b) - 1 \geq 0$
$\Leftrightarrow y_i(w_i x_i + b) = 1$
$\Leftrightarrow w_i \,.x_i + b = y_i$
$\Leftrightarrow b = y_i - w_i \,.x_i$ (16)

So $\boldsymbol{b} = \frac{1}{N_{SV}} \sum_{I=1}^{N_{SV}} (\boldsymbol{y_i} - \boldsymbol{w_i} \,.\boldsymbol{x_i})$ .SV is a set of support vectors where $\boldsymbol{i} \in \boldsymbol{SV}$ when $\boldsymbol{\alpha_i} \neq \boldsymbol{0}$ and $\boldsymbol{N_{SV}}$ is the number of support vectors. Then, when there is imperfect separation of data classes due to the existence of data samples between $\boldsymbol{H_1}$ and $\boldsymbol{H_2}$ as shown in the following picture:

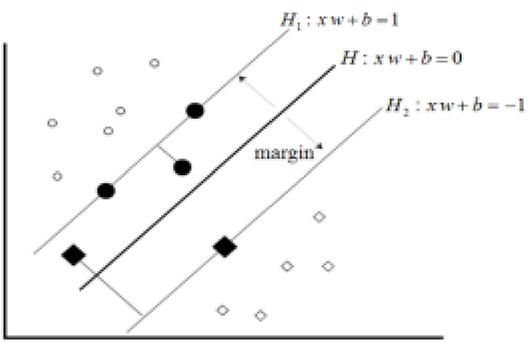


**Figure 2.** Cases of imperfect data separation

In this problem, the soft margin technique is introduced to handle this problem. In soft margin, the equation is modified by including the slack variable $\boldsymbol{\xi_i}$ with $\boldsymbol{\xi_i} > \boldsymbol{0}$ . The equation is as follows:

$$y_i(w_i x_i + b) \geq 1 - \xi_i \,, i = 1, \ldots, n. \quad (17)$$

Then, to the objective function, a positive parameter $\boldsymbol{C}$ is added, this parameter is useful for controlling the trade off between margin and error in classification. A large value of $\boldsymbol{C}$ means giving a large penalty to classification errors. The equation becomes:

$$\min_{w.b} \frac{1}{2} ||w||^2 + C \sum_{i=1}^{n} \xi_i \quad (18)$$

With the following obstacles:

$$y_i(w_i x_i + b) - 1 + \xi_i \geq 0 \,, i = 1, \ldots, n \quad (19)$$

Then, the primal form above is converted into dual form with the Lagrange multiplier $\boldsymbol{\alpha} \in \boldsymbol{R^{n \times 1}}$ , we get the following equation:

$$Max\ L = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i x_j \quad (20)$$

With constraints

$$0 \leq \alpha_i \leq C, \sum_{i=1}^{n} \alpha_i y_i = 0 \text{ dan } i = 1,2, \ldots, n. \quad (21)$$

*Non-Linier Support Vector Machine*

Non-Linear Support Vector Machine is applied to data that cannot be separated linearly, namely using the kernel method approach so that the sample data used can be separated linearly. The kernel method works by mapping input data into a higher dimensional feature space.

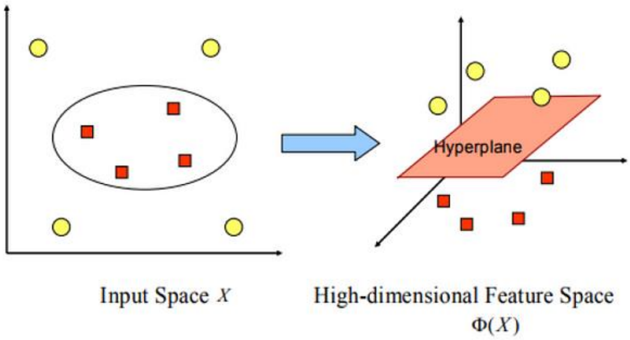


**Figure 3.** Data transformation in feature space

The image above illustrates data in 2nd dimension, where the data cannot be separated linearly by a *hyperplane*. So, data mapping is carried out into a higher dimensional space so that the two classes can be separated linearly by a *hyperplane*. For example, $\boldsymbol{u} = (\boldsymbol{u_1}, \boldsymbol{u_2})$ is the input data on $\boldsymbol{R^2}$ and $\boldsymbol{\phi}(\boldsymbol{u}) = (\boldsymbol{1}, \sqrt{\boldsymbol{2}}\boldsymbol{u_1}, \sqrt{\boldsymbol{2}}\boldsymbol{u_2}, \boldsymbol{u_1^2}, \boldsymbol{u_2^2}, \sqrt{\boldsymbol{u_1}}\boldsymbol{u_2})$ is input data in a higher dimensional space. So, the resulting input data mapping to higher dimensional space will separate linearly so that the optimal *hyperplane* can be found.
Suppose $\boldsymbol{x} \rightarrow \boldsymbol{\phi}(\boldsymbol{x})$ then the equation can be written as follows:

$$Max\ L = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \phi\,(x_i) \phi(x_j) \quad (22)$$

With constraints:

$$\alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0 \text{ and } i = 1,2, \ldots, n. \quad (23)$$

Weight and bias parameters can be calculated using the equation:

$$w = \sum_{i=1}^{N_{SV}} \alpha_i y_i \phi(x_i) \ \text{ and } b = \frac{1}{N_{SV}} \sum_{i=1}^{N_{SV}} (y_i - w^T \phi(x_i)) \quad (24)$$

And the optimal *hyperplane* is changed to:

$$f(x) = w\ \phi(x) + b = 0 \quad (25)$$

If the input data samples at the training stage are large, then calculating the dot product $\boldsymbol{\phi^T}(\boldsymbol{x_i})\boldsymbol{\phi}(\boldsymbol{x_j})$ will take a long time. So, it requires a calculation method without knowing the function $\boldsymbol{\phi}$. Suppose $K$ is a function nature.

$K(u, v) = \phi^T(u)\phi(v)$

Where $\boldsymbol{u}, \boldsymbol{v} \in \boldsymbol{R^n}$ and $\boldsymbol{\phi}: \boldsymbol{R^n} \rightarrow \boldsymbol{R^m}, \boldsymbol{n} < \boldsymbol{m}$ . This $K$ function is called the kernel function. Various types of kernel functions that are often used in Support Vector Machines (SVM) are Linear kernels, Radial Basis Function (RBF), and Polynomial (Sari et al., 2020). The selection of the correct kernel function and parameter tuning is very important because it can have a significant impact on the level of accuracy obtained in SVM analysis (Ginanjar, 2019). The frequently used kernel functions are as follows:

a) Linear Kernel

$$K(u, v) = u^T v$$

b) Polynomials Kernel

$$K(u, v) = (1 + u^T v)^{\,d}, d \geq 1$$

c) RBF (Radial Basis Function) Kernel

$$K(u, v) = \exp\left(-\gamma \left\| u - v \right\|^{2}\right), \gamma > 0$$

By using the kernel function, the equation becomes:

$$Max\ L = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_I, x_j) \quad (26)$$

With constraints

$$\boldsymbol{\alpha_i \geq 0}, \boldsymbol{\sum_{i=1}^{n} \alpha_i y_i = 0} \text{ and } \boldsymbol{i = 1, 2, \ldots, n}. \quad (27)$$

The bias parameters can be calculated using

$$b = \frac{1}{N_{sv}} \sum_{i=1}^{N_{sv}} \left( y_i - \sum_{i=1}^{N_{sv}} a_i\, y_i K(x_i, x) \right) \quad (28)$$

And the optimal *hyperplane*, namely

$$f(x) = \sum_{i=1}^{N_{sv}} \alpha_i y_i\, K(x_i, x) + b \quad (29)$$

## Result and Discussion

The data analyzed in this study comes from customers who used nail art services at the Blinkz Nails studio during the period 2022 to 2024. A total of 103 nail art designs were used in this analysis. The aim of this research is to categorize nail art designs into three classes: plain nail art (class=0), 3D nail art (class=1), and hand painting nail art (class=2).

Before starting further analysis using the *Support Vector Machine* (SVM) method, the first step that must be taken is data preparation, which includes feature extraction. In the context of Nail Art Design data, the feature extraction process is carried out using the GLCM (*Gray Level Co-occurrence Matrix*) method to produce texture features in the form of correlation, homogeneity, contrast and energy. GLCM is a technique used to analyze texture in images. In GLCM, there is a representation of the relationship between two adjacent pixels, taking into account the gray intensity, distance and angle between them. Various angles can be used in GLCM, but for Nail Art Design data, the relevant angles are 0°, 45°, 90°, and 135°.

After the feature extraction process is complete, we will get a series of attributes that will be used as analysis objects using the Support Vector Machine (SVM) method. These attributes are as follows:

**Table 1.** Attributes Description of the Nail Art Design Dataset

| Attributes | Description | Type |
|---|---|---|
| Contrast_0 | Shows the difference in intensity between adjacent pixels in an image at an angle of 0° | Numeric |
| Contrast_45 | Shows the difference in intensity between adjacent pixels in an image at an angle of 45° | Numeric |
| Contrast_90 | Shows the difference in intensity between adjacent pixels in an image at an angle of 90° | Numeric |
| Contrast_135 | Shows the difference in intensity between adjacent pixels in an image at an angle of 135° | Numeric |
| Correlation_0 | shows the correlation and linear relationship between pixels and the gray level in the image at an angle of 0° | Numeric |
| Correlation_45 | shows the correlation and linear relationship between pixels and the gray level in the image at an angle of 45° | Numeric |
| Correlation_90 | shows the correlation and linear relationship between pixels and the gray level in the image at an angle of 90° | Numeric |
| Correlation_135 | shows the correlation and linear relationship between pixels and the gray level in the image at an angle of 135° | Numeric |
| Energy_0 | Shows the uniformity of pixels in the image at an angle of 0° | Numeric |
| Energy_45 | Shows the uniformity of pixels in the image at an angle of 45° | Numeric |
| Energy_90 | Shows the uniformity of pixels in the image at an angle of 90° | Numeric |
| Energy_135 | Shows the uniformity of pixels in the image at an angle of 135° | Numeric |
| Homogeneity_0 | Shows the intensity of togetherness between pixels in the co-occurrence matrix at an angle of 0° | Numeric |
| Homogeneity_45 | Shows the intensity of togetherness between pixels in the co-occurrence matrix at an angle of 45° | Numeric |
| Homogeneity_90 | Shows the intensity of togetherness between pixels in the co-occurrence matrix at an angle of 90° | Numeric |
| Homogeneity_135 | Shows the intensity of togetherness between pixels in the co-occurrence matrix at an angle of 135° | Numeric |

*Descriptive Statistics*

In this study, 16 numerical features were used. The aim of this research is to group nail art designs based on the factors that influence them. The table below shows nail art design categories, including plain nail art, 3D nail art, and hand painting nail art, from a total of 103 data investigated.

**Table 2.** Total Nail Art Design

| Design Nail Art | Quantity |
|---|---|
| Plain Nail Art | 26 |
| 3D Nail Art | 51 |
| Hand Painting Nail Art | 26 |
| Total | 103 |

Table 2 explain that of the 103 data on nail art designs,26 of them are plain nail art, 51 of them are 3D nail art designs, while the remaining 26 are hand painting nail art designs. The next step involves dividing this dataset into training data and test data in a ratio of 60%:40%, 70%:30%, and 80%:20%.

The data that has been divided will be processed using the linear and non-linear *Support Vector Machine* (SVM) method using the Python programming language to determine the level of accuracy achieved.

*SVM Analysis using Linear Support Vector Machine*

The results of data processing and testing using Support Vector Machine Linear for three data sharing schemes are as listed in Table 3 below.

**Table 3.** Level of Accuracy Using the Linear SVM Method

| Division Scheme | Level of accuracy |
|---|---|
| 60/40 | 80% |
| 70/30 | 77.5% |
| 80/20 | 81.25% |

*SVM Analysis using Non-linear Support Vector Machine*

For non-linear SVM, two kernels will be used for comparison, namely the Polynomial kernel and the RBF (*Radial Basis Function*) kernel. The parameters that will be varied are the value *d* (degree) for the Polynomial kernel, and the value γ (gamma) for the RBF kernel. This aims to obtain an optimal level of accuracy. The results of data processing and testing using non-linear SVM for the three data division schemes can be found in Table 4, Table 5, and Table 6.

Based on Table 4 it can be seen that the test results with the Polynomial kernel obtained best results at parameter values d = 5 with an accuracy level of 77,5%, then using the RBF kernel, the accuracy level is obtained by 80% for $\gamma = 2$, $\gamma = 2{,}5$, $\gamma = 3$, and $\gamma = 3{,}5$.

Based on Table 5 it can be seen that the test results with the Polynomial kernel obtained best results at parameter values $d = 5$ with an accuracy level of 83.3%, then using the RBF kernel, the accuracy level is obtained by 90% for $\gamma = 2.5$.

**Table 4.** Level of Accuracy with Division Scheme 60/40

| Polynomial Kernel | | | | | |
|---|---|---|---|---|---|
| Parameter Kernel | d = 1 | d = 2 | d = 3 | d = 4 | d = 5 |
| Level of accuracy | 72.5% | 75% | 75% | 75 % | 77.5% |
| RBF (Radial Basis Function) Kernel | | | | | |
| Parameter Kernel | γ = 1.5 | γ = 2 | γ = 2.5 | γ = 3 | γ = 3.5 |
| Level of accuracy | 72.5% | 80% | 80 % | 80% | 80% |

**Table 5.** Level of Accuracy with Division Scheme 70/30

| Polynomial Kernel | | | | | |
|---|---|---|---|---|---|
| Parameter Kernel | d = 1 | d = 2 | d = 3 | d = 4 | d = 5 |
| Level of accuracy | 76.6% | 76.6% | 76.6% | 80% | 83.3% |
| RBF (Radial Basis Function) Kernel | | | | | |
| Parameter Kernel | γ = 1.5 | γ = 2 | γ = 2.5 | γ = 3 | γ = 3.5 |
| Level of accuracy | 83.3 % | 83.3 % | 90 % | 85% | 85 % |

**Table 6.** Level of Accuracy with Division Scheme 80/20

| Polynomial Kernel | | | | | |
|---|---|---|---|---|---|
| Parameter Kernel | d = 1 | d = 2 | d = 3 | d = 4 | d = 5 |
| Level of accuracy | 80% | 80% | 80% | 85% | 75% |
| RBF (Radial Basis Function) Kernel | | | | | |
| Parameter Kernel | γ = 1.5 | γ = 2 | γ = 2.5 | γ = 3 | γ = 3.5 |
| Level of accuracy | 90% | 95% | 95% | 95% | 95% |

Based on Table 6 it can be seen that the test results with the Polynomial kernel obtained best results at parameter values $d = 4$ with an accuracy level of 85%, then using the RBF kernel, the accuracy level is obtained by 95% for $\gamma = 2, \gamma = 2.5, \gamma = 3$, and $\gamma = 3.5$.

Based on the results obtained, the non-linear SVM method shows a higher level of accuracy than linear SVM. This can be seen from the three data sharing schemes where linear SVM only reaches the highest accuracy level of 81.25%, while non-linear SVM reaches the highest accuracy level of 95%. In particular, RBF kernels in non-linear SVMs tend to produce better accuracy rates than polynomial kernels. It can be seen from the test results on the three data division schemes, the RBF kernel gives the best results on the 60/40 and 70/30 division schemes with the highest accuracy levels of 80% and 90% respectively. Meanwhile, in the 80/20 data sharing scheme, the RBF kernel reaches an accuracy level of 95%.

## Conclusion

Research shows that SVM can be applied not only to general data, but also to specific data such as nail art designs. Using SVM, nail art design data can be accurately classified into relevant categories, demonstrating the flexibility and usefulness of this method in various contexts. After carrying out the training and testing process using the SVM method with various parameter variations, in the SVM analysis using linear SVM in the 60/40 division scheme the accuracy was 80%, in the 70/30 scheme the accuracy was 77.5%, and in the 80/20 scheme obtained an accuracy of 81.25%. Meanwhile, SVM analysis using Non-linear SVM in the 60/40 scheme obtained an accuracy of 77.5% using a polynomial kernel and an accuracy of 80% using the RBF kernel, then in the 70/30 scheme the accuracy was obtained of 83.3% using a polynomial kernel and an accuracy of 90% using an RBF kernel, and in the 80/20 scheme an accuracy of 85% was obtained using a polynomial kernel and an accuracy of 95% using an RBF kernel. So, the best accuracy rate of 95% is obtained in the 80/20 data scheme for the non-linear SVM model with the RBF kernel. The accuracy results obtained from this research also show that SVM provides good performance in classifying nail art designs.

**Acknowledgments**
The author would like to express deepest gratitude to all those who have helped, guided, provided support, and advice for the implementation of this work. Hopefully this work can be useful for everyone who reads it.

**Author Contributions**
Conceptualization: Y.R and J.P; data curation: Y.R and J.P. funding acquisition: Y.R and J.P methodology: Y.R and J.P visualization: Y.R and J.P writing – original draft: Y.R and J.P writing – review & editing: Y.R and J.P

**Funding**
This research was independently funded by Ahmad Dahlan University through the BPD DIY Syariah business capital assistance program.

**Conflicts of Interest**
No Conflicts of interest.

## References

Abbas, S. A. (2018). Entrepreneurship and information technology businesses in economic crisis. *Entrepreneurship and Sustainability Issues*, *5*(3). https://doi.org/10.9770/jesi.2018.5.3(20)

Agarwal, S. (2014). Data mining: Data mining concepts and techniques. In *Proceedings - 2013 International Conference on Machine Intelligence Research and Advancement, ICMIRA 2013*. https://doi.org/10.1109/ICMIRA.2013.45

Ardani, J. A., Utomo, C., & Rahmawati, Y. (2021). Model ownership and intellectual property rights for collaborative sustainability on building information modeling. *Buildings*, *11*(8). https://doi.org/10.3390/buildings11080346

Atik, C., Kut, R. A., Yilmaz, R., & Birant, D. (2023). Support Vector Machine Chains with a Novel Tournament Voting. *Electronics (Switzerland)*, *12*(11). https://doi.org/10.3390/electronics12112485

Avci, C., Budak, M., Yagmur, N., & Balcik, F. B. (2023). Comparison between random forest and support vector machine algorithms for LULC classification. *International Journal of Engineering and Geosciences*, *8*(1). https://doi.org/10.26833/ijeg.987605

Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, *2*(2). https://doi.org/10.1023/A:1009715923555

Chang, C. C., & Lin, C. J. (2011). LIBSVM: A Library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, *2*(3). https://doi.org/10.1145/1961189.1961199

Charleonnan, A., Fufaung, T., Niyomwong, T., Chokchueypattanakit, W., Suwannawach, S., & Ninchawee, N. (2017). Predictive analytics for chronic kidney disease using machine learning techniques. *2016 Management and Innovation Technology International Conference, MITiCON 2016*, MIT80–MIT83. https://doi.org/10.1109/MITICON.2016.8025242

Deng, N., Tian, Y., & Zhang, C. (2012). *Support vector machines: optimization based theory, algorithms, and extensions*. CRC press.

Divayana, D. G. H., Suyasa, P. W. A., & Widiartini, N. K. (2021). An innovative model as evaluation model for information technology-based learning at ICT vocational schools. *Heliyon*, *7*(2). https://doi.org/10.1016/j.heliyon.2021.e06347

Elshewey, A. M., Shams, M. Y., El-Rashidy, N., Elhady, A. M., Shohieb, S. M., & Tarek, Z. (2023). Bayesian Optimization with Support Vector Machine Model for Parkinson Disease Classification. *Sensors*, *23*(4). https://doi.org/10.3390/s23042085

Feta, N. R., & Ginanjar, A. R. (2019). Komparasi fungsi kernel metode support vector machine untuk pemodelan klasifikasi terhadap penyakit tanaman kedelai. *BRITech, Jurnal Ilmiah Ilmu Komputer, Sains dan Teknologi Terapan, 1*(1), 33-39. Retrieved from https://shorturl.at/P6uJ2

Gandhi, R. (2018). Support Vector Machine — Introduction to Machine Learning Algorithms. *Towards Data Science, 7*(06). Retrieved from https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

Jain, A., Ahirwar, M., & Pandey, R. (2019). A Review on Intutive Prediction Of Heart Disease Using Data Mining Techniques. *International Journal of Computer Sciences and Engineering*, *7*(7), 109–113. https://doi.org/10.26438/ijcse/v7i7.109113

Jana, D. K., Bhunia, P., Adhikary, S. Das, & Mishra, A. (2023). Analyzing of salient features and classification of wine type based on quality through various neural network and support vector machine classifiers. *Results in Control and Optimization*, *11*. https://doi.org/10.1016/j.rico.2023.100219

Jeong, S.-E., & Kim, J.-M. (2015). A study on check pattern of nail art. *Journal of the Korea Fashion and Costume Design Association*, *17*(1), 53–68. Retrieved from https://koreascience.kr/article/JAKO201509957115064.page

Kim, J., & Jeong, S. (2014). A Study on Expression Techniques of Nail Art: Focused on Nail Holic in 2012. *Fashion Business*, *18*(6), 100–115. https://doi.org/10.12940/jfb.2014.18.6.100

Krisnawati, M., Cahyono, A., Syarif, I., Naam, F., & Ariyanti, E. E. (2022). Nail Art : Sejarah, Bentuk, Warna dan Teknik Pembuatannya. *Prosiding Seminar Naisional Pasca Sarjana*, 641–645. Retrieved from https://proceeding.unnes.ac.id/snpasca/article/view/1543

Nayak, J., Naik, B., & Behera, H. S. (2015). A Comprehensive Survey on Support Vector Machine in Data Mining Tasks: Applications & Challenges. *International Journal of Database Theory and Application*, *8*(1), 169–186. https://doi.org/10.14257/ijdta.2015.8.1.18

Niu, C., Lv, M., Chen, K., & Wang, G. (2024). Optimization of Urban Mass Transit System Based on Support Vector Machine and Ant Colony Algorithm. *Computer-Aided Design and Applications*, *21*(S3). https://doi.org/10.14733/cadaps.2024.S3.242-257

Octaviani, P. A., Wilandari, Y., & Ispriyanti, D. (2014). Penerapan Metode Klasifikasi Support Vector Machine (SVM) pada Data Akreditasi Sekolah Dasar (SD) di Kabupaten Magelang. *Jurnal Gaussian,* *3*(4), 811-820. https://doi.org/10.14710/j.gauss.3.4.811-820

Puspitasari, A. M., Ratnawati, D. E., & Widodo, A. W. (2018). Klasifikasi penyakit gigi dan mulut menggunakan metode Support Vector Machine. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 2*(2), 802-810. Retrieved from https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/967

Ritonga, A. S., & Purwaningsih, E. S. (2018). Penerapan Metode Support Vector Machine (SVM) Dalam Klasifikasi Kualitas Pengelasan Smaw (Shield Metal Arc Welding). *Ilmiah Edutic*, *5*(1), 17–25. https://doi.org/10.21107/edutic.v5i1.4382

Rizal, R. A., Girsang, I. S., & Prasetiyo, S. A. (2019). Klasifikasi Wajah Menggunakan Support Vector Machine (SVM). *REMIK (Riset Dan E-Jurnal Manajemen Informatika Komputer)*, *3*(2), 1. https://doi.org/10.33395/remik.v3i2.10080

Saputra, R. A., Puspitasari, D., & Baidawi, T. (2022). Deteksi Kematangan Buah Melon Dengan Algoritma Support Vector Machine Berbasis Ekstraksi Fitur GLCM. *Jurnal Infortech*, *4*(2), 200–206. https://doi.org/10.31294/infortech.v4i2.14436

Sari, E. A., Saragih, M. T. B., Shariati, I. A., Sofyan, S., Al Baihaqi, R., & Nooraeni, R. (2020). Klasifikasi Kabupaten Tertinggal di Kawasan Timur Indonesia dengan Support Vector Machine. *JIKO (Jurnal Informatika Dan Komputer)*, *3*(3), 188–195. https://doi.org/10.33387/jiko.v3i3.2364

Somantri, O., Wiyono, S., & Dairoh, D. (2016). Metode K-Means untuk Optimasi Klasifikasi Tema Tugas Akhir Mahasiswa Menggunakan Support Vector Machine (SVM). *Scientific Journal of Informatics*, *3*(1), 34–45. https://doi.org/10.15294/sji.v3i1.5845

Wibawa, A. P., Guntur, M., Purnama, A., Akbar, M. F., & Dwiyanto, F. A. (2018). Metode-metode klasifikasi. In *Prosiding Seminar Ilmu Komputer dan Teknologi Informasi* (Vol. 3, No. 1). Retrieved from https://core.ac.uk/download/pdf/268075072.pdf

Yao, Y., Liu, Y., Yu, Y., Xu, H., Lv, W., Li, Z., & Chen, X. (2013). K-SVM: An effective SVM algorithm based on K-means clustering. *Journal of Computers (Finland)*, *8*(10), 2632–2639. https://doi.org/10.4304/jcp.8.10.2632-2639