RECOGNITION HAND WRITING WITH SMOOTH SUPPORT VECTOR MACHINE AND DIAGONAL BASED FEATURE EXTRACTION

Muhammad Fadli[1], Kania Evita Dewi[2]

[1,2] Program Studi Teknik Informatika – Universitas Komputer Indonesia
Jl. Dipatikukur 112-114 Bandung
Email: mfadli101295@gmail.com[1], kania.evita.dewi@email.unikom.ac.id[2]

ABSTRACT

This study aims to determine the accuracy of handwriting recognition using the Smooth Support Vector Machine (SSVM) classification method and the diagonal based featured extraction. The stages in this study are the handwritten character data collection consisting of the characters A-Z, a-z and numbers 0-9. In this research before carrying out the classification process, handwriting images will go through a preprocessing phase consisting of grayscale, threshold, segmentation, scaling and extraction of diagonal based feature extraction. Then the training and testing process is carried out using the smooth vector machine method. Handwritten character image samples were obtained from 30 correspondent and will be used as training data and test data. Based on the results of test conducted on test data it obtained the best accuracy of 72.6%.

Keywords: Artificial Intelligence, Smoooth Support Vector Machine, Diagonal Based on Feature Extraction, Feature Extraction, handwriting image recognition, image processing, SSVM.

1. INTRODUCTION

Handwritten characters are quite difficult to recognize by machine because everyone has different writing styles and someone's handwriting is also very vulnerable to the similarity between uppercase and lowercase characters, making it difficult to distinguish. Based on these differences many studies that discuss handwriting.

In previous studies the Support Vector Machine (SVM) method as a classification process and the Zoning method for feature extraction have been carried out and obtained an accuracy of 77.6% [1]. Support Vector Machine (SVM) in making predictions using high-dimensional data and large amounts of data becomes less efficient [2]. Therefore a smooth technique method was developed which replaces the SVM plus function with an integral function of the sigmoid neural network, hereinafter known as the Smooth Support Vector Machine (SSVM) [2]. When compared with SSVM, SVM has a longer running time and smaller accuracy than SSVM [2]. Smooth Support Vector Machine (SSVM) has been widely applied into various problems such as for the classification of the district / city human development index in Indonesia with a prediction accuracy of 84.77% [3], the classification of diabetes mellitus using the Smooth Support Vector Machine (SSVM) method produces an excellent accuracy of 97.11% [4], but from this research the Smooth Support Vector Machine (SSVM) method has not been applied to handwriting recognition.

This research applies to handwriting recognition using the Smooth Support Vector Machine (SSVM) algorithm and Diagonal Based Feature Extraction. Referring to previous studies that have used the diagonal feature extraction feature extraction method with the same discussion that is handwriting recognition using the diagonal feature extraction method and k-nearest neighbor obtained an accuracy of 90% [5] and diagonal feature extraction based handwritten character system using neural the network obtained a very high accuracy of 98% for 54 features and 99% for 69 features [6].

Based on this explanation, an accuracy test and calculation will be examined and calculation if this combination is applied to the handwriting problem in order to determine the accuracy of the Smooth Support Vector Machine (SSVM) with the Diagonal Based Feature Extraction feature extraction method.

2. THE CONTENT OF RESEARCH

2.2 System Analysis

The system to be built has several stages. These stages are the training stage, the testing phase and the classification stage using SSVM and the resulting accuracy. An overview of the system can be seen in Figure 3.1 below.

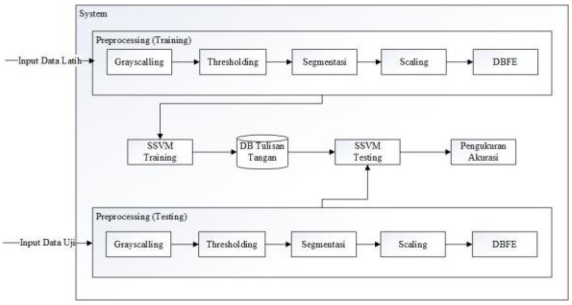


Figure 1. System Overview

2.3 Input Data Analysis

The input data needed to run the system is in the form of handwriting written on A4 HVS paper. To make it easier for researchers to visualize the values generated from each process, the researchers used data entered in the size of 180 pixels x 180 pixels that have been scanned and formatted (. JPG) as data to be processed for the next process. For training data in the form of handwriting composed of characters A-Z, a-z and 0-9 written on paper with a plain white background. Can be seen in Figure 2.

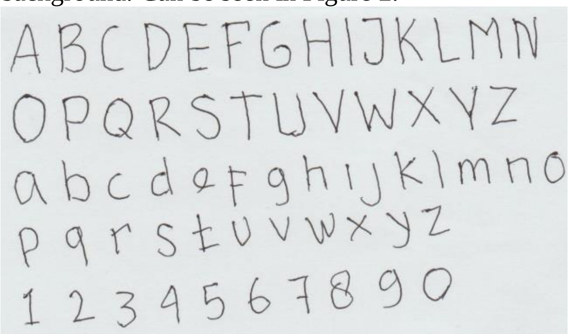


Figure 2. Examples of All Hand Character Images

In the preparation of this research, the input data used is a handwritten document that has been sized to 180 pixels x 180 pixels and has a format (JPG) consisting of only three characters, namely letters A, B, and C. The intention is to facilitate researchers in visualizing the values generated from each process.

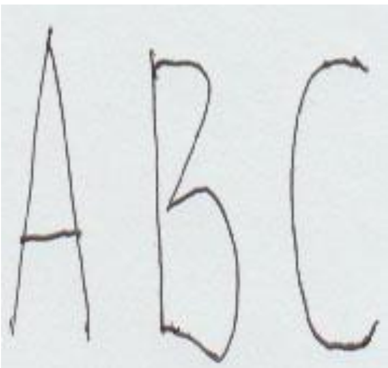


Figure 3. Image Input Data Size is 180x180 pixel

The handwritten image which is the input data which is used as the training data shown in Figure 3, has an RGB value which will be used to calculate the preprocessing stages.

2.4 Method Analysis

In this section that explains the method analysis that occurs in the implementation of Diagonal Based Feature Extraction (DBFE) and Smooth Support Vector Machine (SSVM).

2.4.1 *Preprocessing* Analysis

The stages of preprocessing consist of grayscaling, thresholding, segmentation, scaling and Diagonal Based Feature Extraction (DBFE).

a. Grayscaling

Changing the color format to grayscale serves to shrink the color range to 0 to 255. This process will be easier when you want to threshold the image into a black and white image. Figure 4 is a grayscale process flowchart.

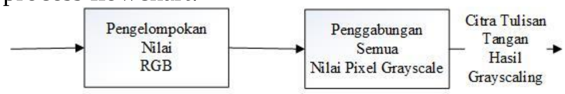


Figure 4. Grayscale Process

The steps are as follows:

1. Image colors are grouped based on red, green and blue values
2. Then use the grayscale formula in equation 1 namely: (0,2989*Red) + (0,5870*Green) + (0,1141 *Blue) (1).
   you will get the grayscale color value of the image.
3. Grayscale values obtained using the RGB value for each pixel.

Suppose the image is in pixels (0,0) has value *Red* = 229, *Green* = 238, *Blue* = 237, then based on equation (1) it becomes:

Gray = (0,2989 * R) + (0,5870 * G) + (0,1141 * B)
= (0,2989 * 229) + (0,5870 * 238) + (0,1141 * 237)
= 68,4481 + 139,706 + 27,0417
= 235,1958
= 235

From the above calculation, the pixel that was valuable *Red* = 229, *Green* = 238, *Blue* = 237 updated to grayscale value = 235. Figure 3.5 below is the result of the grayscale process.

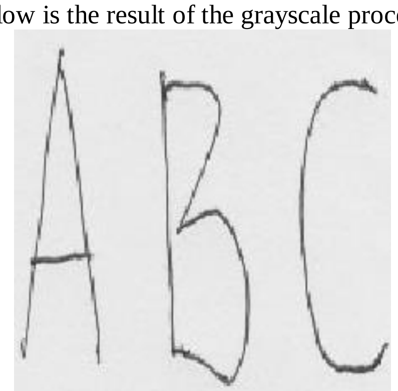


Figure 5. Image After Grayscale

Berikut adalah gambar matriks warna *Red* (R), *Green* (G), dan *Blue* (B) dari hasil proses *grayscale.*

Table 2. Matrix table of grayscale image values

Y\X	0	1	2	3	4	…	179
0	235	235	235	234	233	…	237
1	235	235	234	234	233	…	238
2	234	234	234	233	233	…	238
3	234	234	234	233	234	…	236
4	235	234	234	234	233	…	233
…	…	…	…	…	…	…	…
179	234	234	234	234	234	…	236

b. Thresholding

The threshold method used in this study uses the Sauvola Threshold method which aims to distinguish objects and backgrounds from the image so that it is more easily recognized at the feature extraction stage. Figure 6 is the Sauvola Threshold process.

Figure 6. The Sauvola Threshold Process

$$m(x,y) = \frac{\sum_{i=min}^{i=max} \sum_{j=min}^{j=max} img(i,j)}{i * j} \quad (2)$$

$$s(x,y) = \sqrt{\frac{\sum_{i=min}^{i=max} \sum_{j=min}^{j=max} (img(i,j) - m(x,y))^2}{(i * j) - 1}} \quad (3)$$

$$T(x,y) = m(x,y) * (1 + k * (\frac{s(x,y)}{R} - 1)) \quad (4)$$

In this study, the parameters used are as follows.

1. Number of neighbors = 21 pixels. A value of 21 is suitable because if the neighbor value is too large, the time required is longer, or if it is too small, the results obtained are not optimal..
2. Konstanta R = 128
3. K = 0,3
4. The value of m (x, y) is obtained from equation 2
5. The value of s (x, y) is obtained from the standard deviation equation 3.

From the handwritten image used, a threshold value in pixels (0.0) with a grayscale value of 235 will be sought. Next, look for any neighboring pixels that correspond to a predetermined number of neighbors of 21. Then from the search results obtained image matrix like table 3., where neighboring pixels (0,0) are from pixel (0,0) to pixel (10,10).

Table 3. Examples of image matrices that will be thresholded

(i,j)	0	1	2	3	4	5	6	7	8	9	10	$\sum Img(i,j)$
0	235	235	235	234	233	233	233	232	235	234	233	2572
1	235	235	234	234	233	233	232	232	234	234	234	2570
2	234	234	234	233	233	232	232	232	233	234	234	2565
3	234	234	234	233	234	232	232	232	233	233	234	2566
4	235	234	234	234	233	232	232	232	233	233	234	2566
5	235	235	235	233	232	234	233	232	233	234	234	2570
6	236	236	235	235	234	234	233	233	234	234	234	2578
7	236	236	236	235	235	234	234	234	235	234	233	2582
8	233	234	235	236	236	235	234	233	233	233	232	2576
9	234	234	235	236	236	235	234	234	233	233	232	2578
10	235	235	235	236	236	235	235	235	233	233	232	2582
$\sum_{i=min}^{i=max} \sum_{j=min}^{j=max} img(i,j)$												28307

The first step is looking for the average value of m (0.0) with equation 2, so the results are as follows.

Information:

i min = smallest value of pixel i
i max = greatest value of pixels i
j min = smallest value of pixel i j
j max = greatest value of pixels j

$$m(x,y) = \frac{\sum_{i=min}^{i=max} \sum_{j=min}^{j=max} img(i,j)}{i * j}$$

$$m(0,0) = \frac{28307}{11 * 11} = 233{,}942$$

According to the rounding rules, if the number behind the comma is more than 5, it will be rounded up from 233,942 to 234 (the value used). Then to find the standard deviation value dipixel x = 0, y = 0 s (0.0) used equation 3 so the results are as follows.

Table 4. Calculation of the average number of pixel images

No	Img (i,j)	m(x,y)	$(img(i,j) - m(x,y))^2$
1	235	234	1
2	235	234	1
3	235	234	1
4	234	234	0
5	233	234	1
6	233	234	1
7	233	234	1
8	232	234	4
9	235	234	1
10	234	234	0
11	233	234	1
12	235	234	1
13	235	234	1
14	234	234	0
15	234	234	0
16	233	234	1
17	233	234	1
18	232	234	4
…	…	…	…
121	232	234	4
$\sqrt{\sum \frac{((img(i,j) - m(x,y))^2}{(i * j) - 1}}$			166

$$s(x,y) = \sqrt{\frac{\sum_{i=min}^{i=max}\sum_{j=min}^{j=max}(img(i,j) - m(x,y))^2}{(i * j) - 1}}$$

$$s(0,0) = \sqrt{\frac{166}{(11 * 11) - 1}}$$

$$= 1{,}176$$
$$= 1$$

The next step is enter the values of m (0,0) and s (0,0) into equation 2.2 to get the threshold value T (0,0). So the results will be: $T(x,y) = m(x,y) * (1 + k * (\frac{s(x,y)}{R} - 1))$

$$T(0,0) = m(0,0) * (1 + 0{,}3 * (\frac{s(0,0)}{128} - 1))$$
$$= 234 * (1 + 0{,}3 * (\frac{1}{128} - 1))$$
$$= 164{,}34$$

Because the two numbers behind the comma are odd and less than 5, they are rounded down to 164. From the above calculation, a threshold value of 164 is obtained. The next step is directly entered in equation 2.5 so that you will get a new pixel value. From the example above, the original pixel with a value of 235 will change to 1 because 235 is greater than 164. Here is a table of calculations using the sauvola threshold.

Table 5. The sauvola threshold calculation results

(x,y)	img	m(x,y)	s(x,y)	T(x,y)	*f(x,y)*
(0,0)	235	234	1	164	1
(1,1)	237	236	2	170	1
(1,2)	238	237	2	172	1
…	…	…	…	…	…
(12,0)	114	143	2	102	0
…	…	…	…	…	…
(179,179)	235	234	1	164	1

Here are the images that have been processed using the Sauvola threshold method.

Figure 6. Sauvola Threshold results

c. Segmentation

In this process, the input used is the black and white image of the sauvola threshold. Next will be cut to get the image of handwritten letters. Here is a block diagram of the segmentation process in Figure 7.

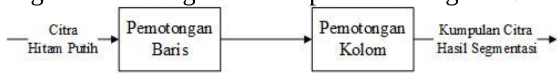
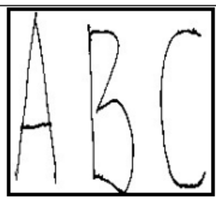
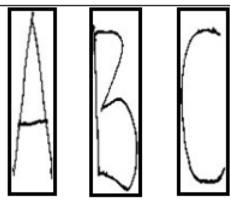


Figure 7. Segmentation process

Cutting is done for each row (horizontal) in the input image first, then cut each column (vertical) in each image from the cut.

Cutting for each line (horizontal) is done by tracing the image pixel from the 0th line pixel. Search continues to find the pixel object, then marked as the initial label cutting. Next do a search again until in one pixel line of the image no object pixel is found, then mark it as the final cutting label. This start label and end label are used as a reference to cut the image of each line (horizontal). Do the same for the next line cut. For cutting each column (vertical) the same as cutting rows, it's just that the image search from the 0th column pixel. Here is a table of handwritten image segmentation results.

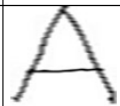
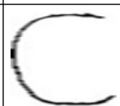
Table 6. The results of the segmentation process

No	Pemotongan Baris	Pemotongan Kolom
1		

d. Scalling

Scaling aims to equalize the image size of segmentation results so that it can be processed by extracting the Diagonal Based feature Extraction feature. In this research, an image scaling size of 100 x 100 pixels is used.

Table 7. Scaling Imagery

Image Ukuran Hasil Segmentasi	Image Hasil Scaling 100 x 100 pixel
	
	
	

e. Diagonal Based Feature Extraction

Diagonal Based Feature Extraction is a feature extraction algorithm that divides the pixel size of images into smaller and even pixels. Character image of the results of the scaling stage measuring 100x100 pixels divided into 25 zones that are evenly distributed. Each zone measures 25x25 pixels. Characteristics are extracted from each zone by moving diagonally from 25x25 pixels each. Each zone has 49 diagonal lines and the foreground pixels in each diagonal line are added to get one sub-feature. These 49 sub-traits will be leveled to get a single trait value and be placed in the appropriate zone. This procedure is repeated for all zones. There will be several zones whose diagonals are blank from foreground pixels. The characteristic value for the zone is zero. Following is the flow of the Diagonal Based Feature Extraction algorithm as shown in Figure 8.

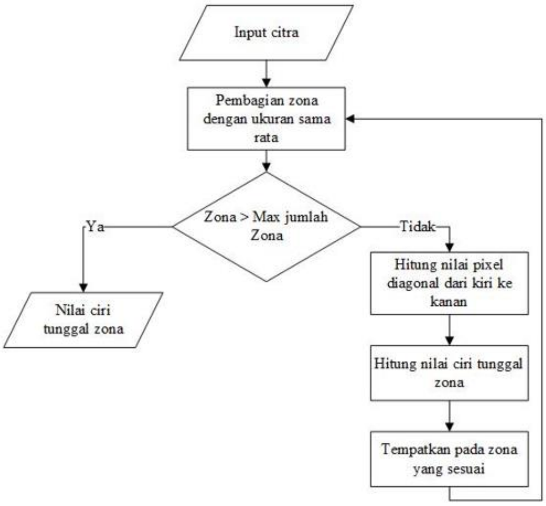


Figure 8. Diagonal Based Feature Extraction

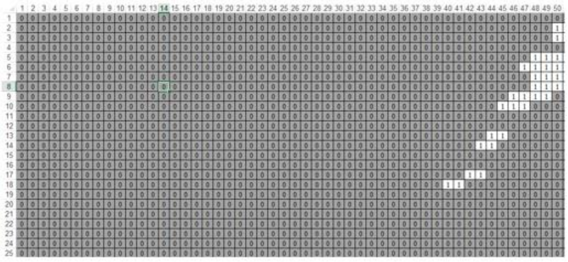
Image image used for character extraction in the form of characters / classes from handwriting recognition in the form of letters A, B and C.

1. Class A

Figure 9. Image image size of 100x100 pixels

Because the image size is too large, it cannot display the pixel value of the image Image size of 100x100 pixels. A white column indicates an image pixel of value (1) and a gray column of value (0).

Table 8. Binary pixel image value of the letter A image



2. Division of Class A Zones

Each character will be divided into small zones. Each character measuring 100 x 100 pixels is divided into 16 small zones with the same size of each zone that is 25 x 25 pixels.

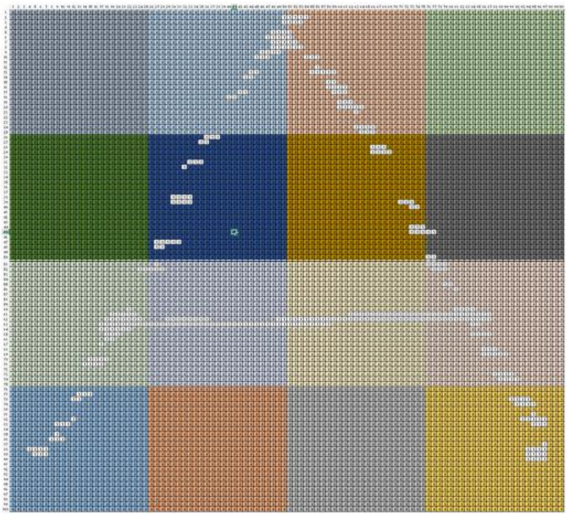


Table 9. Zoning division

Information:

Table 10. The name of the division zone

ZONA 1	ZONA 2	ZONA 3	ZONA 4
ZONA 5	ZONA 6	ZONA 7	ZONA 8
ZONA 9	ZONA 10	ZONA 11	ZONA 12
ZONA 13	ZONA 14	ZONA 15	ZONA 16

3. Extraction Diagonl For Each Zone

a. Diagonal extraction zone 1

Add the pixel values on each diagonal moving from left to right. Diagonal number in this zone.

a. **Table 11.** Diagonal extraction zone 1

JUMLAH PIXEL DIAGONAL	ZONA 1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
JUMLAH PIXEL DIAGONAL		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TOTAL JUMLAH PIXEL DIAGONAL	0																									

The total number of pixels of 49 diagonals is 0. To get a single value for zone 1, the total number of diagonal pixels divided by many diagonals is 49, a value of zone 1 is 0 and is placed in the appropriate zone. can be seen in table 12.

Table 12. Single characteristic value for each zone

0	ZONA 2	ZONA 3	ZONA 4
ZONA 5	ZONA 6	ZONA 7	ZONA 8
ZONA 9	ZONA 10	ZONA 11	ZONA 12
ZONA 13	ZONA 14	ZONA 15	ZONA 16

Diagonal extraction for zone 2, zone 3 and so on up to zone 16 steps are the same as calculating the value in zone 1. So that the single characteristic value of each zone can be seen in table 13.

Table 13. Single grade zone A trait value

0	0,6122448979	0,9795918367	0
0	0,4897959183	0,3673469387	0,0816326530
0,8979591836	0,7959183673	0,9591836734	1,2244897959
0,4081632653	0	0	0,6122448979

To calculate the value of a single feature for class B zone is the same as the steps to calculate the value of a single feature in class A. So the value of a single feature for zone B can be seen in table 14.

Table 14. Single-grade zone value of class B

2,4285714286	1,1836734694	1,5510204082	0
2,0816326531	1,8775510204	1,9387755102	0
2,8775510204	0,8163265306	0,4081632653	2,4693877551
1,8163265306	1,2653061224	1,2857142857	2,3673469388

The values owned by each A, B and so on are vector values. This vector value will be used as input to enter the processing stage using the Smooth Support Vector Machine algorithm.

2.4.2 SSVM Method

SVM is a classification method that divides vector space into 2 parts, namely positive classes and negative classes by hyperplants. Based on that, SVM multiclass is used [16]. The following stages of the linear SSVM method using the Newton Armijo algorithm can be seen in Figure 10.

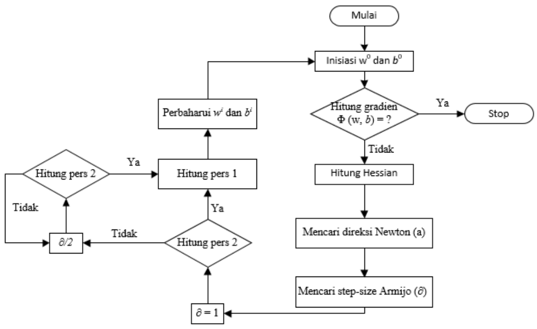


Figure 10. The stages of the SSVM method use the Newton Armijo algorithm

Information :

w^b : The size of the input
b^0 : Number
D : Diagonal matrix measuring (m x m) with values [1, -1]
A : Sized matrix (m x n)
m : Amount of data
n : the features
w : normal vector sized (n x 1)
e : sized vector (m x 1)
y : The parameter determines the location of the separator field with respect to the origin
v : Positive parameters that balance the weight of the training error and the margin maximation term.

The stages of the Newton Armijo algorithm are as follows:

1. Inisialisasi Data

A=

$$\begin{bmatrix} 0 & 0,6122448979 & 0,9795918367 & 0 & \ldots & 0,6122448979 \\ 2,4285714286 & 1,1836734694 & 1,5510204082 & 0 & \ldots & 2,3673469388 \end{bmatrix}$$

(A adalah matrik berukuran (2 x 16)

D = $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ *(diagonal matrix of size (2 x 2))*

e = $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ *(e is a vector of lots of data (2 x 1))*

2. Calculate the Hessian Matrix Gradient

Initial point for calculating hessian matrix gradients using the following formulation:

Known: $w = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \ldots \\ 0 \end{bmatrix}$ and $y = 0$

(w matrix size (16 x 1) because of the large amount of data i.e. 16))

$$\lim_{\alpha \to \infty} \nabla \Psi_\alpha (w, y) = \begin{bmatrix} w - vA^T D(e - D(Aw - ey)) \\ y + ve^T D(e - D(Aw - ey)) \end{bmatrix}$$

Using the initial point is calculated $\boldsymbol{e} - \boldsymbol{D}(\boldsymbol{Aw} - \boldsymbol{ey})$ after that calculate the value $\boldsymbol{w} - \boldsymbol{v}A^T\boldsymbol{D}\big(\,\boldsymbol{e} - \boldsymbol{D}(\boldsymbol{Aw} - \boldsymbol{ey})\big)$ then calculate the value $\boldsymbol{y} + \boldsymbol{v}e^T\boldsymbol{D}\big(\,\boldsymbol{e} - \boldsymbol{D}(\boldsymbol{Aw} - \boldsymbol{ey})\big)$ so the result of

$$\lim_{a\to\infty} \nabla\Psi_\alpha\,(w,y)^{\square} = \begin{bmatrix} 4,8571428572 \\ 1,1428571430 \\ 1,1428571430 \\ 0,0000000000 \\ 4,1632653062 \\ 2,7755102042 \\ 3,1428571430 \\ -0,1632653060 \\ 3,9591836736 \\ 0,0408163266 \\ -1,1020408162 \\ 2,4897959184 \\ 2,8163265306 \\ 2,5270122448 \\ 2,5714285714 \\ 3,5102040818 \\ 0 \end{bmatrix}$$

3. Checking Gradients > 0

$$\left\|\lim_{a\to\infty} \nabla\Psi_\alpha\,(w,y)^{\square}\right\|_2^2 =$$

$$\begin{bmatrix} 4,8571428572 & 1,1428571430 & 1,1428571430 & \dots & 0 \end{bmatrix} \begin{bmatrix} 4,8571428572 \\ 1,1428571430 \\ 1,1428571430 \\ 0,0000000000 \\ 4,1632653062 \\ 2,7755102042 \\ 3,1428571430 \\ -0,1632653060 \\ 3,9591836736 \\ 0,0408163266 \\ -1,1020408162 \\ 2,4897959184 \\ 2,8163265306 \\ 2,5270122448 \\ 2,5714285714 \\ 3,5102040818 \\ 0 \end{bmatrix}$$

= 117,4861657344

It turns out that the gradient> 0 (117,4861657344> 0) so that the process continues at the Newton Direction stage.

4. Calculate Gradients from the Hessian Matrix

At the Newton Direction stage, the Hessian matrix is calculated using the following formulation:

Known : $\quad \boldsymbol{e} - \boldsymbol{D}(\boldsymbol{Aw} - \boldsymbol{ey}) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

Formula : $S_\infty(\boldsymbol{x}) = \lim\limits_{a\to\infty}\left(\frac{1}{1+\varepsilon^{-ax}}\right) = \frac{1+sign(x)}{2}$

$$S_\infty\left(\,\boldsymbol{e} - \boldsymbol{D}(\boldsymbol{Aw} - \boldsymbol{ey})\right) = \begin{bmatrix} \frac{1+sign(1)}{2} \\ \frac{1+sign(1)}{2} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$diag\ (S_\infty\big(\,\boldsymbol{e} - \boldsymbol{D}(\boldsymbol{Aw} - \boldsymbol{ey})\big)) = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}$$

Calculate the hessian matrix using the formula

$H_{11} = A^T\ diag\ (S_\infty\big(\,\boldsymbol{e} - \boldsymbol{D}(\boldsymbol{Aw} - \boldsymbol{ey})\big))$ **A,**

$H_{12} = -A^T\ diag\ (S_\infty\big(\,\boldsymbol{e} - \boldsymbol{D}(\boldsymbol{Aw} - \boldsymbol{ey})\big))$ **e,**

$H_{21} = -e^T\ diag\ (S_\infty\big(\,\boldsymbol{e} - \boldsymbol{D}(\boldsymbol{Aw} - \boldsymbol{ey})\big))$ **A,**

$H_{22} = e^T\ diag\ (S_\infty\big(\,\boldsymbol{e} - \boldsymbol{D}(\boldsymbol{Aw} - \boldsymbol{ey})\big))$ **e.** After grades H_{11}, H_{12}, H_{21}, H_{22} then calculate into the formula below.

$$\lim_{a\to\infty} \nabla^2\Psi_\alpha\,(w,y) = \begin{bmatrix} \frac{\partial^2\Psi_\alpha(w,\gamma)}{\partial^2 w^2} & \frac{\partial^2\Psi_\alpha(w,\gamma)}{\partial w\partial\gamma} \\ \frac{\partial^2\Psi_\alpha(w,\gamma)}{\partial^2\partial w} & \frac{\partial^2\Psi_\alpha(w,\gamma)}{\partial^2\gamma^2} \end{bmatrix}$$

$$= \mathrm{I} + \upsilon \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix}$$ maka hessian yang terbentuk adalah sebagai berikut :

$$= \begin{bmatrix}
12,795918368 & 5,7492711372 & & 11,4985422744 & 4,8571428572 \\
5,7492711372 & 4,5518533944 & & 6,3540191586 & 3,5918367346 \\
7,5335276970 & 4,8713036234 & \cdots & 8,5431070388 & 5,0612244898 \\
0 & 0 & \cdots & 0 & 0 \\
10,110787172 & 4,927946689 & \cdots & 9,8558933780 & 4,1632653062 \\
9,1195335278 & 5,0445647646 & \cdots & 9,4893794252 & 4,7346938774 \\
9,4169096210 & 5,0395668470 & \cdots & 9,6293211162 & 4,6122448978 \\
0 & 0,0999583506 & \cdots & 0,0999583506 & 0,1632653060 \\
13,976676385 & 7,9117034568 & \cdots & 14,7238650562 & 7,5510204080 \\
3,9650145772 & 2,9071220324 & \cdots & 4,8396501456 & 3,2244897958 \\
1,9825072886 & 2,1407746770 & \cdots & 3,1070387336 & 1,3673469387 \\
11,994169096 & 7,3452728028 & \cdots & 13,1911703456 & 7,3877551020 \\
8,8221574344 & 4,7996668054 & \cdots & 9,0995418576 & 4,4489795918 \\
6,1370297374 & 2,9911573510 & \cdots & 5,9823147020 & 2,5270122448 \\
6,2448979592 & 3,0437317784 & \cdots & 6,0874635568 & 2,5714285714 \\
11,498542274 & 6,3540191586 & & 12,9583506874 & 5,9591836734 \\
4,8571428572 & 3,5918367346 & & 5,9591836734 & 5
\end{bmatrix}$$

(matriks berukuran (17 x 17))

5. Determine Newton Direction (d^i)

For example $\mathrm{H} = \boldsymbol{\nabla^2}\Psi_\alpha\,(\boldsymbol{w^i}, \boldsymbol{y^i})$ and $\mathrm{G} = \boldsymbol{\nabla}\ \ \Psi_\alpha\,(\boldsymbol{w^i}, \boldsymbol{y^i})$ then

d^i=$\boldsymbol{H^{-1}}(\boldsymbol{-G})$=

3. CLOSURE

3.1 Conclusion

Based on the test results of the handwriting recognition system using the Smootth Support Vector Machine method and the Diagonal Based Feature Extraction, the best accuracy is obtained at 72.6%. This accuracy is influenced by training parameters, training data and test data used.

3.2 Suggestion

So that further research on handwriting recognition using the Smootth Support Vector Machine and Diagonal Based Feature Extraction method has higher accuracy, the following are suggestions that can be taken into consideration, namely:

1. Other handwriting image segmentation methods are needed.
2. Another better feature extraction method is needed for images such as template matching or the latest one.
3. Smootth Support Vector Machine method is needed using Newton Armijo method with a different kernel.

BILBIOGRAPHY

[1] U. Rohwana, M. Isa, J. Matematika, F. Matematika, and P. Alam, “Pengenalan Tulisan Tangan Huruf Latin Bersambung Secara Real Time Menggunakan Algoritma Learning Vector Quantization,” J. Sains Dan Seni Pomits, vol. 2, no. 1, pp. 1–6, 2013.

[2] L. Herman, Syafie and D. Indra, “Pengenalan Angka Tulisan Tangan Mengunakan Jaringan Syaraf Tiruan,” Ilk. J. Ilm., vol. 10, no. 2, pp. 201–206, 2018.

[3] F. P. Putri and A. Kusnadi, “Pengenalan Tulisan Tangan Offline Dengan Algoritma

Generalized Hough Transform dan Backpropagation," Ultim. Comput., vol. 10, no. 1, pp. 5–12, 2018.
[4] E. Suryanto and S. W. Purnami, "Perbandingan Reduced Support Vector Machine dan Smooth Support Vector Machine untuk Klasifikasi Large Data," J. Sains dan Seni ITS, vol. 4, no. 1, pp. D25–D30, 2015.
[5] R. A. Nugroho, Tarno, and A. Prahutama, "Klasifikasi Pasien Diabetes Mellitus Menggunakan Metode Smooth Support Vector Machine (Ssvm)," Gaussian, vol. 6, pp. 439–448, 2017.
[6] T. W. Fauzi, Fatkhurokhman, Darsyah, Moh.Yamin, Utami, "Smooth Support Vector Machine (Ssvm) Untuk Pengklasifikasian Indeks Pembangunan Manusia Kabupaten / Kota Se-Indonesia," Statiska, vol. 5, no. 2, pp. 14-23, 2017.
[16] N. I. Widiastuti, E. Rainarli, and K. E. Dewi, "Peringkasan dan Support Vector Machine pada Klasifikasi Dokumen," *J. Infotel*, vol. 9, no. 4, p. 416-421, 2017.