IMPLEMENTATION OF MOTION CHAIN CODE AND PRINCIPAL COMPONENT ANALYSIS FOR INDONESIAN SIGN LANGUAGE RECOGNITION

Widyan Fakhruzzaman[1], Nelly Indriani Widiastuti[2]

Teknik Informatika – Universitas Komputer Indonesia
Jl. Dipatikukur 112-114 Bandung
Email: fakhruzzaman.w@gmail.com[1], nelly.indriani@email.unikom.ac.id[2]

ABSTRACT

Sign language is a language that doesn't using voice but uses hand gesture and facial expression that needed by the deaf for comunication and information. Communication limitations with normal people become a social problem for the deaf due to differences in communication methods. Based on this, a system is needed to recognize and translate sign language into text or voice so relationship between them becomes better. One of process that must be done in sign language recognition is the feature extraction. In this research, Motion Chain Code (MCC) algorithm will be used for dynamic sign language and Principal Component Analysis (PCA) algorithm for static sign language. MCC is an algorithm that can representation of hand trajectory while PCA is a method that takes important features from high-dimensional data by reducing the data to low-dimensional. This research will recognize 26 letters of sign language movement with video input using the Multiclass Support Machine (SVM) for classification. Based on the research and testing that has been done, the highest accuracy obtained for recognize static and dynamic sign language is 69,23% using training data as much as 1040 video and tested using 520 videos, each alphabet has 40 video training and 20 video tesing. In testing of static sign language obtained accuracy 67,25% and for dynamic sign language obtained an accuracy of 90%.

Keyword: Sign Language, Motion Chain Code, Principal Component Anlaysis, Multiclass, Support Vector Machine

1. INTRODUCTION

Sign language is a language that doesn't using voice but uses hand gesture and facial expression that needed by deaf people for comunication needs and information needs. Communication limitations with normal people become a social problem for deaf people due to differences in communication methods. Based on this, a system is needed that can recognize and translate sign language into text or voice so that relationship between them becomes better.

In the realm of Indonesian sign language recognition with input images or video, it has been tried in several research. One of research that has been done by Wildan [1], which used the Viola and Jones feature extraction method to recognize Indonesian sign language alphabet. From the results of the design and testing, the accuracy is 55%. Another research that has been done by Novrizal [2], which used the Principal Component Analysis (PCA) method and used the Naïve Bayesian classification method, obtained accuracy of 69%. Based on previous research, it is needed an algorithm that can improve the accuracy of the recognition of sign language well.

Sign language recognition system have an important process to do. One of process that must be done is the feature extraction. Future extraction purposed to obtain information contained in an image that are used as references to distinguish one image with the other image. In research conducted by Atik and friends [3], the use of Principal Component Analysis (PCA) method as a feature of Indonesian sign language recognition has an accuracy of 80,42%. Another research that has been done by Arshad Malik and friends [3], which used the PCA method for recognition of Pakistan Sign Language (PSL) with an accuracy rate of 85%. Another research by Mahmoud Zaki and friends [4], which used the Motion Chain Code (MCC) algorithm for representation hand movement patterns in dynamic sign language in Arabic has an accuracy of 88%. From these researchs, PCA and MCC seem to provide good results for sign language recognition.

Based on the problems and solutions that have been explained, this research will be combine PCA method as a feature for static sign language and MCC algorithm as a feature for dynamic sign language for the recognition of Indonesian Sign Language which refers to Sistem Isyarat Bahasa Indonesia (SIBI).

2. THE CONTENT OF RESEARCH

2.1. Problem Analysis

As for the ongoing communication flow beetwen deaf people and normal people in Figure 1.

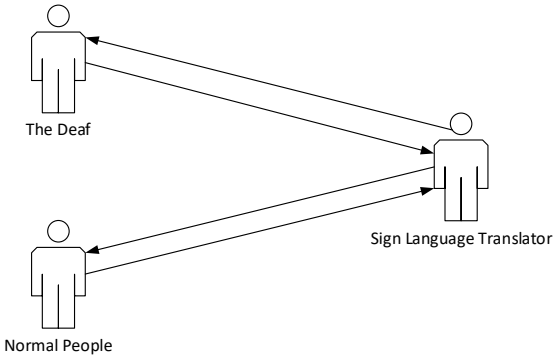


Figure 1. Comunication Flow

2.2. Analysis The System

The system to be built has main process, included preprocessing, MCC, PCA, SVM training and SVM testing. The description of the system can be seen in Figure 2.

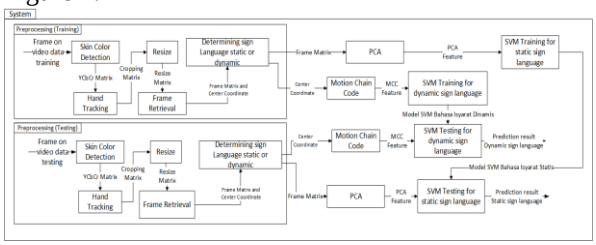


Figure 2. System Overview

Based on Figure 2, input data used for training and testing is a video with extention .mp4 that has a 320x240 resolution and duration of video is about 2 to 4 seconds. In the training process the first one is preprocessing stage. In preprocessing several processes will be carried out, namely skin color detection, hand tracking, resize, frame retrieval, and determination of static or dynamic sign language. If the result of determining sign language is a static sign language, a matrix of frame will be taken to continue the PCA feature extraction process and the features from PCA will be used in SVM training for static sign language. Then if the result of determining sign language is dynamic sign language, then the center coordinate of each frame will be taken. The center coordinate will be used for the extraction of hand movement patterns using the MCC algorithm. The results of the MCC will be used in SVM training for dynamic sign languages.

In the testing process will be carried out the same process as training, namely preprocessing, extraction of PCA features if the result of determining sign language is static and tested using SVM Multiclass using a static sign language prediction model. Then if the result of determining sign language is dynamic, then the MCC process will be carried out and tested using Multiclass SVM using a dynamic sign language prediction model.

2.3. Preprocessing

Preprocessing purposed to process input data before preprocessing it to the next stage.

a. Skin Color Detection

Skin color detection is a process that separates region of object in image based on differences in skin color. In Figure 3 is an overview the process of skin color detection.

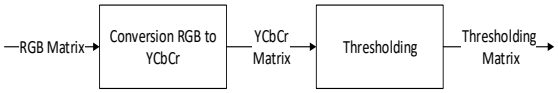


Figure 3. Skin Color Detection

The process of converting from RGB to YCbCr using equation 1 [5].

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 0{,}299 & 0{,}587 & 0{,}144 \\ -0{,}169 & -0{,}331 & 0{,}500 \\ 0{,}500 & -0{,}419 & -0{,}081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1)$$

From the conversion of RGB to YCbCr, the thresholding process for the YCbCr matrix is done using equation 2.

$$\begin{cases} Y > 80 \\ 85 < Cb < 135 \\ 135 < Cr < 180 \end{cases} \quad (2)$$

The thresholding process is done so that the matrix only has two values, the value is 0 and 255. From the thresholding equation, to get a value of 255 all conditions must be met [5].

b. Hand Tracking

This Hand Tracking aims to search the hand area with the largest area and the area is designated as the Region of Interest (ROI). Then the width and length of the ROI will form a square (bounding area) following the largest contour. The flow of the hand tracking process can be seen in Figure 4.

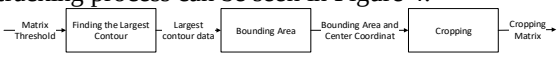


Figure 4. Hand Tracking Process

The input data for hand tracking is the matrix thresholding of YCbCr. In the application of hand tracking, the Emgu CV library is used. Bounding of the results from the hand tracking process can be seen in Figure 5.

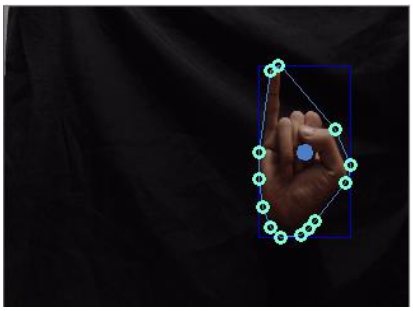


Figure 5. Bounding Area

After the bounding area is obtained, the next step is cropping process using the ROI function in the Emgu CV library.

c. Resize

Resize is the process of changing the resolution or horizontal and vertical size of an image, it aims to equalize the size of the sign language that becomes the same size for training and testing. The method used for the resize process is the linear interpolation method. Linear interpolation is a method used to determine the value of something that is in an interval or in two points that are in line. The workings of the

linear interpolation method is to take four sides of the pixel values of the image to be resized. Then look for the middle pixel value between the two sides of the pixel. To find the pixel value, equation 3 will be used.

$$i = A + \frac{w}{W}(B - A)$$
$$j = C + \frac{w}{W}(D - C) \quad (3)$$
$$y = i + \frac{h}{H}(j - i)$$

Where:

i, j, y= pixel value to be searched

A, B, C, D= the value of the intensity obtained from the previous image

w= width of the desired image size

h= height of the desired image size

W= the width of the distance between the color values to be searched with the previous color (for example the distance A with i)

H= the height of the distance between the color values to be searched with the previous color (for example the distance Y with i)

d. Frame Retrieval

Frame retrieval purposed to take a matrix of frames that only contain sign language movements. The flow of the frame retrieval process can be seen in Figure 6.

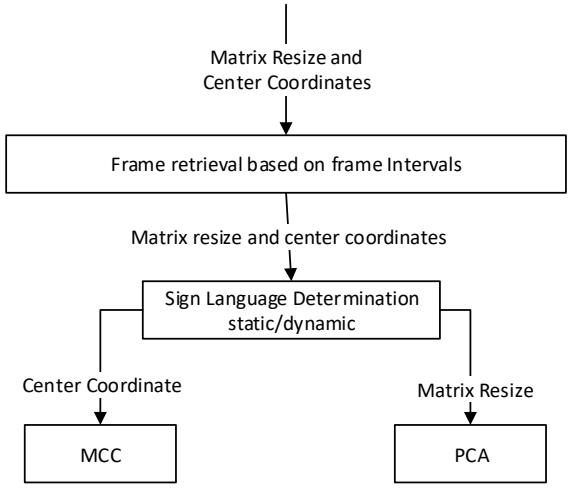


Figure 6. Flow of the Frame Retrieval

The first stage is taking frames based on intervals using equation 4.

$$Iv = \frac{Jum_frame\ - 5}{n} \quad (4)$$

Dimana:

Iv= frame interval

Jum_frame = total number of frames

n = number of frame that will be taken

Example, there are 65 frames in one video, and 8 frames will be taken for analysis. In application, the frame will be taken as much as 20 frames. In the previous research has distinguished frame retrieval between 10, 20, and 30 frames and the result obtained when frames are taken as much as 20 and 30 frame, accuracy tend to increase [7].

$$Iv = \frac{Jum_frame\ - 5}{n}$$
$$= \frac{65\ - 5}{8}$$
$$= 7{,}5 \text{ (rounded down = 7)}$$

Based on the results of calculations obtained, every 7 frames interval will take frames. The frames obtained are 7th, 14th, 21st, 28th, 35th, 42nd, 49th, and 56th frames and the center coordinates will be taken. The next process is the determination of sign language movements including static or dynamic sign language movements. Determination is based on the distance from equation 5.

$$s = x_{max} - x_{min} \quad (5)$$

Where:

s= Length

x_{max} = maximum value of coordinates X

x_{min} = maximum value of coordinates Y

The results of the distance calculation are checked for conditions using the equation 6.

$$signtype = \begin{cases} Dinamis & s > 15 \\ Statis & s \leq 15 \end{cases} \quad (6)$$

Examples of coordinates in alphabet A in Table 1.

Table 1. Coordinate Alphabet A

. Frame	Center Coordinate	
	X	Y
1	181	133
2	183	119
3	183	119
4	184	119
5	184	119
6	185	119
7	185	117
8	184	116

Based on Table 1 obtained the value of x_{max}= 185 and the value of x_{min} = 181, the distance can be calculated as follows.

$s = x_{max} - x_{min}$

$s = 185 - 181$

$= 4$

The result of calculating the center coordinate distance (s) for the letter A is 4. Then check the condition and get the result that 4 <15, then the letter A is a static sign language.

2.4. Motion Chain Code

In this research the Motion Chain Code (MCC) was used to extract hand movements by subtracting the hand centroid in two frames [4]. The output value of the MCC will be used as a dynamic sign language feature. In Figure 7 can be seen as the life process of the MCC.

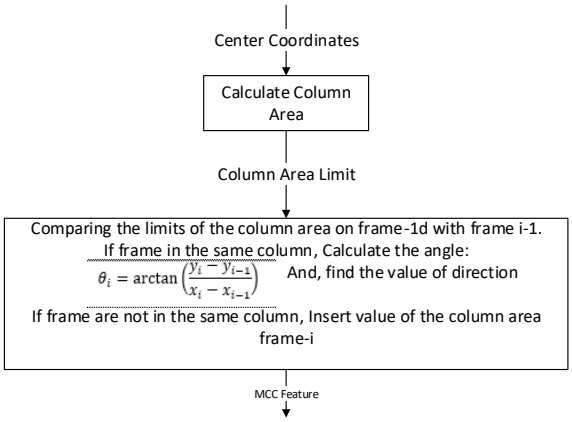


Figure 7. Flow of the Motion Chain Code Algorithm

The first step is to determine the boundary of the column area using the equation 7.

$$b_1 = \frac{x_{min} + x_{max}}{2}$$

$$b_2 = \frac{b_1 + x_{max}}{2} \quad (7)$$

$$b_3 = \frac{b_2 + x_{max}}{2}$$

Where:

b_1, b_2, b_3= column area limit

x_{min} = minimum value of X coordinates

x_{max} = maximum value of Y coordinates

Example of the center coordinate value in alphabet J in Table 2.

Table 2. Center Coordinat J

Number of Frame	Center Coordinates	
	X	Y
1	103	126
2	96	112
3	113	113
4	111	111
5	161	161
6	157	157
7	86	149
8	87	152

Following the calculation to determine the column area.

$$b_1 = \frac{x_{min} + x_{max}}{2} = \frac{79 + 103}{2} = \frac{182}{2} = 91$$

$$b_2 = \frac{b_1 + x_{max}}{2} = \frac{91 + 103}{2} = \frac{194}{2} = 97$$

$$b_3 = \frac{b_2 + x_{max}}{2} = \frac{97 + 103}{2} = \frac{200}{2} = 100$$

Based on the results of the calculation, the equation for each column is obtained using the boundaries of the column area that has been obtained as follows.

$$Columnarea = \begin{Bmatrix} 9, & 0 < x < 91 \\ 10, & 91 < x < 97 \\ 11, & 97 < x < 100 \\ 12, & 100 < x < x_n \end{Bmatrix} \quad (8)$$

The column area has values of 9, 10, 11 and 12 due to the difference in value with the direction of motion (O) which has values from 1 to 8 [5] which can be seen in equation 9.

$$O = \begin{Bmatrix} 1, & \theta < 22{,}5 \text{ dan } \theta > 337{,}5^o \\ 2, & 22{,}5^O < \theta < 67{,}5^o \\ 3, & 67{,}5^O < \theta < 110{,}5^o \\ 4, & 110{,}5^O < \theta < 157{,}5^o \\ 5, & 157{,}5^O < \theta < 202{,}5^o \\ 6, & 202{,}5^O < \theta < 242{,}5^o \\ 7, & 242{,}5^O < \theta < 292{,}5^o \\ 8, & 292{,}5^O < \theta < 337{,}5^o \end{Bmatrix} \quad (9)$$

After obtaining the column area, the next step is to check the condition of each coordinate. If the coordinates (x_i) nd the previous coordinates (x_{i-1}) are not in the same column, then specify the number of column (x_i) using equation 8. If the coordinates (x_i) nd the previous coordinates (x_{i-1}) are in the same column, berada pada kolom yang sama, then calculate the angle value using equation 10 [4].

$$\theta_i = \arctan\left(\frac{y_i - y_{i-1}}{x_i - x_{i-1}}\right) \quad (10)$$

Where:

θ_i = Angle values

x_i= X coordinates value

y_i= Y coordinates value

x_{i-1}= previous X coordinates value

y_{i-1}= previous Y coordinates value

Example checking condition of the center coordinate of the alphabet J frame-2 with frame-1 in Table 3 if the two frames are in the same column.

Table 3. Coordinat Frame-1 with frame-2

Number of Frame	Center Coordinates
1	X = 103 Y = 126
2	X = 96 Y = 112

Based on the equation of the column area in equation 8, the coordinate values on frame-2 and frame-1 in table 3 are not in the same column. Then enter the column area value which is 12 for frame-2 into MCC data: (12).

Another example checks the condition of the central coordinate of the alphabet J frame-7 with Frame-8 in Table 4.

Table 4. Coordinat Frame-7 with frame-6

Number of Frame	Center Coordinates
7	X = 86 Y = 149
6	X = 87 Y = 152

Based on the column area equation, the X coordinate value in table 3 is in the same column. Calculate angle values as follows.

$$\theta_i = \arctan\left(\frac{y_i - y_{i-1}}{x_i - x_{i-1}}\right)$$

$$\theta_5 = \arctan\left(\frac{y_8 - y_7}{x_8 - x_7}\right)$$

$$\theta_5 = \arctan\left(\frac{152 - 149}{87 - 86}\right)$$

$= \arctan\left(\frac{3}{1}\right)$

$= \arctan(3)$

$=71^{o}$

Based on the equation of the motion angle value, then the angle 71^{o} has the value of the direction of movement= 3. Enter the value 3 into the MCC data: (12, 8, 2, 9, 8, 7, 3).

The loop has been completed, the feature data from the MCC process for the J letter sign language obtained are (12, 8, 2, 9, 8, 7, 3). This data will be used as a feature for dynamic sign language movements during training and testing process.

2.5. Principal Component Analysis

Principal Component Analysis (PCA) is used to reduce the dimensions of the data observed into small dimensions without losing significant information in describing the entire data. For more details, see Figure 8.

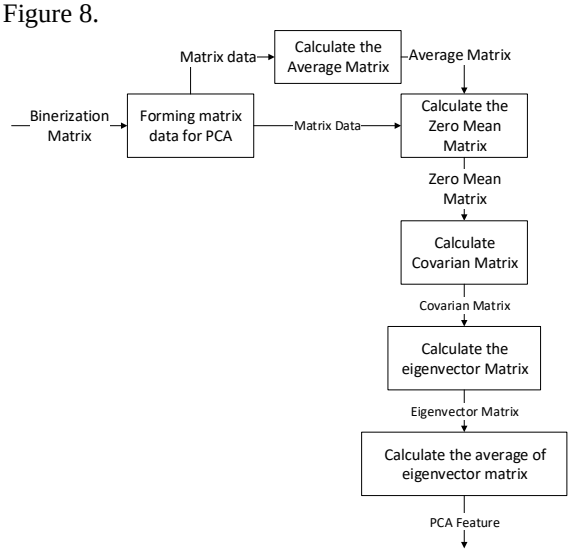


Figure 8. PCA Feature Extraction

Based on Figure 8 can be explained, the first step is to form a data matrix using a matrix from the results of binerization matrix and converted into a one-dimensional matrix measuring 6400x1 and combined into a MxN-sized matrix of 6400x8, where M is the matrix dimension and N is the number of matrices combined.

Table 5. Matrix Data

M / N	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0
14	0	0	0	0	0	1	1	1
15	0	0	0	0	0	1	1	1
16	0	1	0	1	0	1	0	0
17	1	1	0	1	1	1	0	0
18	1	1	0	1	1	1	0	0
19	1	1	1	1	1	1	0	0
20	1	1	1	1	1	0	0	0
	…	…	…	…	…	…	…	…
6400	0	0	0	0	0	0	0	0

Then the average matrix calculation is done using equation 11.

$$\bar{x}_N = \frac{\sum_{i=1}^{M} x_i}{M} = \frac{x_1 + x_2 + \cdots + x_M}{M} \qquad (11)$$

Dimana:

$\bar{x}_N$ = the average of n column

M = matrix dimension

Example calculation of matrix averages $\bar{x}_1$:

$$\bar{x}_N = \frac{x_1 + x_2 + \cdots + x_M}{M}$$

$$\bar{x}_1 = \frac{0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + \cdots + 0}{6400}$$

$$\bar{x}_1 = \frac{3719}{6400} = 0{,}5810938$$

This calculation is done from $\bar{x}_1$to $\bar{x}_N$. The average results can be seen in table 6.

Table 6. Average Value

$\bar{x}_N$	$\bar{x}_1$	$\bar{x}_2$	$\bar{x}_3$	$\bar{x}_4$	$\bar{x}_5$	$\bar{x}_6$	$\bar{x}_7$	$\bar{x}_8$
$\bar{x}$	0.581 09	0.581 09	0.570 31	0.571 25	0.570 15	0.572 03	0.566 56	0.562 65

After getting the average matrix, the next step to calculate the zero matrix means using equation 12.

$$Y = D - \bar{x} \qquad (12)$$

Where:

D = matrix data

$\bar{x}$ = average matrix

An example of calculating a zero mean matrix is as follows.

$Y = D - \bar{x}$

$$Y = \begin{bmatrix} D_{11} - \bar{x}_1 & D_{12} - \bar{x}_2 & \cdots & D_{1,n} - \bar{x}_n \\ D_{21} - \bar{x}_n & D_{22} - \bar{x}_2 & \cdots & D_{1,n} - \bar{x}_n \\ \cdots & \cdots & \cdots & \cdots \\ D_{m,1} - \bar{x}_1 & D_{m,1} - \bar{x}_2 & \cdots & D_{m,n} - \bar{x}_n \end{bmatrix}$$

$$= \begin{bmatrix} 0 - 0.58109 & 0 - 0.58109 & \cdots & 0 - 0.56265 \\ 0 - 0.58109 & 0 - 0.58109 & \cdots & 0 - 0.56265 \\ \cdots & \cdots & \cdots & \cdots \\ 0 - 0.58109 & 0 - 0.58109 & \cdots & 0 - 0.56265 \end{bmatrix}$$

$$Y = \begin{bmatrix} -0.58109 & -0.58109 & \cdots & -0.56265 \\ -0.58109 & -0.58109 & \cdots & -0.56265 \\ \cdots & \cdots & \cdots & \cdots \\ -0.58109 & -0.58109 & \cdots & -0.56265 \end{bmatrix}$$

After obtaining the zero mean matrix, the next step is to calculate the covariance matrix using equation 13 [6].

$$C = \frac{1}{M-1} Y^T Y \qquad (13)$$

Where:

C= Covarian Matrix

M=Matrix Dimension

Y= Zero Mean Matrix

Examples of covariance matrix calculations are as follows.

$$C = \frac{1}{M-1} Y^T Y$$

C

$$= \frac{1}{6400-1} \begin{bmatrix} D_{1,1} & D_{2,1} & \cdots & D_{6400,1} \\ D_{1,2} & D_{2,2} & \cdots & D_{6400,2} \\ \cdots & \cdots & \cdots & \cdots \\ D_{1,8} & D_{2,8} & \cdots & D_{6400,8} \end{bmatrix} \text{x} \begin{bmatrix} D_{1,1} & D_{1,2} & \cdots & D_{1,8} \\ D_{2,1} & D_{2,2} & \cdots & D_{2,8} \\ \cdots & \cdots & \cdots & \cdots \\ D_{6400,1} & D_{6400,2} & \cdots & D_{6400,8} \end{bmatrix}$$

$$C = \frac{1}{6399}\begin{bmatrix} -0.54313 & -0.54313 & \cdots & -0.54313 \\ -0.52392 & -0.52392 & \cdots & -0.52392 \\ \cdots & \cdots & \cdots & \cdots \\ -0.63397 & -0.63397 & \cdots & -0.63397 \end{bmatrix} *$$

$$\begin{bmatrix} -0.54313 & -0.52392 & \cdots & -0.63397 \\ -0.54313 & -0.52392 & \cdots & -0.63397 \\ \cdots & \cdots & \cdots & \cdots \\ -0.54313 & -0.52392 & \cdots & -0.63397 \end{bmatrix}$$

$$C = \frac{1}{6399}\begin{bmatrix} 1557.9 & 1450.9 & \cdots & 1384.48 \\ 1450.91 & 1557.9 & \cdots & 1391.48 \\ \cdots & \cdots & \cdots & \cdots \\ 1384.48 & 1391.4814 & \cdots & 1574.87 \end{bmatrix}$$

$$C = \begin{bmatrix} \frac{1557.9}{6399} & \frac{1450.9}{6399} & \cdots & \frac{1384.48}{6399} \\ \frac{1450.91}{6399} & \frac{1557.9}{6399} & \cdots & \frac{1391.48}{6399} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{1384.48}{6399} & \frac{1391.48}{6399} & \cdots & \frac{1574.87}{6399} \end{bmatrix}$$

$$C = \begin{bmatrix} 0.24346 & 0.22674 & \cdots & 0.21636 \\ 0.22674 & 0.24346 & \cdots & 0.21745 \\ \cdots & \cdots & \cdots & \cdots \\ 0.21636 & 0.21745 & \cdots & 0.24611 \end{bmatrix}$$

After obtaining a covariance matrix, the next step is to determine the eigenvector matrix by looking first eigenvalue using equation 14 [7].

$$C - Z = |C - \lambda I|$$
$$|C - \lambda I| = 0 \qquad (14)$$

Where:

C = Covarian Matrix

Z = Eigenvector Matrix

λ = scalar

I = Indentity Matrix

Examples of eigenvalue calculations are as follows.

$$C - Z = |C - \lambda I|$$
$$|C - \lambda I| = 0$$
$$\begin{bmatrix} C_{11} & C_{12} & \ldots C_{1,n} \\ C_{21} & C_{22} & \ldots C_{2,n} \\ \ldots & \ldots & \ldots \ldots \\ C_{m,1} & C_{m,2} \ldots & C_{m,n} \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \ldots & 0 \\ 0 & 1 \ldots & 0 \\ \ldots & \ldots\ldots & \ldots \\ 0 & 0 \ldots & 1 \end{bmatrix} = 0$$
$$\begin{bmatrix} 0.24346 & 0.22674 & \ldots 0.21636 \\ 0.22674 & 0.24346 & \ldots 0.21745 \\ \ldots & \ldots & \ldots \ldots \\ 0.21636 & 0.21745 & \ldots 0.24611 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \ldots & 0 \\ 0 & \lambda \ldots & 0 \\ \ldots & \ldots\ldots & \ldots \\ 0 & 0 \ldots & \lambda \end{bmatrix} = 0$$
$$\begin{bmatrix} 0.24346 - \lambda & 0.22674 & \cdots & 0.21636 \\ 0.22674 & 0.24346 - \lambda & \cdots & 0.21745 \\ \cdots & \cdots & \cdots & \cdots \\ 0.21636 & 0.21745 & \cdots & 0.24611 - \lambda \end{bmatrix} = 0$$

Then the equation is obtained as follows.

$$-\lambda^8 + 1.95866\lambda^7 - 0.29188\lambda^6 + 0.01857\lambda^5 - 0.0006435\lambda^4 + 0.00001\lambda^3 - 1.574568e7\lambda^2 + 1.033097e - 9\lambda - 2.861807e - 12 = 0$$

From the matrix equation above, the following eigenvalue is obtained.

$\lambda_1 = 1.801801$ $\lambda_2 = 0.47871$ $\lambda_3 = 0.20153$ $\lambda_4 = 0.10211$
$\lambda_5 = 0.04943$ $\lambda_6 = 0{,}03455$ $\lambda_7 = 0.03275$ $\lambda_8 = 0.01818$

Then the eigenvector matrix is obtained in Table 7.

Table 7. Eigenvector Matrix

i / j	1	2	3	4	5	6	7	8
1	0.35539	0.35491	0.35652	0.35333	0.35458	0.35619	0.34981	0.34758
2	-0.21589	-0.14241	-0.24864	-0.40133	-0.24890	0.17614	0.46996	0.62960
3	-0.27672	-0.15426	-0.30472	0.21134	0.42655	-0.15671	0.64060	-0.38110
4	-0.24481	-0.47817	-0.14535	0.29119	0.48233	0.11404	-0.43136	0.41687
5	-0.41787	0.44785	-0.01330	0.44255	-0.21118	-0.53116	-0.02484	0.31850
6	-0.23109	0.33099	-0.47970	0.25703	-0.21005	0.66297	-0.14447	-0.19065
7	-0.00697	-0.53314	0.30863	0.51336	-0.54193	0.13804	0.19685	-0.07366
8	0.68009	-0.08339	-0.60955	0.24328	-0.12381	-0.25262	0.00943	0.14339

hen the result of the eigenvector matrix is calculated the average value of the eigenvector for each eigenvalue (the average of each column). In Table 8 the data is seen the average eignvector results for the alphabet I.

Table 8. The Result of PCA Feature Extraction

Huruf	PCA Values							
I	-0.0447	-0.0322	-0.1420	0.2388	-0.009	0.0633	0.1332	0.1513

2.6. Classification with SVM Multiclass

The results of the PCA feature extraction calculation will be used for training static sign language and the results of MCC will be used for dynamic sign language training. In Figure 9 can be explained the flow of the training process using SVM Multiclass [6].

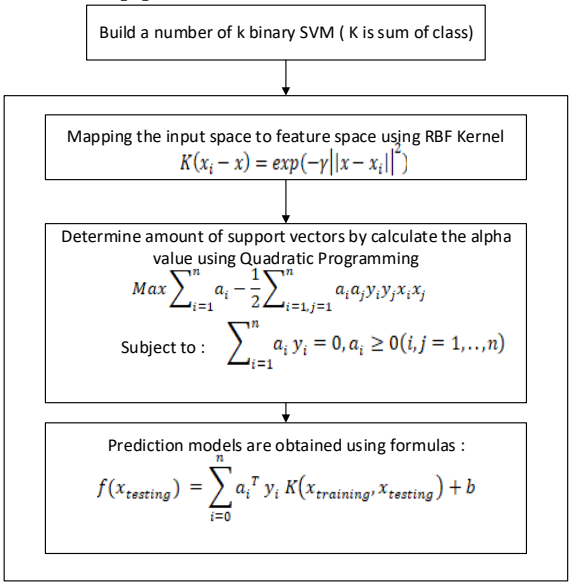


Figure 9. Flows SVM Training

In the testing process after obtaining a separation field, then it can determine the sign language test data included in the positive or negative class using the following decision functions [8].

$f(x_{testing}) = \sum_{i=0}^{n} a_i{}^T y_i K(x_{training}, x_{testing})$+b

he input data used is test data from PCA feature extraction for static sign language and MCC results for dynamic sign language. The calculation results from the separator data are the result of class prediction. In the Figure 10 can be explained the process flow testing SVM Multiclass One vs. All.

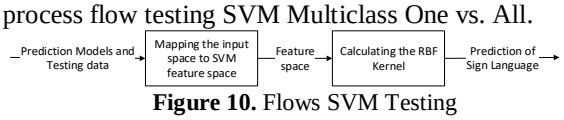


Figure 10. Flows SVM Testing

2.7. Experiment Result

In this research the test was carried out based on the number of training data and test data used.

a. Test Scenario 1

In scenario 1, the accuracy of the Motion Chain Code (MCC) method will be tested as a feature for dynamic sign language and the Principal Component Analysis (PCA) method as a feature for static sign language. Tests were carried out to use 520 videos of 520 video test data training data. Each sign language gesture letter has 20 training data and 20 test data. The confusion matrix for test scenario 1 can be seen in Table 9.

Table 9. Confusion Matrix Testing Scenario 1

Class		Classification Result	
		P	N
Target	T	327	104
	F	89	0

Then calculate the accuracy of the confusion matrix obtained for static sign language using the following equation.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} * 100\%$$
$$= \frac{327}{(327 + 104 + 89 + 0)} * 100\%$$
$$= 62{,}8\%$$

Accuracy for static and dynamic sign language using 520 training data and 520 testing data, each of alphabet has 20 training data and 20 testing data with an accuracy of 62.8%. For dynamic sign language, the accuracy is 82.5% while for static sign language, the accuracy is 56.5%.

b. Test Scenario 2

In the test scenario 2, the accuracy of the Motion Chain Code (MCC) method will be tested as a feature for dynamic sign language movements and the Principal Component Analysis (PCA) method as a feature for static sign language movements. Tests were carried out to use 1040 video training data and 520 video test data. Each alphabet has 40 training data and 20 testing data. The confusion matrix table for test scenario 2 can be seen in Table 10.

Table 10. Confusion Matrix Testing Scenario 2

Class		Classification Result	
		P	N
Target	T	360	92
	F	68	0

The results of the calculation of accuracy of scenario 2 testing are as follows

$$Accuracy = \frac{TP + FN}{TP + TN + FP + FN} * 100\%$$
$$= \frac{360 + 0}{(360 + 68 + 92 + 0)} * 100\%$$
$$= 69{,}23\%$$

In testing of static and dynamic sign language that has been done using 1040 video training data and 520 video test data, each alphabet has 40 video training data and 20 test data videos. In the experiments result for static sign language, correctly classified as many as 324 testing from 480 testing data. The Accuracy obtained for static sign languages is 67.5%. In the experiments result for dynamic sign language, correcly classified as many as 36 testing data from 40 testing data. The accuracy obtained for dynamic sign language is 90%. Based on the experiment result have been carried out, the accuracy obtained for sign language from the alphabet A-Z in experiment scenarion 2 is 69.23% with precision of 79.65% and recall of 100%.

3. CLOSURE

3.1. Conclusion

Based on the experiment result carried out, the amount of training data is very influential on the accuracy obtained. The more training data the greater accuracy you get. Based on the test results it can be concluded that by implementing the Motion Chain Code algorithm and Principal Component Analysis to recognize Indonesian sign language. The highest accuracy obtained for recognize static and dynamic sign language is 69% with training data as much as 1040 video and tested using 520 videos, each alphabet A-Z has 40 training data and 20 testing data. In testing of static sign language obtained accuracy 67,25% and for dynamic sign language obtained accuracy of 90%. PCA feature can be used to recognize static sign but has not been able to properly recognize due to data factors practice and there are several letters of almost the same shape that are difficult to recognize, but can recognize well dynamic sign language letters using features from MCC.

3.2. Suggestion

Based on the results of research and testing can be given suggestions for subsequent research to try several other feature extraction methods for the introduction of static sign language. Try the MCC algorithm by adding another dynamic sign language.

BILBIOGRAPHY

[1] W. Fauzan, "Finger Spelling untuk Kata Khusus dengan Bahasa Isyarat Tangan Menggunakan Metode Viola and Jones," *Skripsi*, UNIKOM, 2017.

[2] N. E. Sihbudi , "Impelemntasi Metode Naive Bayesian untuk Menerjemahkan Bahasa Indonesia," *Skripsi*, UNIKOM, 2017.

[3] A. Mardiyani, "Pengenalan Bahasa Isyarat Menggunakan Metode PCA dan Haar Like Feature," *ITS*, 2012.

[4] A. M. Malik, N. Kousar, T. Abdullah , M. Ahmed, F. Rasheed, "Pakistan Sign Language Using PCA and KNN," *International Journal of Advanced Computer Sience and Application (IJACSA)*, vol. 9, p. 4, 2018.

[5] M. Z. Abdo, S. A. El-Rahman, A. M. Hamdy and E. M. Saad, "Enhancement of Motion Chain Code for Arabic Sign Language Recognition," *IJACSA*, vol. 6, p. 12, 2015.

[6] G. Kukharev and A. Navosielski, "Visitor Identification Real Time Face Recognition System," *Winter School on Computer Graphics (WSCG)*, pp. 157-164, 2004.

[7] B. Putra AR, Computer Vision & Aplikasinya menggunakan C# & EmguCV, Yogyakarta: Penerbit Andi, 2017.

[8] R. Febriana, "Analisis dan Implementasi Aplikasi Pengenalan Gerak Bibir Menjadi Teks Menggunakan Metode Support Vector Machine," *Skripsi, Universitas Telkom*, 2015.

[9] Linear Algebra, "Eignvalue and Eignvevtor," [Online]. Available: http://algebra.math.ust.hk/eigen/01_definition/lecture2.shtml. [Accessed 19 July 2018].

[10] R. Rifkin, A. Klautau, "In Defense of One Vs All Classification," *J. Mach. Learn*, vol. 5, pp. 101-141, 2004.