IMPLEMENTATION SCALE INVARIANT FEATURE TRANSFORM-HISTOGRAM OF GRADIENTS AND MULTICLASS SVM FOR HANDTRACKING AT MOUSE POINTER CONTROL

Anggy Edo Prasetya[1], Irfan Maliki, S.T, M.T[2]

[1,2]Informatics Engineering-University Computer Indonesia
Jln. Dipatiukur No. 112-116, Lebakgede, Coblong, Kota Bandung, West Java 40132
E-mail : anggyedoo@gmail.com[1], irfanmaliki007@gmail.com[2]

ABSTRACT

More interactive dialogue techniques enabling communication between humans and computers that have many variations. Techniques such interaction is direct physical engagement with the computer system and input devices or so-called Human-Computer Interaction (Human Computer Interaction) One interaction including Human-Computer Interaction is handtracking to control a pointer event. To make the process handtracking needed some method in the beginning (preprocessing) such as RGB to HSV, search HSV color red (color detection), opening (erosion and dilation), closing (dilation and erosion), Different of Gaussian (DOG) of SIFT, looking for ROI hand and do Image Moments and to conduct the necessary handrecognition Histogram of Gradients method for feature extraction and Support Vector Machine for classification of a number of events such as moving the pointer is a pointer, left click, right click, scroll and drag and drop. Based on the results of the evaluation system testing has been done for all test scenarios such as moving the pointer to the far right of the desktop screen, left-click the icon of windows, right click on the desktop screen, drag and drop the program word and scrolling on PDF documents obtained average accuracy of 78% ,

Keywords : Handtracking, Handrecognition, Color detection, SIFT, HOG, SVM, Computer Vision

1. PRELIMINARY

Human computer interaction is usually done in voice recognition, touch screen, gesture recognition and handtracking. Handtracking is an introduction to the shape and movement of the human hand is made to achieve certain goals such as the introduction of sign language, the interaction of humans and robots and to control the mouse pointer directly with the human hand through the web camera [1]. Computers in operation generally use certain hardware or tools to enter the command on the computer for example, mouse, keyboard and webcam [2]. Research on handtracking will try to maximize the performance of the input tool to interact between humans and computers [3].

Research on the hand tracking generally using skin color detection method as segmentation method to detect the hand through a range of values of human skin color in the YCbCr color space as studies using pyramidal algorithm Lucas Kanade as handtracking process [4]. The use of skin color detection methods to get good results only if they display a specific area on the palm for human skin tones, but it will get poor results as well as the area hands down or areas containing the human skin color more like the face shown. Because these methods will detect other areas such as regional palms face down and the area causing handtracking process failed.

Previously there handtracking for research on human interaction and robot uses an algorithm adaboost with SIFT (Scale-invariant feature transform) to provide the level of accuracy of 90% [5]. Subsequent research has been done using adaptive algorithms HOG HOG (Histogram of Gradients) -LBP (Local Binary Pattern) to detect the palm of the hand (palm tracking) to provide precision accuracy of 95.2% [6]. Referring to the studies using the algorithm SIFT (Scale-invariant feature transform) -HOG (Histogram of Gradients) and multiclass SVM (Support Vector Machine) can get an accuracy of 97% [7].

Therefore, based on the exposure of problems which have been described this research is expected to get the results of hand tracking either by using algorithms SIFT and HOG as extraction features and multiclass SVM as the classification in the different cases are hand tracking on mouse pointer that can perform event mouse pointer with hand.

2. RESEACRH CONTENT

In this section we discuss the research methods, the theoretical basis, system analysis, and test results.

2.1 Research Methods

The research method uses research methods based on case studies. The case study, or research the case (Case study), is a research subject status based on research regarding typical of the whole personality [8], The step - step study as shown in Figure 1

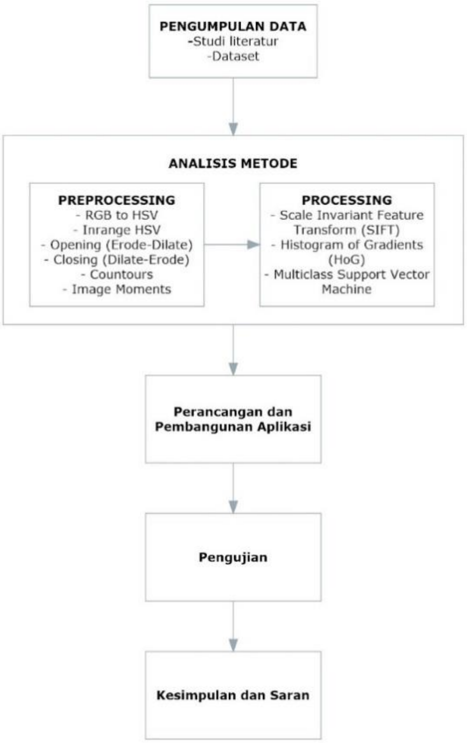

Figure 1 Step - step study

2.2 Computer Vision

At its essence, Computer Vision try to mimic how the human visual system (Human Vision). Human Vision very complex indeed. Man looks at the object through the sense of sight (eyes), then the object image transmitted to the brain to be interpreted so that people understand what objects are visible in his eyes. The result of this interpretation allows for decision-making [9],

2.3 Image HSV

HSV stands for Hue, Saturation, and Value. The advantage is a lot of colors similar to those taken by the human senses. While the colors are made of other models such as RGB is the result of a mixture of primary colors. The hallmark of HSV are as follows:

1. Hue , is the actual color like red, violet and yellow. Used to determine redness (Redness), sulfur (greeness) and others.
2. Saturation , or referred to as Chroma is the purity or strength of color.
3. Value , brightness of color. The value ranges from 0-100%. If the value is 0 then the color will be black, the greater the value, the brighter and new variations will appear.

The following is the calculation of the HSV value from converting RGB images to HSV images [10]:

$$r = \frac{R}{(R+G+B)}, g = \frac{G}{(R+G+B)}, b = \frac{B}{(R+G+B)} \quad (1)$$

$$V = max(r, g, b) \quad (2)$$

$$S = \begin{cases} 0, \ jika\, V = 0 \\ 1 - \frac{\min(r,g,b)}{V}, \ jika\, V > 0 \end{cases} \quad (3)$$

$$H = \begin{cases} 0, \ jika\, S = 0 \\ \frac{60*(g-b)}{(S*V)}, \ jika\, V = r \\ 60 * \left[2 + \frac{(b-r)}{(S*V)}\right], \ jika\, V = g \\ 60 * \left[4 + \frac{(r-g)}{(S*V)}\right], \ jika\, V = b \end{cases} \quad (4)$$

$$H = H + 360, \ jika\, H < 0 \quad (5)$$

R, *G*, *B* are values from RGB images that will be converted to HSV imagery.

2.4 Erode

Erode (Erosion) operation is a process of minimizing the image structure. Erosion operations can be formulated as follows [11] :

$A \ominus B = \{ p \in z^2 \mid (a + b) \in |, \ b \in B\}$(6)

Where A is f (x, y) of the original image and B are elements structure. Structure element that is generally used in the operation of erosion is a form box. Other forms of Structure elements elipse form, lines, and other disc.

Results erosion are the logical AND operation of each of the coordinates of A and B. The following is the result of the operation erode contained in Figure 2

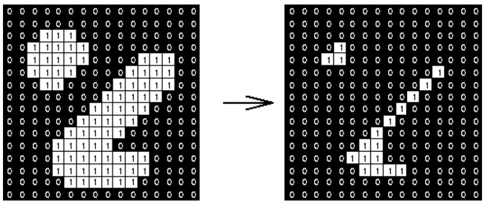

Figure 2 Results of Operations Erode / Erosion

2.5 Dilate

Operation dilate (dilation) is the process of widening of the pixels is 1. The operation of dilation can be formulated as follows [11]:

$A \oplus B = \{ z \mid z = a + b, a \in A, b \in B \}$ (7)

Where A is f (x, y) of the original image and B are elements structure. Structure elements commonly used in the operation are also generally rectangular dilation.

The results of dilation in the form of the sum of all coordinate pair of A and B. The following is the result of a dilate operation contained in Figure 3

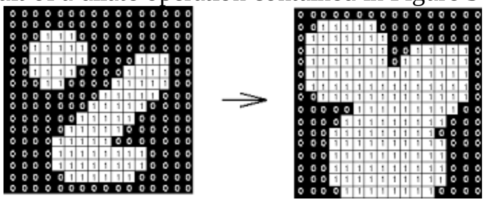

Figure 3 Results of Operations Dilate / dilation

2.6 Histogram of oriented Gradients

The histogram of oriented Gradients is a common way to obtain a descriptor for a particular object detection. For example, to detect the presence of human (Human detection) such as the study conducted by Dalal and Trigs [12], The process of this algorithm can be described as follows [1] :

1. *Preprocessing*, Perform intensity normalization or otherwise alter the image into greyscale image (gray scale image).
2. Counting edge map. Estimating the x and y directions on the image and then compute the gradient magnitudes and gradient angle for each pixel of the image. The following is a calculation formula for the gradient magnitudes and gradient angle:

$$|G| = \sqrt{I_x^2 + I_y^2} \qquad (8)$$

Where is | G | are absolute numbers from gradient *magnitudes* and *I* are *greyscale* image, images from the results of the previous stages. There is a matrix against the x-axis and matrix of the y-axis. and can be calculated by the calculation formula as follows:

$$I_x = I * D_x \; dan \; I_y = I * D_y \qquad (9)$$

D_x is a mask , meanwhile mask each calculated by convolution (sign *). Then the gradien angle into coordinat axis with the angle between 0 to 180 can be calculated by the calculation formula as follows :$[-1\ 0\ 1]D_y\begin{bmatrix}-1\\0\\1\end{bmatrix}$

$$\theta = \arctan(\frac{I_x}{I_y}) \qquad (10)$$

3. *Spatial binning*, The next stage is calculating a histogram of the gradient angle to each cell. Each pixel in a cell has its own histogram value - itself based on the value generated in the calculation of normalized gradient then performed on every block. Cell has a size of 8x8 pixels in an image. While the block has a size of 2x2 cell.
4. *Normalize voting values for generating a descriptor*, The next block feature normalization value is obtained by calculating the following formula [13]:

$$norm = \frac{v(n)}{\sqrt{(\sum_{k=1}^{block*l} v(k)^2)+1}} \qquad (11)$$

V value is the value of the gradient magnitudes, while n is the number of bins and blocks (2x2 cell) is a cell number, while l is the number of blocks that do not overlap. Features normalized block to reduce the effects of changes in the brightness of the object on the block.

5. *Augment all block vectors consecutively*, Once the feature blocks are normalized, normalized value of each block will be combined into a single vector (vector 1 dimension) the results of these vectors can be referred to as a feature vector histogram of oriented Gradients.

Here is an illustration of how the histogram of oriented Gradients contained in Figure 4

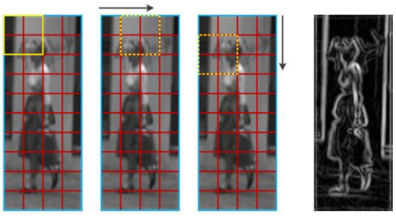

Figure 4 Histogram of oriented Gradients

2.7 Multiclass Support Vector Machine

There are two ways to implement the multiclass SVM by merging several binary SVM and the incorporation of all the data that consists of multiple classes into shape optimization problems. Here's a common method used to implement the multiclass SVM approach as follows [14]:

1. Method one-against-all

By using this method, will be built binary SVM models k (k is the number of classes). For example, there are problems with 4 grade classification. For training to use 4 pieces of binary SVM as shown in Table 1 and its users in the class classifying new data can be seen in the following equation [15]:

$$Kelas\ x = \arg\max_{i=1..k}((w^{(i)})^T.\varphi(x) + b^{(i)}) \quad (12)$$

By determining the greatest hyperplane on the value of x will classify the class.

table 1 4 binary SVM method One-against-all

$y_i = 1$	$y_i = -1$	hypothesis
class 1	Not a class 1	$f^1(x) = (w^1)x + b^1$
Grade 2	Not a class 2	$f^2(x) = (w^2)x + b^2$
Grade 3	Not a grade 3	$f^3(x) = (w^3)x + b^3$
4th grade	Not a class 4	$f^4(x) = (w^4)x + b^4$

2.9 Analysis System

The system is built consists of 2 stages. The first stage is the stage of tracking Hand namely the detection location of the position of ROI (Region of Interest) last hand will be conducted Hand stage recognition which at this stage will be carried out training and testing. Stages of training will be conducted by HOG feature extraction and make the SVM training data. Stages of testing will include SVM training data and make predictions label. Overview of the general description of the system can be seen in Figure 5

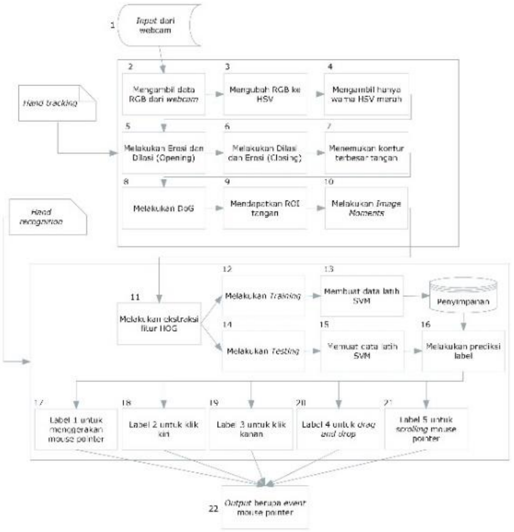
1
Input dari webcam
Hand tracking
2
Mengambil data RGB dari *webcam*
3
Mengubah RGB ke HSV
4
Mengambil hanya warna HSV merah
5
Melakukan Erosi dan Dilasi (Opening)
6
Melakukan Dilasi dan Erosi (Closing)
7
Menemukan kontur terbesar tangan
8
Melakukan DoG
9
Mendapatkan ROI tangan
10
Melakukan *Image Moments*
Hand recognition
11
Melakukan ekstraksi fitur HOG
12
Melakukan *Training*
13
Membuat data latih SVM
Penyimpanan
14
Melakukan *Testing*
15
Memuat data latih SVM
16
Melakukan prediksi label
17
Label 1 untuk menggerakan mouse pointer
18
Label 2 untuk klik kiri
19
Label 3 untuk klik kanan
20
Label 4 untuk *drag and drop*
21
Label 5 untuk *scrolling* mouse pointer
22
Output berupa *event* mouse pointer

Figure 5 Overview general description of the system

Explanation of the stages - the stages above are as follows:

1. *Input* from webcam

An initial stage to get the image from the webcam.

2. Taking the RGB data from the webcam

The process in this stage is to convert the RGB values at each pixel image into HSV image value. For example, take a sample of one frame of RGB image tracking Hand stage with RGB image measuring 320 x 240 as shown in Figure 6

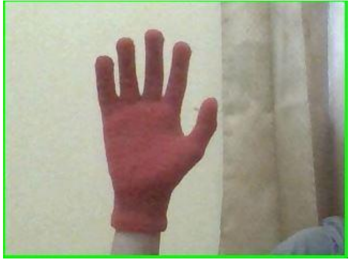

Figure 6. Example of RGB image

3. Changing RGB to HSV

Is the process of converting data martiks RGB to HSV. Performed for the needs of decision-red HSV color values. For example, take the value of RGB pixels (1,1), then the process of calculating the RGB to HSV using equations 1 to 5 as follows:

$$r = \frac{R}{(R+G+B)}, g = \frac{G}{(R+G+B)}, b = \frac{B}{(R+G+B)}$$

$$r = \frac{0}{(0+255+0)}, g = \frac{255}{(0+255+0)}, b = \frac{0}{(0+255+0)}$$

$$r = 0 \quad g = 1 \quad b = 0$$

$$V = max(r, g, b)$$

$$V = 1$$

$$S = \begin{cases} 0, & jika\, V = 0 \\ 1 - \dfrac{\min(r,g,b)}{V}, & jika\, V > 0 \end{cases}$$

$$S = 1 - \frac{0}{1}$$

$$S = 1$$

$$H = \begin{cases} 0, & jika\, S = 0 \\ \dfrac{60 * (g-b)}{(S * V)}, & jika\, V = r \\ 60 * \left[2 + \dfrac{(b-r)}{(S * V)}\right], & jika\, V = g \\ 60 * \left[4 + \dfrac{(r-g)}{(S * V)}\right], & jika\, V = b \end{cases}$$

$$H = 60 * \left[2 + \frac{(0-0)}{(1 * 1)}\right]$$

$$H = 120$$

$$H = H + 360, \quad jika\, H < 0$$

$$H = 120$$

So the HSV values obtained from pixel (1.1) that H = 120, S = 1 and V = 1. Sample results from the conversion of RGB to HSV image imagery can be seen in Figure 7 as follows:

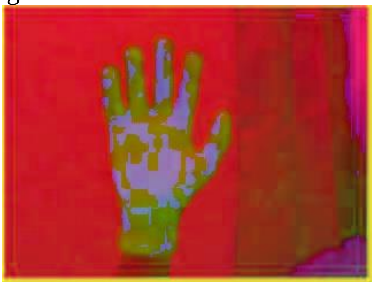

Figure 7 Example Image HSV

4. Taking only the HSV color red

Is the process of making only the red color HSV. Carried out for the needs of the hand segmentation. In this process will be carried out taking only the HSV HSV color red and other colors for segmentation ignore the hand. Red color HSV-making process is based on the scales of red HSV values. The following are the scales of red HSV color values:

a. Hue value is the range of values from 0 to 10 or 170 to 200.
b. Saturation value of the range of values from 70 to 255.
c. Value Value the range of values from 50 to 255.

5. Doing erosion and dilation (opening)

A thickening process matrix data that has been segmented on the need to widen the area ROI hand so the classification process more get good accuracy. Examples of opening operating results contained in Figure 8 as follows:

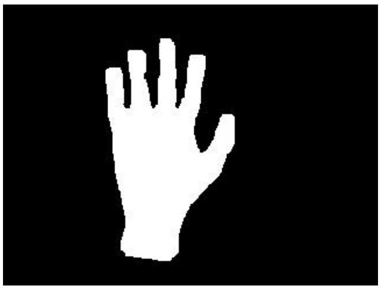

Figure 8 Example of opening operation

6. Perform dilation and erosion (Closing)

Penepisan process is already segmented matrix data for ROI area needs more hands so that the classification process to get good accuracy. Examples of closing operating results contained in Figure 9 as follows:

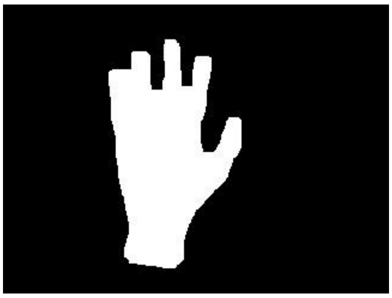

Figure 9 Example of closing operations

7. Find the biggest contour of the hand

The contours of the search process is a data matrix. Segmentation is done to hand in order to get good results. Examples of the results of the hand are looking largest contour in Figure 10 as follows:

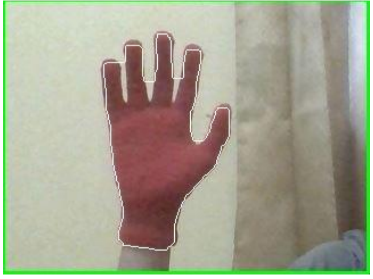

Figure 10 biggest Contour hands

8. Perform DOG

An edge detection process using dogs for hand segmentation in order to obtain good results. Sample results from the image matrix DOG found in Figure 11 as follows:

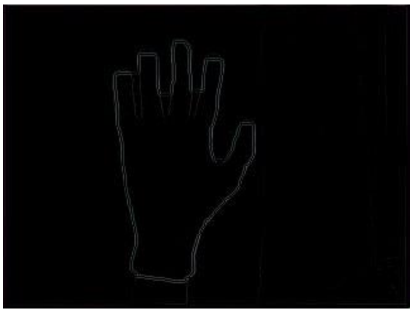

Figure 11 The results of the image matrix DOG

9. Getting ROI hand

Is the process of getting data matrix ROI only on the hand. Carried out for the needs of HOG feature extraction, created training data and SVM classification. Examples of ROI hand results in Figure 12 are as follows:

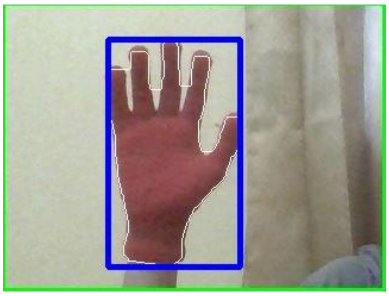

Figure 12 Results of ROI hand

10. Perform Image Moments

A process to obtain the midpoint of the ROI hand. Needed for the needs of the movement of the point x and y of the mouse pointer. Examples of these methods results Image moments can be seen in Figure 13 as follows:

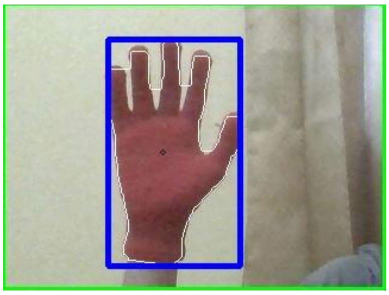

Figure 13 The results of the method of moments Image

11. Perform feature extraction HOG

A feature extraction process to obtain each image will be used as training data. As an example for this phase will take a sample of training data RGB image matrix size of 200 x 200 which can be seen in Figure 14 as follows:

Figure 14 training data sample size of 200 x 200

Here is the process HOG method:

a. Change the RGB image into grayscale image

In the process of HOG method required a gray-scale image (Grayscale) for example will take an RGB value matrix coordinate (1,1) will then be calculated to obtain the value Grayscale as follows:

$$Grayscale = (0.299 * 198) + (0.587 * 219) + (0.114 * 216)$$

$$Grayscale = (59.202) + (128.553) + (24.624)$$
$$Grayscale = 212$$

Sample results from the conversion of RGB image Grayscale image to the training data can be seen in Figure 15 as follows:

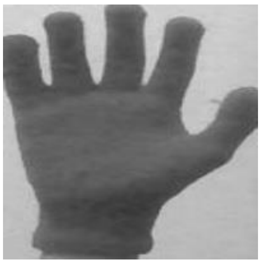

Figure 15 Sample image Grayscale

b. Calculate *edge map*

In this process the calculation of the gradient *magnitudes* and *angle* for each gray image (*Grayscale*) before the pixel will be done before convolution with the *mask*. D_x that is and D_y that is to find the value of I_x and $I_y$with a *Grayscale* matrix measuring 4 x 4 as a sample to find the *magnitude* and *angle* gradient values like equation 8 as follows :

$$I_x = I * D_x \; dan \; I_y = I * D_y$$

$$I_x = \begin{bmatrix} 212\;212\;213\;214 \\ 207\;208\;209\;210 \\ 203\;204\;205\;206 \\ 203\;204\;204\;204 \end{bmatrix} * [-1\;0\;1]$$

$$I_y = \begin{bmatrix} 212\;212\;213\;214 \\ 207\;208\;209\;210 \\ 203\;204\;205\;206 \\ 203\;204\;204\;204 \end{bmatrix} * \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

For example, a convolution calculation only on the coordinate (2,2) to $I\,x$ and $I\,y$ the result is :

$$I_x = |\,(207 * -1) + (208 * 0) + (209 * 1)\,|$$
$$I_x = 2$$
$$I_y = |\,(212 * -1) + (208 * 0) + (204 * 1)\,|$$
$$I_y = 8$$

Furthermore, calculation of magnitudes and angles gradients will be carried out as in equations 9 and 10 as follows:

$$|G| = \sqrt{I_x^2 + I_y^2}$$

$$\theta = \arctan\left(\frac{I_x}{I_y}\right)$$

$$|G| = \sqrt{2^2 + 8^2}$$
$$|G| = \sqrt{4 + 64}$$
$$|G| = 8\;(dibulatkan)$$
$$\theta = \arctan\left(\frac{2}{8}\right)$$
$$\theta = \arctan(0.25) \sim 14°$$

Then the value of the gradient *magnitudes* and *angle* is 8 and 14 ° for coordinates (2.2)

c. Perform *Spatial binning*

In this process will be calculated from the gradient angle histogram that has been obtained is processed prior to each - each cell. Each pixel in a cell will have its own histogram value - itself based on the value generated in the calculation of normalized gradient then performed on each block. Cell for this research has a size of 7 x 7, which will continue to move to the right and down until the end of the picture cell as shown in Figure 16 as follows:

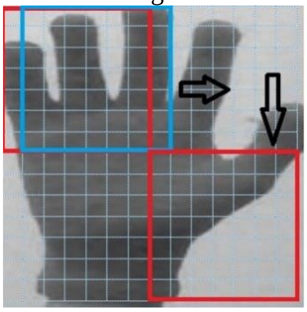

Figure 16 Cell moves to the end of the picture

Here is an example of 9 bins with the orientation of 0-180 ° as shown in Figure 17 below:

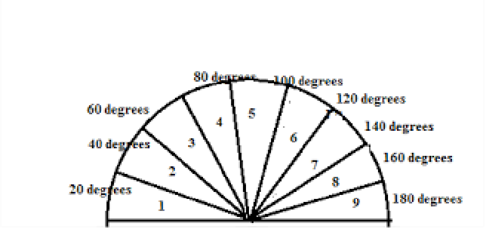

Figure 17 Figure 9 bins orientation of 0-180 °

Results of granting all voting in the form of a histogram of image training data contained in Figure 18 as follows:

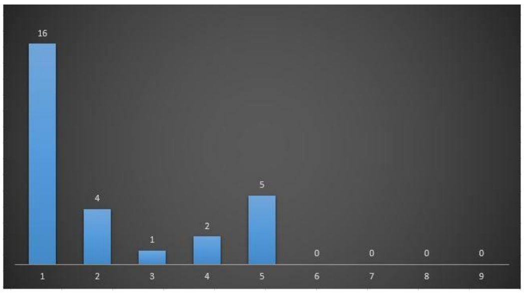

Figure 18 Histogram data training

d. *Normalize voting values for generating a descriptor*

The next step will be the calculation of the normalization of the block feature value for each gradient value magnitude in the previous process in spatial binning then the calculation formula is like equation 11 as follows:

$$norm = \frac{v(n)}{\sqrt{(\sum_{k=1}^{block*l} v(k)^2) + 1}}$$

$$= \frac{28 * (9)}{\sqrt{(\sum_{k=1}^{7*7*16} 28 * (1)^2) + 1}}$$

$$= \frac{252}{\sqrt{(\sum_{k=1}^{784} 28) + 1}}$$

$$= \frac{252}{\sqrt{(4506251681)}}$$

$$= \; 0.003754$$

Normalization of block feature values is carried out aimed at reducing the effect of changes in object brightness on each block.

12. Perform training

It is a process of training training data from the extraction of HOG features using SVM. The following is the result of the HOG vector feature from the training data that has been combined can be seen in Table 2 as follows:

Tabel 2 Results of the HOG vector feature training data images

0: 0.003754	1: 0.002716	2: 0.003051	3: 0.002171	4: 0.000834
5: 0.001855	6: 0.001639	7: 0.001461	8: 0.000664	9: 0.004589
10: 0.001506	11: 0.003736	12: 0.003374	...	7055: 0.006111

13. Create SVM training data

It is a process of storing the results of SVM training data which will be loaded when the *testing* phase begins.

14. Perform *testing*

Is a process of testing *real-time* images and matching with SVM training data.

15. Contains SVM training data

Is a process of loading SVM training data that has been previously made.

16. Predict labels

It is a SVM prediction process to predict the numbers that come out from entering *real-time* images during *testing* . as an example the test data that has found its *support vector* value (0.043258588, -4) will be entered into equation 12 as follows:

$$
\begin{aligned}
&Kelas\ x \\
&= \arg\max_{k=1..5}\left(\begin{bmatrix}336.4060992\\571.7978874\end{bmatrix}^T.\begin{bmatrix}0.043258588\\-4\end{bmatrix}\right. \\
&+ 83.96716895, \begin{bmatrix}7.264950745\\13.52758601\end{bmatrix}^T.\begin{bmatrix}0.043258588\\-4\end{bmatrix} \\
&+ 2.040385033, \begin{bmatrix}11.67556696\\19.36271229\end{bmatrix}^T.\begin{bmatrix}0.043258588\\-4\end{bmatrix} \\
&+ 2.988421626), \begin{bmatrix}54.91771231\\96.38651963\end{bmatrix}^T.\begin{bmatrix}0.043258588\\-4\end{bmatrix} \\
&+ 14.44890498), \begin{bmatrix}28.24582215\\48.26470044\end{bmatrix}^T.\begin{bmatrix}0.043258588\\-4\end{bmatrix} \\
&\left.+ 7.226943864\right)
\end{aligned}
$$

$$
\begin{aligned}
&Kelas\ x \\
&= \arg\max_{k=1..5} \begin{matrix}(-2188.671928, -51.7556875, \\ -73.957359 \\ , -368.7215109, -184.6099835)\end{matrix}
\end{aligned}
$$

$$Kelas\ x = -51.7556875$$

As can be seen in the equation above the value of the largest *hyperplane* is -51.7556875 which is where the *hyperplane* value index belongs to the class / label 2 that is for *the* mouse pointer's left-click *event*.

17. Label 1 to move the mouse pointer

A process event if the mouse pointer predictive value that comes out is 1.

18. Label 2 for a left click

A mouse pointer process event if the number of predictions that came out was 2.

19. Label 3 for right click

A process event if the mouse pointer predictive value that comes out is 3.

20. Label 4 to drag and drop

A process event if the mouse pointer predictive value that comes out is 4.

21. Label 5 for scrolling the mouse pointer

A process event if the mouse pointer prediction exit number is 5.

22. *Output* in the form of mouse pointer events

As described earlier, the output of the system is in the form of event mouse pointer mouse pointer movement, left click, right click, drag and drop and scrolling the mouse pointer.

2.10 Evaluation Testing System

In performance testing, it can be concluded that the SIFT-HOG + SVM method has different accuracy in each test scenario. Based on the results of the accuracy testing of all testing scenarios that have been carried out the results can be seen in Table 3 as follows:

Table 3 Results of accuracy of all test scenarios

Testing Scenario	SIFT-HOG+SVM Method
MOVE training data	80%
CLICK1 training data	90%
CLICK2 training data	70%
DRAG training data	100%
SCROLL training data	50%
Average accuracy	**78%**

From these data it can be concluded that the results of good handtracking with the average percentage of accuracy for all test scenarios are 78%.

3. CLOSING

3.1 Conclusion

Based on the results of research on the implementation of the SIFT-HOG algorithm and SVM in controlling the pointer, it can be concluded that:

1. Good *handtracking* results with an average percentage of accuracy for all test scenarios that is 78%.

3.2 Suggestion

Besides the advantages of the system there would be a shortage, therefore the future development of this research is the advice - the advice that can be given as follows:

1. Using the method of ICP (Iterative Closest Point) to perform handtracking process.
2. Using the method of CNN (Convolutional Neural Network) for the classification of a hand gesture in order to achieve high accuracy.
3. Creating uniqe or different each - each dataset of hand gesture that does not degrade the accuracy of the results.

BIBLIOGRAPHY

[1] R. Klette, Concise Computer Vision, New York: Springer, 2014.

[2] VH Amiraj Dhawan, "Implementation of Hand Detection based Techniques for Human Computer Interaction," International Journal of Computer Applications, vol. 72, 2013.

[3] DM Iqbal, "Computer Vision Sensing Visual for variety," Foshema Research Group, Depok, 2016.

[4] H. charisma, "Implementation Handtracking In Control Mouse Pointer Using Pyramidal Lucas-Kanade algorithm," Thesis, 2011.

[5] C.-c. wang and K.-. C.-. Wang, "Hand Posture Recognition Using Adaboost with SIFT for Human Robot Interaction," Graduate Institute of Networking and Multimedia.

[6] Q. J. Gao and Cao, "Adaptive HOG-LBP Based Learning for Palm Tracking," International Conference on Computer and Information Application, 2012.

[7] FJ Ansari, "Hand Gesture Recognition using fusion of SIFT and HOG with SVM as a classifier," International Journal of Engineering Technology Science and Research, vol. 4, no. 9, 2017.

[8] M. Nazir, Methods 2009.

[9] R. Munir, digital image processing with algorithmic approach 2004.

[10] A. Fauzan, "Color Space Hue Saturation Value (HSV) and The conversion process," January 3, 2015. [Online]. Available:http://www.charisfauzan.net/2015/01/ruang-warna-hue-saturation-value-hsv.html, [Accessed 22 April 2018].

[11] AS Abdul Kadir, Theory and Application of Image Processing, 2013.

[12] BT Navneet Dalal, "Histograms of Oriented Gradients for Human Detection," 2005. [Online] .Available:https://lear.inrialpes.fr/people/triggs/pubs/Dalal-cvpr05.pdf [Accessed 16 April 2018].

[13] MIT Taichiro Tokumori, "Histograms of Oriented Gradients (HOG)," University of the Ryukyus, Japan.

[14] K. Sembiring, "Application of Support Vector Machine Technique for Intrusion Detection in Networks," Journal of the ITB 2007.

[15] PA Octaviani, "Application of Classification Method of Support Vector Machine (SVM) on Accreditation Data Elementary School (SD) in Magelang District," Diponegoro University, Semarang, 2014.