Matrix: Jurnal Manajemen Teknologi dan Informatika
Volume 12 Issue 1 Year 2022 Pages 7-18

DOI: http://dx.doi.org/10.31940/matrix.v12i1.7-18
URL: http://ojs2.pnb.ac.id/index.php/matrix

PowerPoint slideshow navigation control with hand gestures using Hidden Markov Model method

Cahya Rahmad [1]*, Arief Prasetyo [2], Riza Awwalul Baqy [3]

[1]*Department of Information Technology, Politeknik Negeri Malang, Indonesia*

*Corresponding Author: *cahya.rahmad@polinema.ac.id*

Abstract: Gesture is the easiest and most expressive way of communication between humans and computers, especially gestures that focus on hand and facial movements. Users can use simple gestures to communicate their ideas with a computer without interacting physically. One form of communication between users and machines is in the teaching and learning process in college. One of them is the way the speakers deliver material in the classroom. Most speakers nowadays make use of projectors that project PowerPoint slides from a connected laptop. In running the presentation, the speaker needs to move a slide from one slide to the next or to the previous slide. Therefore, a hand gesture recognition system is needed so it can implement the above interactions. In this study, a PowerPoint navigation control system was built. Digital imaging techniques use a combination of methods. The YCbCr threshold method is used to detect skin color. Furthermore, the morphological method is used to refine the detection results. Then the background subtraction method is used to detect moving objects. The classification method uses the Hidden Markov Model (HMM). With 526 hand images, the result shows that the accuracy of the confusion matrix is 74.5% and the sensitivity is 76.47%. From the accuracy and sensitivity values, it can be concluded that the Hidden Markov Model method can detect gestures quite well as a PowerPoint slide navigation control.

Keywords: background subtraction, hand gesture, hidden Markov model, slide powerPoint, YCbCr

History Article: Submitted 10 January 2022 | Revised 18 January 2022 | Accepted 9 March 2022

How to Cite: C. Rahmad, A. Prasetyo, and R. A. Baqy, "PowerPoint slideshow navigation control with hand gestures using Hidden Markov Model method," *Matrix: Jurnal Manajemen Teknologi dan Informatika*, vol. 12, no 1, pp. 7-18, 2022

Introduction

Gestures are the easiest and most expressive communication methods between humans and computers—especially gestures that focus on hand and facial movements. Users can use simple gestures to communicate their ideas with a computer without physically interacting. Techniques such as gesture recognition systems will capture the user's ideas without physical contact and increase the bond between the user and the machine. One form of application between users and machines is in college's teaching and learning process [1].

Along with the development of technology, the teaching and learning process in college will also develop. One of them is the way the speaker delivers material in the classroom. Most presenters nowadays make use of projectors that project PowerPoint slides from a connected laptop. When running his presentation, the speaker needs to slide from one slide to the next or the previous slide. To achieve this, ordinary presenters use several methods, by moving themselves, with the help of operators, or with tools. If the speaker moves the slide by himself, the speaker must walk closer to his laptop and press the keyboard button to operate it. Meanwhile, when using operator assistance, other people's help is needed, and misunderstandings often occur between presenters and operators. And finally, with tools. This tool is shaped like a pen that has a forward and backward button. The problem that often arises with this tool is battery life and the tool suddenly not running. Another method is needed for the speaker to interact with the hardware or laptop.

In the context of Human-Computer Interaction (HCI), gestures can be used to interact with machines. The hand gesture system is fascinating today because of its ease as a form of interaction carried out by humans. Gesture detection or hand movement can be done with a camera/webcam installed on a laptop [2].

The method used is the Hidden Markov Model (HMM). The HMM method is the most widely used hand gesture recognition method. HMM is a statistical model in which the system being modeled is assumed to be a Markov process with unknown parameters. HMM represents the statistical behavior of a sequence of symbols that can be observed using a hidden state network with transition and emission probabilities. HMM can be used for pattern recognition once hidden parameters are identified using observable data [3]. Based on the explanation above, a hand motion detection application will be made using the Hidden Markov Model method for presenters in navigating PowerPoint slides. So, the presenters do not need to face the problems previously mentioned.

Previous research was conducted by Ying Yin and Randall Davis with the title Real-Time Continuous Gesture Recognition for Natural Human-Computer Interaction. This research was conducted using the Hidden Markov Model method. The Hidden Markov Model method is used to identify human poses [4]. Yanmin Zhu and Bo Yuan with the title Real-Time Hand Gesture Recognition with Kinect for Playing Racing Video Games. This research was conducted using the Hidden Markov Model method. The Hidden Markov Model method is used to control the game Need for Speed [5]. Aakash Anuj, Tanwi Mallick, Partha Pratim Das, and Arun Kumar Majumdar with the title Robust Control of Applications by Hand-Gestures. This research was conducted using the Hid-den Markov Model method and Kinect sensor. The Hidden Markov Model method is used to navigate PowerPoint [2]. Joko Sutopo, Mohd Khanapi Abd Ghani, M.A. Burhanuddin, and Zulwati with the title Gesture Recognition of Dance Chain Code and Hidden Markov Model. This research was conducted using the Hidden Markov Model method and the Chain Code feature extraction. The method is used to classify dance movements [6]. Roberto Pangihutan Situmeang conducted previous research with the title Implementation of the Hidden Markov Model Algorithm for Facial Gesture Recognition. This research was conducted using the Haar Cascade and Hidden Markov Model methods to classify facial cues [7]. Based on the research that has been done above, it will be possible to create a system that can combine hand gestures with the Hidden Markov Model meth-od to create a system that can replace presentation slides on PowerPoint.

Methodology

This study uses a dataset from the Cambridge Hand Gesture Dataset [8], which provides a collection of hand images intending to foster the development of hand pose recognition research. This dataset consists of 900 series of pictures from 9 gesture classes, consisting of 3 simple hand shapes and 3 simple movements. The dataset also has 5 different types of lighting. The dataset is divided into 3 labels based on the formed hand grapples. F label for hands with tight fingers (Flat), S label for hands with open fingers (Spread), and V label for hands with fingers forming the letter V, just like shown in Figure 1.

Figure 1. Gesture F, gesture S and gesture V

In designing this research, a soft device development method is used, namely the spiral model. The Spiral Model is an SDLC model, which combines architecture and prototyping in stages. This model combines the SDLC Waterfall and Iterative models with a significant emphasis on risk analysis. The main problem with the spiral model is determining the right time to make a move to the next stage. The shift to the next stage is still carried out as planned, even though the work in the previous stage has not been completed. The plan is prepared based on statistical data, which has been received during previous work or even from the experience of a personal developer [9]. First threshold Iteration: Looks for a threshold value. The threshold is used to detect skin color. Different types of thresholds will be compared with each other and tested under certain conditions. The Second is Morphological Iteration: Looking for the right morphological method. Morphology is used to improve the threshold results. Various types of morphology will be compared with each other and tested under certain conditions. The Third is Dataset Iteration: Testing the dataset with the previous methods. The dataset will be separated into test data and training data. Then tested the accuracy with the HMM method. The Fourth is Hand Detection Iteration: Determine hand position by filtering and classifying blob. Test the dataset with real-time imagery. Adding a background subtraction method together with the threshold method. The Fifth is Motion Detection Iteration: Determines the centroid point to detect the user's hand movement. The difference between the points can command the slide to shift. The initial stage of each iteration is a literature study and application analysis based on previous work. The final stage of each iteration is to optimize the application to be lighter and more responsive.

The data is separated into test data and training data. Each data will be pre-processed by looking for a threshold value. The threshold value is used to separate the background color and foreground color which in this study is used to detect skin color [10]. In this study, the value of 64 is used. Different types of threshold will be compared with each other and tested under certain conditions [9]. In the Table 1, the authors compare various color spaces with their respective limits to obtain 3 threshold search methods.

Table 1. Threshold methods

No	Color Space	Threshold Value	Average computation time
1	RGB [11]	R > 95; G > 40; B > 20; Max(R,G,B) - Min(R,G,B) > 15; |R-G| > 15; R > G; R > B;	21,3117 ms / frame
2	HSV [12]	(0 < H < 0,24 || 0,74 > H > 1); 0,16 > S > 0,79;	44,1064 ms / frame
3	YCbCr [13]	78 <= Cb <= 126; 132 <= Cr <= 172;	18,9543 ms / frame

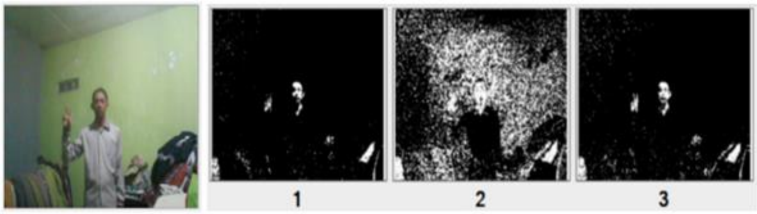


Figure 2. Threshold methods

Based on experiments with various color spaces and threshold values above in Figure 2, the authors decided to use method number 3, namely the YCbCr color space with a threshold value where the Cb value is between 78 to 126 and the Cr value between 132 to 172. The method is chosen based on the resulting image of the threshold that can capture skin color well and has fast computation time.

The next preprocessing is finding the right morphological method. Morphology is used to improve the threshold results. Various types of morphology will be compared with each other and tested under certain conditions. Morphological operations are performed on binary images to remove background noise in the image. The morphological operations used include dilation and then erosion. Thus, only hands that are colored skin are formed from the threshold result.

Then the median filtering was also carried out. The median filter replaces each pixel with a median or "middle value" pixel in the rectangular environment around the center pixel. The pixels of the image border is not changed so that the edge values can be preserved. In a binary image, because the value in the image consists of 2 values, '0' and '1', the more values are automatically taken as the middle value. Since unwanted noise in binary images appears in the form of tiny white dots on a black background, these white noise pixels will be removed with a median filter, and since the edges are preserved, the contours of the hand are mostly obtained. Thus, facilitating the calculation of the geometry of the hand and radius [14]. In looking for morphological methods, the authors compare various morphological methods to obtain 3 morphological methods.

Table 2. Morphological methods

No	Morphological Method	Information
1	Dilation and erosion [15]	Lots of noise, thin gestures
2	Dilation, erosion, and fill holes [16]	Lots of noise, a form of broken gesture
3	Dilation, erosion and filter median [14]	Little noise, medium gesture form

Based on the experiment with the 3 methods above in Table 2, the authors decided to use method number 3 by using dilation morphology, erosion, and median filter. The choice of method was decided based on the results of the threshold of method 3, which is cleaner with less noise (white spots) and medium hand gesture form. not thin and not dashed

After the dataset is preprocessed using the threshold method and the morphology that had previously been determined. Accord.Net has provided a function to detect a collection of white dots that form a blob, like in Figure 3 below. From the collection of blobs obtained, only the largest blob is extracted as a hand shape. The extraction result is in the form of a binary image.

Figure 3. Binary image

During the pre-processing to blob extraction, researchers realized that not all images were processed/extracted properly. So the researchers decided to sort out the images that could be processed well with those that did not. From 900 data, 526 data were obtained, which could be processed and extracted the shape of the hand. Then each hand shape is divided as training data and test data. 17 test data were taken for each hand shape. Then obtained 475 data as training data and 51 data as test data. The obtained binary image is resized to be smaller. The size of the resize will affect the accuracy of the classification. Then the image is converted into a 1D matrix form. 1D matrix form can be seen in Figure 4.

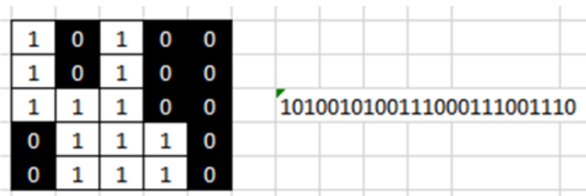


Figure 4. One dimensional matrix

The series of numbers formed will be used in classification using the Hidden Markov Model (HMM). The learning process on HMM can be done repeatedly according to the wishes of the re-searcher, which will also affect accuracy. The HMM learning process of training data is carried out every time the application runs for the first time. So that the determination of the size of the resize and the number of iterations greatly affects the speed of the application when it is first to run.

The research process endeavor is the dataset is taken from the "Cambridge Hand Data Set." Then the data is separated into test data and training data. Each data will be pre-processed and the same feature extraction. HMM will classify the results of the extraction of both. The classification results will be used as an assessment of classification accuracy. Second, the data is retrieved from the laptop webcam. Then do pre-processing and feature extraction as before. The extraction results will be classified with the extraction results from the previous training data. The classification results will determine whether the PowerPoint slides are shifted or not. More or less, the process flow can be seen in Figure 5.

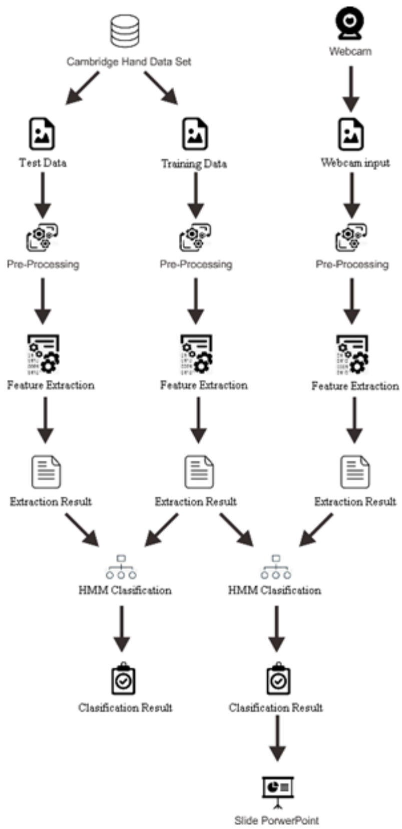


Figure 5. Initiative process

Results and Discussions

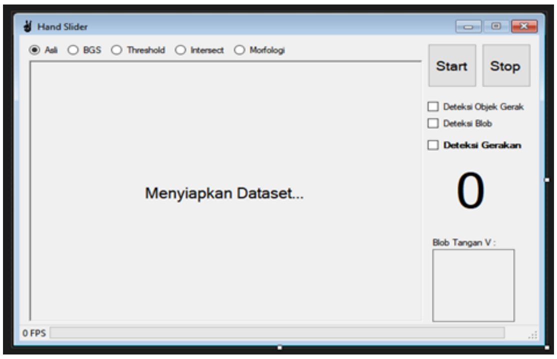


Figure 6. User interface

The background subtraction method is applied to distinguish the background and moving objects. In simple terms, this method works by parsing pixels from the background image and pixels from the foreground image. The background image and the foreground image are converted into a grayscale image, then look for the difference in each pixel of the two images. If the difference in the value of the two pixels exceeds the predetermined value (in this study, the value 64 is used), then the pixel is considered part of a moving object. This method can help reduce complex backgrounds.

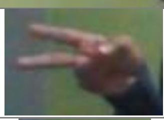
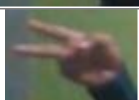
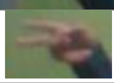
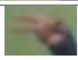
Figure 7. Background subtraction method

The resulting image from the background subtraction in Figure 7 will then be combined with the threshold method. Merging is done by the AND operation. Only white pixels from both the image, the background image, and the foreground image will be displayed as merged pixels. So that the image that is formed is an image which is a combination of white pixels from the two images. Result can be seen in Figure 8.

Figure 8. Combining background subtraction and threshold methods. Background subtraction (A), Threshold (B), Combining result (C)

After being combined, the images are subjected to a morphological process. Detecting the hand's position is done in two ways, namely filtering the blobs and classifying the blobs. The blob is filtered by limiting the width of the blob and the length to width ratio of the blob.

Table 3. Hand distance measurement with camera

Distance	Hand Image	Hand *Blob*
30 cm		
50 cm		
100 cm		
150 cm		
200 cm		
250 cm		

Based on Table 3 above, the hand gesture blob at a distance of 30 cm is not detected properly where the palm part is not fully detected. At a distance of 50 cm to 150 cm, blobs of hand gestures and fingers are well-formed. Meanwhile, at a distance of 200 cm to 250 cm, the fingers begin not to form properly. Therefore, the hand distance to the camera is limited to approximately 50 cm to 150 cm. The detection distance is measured based on the size of the hand blob obtained. At a distance of 50 cm, the blob size is defined as a minimum height of 15 pixels and a minimum width of 35 pixels. At a distance of 150 cm, the blob size is determined with a maximum height of 65 pixels and a maximum width of 125 pixels. The blob ratio will also be measured so that only the horizontal blob is processed. The blob ratio is obtained by dividing the width of the blob and the height of the blob. A blob with a ratio value between 1.25 and 3.0 is selected, which will then be classified.

Then the blob is classified by the HMM method. The Hidden Markov Model, better known as the Hidden Markov Model (HMM), is a statistical model of a system assumed to be a Markov process with unknown parameters. The hidden Markov model has been known to use reinforcement learning and recognition of temporary strands such as words, handwriting, gestures, music, and bioinformatics [17]. In the Hidden Markov Model, a general model is used in modeling the problem. Equation 1 is a general model used in the Hidden Markov Model.

$$\lambda = (A, B, \pi) \tag{1}$$

There is a symbol of lamda (λ) as the Markov model, A as the transition probability, B is the probability of observation, and the symbol phi (π) is the probability of the initial state [18]. Three fundamental problems must characterize the Hidden Markov Model: Problem 1 (likelihood): Determine the proximity value P (O | λ), from HMM λ = (A, B) and the series of observations O. Problem 2 (Decoding): Determining the best-hidden state sequence, from HMM λ = (A, B) and a series of observations O. Problem 3 (Learning): Determine the parameters of HMM A and B from a set of HMM states and a series of observations O.

Each classified blob will form a bounding box around it with a different color, presented in Figure 9. Label F will have a bounding box in red, label S in green and label V in blue. The classified blob with the label V shall be considered a hand.

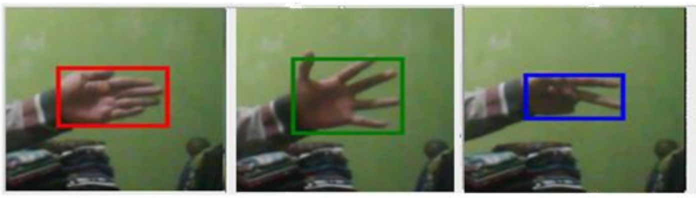


Figure 9. Blob detection

After getting the hand position, the hand centroid point is obtained by looking for the center of gravity point from the Accord.NET library. The point is then stored in the variable to compare the difference with the centroid point of the next hand. This center of gravity will also determine which hand the user is pointing at. The result of hand detection is shown in Figure 10. The application can detect movement to the right, left, up, and down so that the coordinate points on the X and Y axes will be saved as variables. Each part of the hand is only allowed to perform one specific movement. For example, if the right hand is detected, the only movement to the left is vice versa. The general movements are up and down movements, and individual parts of the hand can perform general movements.

Then the next new frame will also detect the position of the hand and the coordinates of X and Y and stored them in the new variable. Next, we will look for the difference between the old X or Y values. The difference limit needs to be set to reduce the sensitivity of detection that is too fast and avoid the detection of other blobs that are too far from the starting point. The difference between the coordinates of the specific motion required to be considered moving is if the point has a value between half the width of the blob and 2 times the width of the blob. Meanwhile, the general movement is if the point has a value between half of the blob's height and 2 times the blob's height.

If the difference between the new X value and the old X is between the difference margin, then the application gives a command to the right or left to the program depending on the direction of the hand movement. If the hand moves to the left, the application sends the command swipe to the next slide—Vice versa. And if the hand moves up, the application sends a command to go to the initial slide. Hands down, it means going to the final slide. The result of movement detection is shown in Figure 11.

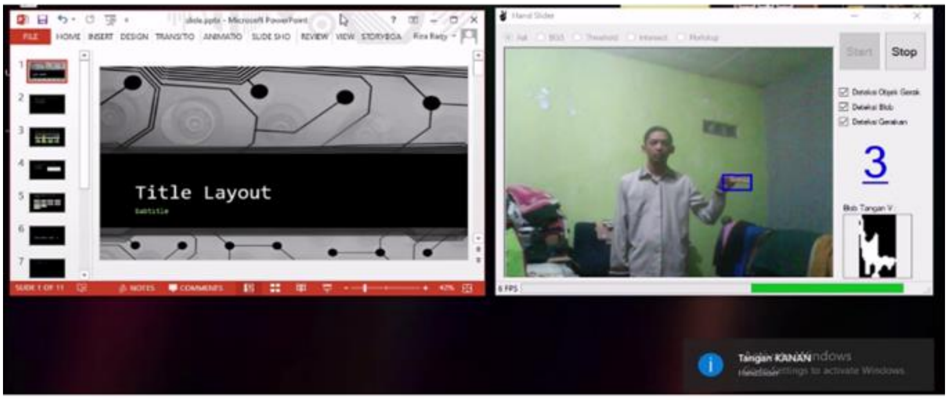


Figure 10. Hand detection

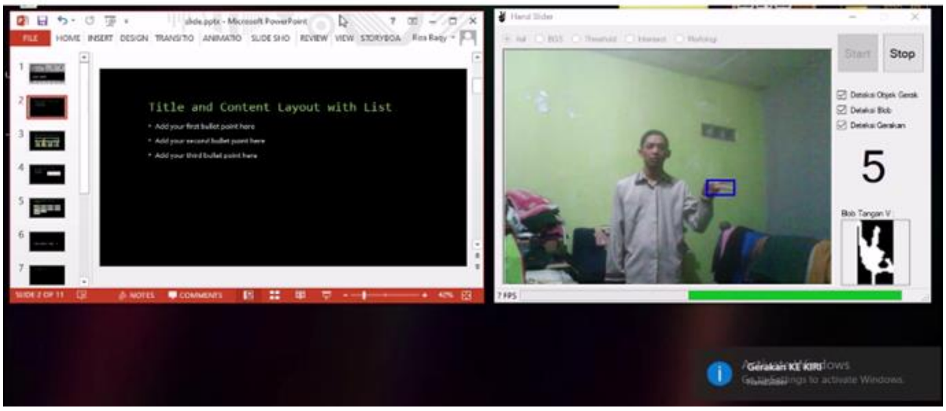


Figure 11. Movement detection

The application has provided a countdown time and notification for the user to determine how long it will take until a motion is detected. A waiting time is provided before making the next movement. Like the difference limit, the waiting time is earmarked to reduce detection errors due to the fast detection process. The duration of time until motion is detected is 5 seconds. The waiting duration is 2.5 seconds if motion is not detected and 5 seconds when motion is detected. A notification will also be displayed intended when the slideshow is displayed on full screen. One form of error detection is when the user actively uses his hands. It is hoped that users will keep a sufficient distance from the camera's detection range so that the camera does not easily detect the user's hand movements. Or by pointing the camera in another direction where the user is not always visible in the camera's distance.

Accuracy testing in Table 4 was carried out using 475 training data and 51 test data. Testing is carried out with the help of applications that are devoted to conducting training and testing. The test will focus on one of the hand labels, namely the V label. There are 2 test parameters, namely the resize blob size and the learning iteration which are shown in Table 5. The resize sizes tested were 5x5, 10x10, 15x15, 20x20, and 25x25 with normal iterations (100 times).

Table 1. Accuracy test results

	Total			F			S			V		
Size	**T**	**F**	**%**	**T**	**F**	**%**	**T**	**F**	**%**	**T**	**F**	**%**
5	28	23	54,90196	11	6	64,70588	10	7	58,82353	7	10	41,17647
10	26	25	50,98039	13	4	76,47059	6	11	35,29412	7	10	41,17647
15	26	25	50,98039	10	7	58,82353	5	12	29,41176	11	6	64,70588
20	30	21	58,82353	10	7	58,82353	7	10	41,17647	13	4	76,47059
25	26	25	50,98039	11	6	64,70588	2	15	11,76471	13	4	76,47059
30	25	26	49,01961	11	6	64,70588	2	15	11,76471	12	5	70,58824
35	25	26	49,01961	11	6	64,70588	2	15	11,76471	12	5	70,58824
40	28	23	54,90196	11	6	64,70588	4	13	23,52941	13	4	76,47059
45	22	29	43,13725	12	5	70,58824	10	7	58,82353	0	17	0
50	17	34	33,33333	17	0	100	0	17	0	0	17	0
100	26	25	50,98039	11	6	64,70588	3	14	17,64706	12	5	70,58824
125	26	25	50,98039	11	6	64,70588	3	14	17,64706	12	5	70,58824

Table 5. Iteration accuracy test results

Iteration	**Total**			**F**			**S**			**V**		
	T	**F**	**%**	**T**	**F**	**%**	**T**	**F**	**%**	**T**	**F**	**%**
50	23	28	45,09804	13	4	76,47059	10	7	58,82353	0	17	0
100	30	21	58,82353	10	7	58,82353	7	10	41,17647	13	4	76,47059
150	30	21	58,82353	10	7	58,82353	7	10	41,17647	13	4	76,47059
200	30	21	58,82353	10	7	58,82353	7	10	41,17647	13	4	76,47059
250	30	21	58,82353	10	7	58,82353	7	10	41,17647	13	4	76,47059

Based on the test results in Table 4, resize with sizes 20x20, 25x25, and 40x40 have a high accuracy value on the V label, 76.47%. But the author will use a size of 20x20 to save and speed up the process of training and testing the application. Then iteration parameter testing is carried out. The number of iterations tested was 50, 100, 150, 200, and 250 times. Based on the test results in Table 5, iterations above 100 times have a high accuracy value on the V label, 76.47%. But the author will use iterations 100 times to save and speed up the process of training and testing the application.

In evaluating the performance of the algorithm, you can use a confusion matrix reference. The Confusion Matrix represents the prediction and actual (actual) conditions of the data generated by the algorithm. We can determine accuracy, precision, sensitivity, and specifications [19]. There are 4 terms as a representation of the result of the classification process on the confusion matrix. The four terms are True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN).

Table 6. Confusion matrix

		Prediction		
Actual		F	S	V
	F	10	4	3
	S	4	7	6
	V	2	2	13

Based on Table 6, it can be determined the calculation of accuracy, precision, and sensitivity to the V label. Label V is positive data while label F and label S are negative data. Results should be clear and concise. The results should summarize (scientific) findings rather than provide data in great detail.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} = \frac{13 + (10 + 4 + 4 + 7)}{10 + 4 + 3 + 4 + 7 + 6 + 2 + 2 + 13} = \frac{38}{51} = \mathbf{74,5}\%$$

$$Sensitivity = \frac{TP}{TP + FN} = \frac{13}{13 + (2 + 2)} = \frac{13}{17} = \mathbf{76,47}\%$$

The efforts to improve accuracy were made by reducing the dataset to 330 datasets, consisting of 300 training data and 30 test data. Reduced data is ambiguous data. For example, the image resulting from the threshold label F resembles the image resulting from the threshold label V. Thus, the data should be eliminated.

Table 7. Confusion matrix improved accuracy

		Prediction		
		F	S	V
Actual	F	7	3	0
	S	0	9	1
	V	1	3	6

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} = \frac{6 + (7 + 3 + 0 + 9)}{7 + 3 + 0 + 0 + 9 + 1 + 1 + 3 + 6} = \frac{25}{30} = \mathbf{83,33}\%$$

$$Sensitivity = \frac{TP}{TP + FN} = \frac{6}{6 + (1 + 3)} = \frac{6}{10} = \mathbf{60}\%$$

Based on Table 7, the elimination resulted in an increase in the accuracy value from 74.5% to 83.33%. But the detection of the V label is getting worse. This is due to the decrease in the sensitivity value from 76.47% to 60%. So that researchers continue to use the initial dataset with a sensitivity of 76.47% in the application.

Motion detection testing was carried out with 3 different subjects (S1-S2-S3) by counting the hand movements needed to shift the entire slide. A total of 5 slides (5 moves) will be tested to detect left and right movements. And for the up and down movements are also 5 movements using the right hand.

Table 8. The test results shifted the slides

Movement Direction	Number of Moves			Minimum Movements	Maximum Movements
	S1	S2	S3		
Left	5	8	10	1	5
Right	10	10	7	1	4
Up	9	6	7	1	3
Down	6	6	8	1	3

Based on the test results in Table 8, the slides can be shifted by performing one motion. The difference in the number of movements is influenced by several factors, which are the lighting position and movement speed. The light source position can affect the skin tone detection received by the camera and can lead to false detection or detection failure. Movement speed that is too fast will cause objects to look blurry, which will result in failed detection.

Conclusion

The result shows that the accuracy of the confusion matrix is 74.5%, and the sensitivity is 76.47%. From the accuracy and sensitivity values, it can be concluded that the Hidden Markov Model method can detect gestures quite well as a PowerPoint slide navigation control.

Acknowledgments

The researcher would like to express his deepest gratitude to every contributor who has helped in completing this research.

References

[1] R. E. Sibai, C. A. Jaoude, and J. Demerjian, "A new robust approach for real-time hand detection and gesture recognition," *Int. Conf. Comput. Appl. ICCA 2017*, pp. 18–25, 2017, doi: 10.1109/COMAPP.2017.8079780.

[2] A. Anuj, T. Mallick, P. P. Das, and A. K. Majumdar, "Robust control of applications by hand-gestures," *5th Natl. Conf. Comput. Vision, Pattern Recognition, Image Process. Graph. NCVPRIPG 2015*, 2016, doi: 10.1109/NCVPRIPG.2015.7490031.

[3] P. K. Pisharady and M. Saerbeck, "Recent methods and databases in vision-based hand gesture recognition: A review," *Comput. Vis. Image Underst.*, vol. 141, pp. 152–165, 2015, doi: 10.1016/j.cviu.2015.08.004.

[4] Y. Yin and R. Davis, "Real-time continuous gesture recognition for natural human-computer interaction," *Proc. IEEE Symp. Vis. Lang. Human-Centric Comput. VL/HCC*, pp. 113–120, 2014, doi: 10.1109/VLHCC.2014.6883032.

[5] Y. Zhu and B. Yuan, "Real-time hand gesture recognition with Kinect for playing racing video games," *Proc. Int. Jt. Conf. Neural Networks*, pp. 3240–3246, 2014, doi: 10.1109/IJCNN.2014.6889481.

[6] J. Sutopo, M. K. A. Ghani, M. A. Burhanuddin, and Zulhawati, "Gesture recognition of dance using chain code and hidden Markov model," vol. 8, no. 3, pp. 195–200, 2019.

[7] R. P. Situmeang, *Implementasi Algoritma Hidden Markov Model Untuk Pengenalan Isyarat Wajah*. Bandung: Universitas Komputer Indonesia, 2017.

[8] T. K. Kim, S. F. Wong, and R. Cipolla, "Tensor canonical correlation analysis for action classification," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2007, doi: 10.1109/CVPR.2007.383137.

[9] V. Osetskyi, "SDLC models explained: Agile, waterfall, v-shaped, iterative, spiral," Aug. 29, 2017. https://medium.com/existek/sdlc-models-explained-agile-waterfall-v-shaped-iterative-spiral-e3f012f390c5 (accessed May 21, 2020).

[10] R. A. Elsayed, M. S. Sayed, and M. I. Abdalla, "Skin-based adaptive background subtraction for hand gesture segmentation," *Proc. IEEE Int. Conf. Electron. Circuits, Syst.*, vol. 2016-March, pp. 33–36, 2016, doi: 10.1109/ICECS.2015.7440242.

[11] F. E. Sandnes, L. Neyse, and Y. P. Huang, "Simple and practical skin detection with static RGB-color lookup tables: A visualization-based study," *IEEE Int. Conf. Syst. Man, Cybern. SMC 2016 - Conf. Proc.*, pp. 2370–2375, 2017, doi: 10.1109/SMC.2016.7844593.

[12] A. Bapat and V. Kanhangad, "Segmentation of hand from cluttered backgrounds for hand geometry biometrics," *TENSYMP 2017 - IEEE Int. Symp. Technol. Smart Cities*, pp. 8–11, 2017, doi: 10.1109/TENCONSpring.2017.8070016.

[13] U. Santa, F. Tazreen, and S. A. Chowdhury, "Bangladeshi hand sign language recognition from video," *20th Int. Conf. Comput. Inf. Technol. ICCIT 2017*, pp. 1–4, 2018, doi: 10.1109/ICCITECHN.2017.8281818.

[14] S. S. Kakkoth and S. Gharge, "Visual descriptors based real time hand gesture recognition," *Int. Conf. Adv. Commun. Comput. Technol. ICACCT 2018*, pp. 361–367, 2018, doi: 10.1109/ICACCT.2018.8529663.

[15] M. A. Rahaman, M. Jasim, T. Zhang, M. H. Ali, and M. Hasanuzzaman, "Real-time Bengali and Chinese numeral signs recognition using contour matching," *IEEE Int. Conf. Robot. Biomimetics, IEEE-ROBIO 2015*, pp. 1215–1220, 2015, doi: 10.1109/ROBIO.2015.7418937.

[16] M. M. Hasan, M. Khaliluzzaman, S. A. Himel, and R. T. Chowdhury, "Hand sign language

recognition for Bangla alphabet based on Freeman Chain Code and ANN," *4th Int. Conf. Adv. Electr. Eng. ICAEE 2017*, pp. 749–753, 2017, doi: 10.1109/ICAEE.2017.8255454.

[17] Eugine Kang, "Hidden Markov Model - Eugine Kang - Medium," Sep. 01, 2017. https://medium.com/@kangeugine/hidden-markov-model-7681c22f5b9 (accessed May 18, 2020).

[18] A. Setiyoaji, L. Muflikhah, and M. A. Fauzi, "Named entity recognition menggunakan hidden markov model dan algoritma viterbi pada teks tanaman obat," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 1, no. 12, pp. 1858–1864, 2017, [Online]. Available: http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/673.

[19] Resika Arthana, "Mengenal accuracy, precision, recall dan specificity serta yang diprioritaskan dalam machine learning | by resika arthana | medium," Apr. 05, 2019. https://medium.com/@rey1024/mengenal-accuracy-precission-recall-dan-specificity-serta-yang-diprioritaskan-b79ff4d77de8 (accessed Aug. 01, 2020)