Applying Hand Gesture Recognition for User Guide Application Using MediaPipe

Indriani[1] Moh.Harris[1] Ali Suryaperdana Agoes[1,*]

[1] *Department of Informatics Engineering, STMIK AMIKBANDUNG, Bandung, Indonesia.*
[*] *Corresponding author. Email: ali@stmik-amikbandung.ac.id*

ABSTRACT

Hand gesture recognition is considered important with development technology in industry 4.0 in Human-Computer-Interactions (HCI) which gives computers the competence to capture and interpret hand gestures the executing command without touching devices physically. The MediaPipe is present as a framework built-in machine learning that has a solution for a hand gesture recognition system. In this research, we develop a simple user guide application using the MediaPipe framework. The user guide is commonly known as documentation about technical communication or a manual in a certain system to assist people. The user guide has step-by-step descriptions about handling a particular system and helps the user deal with user frustration by giving them the means to be identified, understand, and disentangle technical problems that frequently occurred by themselves. In our experiment, we captured a real-time image using Kinect, then trained a variety of hand gesture data, identified each hand gesture, and recognized hand gestures to convey information based on hand gestures in the system user guide application. The user can archive information user guide based on hand gestures that have been recognized. We proposed using hand gesture recognition using MediaPipe in our application to improve the convenience of utilization the user guide application and change user guide application that is still manual become a more interactive application.

Keywords: *Hand Gesture Recognition, MediaPipe, Kinect, User Guide Application.*

1. INTRODUCTION

We are now in an era of industry 4.0 or the Fourth Industrial Revolution which requires automation and computerized that are realized from the consolidation between various physical and digital technologies such as sensors, embedded systems, Artificial Intelligence (AI), Cloud Computing, Big Data, Adaptive Robotic, Augmented Reality, Additive Manufacturing (AM), and Internet of Things (IoT). [1]. The enhanced digital technology connectivity made technology a crucial requirement in carrying out our daily activities like doing tasks or work, shopping, communication, entertainment, and even searching for information or news [2]. The technology works more using the machines and advances in interaction with using a broad range of gestures to recognize, communicate, or interact with each other.

The gesture is known as a form of non-verbal communication or non-vocal communication where utilize of the body's movement that can convey a particular message originating from parts of the human body, the hand or face are the most commonly adopt [3]. Gesture-based interaction introduced by Krueger as a new type of Human-Computer Interaction (HCI) in the middle 1970s has become a magnetic area of the research. In the Human-Computer-Interaction (HCI), building interfaces of applications with managing each part of the human body to communicate naturally are the great attention to do research, especially the hands as the most effective-alternative for the interaction tool, considering their ability [4].

Through Human-Computer-Interaction (HCI), recognizing hand gestures could help achieve the ease and naturalness desired [5]. When interacting with other people, hand movements have the meaning to convey something with its information. Ranging from simple hand movements to more complex ones. For example, we can use our hand to point something (object or people) or use different simple shapes of hand or hand movements expressed through manual articulations combined with their grammar and lexicon as well-known as sign languages. Hence, using hand gestures as a device then integration with computers can help people communicate more intuitively [5].

Currently, many frameworks or library machine learning for hand gesture recognition have been built to make it easier for anyone to build AI (Artificial Intelligence) based applications. One of them is MediaPipe. The MediaPipe framework is present by Google for solving the problem using machine learning such as Face Detection, Face Mesh, Iris, Hands, Pose, Holistic, Hair segmentation, Object detection, Box Tracking, Instant Motion Tracking, Objection, and KIFT. MediaPipe framework helps a developer focus on the algorithm and model development on the application, then support environment application through results reproducible across different devices and platforms which it is a few advantages of using features on the MediaPipe framework [6].

In this paper, we focus on developing a manual user guide application with improving architecture application by applying hand gesture recognition using the MediaPipe framework and camera of Kinect for capture hand pose to recognized. Using hand gesture recognition will improve our user guide application more interactive.

2. RELATED WORK

2.1 Hand Gesture Recognition

Gesture recognition is an essential topic in computer science and builds technology that aims to interpret human gestures where anyone can use simple gestures to interact with the device without touching them directly. The entire procedure of tracking gestures to their representation and converting them to some purposeful command is known as gesture recognition [7]. Identify from explicit hand gestures as input then process these gestures representation for devices through mapping as output is the aim in hand gestures recognition.

Recognition of the hand gesture in kinds of literature based on extracted features is divided into three groups, as follows:

- *High-Level Features-Based Approaches:* Aim to figure out the position of the palm and joint angles such as the fingertips, joint location, or anchor points of the palm [8][9][10][11]. Whereas, effect collisions or occlusions on the image are difficult to detect after features are extracted [12], and sensitivity segmentation performance on 2D hand image [4] are the problem that occurred frequently. The gestures are defined from the results with a set of rules and conditions from the vectors and joints of the hands [13].
- *Low-Level Feature-Based Approaches:* Utilized these features for could be extracted quickly for robust to noise. Zhou [14] discovered recognition of the hand shape as a cluster-based signature using a novel distance metric called Finger Earth's Distance. Stanner [15] determines the bounding region of the hand elliptically for implement hand recognition based on principal axes. Yang [16] did research using the optical flow of the hand region as a low-level feature. Low-Level Feature-Based is not efficient when cluttered background [4].
- *3D Reconstruction-Based Approaches:* Use the 3D model of features for achieving the construe of hand completely. Research [17] showed that successfully segmenting the hand in skin color needs similarity and high contrast of the background related to the hand through structured light to bring in 3D of depth data. Another one [18] uses a stereo camera to track numerous interest points of the superficies of the hand which results in difficulty for handle robust 3D reconstruction, despite data contains 3D has valuable information that can help dispose of vagueness. See [19] [20] for more 3D reconstruction-based approach.

From kinds of literature, there are three Hand gesture recognition methods, as follow:

- *Machine Learning Approaches:* The resulting output came from the stochastic process and approach based on statistical modeling for dynamic gestures such as PCA, HMM [21][22][23][24], advanced particle filtering [26], and condensation algorithm [25].
- *Algorithm Approaches:* Collection of encoded conditions and restraints manually for defining as gestures in dynamic gestures. Galveia [27] applied a 3rd-degree polynomial equation to determine the dynamic component of the hand gestures (create a 3rd-degree polynomial equation, recognition, reduced complexity of equations, and comparison handling in gestures library).
- *Rule-based Approaches:* Suitable for dynamic gestures either static gestures which are contained a set of pre-encoded rules and features inputs [4]. The features of input gestures are extracted and compared to the encoding rules that are the flow of

the recognized gestures. Matching between gestures with rule and input which is outputted approved as known gestures [28].

2.2 MediaPipe Framework

Today, there are many frameworks or libraries of machine learning for hand gesture recognition. One of them is MediaPipe. The MediaPipe is a framework designed to implement production-ready machine learning that must build pipelines to perform inference over arbitrary sensory data, has published code accompanying research work, and build technology prototypes [6]. In MediaPipe, graph modular components come from a perception pipeline along with the function of inference model function, media processing model, and data transformations [29]. Graph of operations are used in others machine learning such as Tensor flow [30], MXNet [31], PyTorch[32], CNTK[33], OpenCV 4.0[34].

Using MediaPipe for hand gesture recognition has been researched by Zhang [35] before, using a single RGB camera for AR/VR application in a real-time system that predicts a hand skeleton of the human. We can develop a combined MediaPipe using other devices. The MediaPipe implements pipeline in Figure 1. consists of two models for hand gesture recognition as follows [29][35][36] :

1. A palm detector model processes the captured image and turns the image with an oriented bounding box of the hand,
2. A hand landmark model processes on cropped bounding box image and returns 3D hand key points on hand.
3. A gesture recognizer that classifies 3D hand key points then configuration them into a discrete set of gestures.

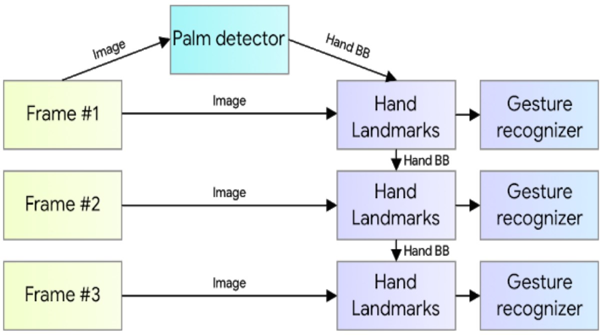


Figure 1 Hand Perception Pipeline Overview [36].

2.2.1 Palm Detector Model

MediaPipe framework has built detect initial palm detector called BlazePalm. Detecting the hand is a complex task. Step one is to train the palm instead of the hand detector, then using the non-maximum suppression algorithm on the palm, where it is modeled using square bounding boxes to avoid other aspect ratios and reducing the number of anchors by a factor of 3-5. Next, encoder-decoder of feature extraction that is used for bigger scene context-awareness even small objects, lastly, minimize the focal loss during training with support a large number of anchors resulting from the high scale variance [35] [36].

2.2.2 Hand Landmark

Achieves precise key point localization of 21 key points with a 3D hand-knuckle coordinate which is conducted inside the detected hand regions through regression which will produce the coordinate prediction directly which is a model of the hand landmark in MediaPipe [35][36]., see in Figure 2.

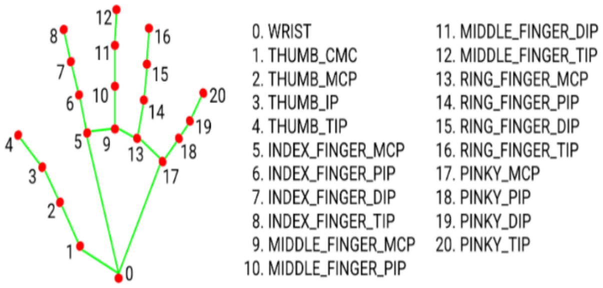


Figure 2 Hand Landmark in MediaPipe [38]

Each hand-knuckle of the landmark has coordinate is composed of x, y, and z where x and y are normalized to [0.0, 1.0] by image width and height, while z representation the depth of landmark. The depth of landmark that can be found at the wrist being the ancestor. The closed the landmark to the camera, the value becomes smaller.

2.2.3 Hand Recognizer

For recognizing hand gestures, the implementation of a simple algorithm is to compute gestures with a determined accumulated angle of the joint state or conditions each finger such as bent finger or straight finger then do map the set of finger states that we got before to set the label of pre-defined gestures like “OK”, “Spiderman”, “Rock” [35][36]. It can be seen in Figure 3 below.

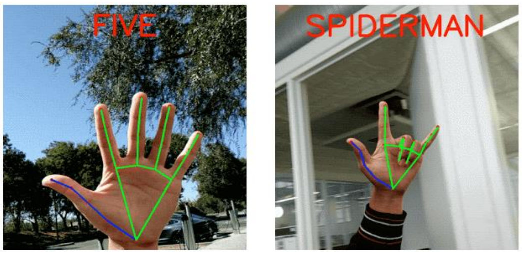


Figure 3 Hand Gesture Recognition [35][36].

Grishchenko et al. [37] challenged developing a library in MediaPipe that presents a novel state-of-the-art human body pose topology that optimized 540+ key points, which consist of 33 poses and 468 facials and 21 per-hand landmarks, which well known as MediaPipe Holistic. Their research built a simple interface of remote control application that has featured user interaction without using a device like a mouse or a keyboard, manipulated object on the screen that depends on the hand detection accuracy including gesture recognition, see Figure 4.

Figure 4 Remote Control Interface using Gesture Recognition [37].

3. RESEARCH METHODS

In this research, the user guide application is a guide for the user to display steps taken by the system by identifying hand gestures as a certain command.

We develop a user guide application that implements hand gesture recognition using Kinect to capture hand pose and then recognize it for running the application. We are using the MediaPipe framework and Python programming language to develop an application user guide. For detail, can see Figure 5 below.

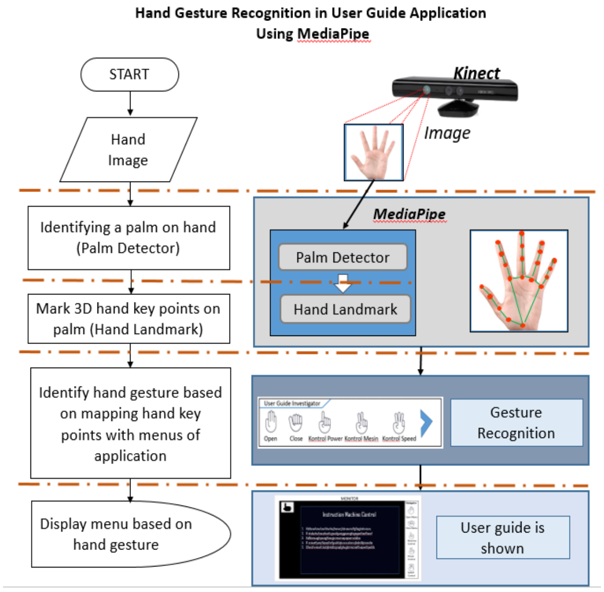
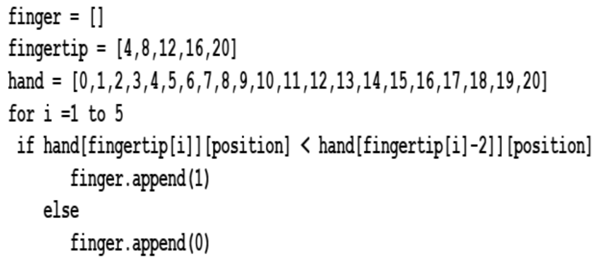


Figure 5 Workflow Method Research.

Numerous inspiring successes research in applying Kinect as a device for articulating human body tracking, pose even recognition systems. We use the Kinect x360 with RGB camera with a resolution of 640x240 pixel for capturing a real-time image for process using MediaPipe framework. MediaPipe will read the image that received from Kinect, then on an image will do palm detection and make hand landmark that made return 3D hand key points and joint it make up like skeleton. 3D key points in the palm that has been marked in the image will be computed and initialized as a tool for reading pose hand and recognition based that will be conveyed information based on hand pose had been initialized before.

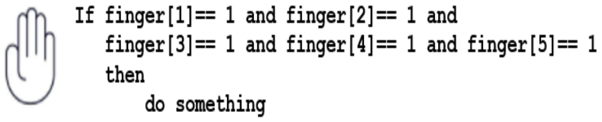
3.1 Identification Hand Gesture in MediaPipe

To identify the poses of the hand differently could be calculated using 21 key points on hand landmarks explained in Section 2.2.2 above. This identifying could be done with, firstly, determine every finger of the hand condition is open or close. For more clearly, see Figure 6 below. Figure 6 shows a pseudocode or algorithm for identifying a condition finger of a hand-related to Figure 2. In Figure 2, we can see that coordinate [4,8,12,16,20] is a coordinate of tips of fingers and declared as fingertips. The Declaration of hand with coordinate 0 until 20 is obtained from 21 key points with a hand-knuckle coordinate in hand landmark. We will compare the coordinate of fingertip based on position x (horizontal) and y (vertical) with middle points [2,6,10,14,18]. If the compare coordinates a fingertip has a value higher than middle points, then set finger with value 1, mean finger in condition open and vice versa.



```
finger = []
fingertip = [4,8,12,16,20]
hand = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]
for i =1 to 5
 if hand[fingertip[i]][position] < hand[fingertip[i]-2]][position]
       finger.append(1)
    else
       finger.append(0)
```

Figure 6 Pseudocode detects a finger of hand condition open or close



```
If finger[1]== 1 and finger[2]== 1 and
   finger[3]== 1 and finger[4]== 1 and finger[5]== 1
   then
       do something
```

Figure 7 Pseudocode detects a hand condition open or close.

In the next step, after determined the condition each finger is open or close as mentioned before, we gathered all values of fingers and compared every condition of fingers of the hand (1 for the finger is open and 0 for the finger is close), see in Figure 7. If the condition has been fulfilled and resulted is true, then the program will execute according to the instruction that has been made.

3.2 Mock-up User Guide Application

To develop a user guide application, we prepared a design mock-up user and a hand pose to recognize and position the information on the mock-up on the user application guide, see Figure 8 and Figure 9 below. The system will capture an image of a hand pose with all fingers had stated open, identifying it then do command to display mock-up as seen in Figure 8 is shown a mock-up open menu that contains hand pose is initialized with a varied menu.

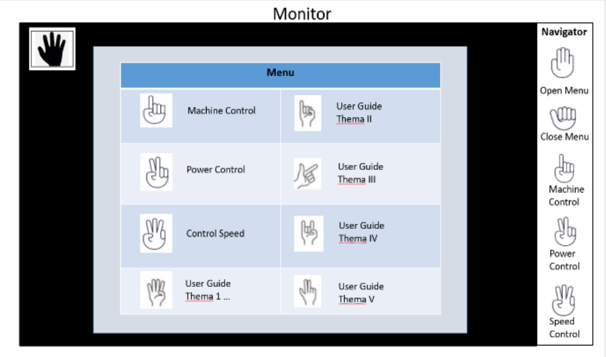
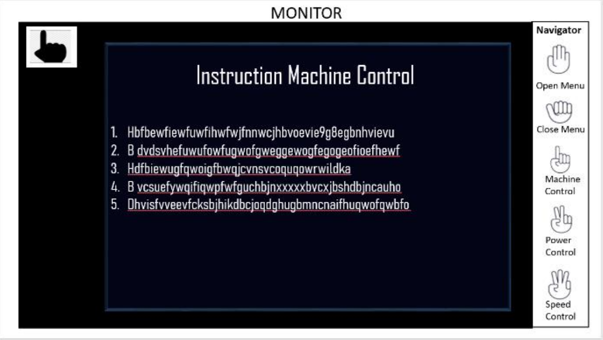


Figure 8 Mock-up Open Menus User Guide Application.



Instruction Machine Control

1. Hbfbewfiewfuwfihwfwjfnnwcjhbvoevie9g8egbnhvievu
2. B dvdsvhefuwufowfugwofgweggewogfegogeofioefhewf
3. Hdfbiewugfqwoigfbwqjcvnsvcoquqowrwildka
4. B vcsuefywqifiqwpfwfguchbjnxxxxxbvcxjbshdbjncauho
5. Dhvisfvveevfcksbjhikdbcjoqdghugbmncnaifhuqwofqwbfo

Figure 9 Mock-up One of Menu of User Guide.

If the user chooses one of the menus in the user guide application, then display information about the problem topic based on the chosen hand pose, as seen in Figure 9. The figure shows a mock-up of Instruction Machine Control that is one of a menu of user guide user application occurred along with information about guide or resolve a problem based on a topic after recognized through hand pose what the user has chosen.

In Figure 8 and Figure 9, the right corner on the mock-up shows varieties of hand pose for the menu and use to help the user as a reference to display information guide based on problem topics that have been provided. Whereas the top left corner is shown capturing hand pose an image on the camera Kinect. Other mock-up menus will have the same flow as Figure 9, which is different. Every menu is the content information to be delivered.

3.3 Description of The Dataset

The main objective of the research is to recognize hand gestures to display one of the menus that a user has chosen through a Kinect. We used 10 captured hand gestures which each hand gesture directly set one menu out, see in Figure 10 below.

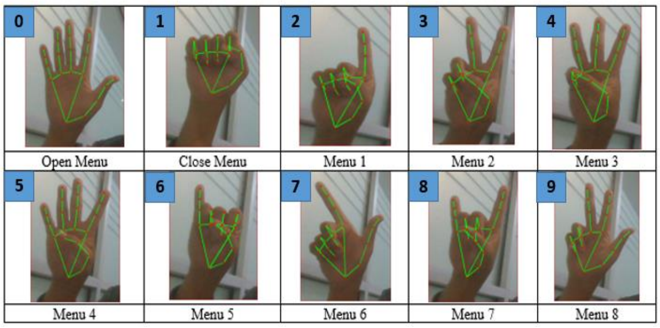
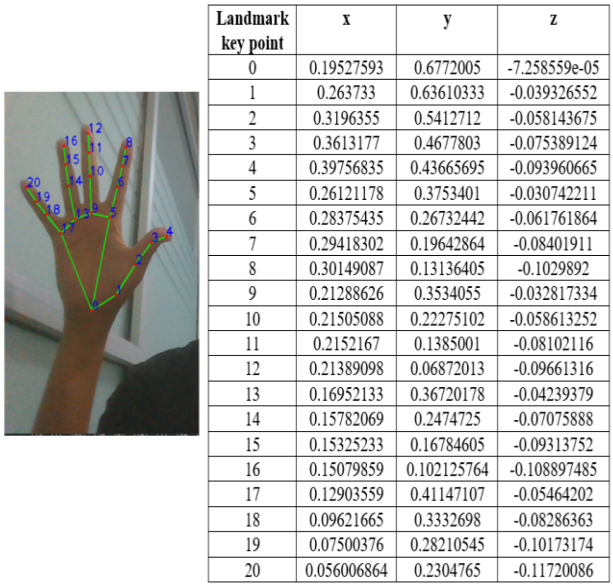


Figure 10 Hand Gestures for Menus.



Landmark key point	x	y	z
0	0.19527593	0.6772005	-7.258559e-05
1	0.263733	0.63610333	-0.039326552
2	0.3196355	0.5412712	-0.058143675
3	0.3613177	0.4677803	-0.075389124
4	0.39756835	0.43665695	-0.093960665
5	0.26121178	0.3753401	-0.030742211
6	0.28375435	0.26732442	-0.061761864
7	0.29418302	0.19642864	-0.08401911
8	0.30149087	0.13136405	-0.1029892
9	0.21288626	0.3534055	-0.032817334
10	0.21505088	0.22275102	-0.058613252
11	0.2152167	0.1385001	-0.08102116
12	0.21389098	0.06872013	-0.09661316
13	0.16952133	0.36720178	-0.04239379
14	0.15782069	0.2474725	-0.07075888
15	0.15325233	0.16784605	-0.09313752
16	0.15079859	0.102125764	-0.108897485
17	0.12903559	0.41147107	-0.05464202
18	0.09621665	0.3332698	-0.08286363
19	0.07500376	0.28210545	-0.10173174
20	0.056006864	0.2304765	-0.11720086

Figure 11 Key points of hand landmark corresponding in coordinate (x,y,z) in MediaPipe for one hand gesture.

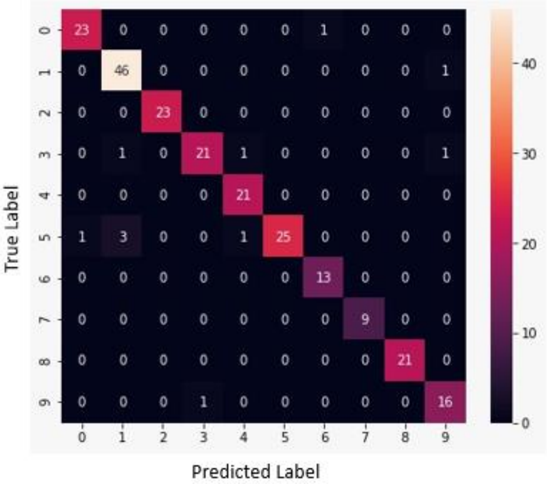
Actually, many datasets that contain images of hand gestures are publicly available to use. In this paper, we recorded our data and created a small dataset of around 900 samples were taken. These datasets consist of 10 varieties of hand gestures, as seen in Figure 10. The various conditions of the hand for datasets samples are such as from both the right and the left of the hand, the position of the palm such as the palm forward to the camera or the back of hand forward to the camera, and variety degree position of the hand where had captured the image from the camera in Kinect.

The collected datasets consist of an index of gestures (ID) is specified, extracted landmark coordinates, relative coordinates, flattening to the one-dimensional array, and normalized values were captured in MediaPipe. Index of gestures (ID) is a reference for labeling 10 varieties of hand gestures, as seen in Figure 10 above. Each gesture has one label ID for an identifier. We label the index from 0 – 9 for a picture from the first-row top left until the second-row bottom right. So, we will know that the hand gesture with the condition all fingers in open has label 0 for Open Menu.

Besides the index of gestures, determined the extracted key points are also made to get valuable 21 key points in each hand gesture that has been captured. The extracted key points are generated from the coordinate x, y, and z from 21 key points of the hand, see Figure 11 above. The coordinate x is shown the landmark position in the horizontal axis, coordinate y is the landmark position in the vertical axis, and z is the landmark depth from the camera. These datasets are simply written in a CSV file and saved to local. This allowed us to obtain training data made of quality data.

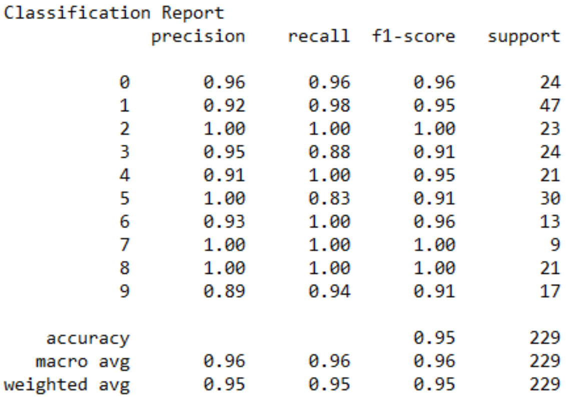
4. RESULT

The measure of the performance on the model in machine learning used Confusion Matrix. In Python, we can use the library scikit-learn to develop a confusion matrix. Experiment datasets were obtained before we used them to predict the hand gestures. The Confusion Matrix was also used to observe an accuracy achieved for the model was made.



True Label \ Predicted Label	0	1	2	3	4	5	6	7	8	9
0	23	0	0	0	0	0	1	0	0	0
1	0	46	0	0	0	0	0	0	0	1
2	0	0	23	0	0	0	0	0	0	0
3	0	1	0	21	1	0	0	0	0	1
4	0	0	0	0	21	0	0	0	0	0
5	1	3	0	0	1	25	0	0	0	0
6	0	0	0	0	0	0	13	0	0	0
7	0	0	0	0	0	0	0	9	0	0
8	0	0	0	0	0	0	0	0	21	0
9	0	0	0	1	0	0	0	0	0	16

Figure 12 Classification performance of hand gesture recognition.



```
Classification Report
                   precision    recall  f1-score   support

           0       0.96      0.96      0.96        24
           1       0.92      0.98      0.95        47
           2       1.00      1.00      1.00        23
           3       0.95      0.88      0.91        24
           4       0.91      1.00      0.95        21
           5       1.00      0.83      0.91        30
           6       0.93      1.00      0.96        13
           7       1.00      1.00      1.00         9
           8       1.00      1.00      1.00        21
           9       0.89      0.94      0.91        17

    accuracy                           0.95       229
   macro avg       0.96      0.96      0.96       229
weighted avg       0.95      0.95      0.95       229
```

Figure 13 Accuracy classification performance of hand gesture recognition.

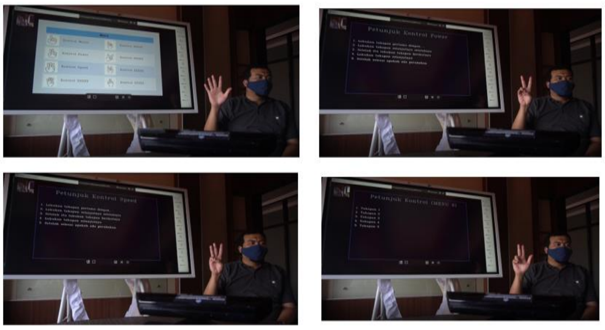


Figure 14 Screenshot s user guide application.

Using MediaPipe for implementing machine learning on hand gesture recognition in a user guide application achieved good performance see Figure 12 and Figure 13. Figure 12 above shows the predicted 10 varieties of hand gestured, and we can also see that a few hand gestures could make a false prediction of hand gestures. Figure 13 above results in validation accuracy of 95% for hand gestures to be recognized. A false prediction of hand gestures was getting the percentage accuracy performance to become descend. This is caused by lighting, a distance between Kinect as a picture catcher and user while using the application, and degrees of angle from Kinect is placed. Figure 14 shows the result of images of deploying the user guide application using hand gestures as a command for display information.

5. CONCLUSION

The Hand gesture recognition system has become an important role in building efficient human-machine interaction. Implementation using hand gesture recognition promises wide-ranging in technology industry. The MediaPipe as one framework based on machine learning plays an effective role in developing this application using hand gesture recognition, with the result has shown an accuracy performance of 95%. We would like to extend our system further to develop collaboration with other devices and other human body parts and experiment with both static and dynamic hand gesture recognition systems.

REFERENCES

[1] Ustunug A, Cevikcan, Industry 4.0: Managing The Digital Transformation, Springer Series in Advanced Manufacturing, Switzerland. 2018. DOI: https://doi.org/10.1007/978-3-319-57870-5.

[2] Pantic M, Nijholt A, Pentland A, Huanag TS, Human-Centered Intelligent Human-Computer Interaction (HCI^2): How Far We From Attaining It?,International Jounal of Autonomous and Adaptive Communications Systems (IJAACS), vol.1 no.2, 2008. pp 168-187. DOI: 10.1504/IJAACS.2008.019799 .

[3] Hamed Al-Saedi A.K, Hassin Al-Asadi A, Survey of Hand Gesture Recognition System. IOP Conferences Series: Journal of Physics: Conferences Series 1294 042003. 2019. DOI: https://doi.org/10.1088/1742-6596/4/042003.

[4] Z.Ren, J.Meng, Yuan J. Depth Camera Based Hand Gesture Regconition and its Application in Human-Computer-Interaction. In Processing of the 2011 8th International Conference on Information, Communication and Signal Processing (ICICS). Singapore. 2011.

[5] S.Rautaray S, Agrawal A. Vision Based Hand Gesture Recognition for Human Computer Interaction: A Survey. Springer Artificial Intelligence Review. 2012. DOI: https://doi.org/10.1007/s10462-012-9356-9.

[6] Lugaresi C, Tang J, Nash H, McClanahan C, et al. MediaPipe: A Framework for Building Perception Pipelines. Google Research. 2019. https://arxiv.org/abs/2006.10214.

[7] Z.Xu, et.al, Hand Gesture Recognition and Virtual Game Control Based on 3D Accelerometer and EMG Sensors, In Processing og IUI'09, 2009, pp 401-406.

[8] C.Chua, H. Guan, Y.Ho, Model-Based 3D Hand Posture Estimation From a Single 2D Image. Image and Vision Computing vol.20, 2002, pp. 191-202.

[9] Y.Li, Hand Gesture Recognition Using Kinect, 2012.

[10] M.Panwar, Hand Gesture Recognition Based on Shape Parameters, In International Conferences: Computing Communication and Application (ICCCA), 2012.

[11] Marco Maisto, An Accurate Algorithm for Identification of Fingertips Using an RGB-D Camera, IEEE Journal on Emerging and Selected Topics in Circuits and System, 2013. pp. 272-283.

[12] E.Holden, Visual Recognition of Hand Motion, Ph.D Thesis Departement of Computer Science, University of Western. 1997.

[13] Cardoso T, Delgado J, Barata J, Hand Gesture Recognition toward Echancing Accessibility. In 6th International Conference on Software Development and Technologies for Enhancing and Fighting Infoexclusion (DSAI). Procedia Computer Sciences vol.67. 2015. pp.419-429. DOI: https://doi.org/10.1016/j.procs.2015.09.287.

[14] Z Ren, Robust Hand Gesture Recognition Based on Finger-Earth Mover's Distance with Commodity Depth Camera. 2011.

[15] Stanner T, Weaver J, Pentland A, Real Time America Sign Language Recognition Using Desk

and Wearable Computer based Video. IEEE Tran on PAMI vol.20. 1998. pp. 1375-1375.

[16] Yang M.H, Ahuja N, Tabb M, Extraction of 2D Motion Trajectories and its Application to Hand Gesture Recognition. IEEE Trans on PAMI vol.29. 2002. pp 1062-1074.

[17] Bray M, Koller-Meier E, Gool L.V, Smart Particle Filtering for 3D Hand Tracking. In Processing of Sixth IEEE International Conference on Face and Gesture Recognition. 2004.

[18] Dewaele G, Devernay F, Horaud R. Hand Motion from 3D Point Trajectories and Smooth Surface Model. In Processing of 8th ECCV. 2004.

[19] Stanger C, Model-Based 3D Tracking of an Articulated Hand, 2001.

[20] Keskin C, Real Time Hand Pose Estimation using Depth Sensors, In IEEE International Conferences on Computer Vision Workshop. 2011.

[21] Lee H, Kim J. An HMM-Based Threshold Model Approach for Gesture Recognition. IEEE Trans on PAMI vol.21. 1999. pp 961-973.

[22] Wilson A, Bobick A, Parametric Hidden Markov Models for Gesture Recognition. IEEE Trans. On PAMI vol.21, 1999. pp.884-900.

[23] Wu Xiayou, An Intelligent Interactive System Based on Hand Gesture Recognition Algorithm and Kinect, In 5th International Symposium on Computational Intelligence and Design.2012

[24] Wang Y, Kinect Based Dynamic Hand Gesture Recognition Algorithm Research, In 4th International Conference on Intelligent Human-Machine System and Cybernetics. 2012.

[25] Doucet A, De Freitas N, Gordon N, Sequential Monte Carlo In Practice. New York: Springer-Verlag.2001

[26] Kwok C, Fox D, Meila, Real Time Particle Filters. In Processing of IEEE.2004.

[27] Galveia B, Cardoso T, Rybarczyk, Adding Value to The Kinect SDK Creating a Gesture Library, 2014.

[28] Su C.M, A Fuzzy Rule-Based Approach to Spatio-Temporal Hand Gesture Recognition. IEEE Trans on System, Man and Cybernetics-Part C: Application and Review no.30, 2000, pp. 276-281.

[29] Lugaresi C, Tang J, Nash H et.al, MediaPipe: A Framework for Perceiving and Processing Reality. Google Research. 2019.

[30] Abadi M, Barham P, Chen J et.al, Tensorflow: A System for Large-Scale Machine Learning, In 12th USENIX Symposium on Operating System Design and Implementation (OSDI), USA, 2016, https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi.

[31] Chen T, Li M, Li Y, MXNet: A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed System, 2015, https://arxiv.org/pdf/1512.01274.pdf.

[32] Pazke A, Gross A, Chintala S, Automatic Differentiation in PyTorch, In 31st Conference on Neural Information Processing System (NIPS), USA, 2017.

[33] Seide F, Agarwal A, CNTK: Microsoft's Open-Source Deep Learning Toolkit, In KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016, DOI: https://doi.org/10.1145/2939672.2945397.

[34] Matveev D, OpenCV Graph API. Intel Corporation. 2018.

[35] Zhag F, Bazarevsky, Vakunov A et.al, MediaPipe Hands: On – Device Real Time Hand Tracking, Google Research. USA. 2020. https://arxiv.org/pdf/2006.10214.pdf.

[36] MediaPipe: On-Device, Real Time Hand Tracking, In https://ai.googleblog.com/2019/08/on-device-real-time-hand-tracking-with.html. 2019. Access 2021.

[37] Grishchenko I, Bazarevsky V, MediaPipe Holositic – Simultaneoue Face, Hand and Pose Prediction on Device, Google Research, USA, 2020, https://ai.googleblog.com/2020/12/mediapipe-holistic-simultaneous-face.html, Access 2021.

[38] MediaPipe Github: https://google.github.io/mediapipe/solutions/hands. Access 2021.