



## Tracked Robot Control with Hand Gesture Based on MediaPipe

Marthed Wameed\*

Ahmed M. ALKAMACHI\*\*

Ergun Ercelebi\*\*\*

\*,\*\**Department of Mechatronics Engineering/ Al-khwarizmi College of Engineering/University of Baghdad/ Baghdad/Iraq*

\*\*\* *Department of Electric and Electronic Engineering/ Gaziantep University/Turkey*

Corresponding Author: \*Email: [Marthad.Wameed1202a@kecbu.uobaghdad.edu.iq](mailto:Marthad.Wameed1202a@kecbu.uobaghdad.edu.iq)

\*\* Email: [Ahmed78@kecbu.uobaghdad.edu.iq](mailto:Ahmed78@kecbu.uobaghdad.edu.iq)

\*\*\* Email: [ercelebi@gantep.edu.tr](mailto:ercelebi@gantep.edu.tr)

(Received 23 November 2022; Accepted 26 April 2023)

<https://doi.org/10.22153/kej.2023.04.004>

### Abstract

Hand gestures are currently considered one of the most accurate ways to communicate in many applications, such as sign language, controlling robots, the virtual world, smart homes, and the field of video games. Several techniques are used to detect and classify hand gestures, for instance using gloves that contain several sensors or depending on computer vision. In this work, computer vision is utilized instead of using gloves to control the robot's movement. That is because gloves need complicated electrical connections that limit user mobility, sensors may be costly to replace, and gloves can spread skin illnesses between users. Based on computer vision, the MediaPipe (MP) method is used. This method is a modern method that is discovered by Google. This method is described by detecting and classifying hand gestures by identifying 21 three-dimensional points on the hand, and by comparing the dimensions of those points. This is how the hand gestures are classified. After detecting and classifying the hand gestures, the system controls the tracked robot through hand gestures in real time, as each hand gesture has a specific movement that the tracked robot performs. In this work, some important paragraphs concluded that the MP method is more accurate and faster in response than the Deep Learning (DL) method, specifically the Convolution Neural Network (CNN). The experimental results shows the accuracy of this method in real time through the effect of environmental elements decreases in some cases when environmental factors change. Environmental elements are such light intensity, distance, and tilt angle (between the hand gesture and camera). The reason for this is that in some cases, the fingers are closed together, and some fingers are not fully closed or opened and the accuracy of the camera used is not good with the changing environmental factors. This leads to the inability of the algorithm used to classify hand gestures correctly (the classification accuracy decrease), and thus response time of the tracked robot's movement increases. That does not present possibility for the system to determine whether the finger is closed or opened.

**Keywords:** *Computer vision, Machine learning, MediaPipe, Hand landmarks.*

### 1. Introduction

Hand gestures at the present time play an important role in communication between humans and robots. Also, Hand gestures have developed significantly, as they were previously used in specific fields but are now used in wide fields, for example, in smart cars, video games, and smart

homes. In addition, Hand gestures is very widely used in translation to communicate between people who have difficulty speaking or hearing [1]. In this paper, the computer vision algorithm, specifically the machine learning algorithm (MediaPipe), is utilized. The techniques of turning images into digital data and altering their nature so that they are easier for machines to grasp while simultaneously

*This is an open access article under the CC BY license*



enhancing the visual information and human perception of it are known as computer vision [2]. Computer vision is the only technique that can extract information from incoming images [3]. The MediaPipe algorithm is used to detect and classify the hand gestures [4]. The goal in this research is to detect hand gestures using the MP algorithm via the camera sensor, classify hand gestures, and transmit the instructions wirelessly. That is to the tank robot to perform certain movements, as each hand gesture indicates a specific movement in the tank robot. There are other ways to detect and classify hand gestures using deep learning, specifically convolutional neural networks (CNN) [5-11]. This method is more complex and consumes more processing time.

The main research contribution in this manuscript is to improve the MediaPipe algorithm in a way that gives the fastest response in classifying hand gestures. That is by making the hand detection only runs on first frame or when hand is missing in an image. That can significantly reduce response time and make the algorithm more accurate in detect hand gestures.

## 2. Related work

In this section, a summary of previous studies are provided. Hand gesture is one of the important topics in our world and is defined as a non-voice communication through which hand movement can be used to transmit instructions to multiple devices without direct contact with those devices. Waskito et.al.[12] depended on deep learning to detect hand gestures in real time by using the convolutional neural networks (CNN) and then control the movement of wheeled robot via hand gestures. Huu et. al. [13] used artificial neural network (ANN) to detect hand gesture and control the smart home in real time via these gestures. Bidyut Juoti et. al. [14], used the machine learning specifically uses

Media Pipe to detect hand gesture to control computer mouse through these hand gestures. In MP, Hand Gestures were detected by two faces. The first was the Hand Detection via palms detector and the second was the Hand Landmark via 21 three dimensional points on every part of the hand gesture as shown in the Figure (1) [4]. Subhangi Adhikary et.al [15] saved the new data which contains 21 points on every part of the Hand Gesture in a file called CSV, and this data were used to train and test the system. MP process reduced the use of data augmentation (i.e. Rotations, Flipping, Scaling) used in deep learning [16]. Prakash et.al. [17] detected hand gestures by fingertips to control mouse operations in the computer in real time using the camera. The region of hand was first segmented using the increasing area technique and then applying morphological operations. Haldera et. al. [16] utilized machine learning algorithm and MP's open source framework to detect sign language and to test this system they use several data for sign language such as American, Italian, Indian and Turkish. Ali Suryaperdana Agoes et. al.[18] employed hand gestures to control a user guide application through MP's open source framework. By comparing the value of coordinates points of the fingers and detecting the finger, closed or opened. If the coordinates value of fingertip's was higher than the finger middle points, set the finger with a value of 1, meaning the finger was in the open condition, and vice versa. Taban et.al.[19] depending on the computer vision, specifically the use of the Viola-Jones algorithm to detect and classify hand gestures, and through these gestures, the lights are turned on and off. The lighting response time to the gesture commands was 0.43 seconds, and the dataset consisted of eight gestures containing 12000 images. In this work, for more efficient and precise computer training, a novel concept proposes using skin detection prior to computer training to automatically establish the size and position of all positive pictures.

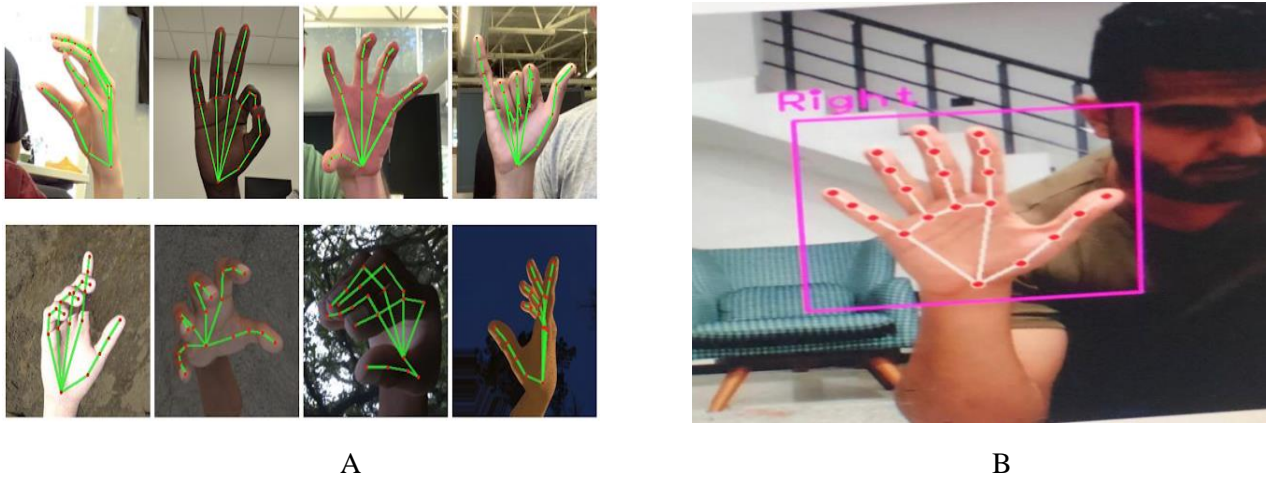


Fig. 1. A. Hand gestures detection [20]. B. Hand landmarks.

### 3. System Design

#### 3.1. General Design

Figure (2) shows the main parts of the system. The system consists of two main parts.

#### 1. Base station

As shown in Figure (3) images of hand gestures are captured in real time by a camera (AverMedia

PW313) connected to a microcontroller (Raspberry Pi 4). The camera and the microcontroller detect and classify hand gestures based on computer vision. Next, the output commands are sent via Bluetooth to the robot to execute these commands. Each hand gesture has a specific command that moves the robot according to those commands, as shown in Table 1.

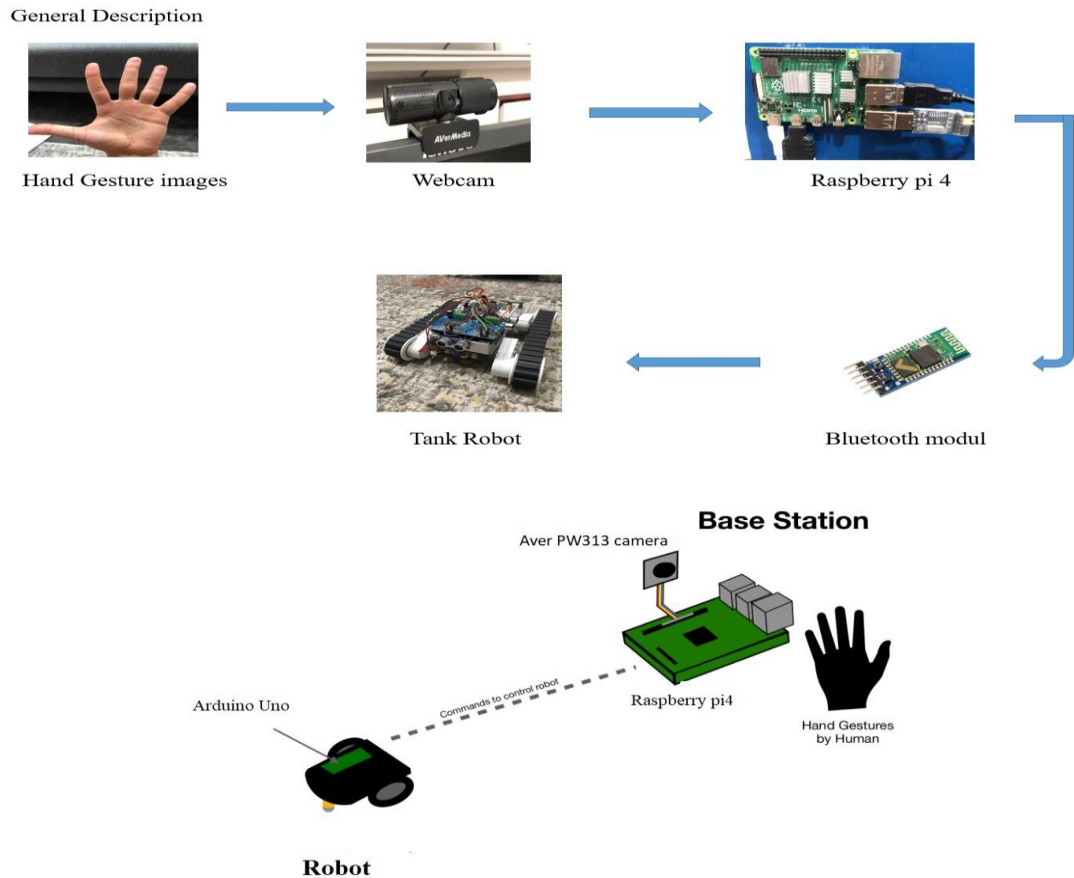
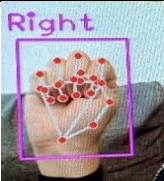





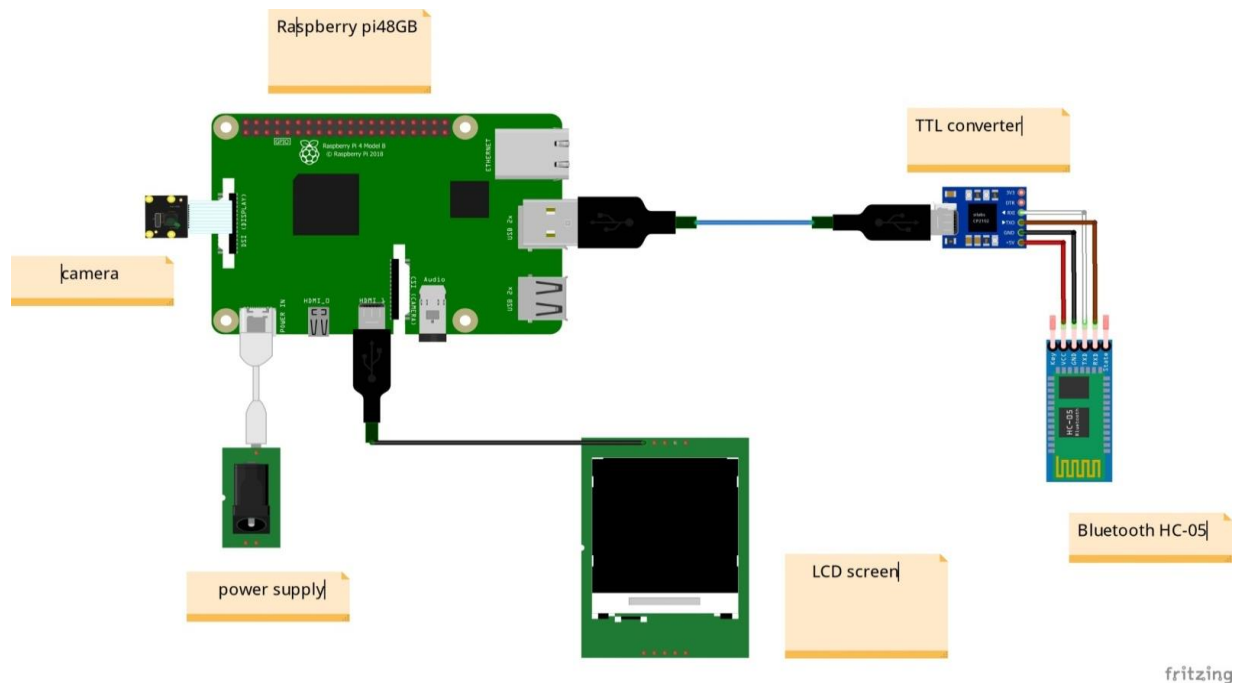


Fig. 2. Overall system description.

**Table 1,**  
The style of hand motions and directives used.

The Form of Hand Gestures					
0	1	2	3	4	5
					
The Commands executed in the Robot					
Stop (Robot is not moving)	Move forward	Move backward	Left (Rotating anti clockwise)	Right (Rotating clockwise)	Stop (Robot is not moving)



**Fig. 3. Base station parts description.**

PyCharm, the Python programming language, the OpenCV, proposed model, and TensorFlow libraries are used in the software design for preprocessing and picture classification in machine learning.

## 2. Tracked Robot

As shown in Figure (4) the tracked robot contains several parts, and among these parts is the

microcontroller (Arduino Mega). The microcontroller receives commands for moving the robot from the Bluetooth connected to itself and sends those commands to a DC driver (Motor Driver BTS7960). The speed and rotation of the DC motors are controlled. Whereas, the DC motors are responsible for the movement of the tracked robot. The robot also contains four ultrasonic sensors to avoid obstacles and be more accurate in its movements.

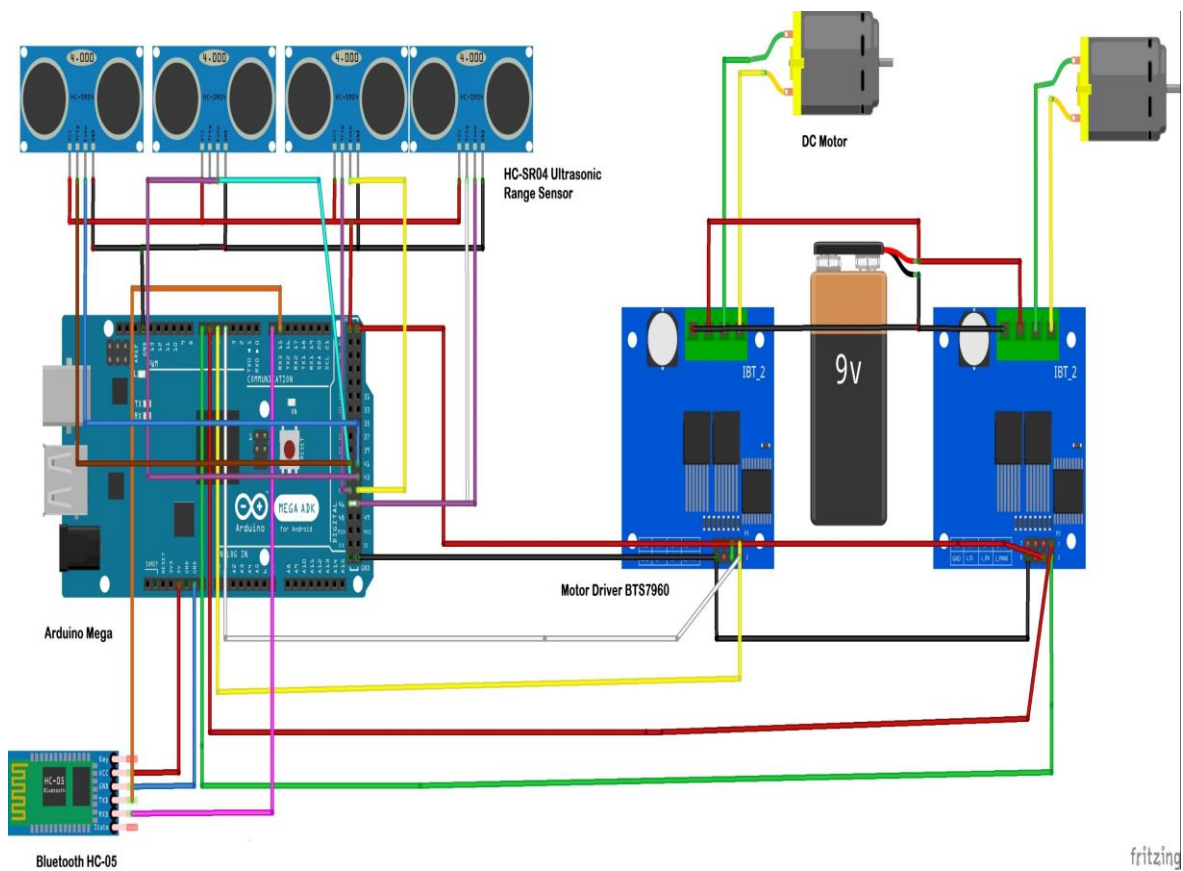
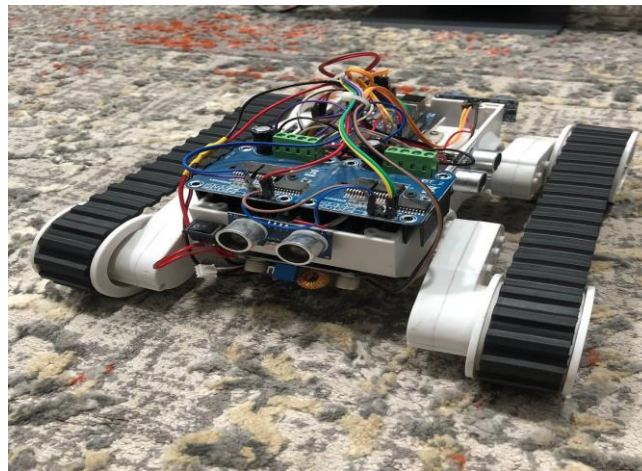


Fig. 4. Tracked robot parts description.

Figure (5) shows the complete process flowchart of the system's work, starting with taking pictures with the camera, passing through

the stages of detecting and classifying hand gestures, and leading to controlling the tracked robot's movement through hand gestures.



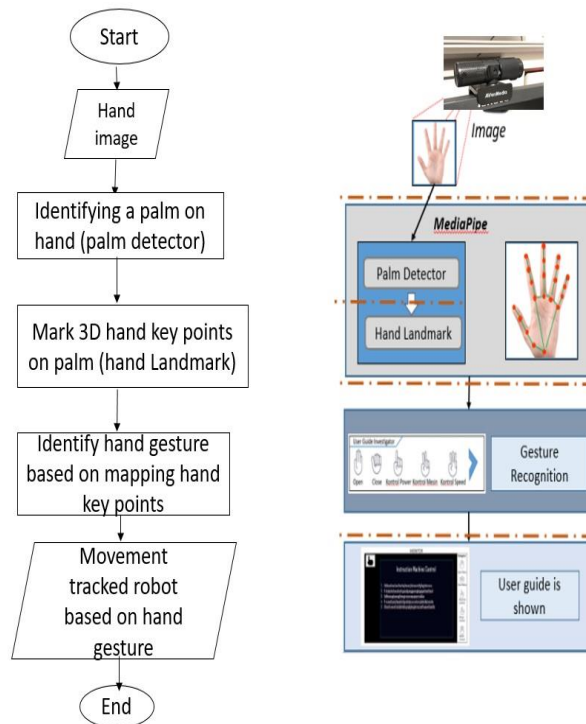


Fig. 5. Software design flowchart.

### 3.2 MediaPipe Framework

In July of 2019, Google published MediaPipe as an open-source tool. This method detects the palm with the use of Single shot detection (SSD), an object identification technique (detection the palm), and the hand landmark model is used to extrapolate the information necessary to recognize the Fingers. The MediaPipe is an algorithm that detects individual hands inside a video stream. The algorithm is developed using machine learning

techniques. Real-time hand tracking is possible by using MediaPipe for everyone with a webcam. By using machine learning, technique in this study extrapolates 21 points 3D from a single picture. Real-time tracking of several hands is made possible by a three-stage approach consisting of a palm detection model, a hand landmark model and gesture recognizer model as shown in Figure (6) [20-23]. The three models, as well as the training datasets, are detailed in more depth below.

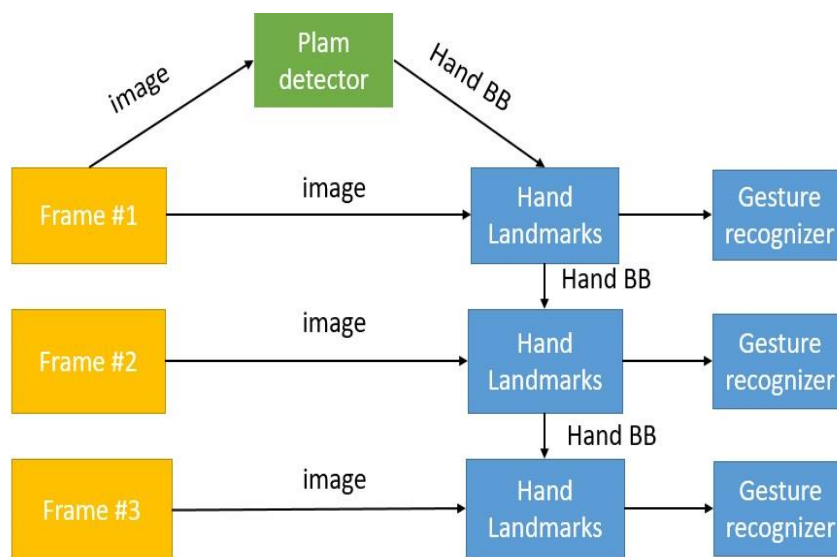


Fig. 6. MP algorithm parts.

### 3.2.1 Datasets

Three distinct data sets are selected for use in the training of the MediaPipe models. The description of the datasets that are used in the training of the model is listed below.

- **The wild dataset:** This dataset contains 6,000 pictures that vary greatly in regard to factors like location, lighting, and the way people's hands look. One drawback of the dataset is that it doesn't capture more complex forms of hand articulation.
- **The house collected dataset:** There are around 10,000 images in this collection, each showing a different hand movement from every possible angle. The disadvantage of this dataset is that it was collected from just 30 persons, and there was very little variation in the backgrounds of those people.
- **Synthetic dataset:** This dataset takes a high-resolution representation of a synthetic hand model and maps it to its 3D coordinates against a variety of backgrounds. Five skin tones are included in the hand model's texturing. One hundred thousand still images were captured from the change video streams between hand positions. Three cameras were used to capture each pose in a randomly generated high dynamic range lighting scenario.

### 3.2.2. Palm Detector Model

Compared to other tasks, detecting hands is more difficult since it must be able to distinguish between different hand sizes and recognize closed portions of hands. It is possible to use the contrast and forms of a person's nose and lips as distinguishing traits in a face detection activity. In contrast, comparable traits are absent in the hand detection task, significantly increasing its difficulty. Thus, the palm detection is trained since

it is simpler to estimate bounding boxes around stiff objects like the palm and the fist [22].

### 3.2.3. Hand Landmark

After revealing the Palm Detector model, the Landmark stage comes. The 21 three-dimensional points are identified on the hand, as shown in Figure (6). Where the X-axis represents the width of the image, the Y-axis represents the height of the image, and the Z-axis represents the depth of the image (the distance between the hand and camera). Comparing between the dimensions of 21 points 3D, the hand gestures are classified.

### 3.2.4 Gesture Recognizer Model

Figure (7) shows 21 points 3D on each part of the palm. To classify hand gestures, the value of coordinates of the points (4, 8, 12, 16, 20) are compared. These points represent the fingertips with the value of coordinates of the points (3, 5, 9, 13, 17) that represent the middle points relative to the x and y axes.

- **The state of the thumb.**

The thumb is open if the value of coordinate of the Point (4), (which represents fingertip ) relative to the X-axis, is greater than the value of coordinate of the Point (3), (which represents the middle point) relative to the X-axis, and vice versa.

- **Second, the state of the all fingers except for the thumb.**

The fingers (index, middle, ring, and pinky) are opens if the values of coordinates of the points (8, 12, 16, 20), (which represent the fingertip) relative to the Y axis are greater than the values of coordinates of the points (5, 9, 13, 17),(which represent the middle point) relative to the Y axis, and vice versa.

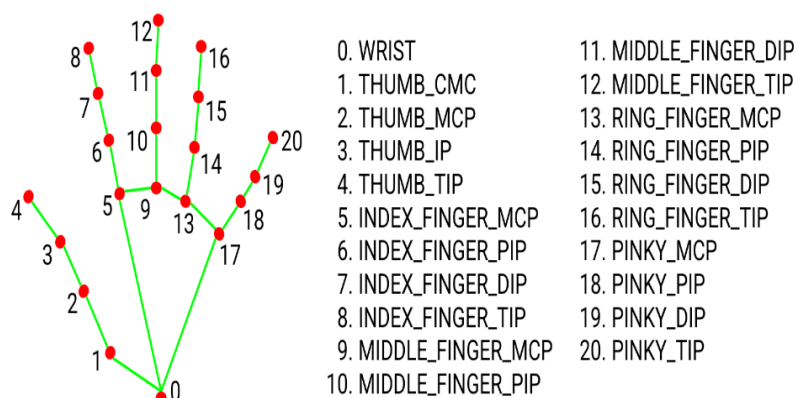


Fig. 7. Hand Landmark [20].

Figure (8) illustrates the graph that we created in MediaPipe in order to monitor hand movements. The graph is segmented into two parts: one to detect hands and another to calculate landmarks [22]. One of the main improvements provided by MediaPipe is the palm detector, which only executes when required (which isn't very frequently) and therefore dramatically lowers computing costs. Instead of using the palm detector on every frame, we accomplish this by determining

the hand location in the current video frames from the computed hand landmarks in the previous frame. The main contribution in this manuscript is to improve the MediaPipe algorithm in a way that makes it the fastest response in detecting hand gestures in real time by making the hand detection only run on the first frame or when the hand is missing in the image. Which significantly reduces response time and makes the algorithm more accurate at detecting hand gestures.

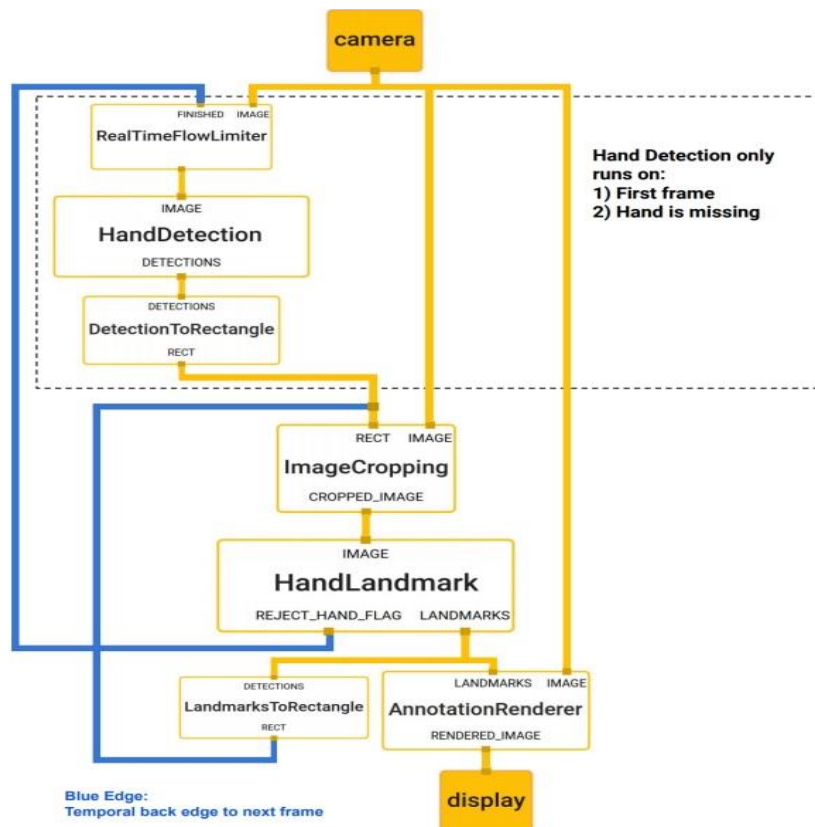


Fig .8. Stricture of MP algorithm.

#### 4. Description of the Data sets for Testing Proposed Model

MediaPipe is capable to recognize significantly more complicated gestures. Six distinct hand gestures are distinguished by the number of raised fingers are identified in this work. These results are categorized into six classes and distinguished by the number of fingers raised (from 0 to 5) as shown in Figure (9). Figure (9) lists six gestures that are employed to assess hand gesture detection. Initial

images for the data collecting phase are captured using the laptop's webcam. Due to the low quality of the captured photos caused by noise, the information are re-collected using Aver PW313 webcam. All of these photos are shot in different lighting scenarios. Five hundred pictures are taken for each class, half in the sunlight and half at night using the standard lamp in the room. The hands' locations (distance between camera and gesture) and angles (between camera and gesture) are also altered.





Fig .9. Explain the dataset consist of six classes.

## 5. Configuration Settings of MediaPipe Algorithm

There are a variety of configuration settings that may be applied. The configuration are explained below:

**Static Image Mode:** a flag that can be used to determine which mode will be utilized while reading the image.

In this study, the proposed model are tested in two modes

- Default mode, depending on the data set that are previously collected for testing the proposed model.
- Undefault mode depending on the data set from real time for testing the proposed model.

**Max\_Num Hands:** Depending on the number of hands being examined, a value for this integer variable must be determined.

For this work, the default value of 2 is used.

**Model\_Complexity:** This flag indicates the accuracy and response time. If the value is 1, this indicates that the system works with high accuracy, but with a longer response time. If it is 0, it indicates the opposite. In this work, the value was set to the default value, which is 1.

**Min\_Detection\_Confidence:** This is the level of confidence that the hand detector uses to set the output threshold.

If the confidence is less than this provided threshold value, the detectors is considered to have failed, and also no output is given.

The number must be between 0.0 and 1.0. In this work, the default value of 0.5 is utilized.

**Mim\_Tracking\_Confidence:** The static mode does not make use of this flag. When performing real-time hand tracking, this value should be set between zero and one. This is done to ensure that the hand is successfully monitored. This uses the default of 0.5. During live tracking, if the confidence is too low, it will move on to the next image without doing the detection. Hand detection is performed on each and every image in static mode [20].

## 6. Experimental Results

### 6.1 Accuracy of Proposed Model Algorithm When Static Image Mode is Default without Control Robot

Table (2) details MP's classification report, and Figure (10) displays the outputs images. The accuracy of proposed model is 0.97. That has a recall and precision of 0.90 and 0.91, respectively. Also, the F1 score is 0.90. The F1-score for each classes is shown in Figure (10).

Additionally, environmental settings such as the distance between hand gestures and the camera being 1 m and the tilt angle between hand gestures and the camera being  $0^\circ$  have been specified.

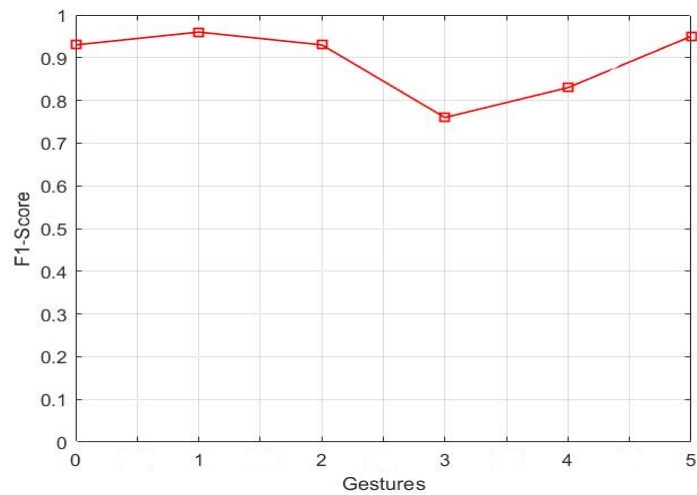


Fig. 10. Clarify F1-score of different classes.

Table 2, Classification report.

Classes	Precision	Sensitivety (Recall)	Accuracy	F1-Score
0	0.97	0.9	0.98	0.93
1	0.96	0.96	0.99	0.96
2	0.90	0.97	0.97	0.93
3	0.81	0.73	0.95	0.76
4	0.83	0.85	0.95	0.83
5	0.93	.990	0.98	0.95
Mean	0.91	0.90	0.97	0.89

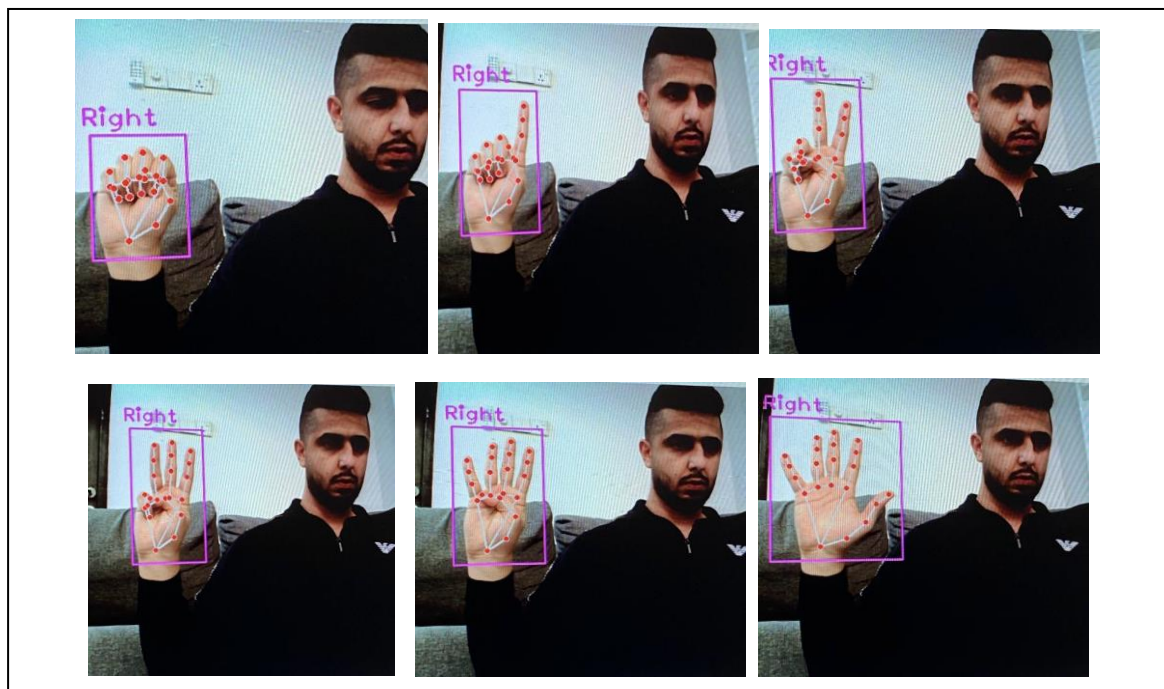


Fig. 10. Results of the proposed model algorithm

Through the above table, the highest value of accuracy is represented in class 1, and the lowest

accuracy is represented by classes 4 and 3, is noted. The accuracy of the classification of hand

gestures depends on the comparison between the dimensions of the points (21 points 3D).

In some cases, the fingers are close together, and some fingers are not fully closed or opened, which leads to the inability of the algorithm used to classify hand gestures correctly, and thus the classification accuracy decreases.

**6.2. Test the Proposed Model when Static Image Mode is Undefault to Control the Robot in Real Time**

Depending on the changing intensity of illumination, the accuracy of the system is calculated this is the first testing for the proposed system. A specific application is used to measure the intensity of lighting based on the phone's camera. In this research, lighting with a variable intensity that is 50 lux, 100 lux, and 150 lux, has been used. Additionally, environmental settings such as the distance between hand gestures and the camera being 1 m and the tilt angle between hand gestures and the camera being 0° have been specified. The experiment is repeated 20 times for each hand gesture and light intensity.

**Table 3, Light intensity variation based accuracy.**

Hand Gesture	Accuracy with Intensity Light (%)		
	50 LUX	100 LUX	150 LUX
Stop	100	100	100
Forward	100	100	100
Backward	100	100	100
Left	75	100	65
Right	65	100	40

According to Table (3), when the illumination level is set to 100 lux, the accuracy for all hand gestures reaches 100%.

Depending on the changing the distance between camera and hand gestures, the accuracy of the system is calculated. This is the second test for the proposed system. The measurement distance is the distance between the hand gestures and the camera. The distances chosen are 1 m, 2 m, and 4 m. additionally, environmental settings such as the intensity of light being 100 lux and the tilt angle between hand gestures and the camera being 0° have been specified. Experiment is repeated 20 times for each hand gesture and specified distance.

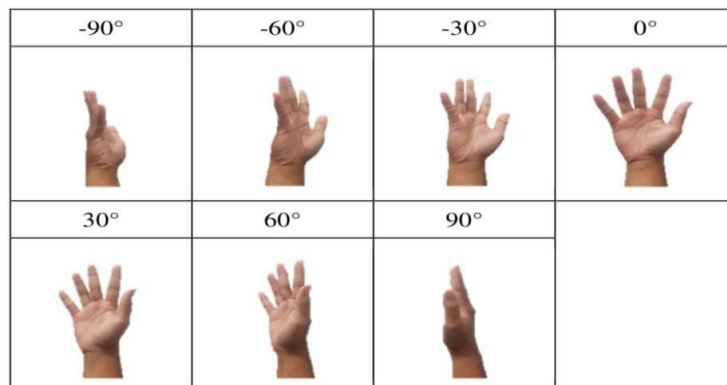
**Table 4, Variations in distance-based accuracy.**

Hand Gesture	Accuracy with Distance (%)		
	1 m	2 m	4m
Stop	100	100	100
Forward	100	100	100
Backward	100	100	100
Left	100	80	70
Right	100	70	70

Table (4) shows that at 1 m, the accuracy reaches 100% for all hand gestures. This means that at this distance, all hand gestures can be correctly classified.

Depending on the changing tilt angle between camera and hand gestures as shown in Figure (11).

The accuracy of the system is calculated. This is the second testing for the proposed system.



**Fig. 11. The tilt angles between the hand and the camera.**

The tilt angle used between the hand gesture and the camera is 0°, 30°, 60° and 90° in this test. Additionally, environmental settings such as the

distance between hand gestures and the camera being 1 m and intensity of light being 100 lux. Each hand gesture and tilt angle is tested 20 times.

**Table 5**  
**Tilt-angle testing accuracy.**

Hand Gesture	Accuracy with Tilt Angle (%)			
	0°	30°	60°	90°
Stop	100	100	100	60
Forward	100	100	100	30
Backward	100	100	100	20
Left	100	100	55	15
Right	100	100	40	12

Table (5) shows high accuracy values, with 100% for each hand gesture when the slope angle is 30° degrees or 0° degrees.

**Table 7,**  
**Actuators Power Consumption Test for each pulse width modulator (PWM).**

Hand Gesture	Left Motor PWM	Consumed Power(w)	Right Motor PWM	Consumed Power(w)
Stop	0	0	0	0
Forward	38	0.808	38	0.8602
Backward	38	0.8064	38	0.808
Left	29	0	29	0.6768
Right	29	0.633	29	0

Using current sensor (INA219) to measure consumed power in the above table, a slight difference in the value of consumed power for the same value of PWM between the right and left motors is noticed. This difference is due to the distribution of loads in the robot.

### 6.2.2 Time Response

Time testing checks how long it takes from when the input is given until the output is run. Hand gestures are used as input, and the robot's motors move as a result. The response time is calculated from the time the input is given, and then the Raspberry Pi 4 processes the data and sends the data to the robot motors via Bluetooth. After that, each hand gesture undergoes a test of response time. Each hand gestures is tested 3 times.

### 6.2.1 Actuators Power Consumption Test

Two electric motors (6v, DC) are used to move the tracked robot, while a 2,200-mAh, 14.8-volt Li Ion battery is used to power the motors. Various hand gestures have different PWM values.

**Table 8,**  
**The Robotic Response time test.**

Hand Gesture	Average Reaction Time (s)
Stop	0.091
Forward	0.121
Backward	0.126
Left	0.162
Right	0.165
Average Total	0.133
Reaction Time (s)	

Table (8) shows that the average time it takes for hand gesture input to trigger motor action on the robot is 0.133 seconds.

Table (9) shows the comparison of the state-of-the-art works of Waskito et.al. [12], Huu et. al.[13], Bidyut Juoti et. al. [14], Subhangi Adhikary et.al [15], A. Halder et.al [16], Prakash et.al. [17], Ali Suryaperdana Agoes et. al. [18] with the model we suggested, which delivered more accurate and better results.

**Table 9,**  
**Comparison of the state-of-the-art works with our proposed model.**

Reference	Objective	Sensors	Neural network architecture	Data Set	Accuracy (%)
Waskito et.al. [12]	Wheeled Robot control	Single camera	CNN	6000(training) 600(validation)	96.67%
Huu et. al.[13]	Smart home control	Single camera	ANN	6000(size352x200) 6000(size 704x400)	91.5%
Bidyut Juoti et. al. [14],	Virtual-mouse control	webcam	Machine learning (MediaPipe)	From Google enormous and diverse	-
Subhangi Adhikary et.al [15]	Communicating between people (disabilities)	Single camera	Machine learning (MediaPipe and random forest classifier)	2194	96.28%.
A. Halder et.al [16],	Communicating between people (disabilities)- sign language (Turkey)	single camera	Machine learning (MediaPipe and support vector machine SVM)	4124	96.22%
Ali Suryaperdana Agoes et. al. [18]	Human-Computer -Interactions (HCI)( User Guide Application)	kinect	Machine learning (MediaPipe)	900	95%
Proposed model	Movement robot control	single camera (AverMedia PW313)	Machine learning (MediaPipe)	900	97%

## 7. Discussion

The accuracy of some commands decreases with changing environmental factors in real time. That is due to the system's algorithm not being able to correctly classify hand gestures. The most important reason for the system's inability to classify hand gestures correctly is that the fingers are close together and some fingers are not fully closed or opened. That may present a delay in the response of the robot's movement to some commands. Thus, changing some irregular hand gestures to more regular gestures to increase accuracy and reduce the robot's response time is recommended.

## 8. Conclusion

This study proposed a model to control the movement of the robot through hand gestures using computer vision, specifically relying on the MP algorithm. The MP algorithm classifies hand gestures by comparing the coordinates of 21 three-dimensional points specified on the hand. The data sets used in this study for testing Proposed Model consisted of 30000 images of various hand gestures

captured under various environmental conditions. Through this model, the robot's movement was very well controlled, but there were some gestures that were not accurately classified in real time. The reason is due to the fact that some cases, the fingers are close together, and some fingers are not fully closed or open. That leads to the inability of the algorithm used to classify hand gestures correctly, and thus the classification accuracy decreases. That gives impossibility for the system to determine whether the finger is closed or opened. Hence, the system depends on the classification of hand gestures based on the comparison between the coordinates of the 3D points specified on the hand. To increase the accuracy of the classification of hand gestures, suggestion of replacing these with others in which the fingers are more clearly defined. That leads to more accurate classification and thus reduces the response time of the robot to commands. In future work, a microcontroller can be used with high specifications for the application of computer vision algorithms, such as a Jetson Nano, as well as a high-resolution camera instead of the one currently used.



## References

- [1] S. Amaliya, A. N. Handayani, M. I. Akbar, H. W. Herwanto, O. Fukuda, and W. C. Kurniawan, "Study on Hand Keypoint Framework for Sign Language Recognition," *7th Int. Conf. Electr. Electron. Inf. Eng. Technol. Breakthr. Gt. New Life, ICEEIE 2021*, pp. 3–8, 2021, doi: 10.1109/ICEEIE52663.2021.9616851.
- [2] F. Hardan and A. R. J. Almusawi, "Developing an Automated Vision System for Maintaining Social Distancing to Cure the Pandemic," *Al-Khwarizmi Eng. J.*, vol. 18, no. 1, pp. 38–50, 2022, doi: 10.22153/kej.2022.03.002.
- [3] Y. G. Khidhir and A. H. Morad, "Comparative Transfer Learning Models for End-to-End Self-Driving Car," *Al-Khwarizmi Eng. J.*, vol. 18, no. 4, pp. 45–59, 2022, doi: 10.22153/kej.2022.09.003.
- [4] A. Osipov and M. Ostanin, "Real-time static custom gestures recognition based on skeleton hand," *2021 Int. Conf. "Nonlinearity, Inf. Robot. NIR 2021*, pp. 1–4, 2021, doi: 10.1109/NIR52917.2021.9665809.
- [5] A. Mujahid *et al.*, "Real-time hand gesture recognition based on deep learning YOLOv3 model," *Appl. Sci.*, vol. 11, no. 9, 2021, doi: 10.3390/app11094164.
- [6] M. Al-Hammadi *et al.*, "Deep learning-based approach for sign language gesture recognition with efficient hand gesture representation," *IEEE Access*, vol. 8, pp. 192527–192542, 2020, doi: 10.1109/ACCESS.2020.3032140.
- [7] H. Y. Chung, Y. L. Chung, and W. F. Tsai, "An efficient hand gesture recognition system based on deep CNN," *Proc. IEEE Int. Conf. Ind. Technol.*, vol. 2019-February, pp. 853–858, 2019, doi: 10.1109/ICIT.2019.8755038.
- [8] Y. S. Tan, K. M. Lim, and C. P. Lee, "Hand gesture recognition via enhanced densely connected convolutional neural network," *Expert Syst. Appl.*, vol. 175, no. November 2020, p. 114797, 2021, doi: 10.1016/j.eswa.2021.114797.
- [9] R. Ahuja, D. Jain, D. Sachdeva, A. Garg, and C. Rajput, "Convolutional neural network based American sign language static hand gesture recognition," *Int. J. Ambient Comput. Intell.*, vol. 10, no. 3, pp. 60–73, 2019, doi: 10.4018/IJACI.2019070104.
- [10] P. Nakjai and T. Katanyukul, "Hand Sign Recognition for Thai Finger Spelling: an Application of Convolution Neural Network," *J. Signal Process. Syst.*, vol. 91, no. 2, pp. 131–146, 2019, doi: 10.1007/s11265-018-1375-6.
- [11] A. G. Mahmoud, A. M. Hasan, and N. M. Hassan, "Convolutional neural networks framework for human hand gesture recognition," *Bull. Electr. Eng. Informatics*, vol. 10, no. 4, pp. 2223–2230, 2021, doi: 10.11591/EEI.V10I4.2926.
- [12] T. B. Waskito, S. Sumaryo, and C. Setianingsih, "Wheeled Robot Control with Hand Gesture based on Image Processing," *Proc. - 2020 IEEE Int. Conf. Ind. 4.0, Artif. Intell. Commun. Technol. IAICT 2020*, pp. 48–54, 2020, doi: 10.1109/IAICT50021.2020.9172032.
- [13] P. N. Huu, Q. T. Minh, and H. L. The, "An ANN-based gesture recognition algorithm for smart-home applications," *KSII Trans. Internet Inf. Syst.*, vol. 14, no. 5, pp. 1967–1983, 2020, doi: 10.3837/tiis.2020.05.006.
- [14] B. J. Boruah, A. K. Talukdar, and K. K. Sarma, "Development of a Learning-aid tool using Hand Gesture Based Human Computer Interaction System," *2021 Adv. Commun. Technol. Signal Process. ACTS 2021*, pp. 2–6, 2021, doi: 10.1109/ACTS53447.2021.9708354.
- [15] S. Adhikary, A. K. Talukdar, and K. Kumar Sarma, "A Vision-based System for Recognition of Words used in Indian Sign Language Using MediaPipe," *Proc. IEEE Int. Conf. Image Inf. Process.*, vol. 2021-Novem, pp. 390–394, 2021, doi: 10.1109/ICIIP53038.2021.9702551.
- [16] A. Halder and A. Tayade, "Real-time Vernacular Sign Language Recognition using MediaPipe and Machine Learning," *Int. J. Res. Publ. Rev.*, no. 2, pp. 9–17, 2021, [Online]. Available: www.ijrpr.com.
- [17] R. Meena Prakash, T. Deepa, T. Gunasundari, and N. Kasthuri, "Gesture recognition and finger tip detection for human computer interaction," *Proc. 2017 Int. Conf. Innov. Information, Embed. Commun. Syst. ICIIECS 2017*, vol. 2018-Janua, pp. 1–4, 2018, doi: 10.1109/ICIIECS.2017.8276056.
- [18] Indriani, M. Harris, and A. S. Agoes, "Applying Hand Gesture Recognition for User Guide Application Using MediaPipe," *Proc. 2nd Int. Semin. Sci. Appl. Technol. (ISSAT 2021)*, vol. 207, no. Issat, pp. 101–108, 2021, doi: 10.2991/aer.k.211106.017.
- [19] D. A. Taban, A. Al-Zuky, S. H. Kafi, A. H.

- Al-Saleh, and H. J. Mohamad, "Smart Electronic Switching (ON/OFF) System Based on Real-time Detection of Hand Location in the Video Frames," *J. Phys. Conf. Ser.*, vol. 1963, no. 1, 2021, doi: 10.1088/1742-6596/1963/1/012002.
- [20] GitHub, "MediaPipe on GitHub." <https://google.github.io/mediapipe/solutions/hands> (accessed Sep. 20, 2022).
- [21] R. E. Valentin Bazarevsky and Fan Zhang, "On-Device, Real-Time Hand Tracking with MediaPipe," *MONDAY, AUGUST 19, 2019*. <https://ai.googleblog.com/2019/08/on-device-real-time-hand-tracking-with.html> (accessed Oct. 01, 2022).
- [22] F. Zhang *et al.*, "MediaPipe Hands: On-device Real-time Hand Tracking," 2020, [Online]. Available: <http://arxiv.org/abs/2006.10214>.
- [23] C. Lugaresi *et al.*, "MediaPipe: A Framework for Perceiving and Processing Reality," *Google Res.*, pp. 1–4, 2019.

## التحكم في الروبوت بإيماءة اليد اعتماداً على الميديا باي

مرثد وميض مجيد\* احمد محروس القامجي\*\*

اركون ارچلبي\*\*\*

\*\*\*قسم هندسة الميكاترونكس/ كلية الهندسة الخوارزمي/ جامعة بغداد/ بغداد / العراق

\*\*\* قسم الهندسة الكهربائية والالكترونية/ جامعة غازي عنتاب/ تركيا

\*البريد الالكتروني: [Marthad.Wameed1202a@kecbu.uobaghdad.edu.iq](mailto:Marthad.Wameed1202a@kecbu.uobaghdad.edu.iq)

\*\*البريد الالكتروني: [Ahmed78@kecbu.uobaghdad.edu.iq](mailto:Ahmed78@kecbu.uobaghdad.edu.iq)

\*\*\* البريد الالكتروني: [ercelebi@gantep.edu.tr](mailto:ercelebi@gantep.edu.tr)

### الخلاصة

تعتبر إيماءات اليد حالياً واحدة من أكثر الطرق دقة وتطوراً للتواصل في العديد من التطبيقات، مثل لغة الإشارة، والتحكم في الروبوتات، والعالم الافتراضي، والبيوت الذكية، ومجال ألعاب الفيديو. تُستخدم عدة تقنيات لاكتشاف إيماءات اليد وتصنيفها، مثل استخدام القفزات التي تحتوي على عدة أجهزة استشعار أو اعتماداً على رؤية الحاسوبية. في هذا العمل، سوف نعتد على رؤية الحاسوبية بدلاً من استخدام القفزات للتحكم في حركة الروبوت، وذلك لعدة أسباب منها أن استخدام القفزات يتطلب توصيلات كهربائية معقدة تقيد حركة المستخدم، وكذلك إصلاح أجهزة الاستشعار المرتبطة بالقفزات مكلفاً إذا تعرضت للتلف، وأخيراً القفزات معرضة لنقل الأمراض الجلدية بين المستخدمين. بناءً على رؤية الحاسوبية يتم استخدام طريقة mediapipe وهي طريقة حديثة تم اكتشافها بواسطة Google. يتم تلخيص هذه الطريقة من خلال الكشف عن إيماءات اليد وتصنيفها من خلال تحديد 21 نقطة ثلاثية الأبعاد على اليد، ومن خلال مقارنة أبعاد تلك النقاط، يتم تصنيف إيماءات اليد. بعد اكتشاف إيماءات اليد وتصنيفها، يتحكم النظام في حركة الروبوت من خلال إيماءات اليد، حيث أن كل إيماءة يد لها امر محددة يقوم من خلالها الروبوت بأداء حركة معينة. في هذا العمل، لخصت بعض الفقرات المهمة إلى أن طريقه MP أكثر دقة وأسرع في الاستجابة من طريقة التعلم العميق (DL)، وتحديدًا شبكة الالتفاف العصبية (CNN). أظهرت النتائج التجريبية في الوقت الفعلي أن دقة هذه الطريقة من خلال تأثير العناصر البيئية مثل شدة الضوء والمسافة وزاوية الميل (بين إيماءة اليد والكاميرا) تقل في بعض الحالات عندما تتغير العوامل البيئية والسبب في ذلك يرجع إلى جودة الكاميرا المستخدمة وكذلك وضعيات إيماءات اليد الغير منتظمة (تقارب الاصابع أو بعض أصابع اليد ليست مغلقة أو مفتوحة بشكل كامل مما يؤدي إلى عدم قدرة الخوارزمية المستخدمة في تصنيف إيماءات اليد بشكل صحيح (انخفاض دقة التصنيف)، وبالتالي زيادة وقت الاستجابة لحركة الروبوت، مما يجعل من الصعب على النظام تحديد ما إذا كان الإصبع مغلقاً أم مفتوحاً.