Hand Gestures Replicating Robot Arm based on MediaPipe

Muneera Altayeb
Communications and Computer Engineering Department
Al-Ahliyya Amman University, Al-Saro, Al-Salt, Amman, Jordan
m.altayeb@ammanu.edu.jo

Article Info

Article history:

Received Feb 6, 2023
Revised Aug 22, 2023
Accepted Sep 2, 2023

Keywords:

Image processing
Arduino
Hand Gesture
Robot Arm
MediaPipe
Landmarks

ABSTRACT

A robotic arm is any variety of programmable mechanical devices designed to operate items like a human arm and is one of the most beneficial innovations of the 20th century, quickly becoming a cornerstone of many industries. It can perform a variety of tasks and duties that may be time-consuming, difficult, or dangerous to humans. The gesture-based control interface offers many opportunities for more natural, configurable, and easy human-machine interaction. It can expand the capabilities of the GUI and command line interfaces that we use today with the mouse and keyboard. This work proposed changing the concept of remote controls for operating a hand-operated robotic arm to get rid of buttons and joysticks by replacing them with a more intuitive approach to controlling a robotic arm via the hand gestures of the user. The proposed system performs vision-based hand gesture recognition and a robot arm that can replicate the user's hand gestures using image processing. The system detects and recognizes hand gestures using Python and sends a command to the microcontroller which is the Arduino board connected to the robot arm to replicate the recognized gesture. Five servo motors are connected to the Arduino Nano to control the fingers of the robot arm; These servos are related to the robot arm prototype. It is worth noting that this system was able to repeat the user's hand gestures with an accuracy of up to 96%.

Corresponding Author:

Muneera Altayeb ,
Department of Communications and Computer Engineering,
Al-Ahliyya Amman University,
Al-Saro, Al-Salt, Amman, Jordan
Email: m.altayeb@ammanu.edu.jo

1. INTRODUCTION

Human-computer communication has always been a challenge, as people have spent more than half of the last century experimenting with different methods to interact with computers in an effort to create more effective and user-friendly interfaces. Hand gesture recognition is one of the most practical and popular solutions for increasing human-computer interaction. It gives computers the ability to capture and interpret hand gestures to execute commands. Its popularity has grown in recent years because of its uses in several applications, including robot control, human-computer interaction (HCI), accessibility assistance, virtual environment control, sign language translation, home automation, and musical composition. Hand gesture recognition systems have evolved tremendously in the last few years because of their ability to interact with machines efficiently and the Electromyography (EMG) hand gesture classification yields precise findings for gesture identification. The main approaches to hand gesture research can be classified into the wearable glove-based sensor approach and the camera vision-based sensor approach. Human gesture recognition is applied in many research areas due to the growing need for a more sustainable, intelligent, and reliable system [1,2]. It is also worth noting that hand gesture recognition based on digital image processing using MATLAB is one of the most popular methods that has shown great success. The idea behind this method is to use digital images captured by cameras and analyze these images to identify hand movements [3].

Chaya et al.[4] provided real-time replication of arm movements using surface EMG signals. The idea of this method is to use features extracted from EMG signals and then classify these features by Support Vector Machines - SVM and Related Vector Machines (RVM) to control the movement of the two main joints wrist and elbow to repeat nine arm movements, the average accuracy was 93.3% using SVM and 67.88% using RVM.

Dheeban et al. [5] introduce a gesture-controlled robotic arm. The robotic arm moves and does the work based on human hand gestures, and this technology mimics human hand activities. This is an accelerometer-based gesture detection system that uses the ZigBee protocol to control the movements of a robotic arm through wireless control. The system consists of two parts: the sending and receiving parts. On the transmitting part, an accelerometer is used to detect the 3-axis movement of the robotic arm (x, y, z). This accelerometer is attached to a sensor glove that can be worn by a person. Based on the movement of a human hand, this accelerometer produces three analog values (x, y, and z). This accelerometer reading is sent to the Arduino Uno board's analog input ports. The sensor glove's fingers have slide potentiometers attached to them. They're also fed into the analog pins on the Arduino board. The sliding potentiometer value varies as the fingers move, and this may be utilized to identify finger motions. The Zigbee module is attached to the Arduino board with the help of an Arduino shield, and then the data obtained from the slide potentiometer and the accelerometer is sent wirelessly through the ZigBee module. The Zigbee module sends the data wirelessly to another ZigBee module, which is placed on the controller in the robotic arm. At the receiving end, a ZigBee module connected to an Arduino Mega through a ZigBee shield receives the data. Using an Arduino mega board and the data collected from the ZigBee, the motors are then moved. The actuation methods employ servo motors. The PWM output ports on the Arduino Mega supply the PWM signal for the servo motors. The values obtained from the related ZigBee module are used to calculate the PWM. The robotic arm's fingers are controlled by five micro servo motors, and the arm's overall movement is provided by three servo motors.

Flex Sensors were used by the authors of [6] to create a Gesture Control Robotic Arm. The robotic arm's mechanical framework is composed of a cheap, readily available plastic sheet. Its constructed in such a way as to accommodate the actuators and the control circuit. The fingers have three connections, which makes their motion resemble that of a human finger. The opposite thumb was created with a pivot joint that allowed it to be attached to the palm. Then, it behaves as a human thumb would. The foundation is then secured to plywood, and the entire component is then fastened to a plastic sheet to keep it in place.A triple-axis accelerometer and five flex sensors are fitted to the human hand glove to regulate hand movement. Using a ZigBee transmitter-receiver module, the entire portion was then made wireless, allowing it to be controlled from a distance. The sensors mounted to the hand glove generate an analog voltage when the user moves the arm or fingers. This analog voltage is sent into the microcontroller's built-in ADC.

In Saurabh et al.'s work [7], the design of a Wireless Gesture Controlled Robotic Arm was developed. The suggested system architecture involves connecting a camera to a laptop computer. A computer is used to operate the gesture recognition system. A pair of wireless communication modules are connected to the robot controller and the gesture recognition system, respectively. The visual data of the different hand movements are obtained via the camera. The picture or video used as input may be noisy, or the recognition of the surrounding area as a hand region may limit performance. The obtained data is enhanced and subsequently processed so that it may be approximated with motions (data) contained in the database. The data is then analyzed to recognize the gesture. Each motion corresponds to a separate instruction for controlling the robot. After that, a wireless module is employed to dispatch these various robots. Then, using the wireless module, these various robot control commands are sent to the robot controller. As a result, the robotic arm will perform movements in response to various human hand motions, allowing for human-robot interaction. The MATLAB tool was used to create the gesture recognition system.

Chaitanya et al. [8] introduced Home Appliance Control Based on Hand Motion Gesture. Their study focuses on the existing use of gestures in household appliances as well as their potential applications in other sectors. In houses, factories, and workplaces, a variety of appliances are utilized. This research focuses on the analysis of a distorted electric field for enhanced proximity detection caused by hand motions. By detecting, tracking, and classifying the user's hand or finger motion in free space, it enables the creation of new user interface applications. The goal of their project was to use free-air hand gesture motion technologies to operate a gadget. It makes use of the hand gesture pad, which operates on the e-field distortion theory. Any gadget may be controlled by the user using this technology. In the electronics business, it may provide dependable help and security. For data analysis, an ARM7-based microcontroller is used, and RF communication is used to communicate gestures from the pad to the device section. To complete the application, the embedded C programming tool is employed. The system is divided into two parts: gesture recognition and controlling part.

Gesture recognition consists of a gesture pad, an LPC2148 microcontroller, an RF encoder, and an RF transmitter. The IC3130, which is based on Microchip's MGC3X30, recognizes gestures. It is based on 3D sensor technology, which employs an electric field (E-field) for improved proximity detection. The gesture sensor's output data is delivered to the microcontroller, which analyzes it before sending it to the RF transmitter. The RF transmitter requires an encoder to maintain isolation between the controller and the transmitter. The RF receiver, the LPC2148 controller, and the relay circuit are used to control household appliances. The decoder circuit decodes the data from the RF transmitter and sends it to the microcontroller for analysis. The device is controlled using the analyzed data.

Authors in [9] proposed a design of a robotic arm that reacts to human hand movements wirelessly, which means replicating the human hand movement. The integrated system consists of a three-dimensional image sensor device with a robotic arm manipulator via the Robot Operating System (ROS), where a 3D camera is used to retrieve the position of the user’s arm. The robotic manipulator consists of Kinetic and Kinova arms, which are connected to a PC using a USB connection. The user can control the robotic arm’s specific joints using his right hand. A Kinect is a sensor that detects its surroundings. It has a variety of sensors, including an RGB camera, a depth sensor, and a microphone array. The infrared picture captured by the depth sensor is converted into a point cloud by the built-in algorithm using an IR camera and an IR structured-light laser projector.

Rao and Gattani [10] provide a study of integrating a system for operating an 8-DoF (degrees of freedom) robotic arm to create telerobotic applications with human-like movements. A human upper limb action was recorded and applied to the robot arm utilizing a motion capture technique. The activities that can be performed by the robotic arm are reaching, picking, holding, and dumping an object. The system consists of a Blender Game Engine, motion capturing using Microsoft Kinect V2, and an interactive platform consisting of a biomimetic robot arm with a shoulder, elbow, wrist, and gripper. The Arbotix-M microcontroller serves as an interpreter for the robot's signals sent via serial connection. An Arduino microcontroller is set up to handle the feedback data from the constructed motors of the robot arm. An I2C connection is made with the Arbotix microcontroller.

Ana Cisnal et al. [11] introduced the RobHand (Robot for Hand Rehabilitation) system, which is an exoskeleton that uses a custom-made EMG real-time embedded technology to provide EMG-driven bilateral assistance. RobHand has been given a threshold non-pattern recognition EMG-driven control that identifies healthy hand motions and copies them on the exoskeleton put on the paretic hand. The system uses a LAUNCHXL-F28069M microcontroller. The EMG-driven mode directs the action of the exoskeleton based on EMG-processed signals acquired from the forearm muscles that govern the hand's opening and closing motion, as desired by the patient. Controlling the exoskeleton requires the detection of hand motion or intention of motion, notably hand opening and closing. This is accomplished by capturing the EMG signals of the muscles that control the flexion and extension of the hand's fingers. This system achieved 96% overall accuracy in detecting hand gestures and indicates the adequate temporal response of the system.

Liu Kai et al. [12] propose an exoskeleton robot replicating human arm reaching movements by proposing a general principle for exoskeleton robots based on postural synergy to produce human-like movements with fewer actuators and so drastically simplify the mechanical system while also fitting the limited bandwidth of the bio-interface, which can account for more than 80% of the variation. A bio-interface with a limited bandwidth can be used to control the upper-limb exoskeletal rehabilitation robot since fewer joints' motions need to be calculated concurrently and continuously. A prototype of a multi-DOF exoskeletal rehabilitation robot with only two actuators has been created as an application of the suggested strategy. Due to the linked design of the five actuated joints, the exoskeleton robot may be controlled by only two actuators and only two channels of peripheral signals. This significantly reduces the complexity of the exoskeleton robot's control system, making it simple for patients to operate, and the decrease in control signal channels will aid in the use of bio-signals such as EMG during real-time control of the exoskeleton robot.

The novelty in the idea of this project is to change the concept of remote controls to being manually automated, eliminating buttons and control levers, and replacing them with a more intuitive approach to controlling a full robotic arm via the operator's hand gesture. The gesture-based control interface offers a lot of opportunities for more natural, intuitively comprehensible, configurable, and easy human-machine interaction, and it can expand the capabilities of the common graphical and command line interfaces that we use today with the mouse and keyboard. As a result, advanced hardware and software approaches to hand-gesture recognition are critical for a wide range of 3D applications, including computer and robot control, interaction with a computer-generated environment (virtual or augmented reality), sign language comprehension, gesture visualization, game control, and the enhancement of disabled people's communication abilities.

2. METHODOLOGY

The idea of the proposed project is to implement a vision-based hand gesture recognizer system and a robot arm that can replicate the user's hand gestures using image processing. The system will detect and recognize the hand gestures using Python and send a certain command to the arm robot to replicate the recognized gesture. The hand gestures will open and close the robotic arm's fingers. The system will use a camera that will be connected to a PC to detect hand motions and image processing through Python. The PC will send a certain command to a microcontroller, Arduino Nano. The Arduino Nano will be connected to the PC via USB. Five servo motors will be connected to the Arduino Nano to control the robot arm fingers; these servos will be connected inside the robot arm prototype. The proposed system relies in its design on building a circuit consisting of hardware and software components as shown in Fig. 1, which describes the system block diagram. Briefly, once the cam detects the user's hand gesture, the PC will process the image and send the command to the Arduino to enable the robot arm to replicate the hand's movements through controlling signals that will be sent to the servomotors.

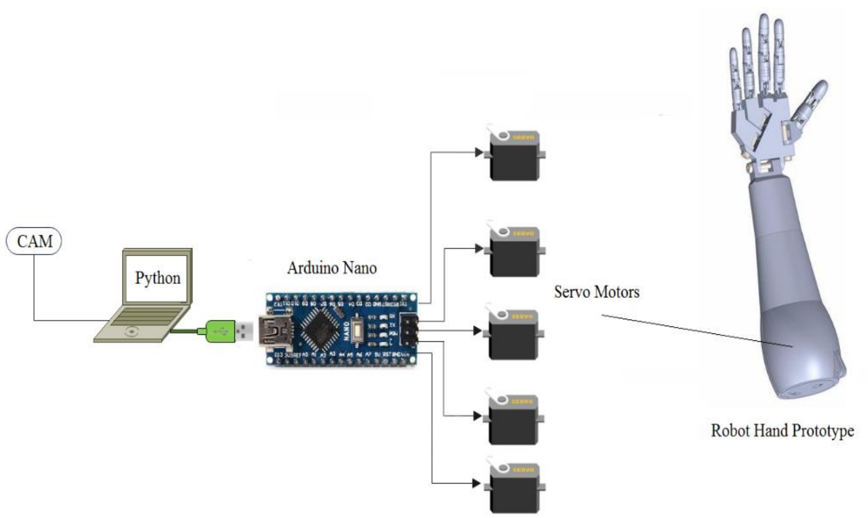

Figure 1. System block diagram

.

2.1. Hardware Requirements

The proposed system for this project is based on a set of components, which are:

- Arduino Nano
- Servo Motors
- NANO IO Shield
- Robot Arm Prototype
- Webcam
- Buck converter (LM2596)
- PC- (RAM 16GB, CPU core i7)

2.2. Overall System Connection

The robot arm prototype is shown in Fig. 2, Each finger is attached to a servo motor, and the servo motor is connected to the Arduino Nano using a NAaNO IO shield board. The Arduino Nano should be connected to the PC to receive the commands according to the human hand gestures captured by the camera.

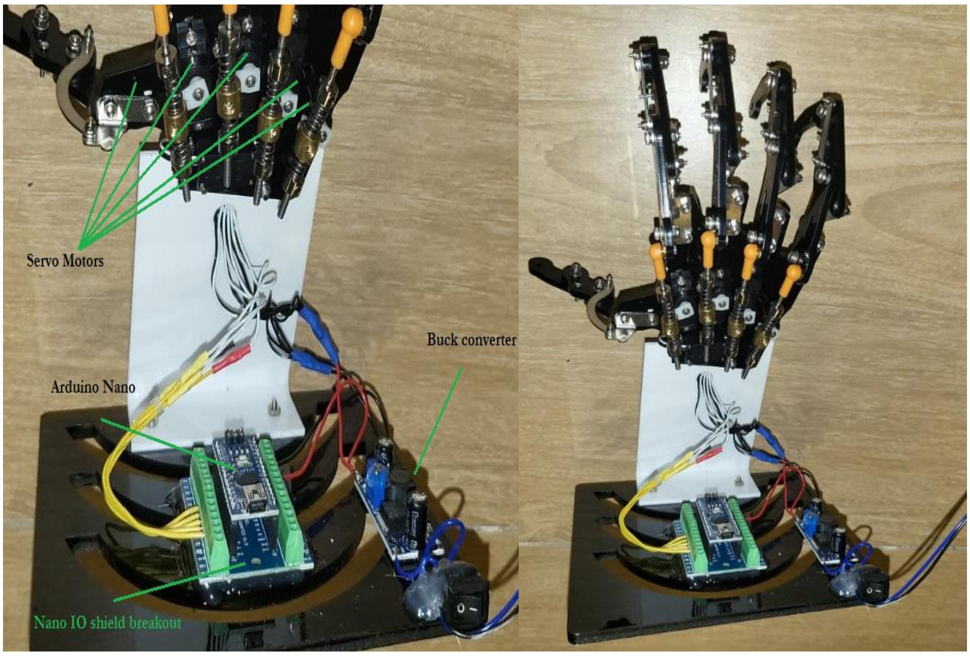

Figure 2. Robot arm prototype.

The servo motors and the Arduino Nano should be powered using +5v. This is done by using a buck converter (LM2596) [13], which is used to convert the 12v (DC) to a lower level of +5v (DC) as shown in Fig. 3. After implementing the power supply circuit, the other components' connection to the Arduino Nano is shown in Fig.4. The Arduino Nano should be connected to the PC in order to receive the commands according to the human hand gesture captured by the PC cam. Because the Arduino contains an internal USB-to-Serial converter, this enables the Arduino to directly interface with the computer by using a mini- b USB cable via the Arduino Nano mini- b USB port. The USB to serial converter, as its name implies, converts the USB connection to the 5V (serial) which TX (D1) and RX (D0) of the Arduino require for communication for sending and receiving commands to/from the PC. Then, 5 servo motors are attached to the robot arm's fingers to move them according to the user hand gesture detected. Each servo has 3 wires; it's powered via 2 wires while the third is the signal wire which is connected with the Arduino Nano pin. These servos are controlled via PWM signals received from the Arduino board.

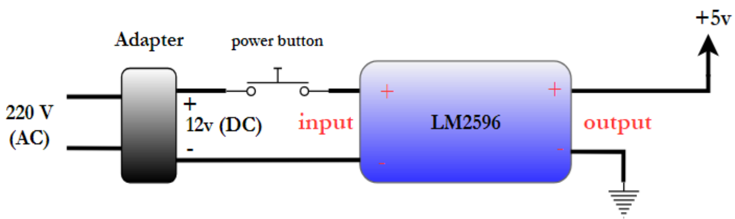

Figure 3 . Power supply schematic circuit

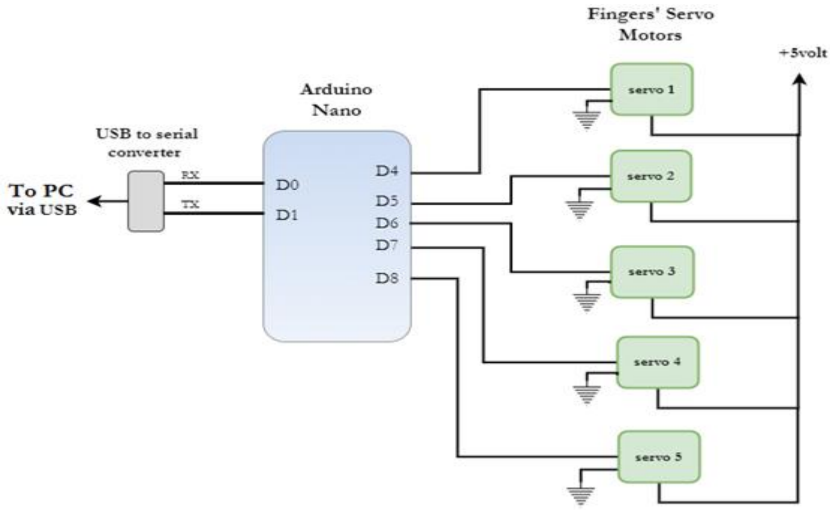

Figure 4. Overall system connection schematic circuit.

2.3. Software Framework:

The project is accomplished by interfacing the software architecture with the hardware implementation. According to the software architecture, the Arduino IDE is used to program the Arduino Nano using the C language, while the PyCharm IDE is used for writing the image capturing and processing programs using the Python language [15,16]. The camera is used to capture the image, which is then processed using Python code. The algorithm used to detect the hand gesture is hand identification, using MediaPipe and OpenCV libraries for hand gesture tracking and image processing. Using real-time picture analysis, the MediaPipe hand's algorithm detects which palm corresponds to which hand [17,18]. MediaPipe splits gesture landmarks into two parts: palm detection and key points landmark detection. The palm detector reduces the image size by cutting out the gesture part from the original image to improve the efficiency of key-point detection. Palm detection is followed by the mapping of 21-point dots as shown in Fig. 5. Then, a line is drawn connecting the points, and the angle between them is calculated. After that, the acquired data from processing the image of the user's hand is sent to the Arduino, which then sends commands to the servo motors to move and replicate the gesture [19].

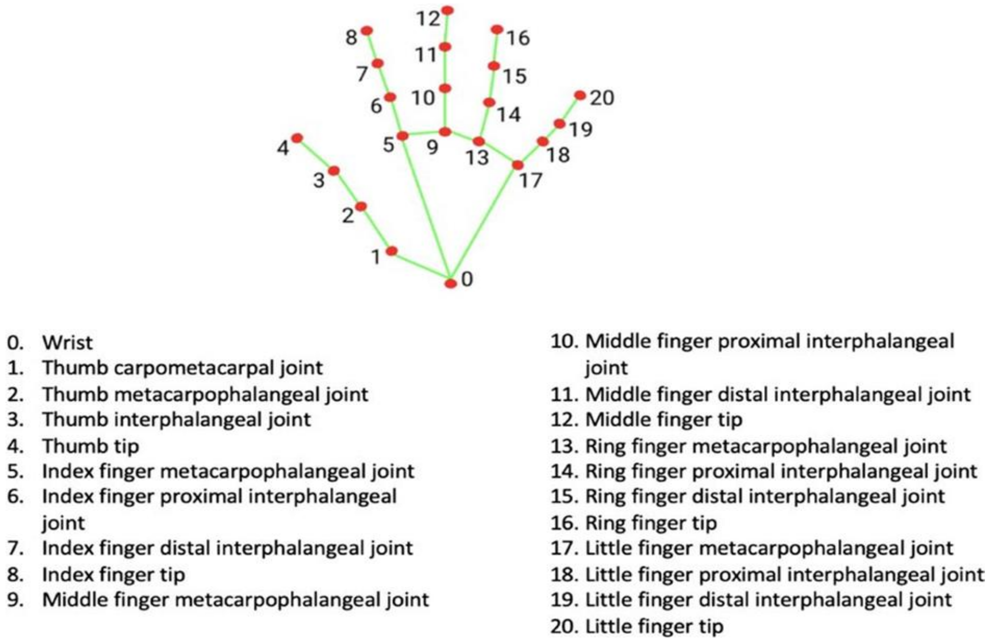

Figure 5. The 21 Hand landmarks.[14]

2.3.1. System Flowchart

The system working flowchart in Fig. 6 describes the system function; when the system starts, the PC should be connected to the Arduino board via a USB cable. Then the robot arm fingers move to the open hand position. Then the system reads the frame from the camera and checks if a hand is detected or not. When a hand is detected, it checks if the hand landmarks are detected. When the hand landmarks are detected, the system calculates each finger angle and sends it to the Arduino serially which then sets the PWM signal for each servo motor to move and replicate the user's hand gesture.

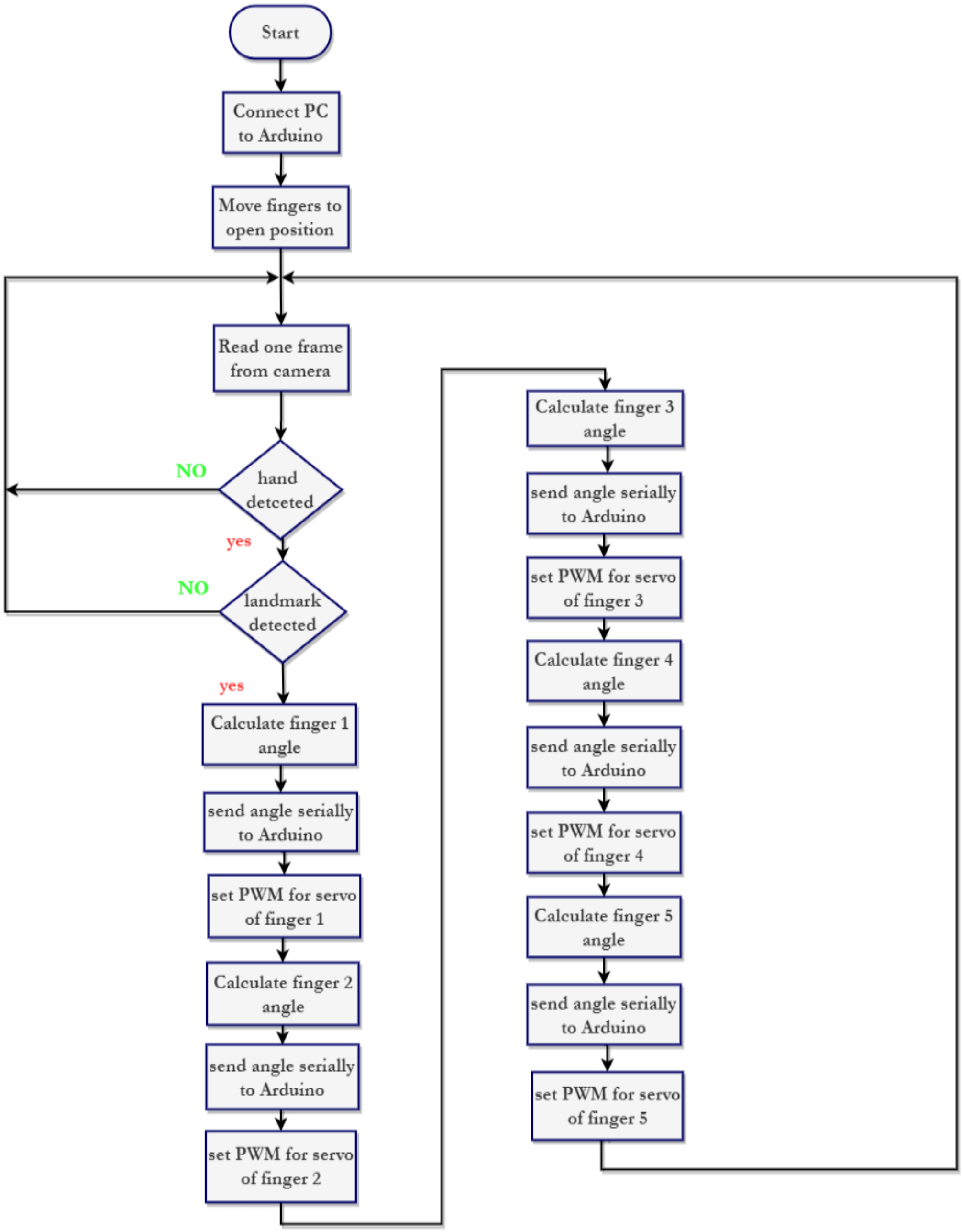

Figure 6. System working flowchart.

When a hand is detected, the system checks the landmarks of the hand as shown in Fig. 7, sequentially calculates each finger angle, and sends it to the Arduino, then sets a PWM signal for each servo motor to move and repeat the user's hand gesture.

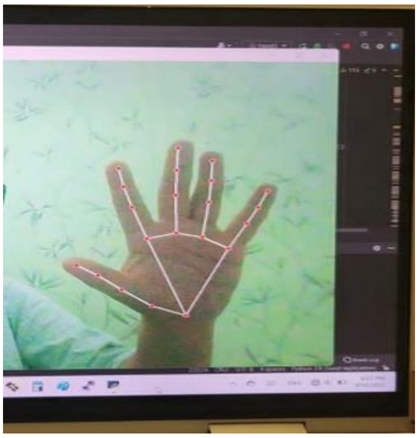

Figure 7. Hand's landmarks

The steps of hand tracking and landmarks estimation in Python are given by the code below:

```
import cv2
import mediapipe
drawingModule = mediapipe.solutions.drawing_utils
handsModule = mediapipe.solutions.hands
capture = cv2.VideoCapture(0)
withhandsModule.Hands(static_image_mode=False
,min_detection_confidence=0.7,min_tracking_confidence=0.7, max_num_hands=2) as hands:
```

The MediaPipe module's function will be used to estimate the hand landmarks from the camera frames obtained through the use of the cv2 module. One small detail that must be taken into account is that while the OpenCV frames collected in the previous call were returned in BGR format, this method requires a picture in RGB format.

```
Resulting_image= hands.process(cv2.cvtColor(frame, cv2.COLOR_BGR2RGB ))
```

3. RESULTS AND DISCUSSION

This robot arm is different because it works in real time without any delay. Artificial intelligence based on image processing techniques has been applied to detect and recognize hand gestures. Accuracy was evaluated by testing the robot arm under different conditions (finger movements, distance, background color). In the finger movements, different movements such as opening/closing were applied to each finger as shown in Fig. 8. On the other hand, the distance test was applied by detecting the hand at different distances (0.5,1.0, 1.5) m. The last test was performed to detect the hand according to the different background colors. It should be noted that each test condition was repeated 10 times to determine the accuracy of the system. Table 1 shows the working accuracy of the robot arm in different conditions, reaching about 96%.

Table 1. The accuracy was obtained for different conditions.

Testing Case	No. of Tests	No. of Correct responses	Ac%
Finger- movements	10	10	100%
Distance 0.5 m	10	10	100%
Distance 1.0 m	10	9	90%
Distance 1.5 m	10	8	80%
Red background	10	10	100%
Black background	10	10	100%
White background	10	10	100%
Accuracy (AVG)			95.72%

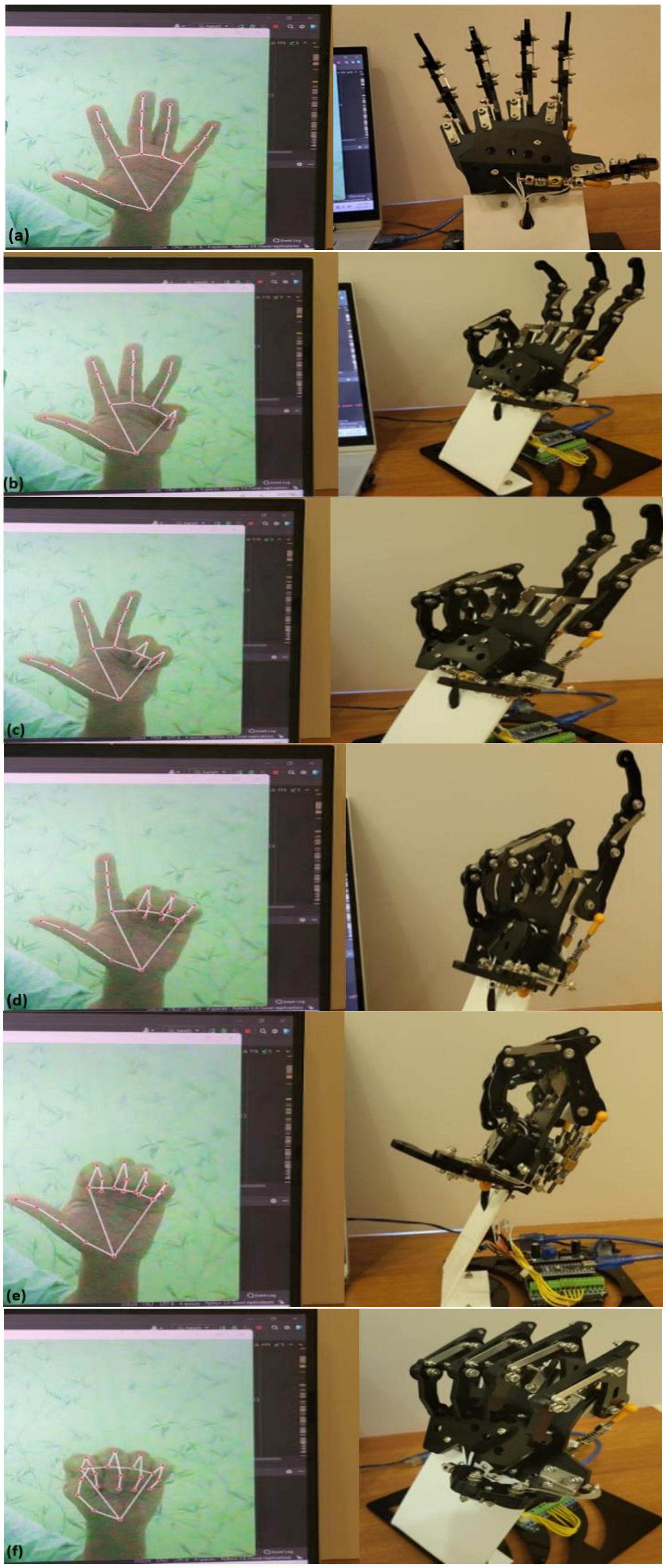

Figure 8. (a) All fingers open, (b) Little finger closed, (c) Little and ring fingers closed, (d) Little, pinky, and middle fingers closed, (e) Thumb closed, (f) All fingers closed.

4. CONCLUSION

Experimental results show that the proposed model of the robot arm using AI-based image processing techniques supported by MediaPipe and OpenCV libraries has very good accuracy, about 96%. In future research, the number of hand gestures that are recognized will be increased and the robotic arm will be allocated to different applications, including serving people with special needs. Building a robotic arm system that can recognize the user's hand gestures through computer vision and image processing is the main goal of this project. Advances in machine learning have huge benefits in all areas of computer-aided systems. Table 2 represents a summary of the performance of the proposed system compared to previous studies as reported in the literature review. The results of the model described in the current work showed its superiority over the results of previous models.

Study/year	Percentage of accuracy
[4]/2019	93.3%
[11]/2021	96%
[12]/2018	80%
[13]/2021	95%
[14]/2020	92%
Proposed	96%

ACKNOWLEDGMENTS

Special thanks to Saad Al-Ammar, Hashim Faisal, and Faisal Shamkhi for contributing to this work by performing experiments to validate the accuracy of the robot arm.

REFERENCES

[1] Oudah, M., Al-Naji, A., & Chahl, J. (2020). Hand gesture recognition based on computer vision: a review of techniques. journal of Imaging, 6(8), 73.

[2] Ansar, H., Jalal, A., Gochoo, M., & Kim, K. (2021). Hand gesture recognition based on auto-landmark localization and reweighted genetic algorithm for healthcare muscle activities. Sustainability, 13(5), 2961.

[3] Khan, T., & Pathan, A. H. (2015). Hand gesture recognition based on digital image processing using MATLAB. International Journal of Scientific & Engineering Research, 6(9), 338-346.

[4] Chaya, N. A., Bhavana, B. R., Anoogna, S. B., & Hiranmai, M. (2019). Real-time replication of arm movements using surface EMG signals. Procedia Computer Science, 154, 186-193.

[5] Ss, Dheeban & Dhanasekaran Velayutha Rajan, Harish & A, Hari & Marimuthu, Prasanna & N, Senthil Kumar. (2016). Gesture Controlled Robotic Arm.

[6] Afzal, W., Iqbal, S., Tahira, Z., & Qureshi, M. E. (2017). Gesture control robotic arm using flex sensor. Applied and Computational Mathematics, 6(4), 171-176.

[7] Khajone, S. A., & Mohod, D. S. (2015). VM Harne Implementation of a Wireless Gesture Controlled Robotic Arm International Journal of Innovative Research in

[8] Chaitanya, D., Suhasini, Y. A., & Nagarjun, G. (2017). Home Appliances Control Based on Hand Motion Gesture. International Journal of Emerging Technologies in Engineering Research, 5(11), 110-114.

[9] Baranowski, L., Kaczmarek, W., Panasiuk, J., Prusaczyk, P., & Besseghieur, K. (2017). Integration of vision system and robotic arm under ROS. Engineering Mechanics, 114-117.

[10] Rao, M. An integrated system design interface for operating 8-DoF robotic arm.2021 The 21st International Conference on Control, Automation and Systems (ICCAS 2021) Ramada Plaza Hotel, Jeju, Korea, Oct. 12∼15, 2021

[11] Cisnal, Ana, et al. "RobHand: A Hand Exoskeleton With Real-Time EMG-Driven Embedded Control. Quantifying Hand Gesture Recognition Delays for Bilateral Rehabilitation." IEEE Access 9 (2021): 137809-137823.

[12] Liu, Kai, et al. "Postural synergy based design of exoskeleton robot replicating human arm reaching movements." Robotics and Autonomous Systems 99 (2018): 84-96.

[13] Altayeb, M., & Al-Ghraibah, A. (2023). Arduino Based Real-Time Face Recognition And Tracking System. International Journal, 12(4).

[14] Harris, M., & Agoes, A. S. (2021, November). Applying Hand Gesture Recognition for User Guide Application Using MediaPipe. In 2nd International Seminar of Science and Applied Technology (ISSAT 2021) (pp. 101-108). Atlantis Press.

[15] Joshi, P. (2015). OpenCV with Python by example. Packt Publishing Ltd.

[16] Hughes, J. M. (2016). Arduino: a technical reference: a handbook for technicians, engineers, and makers. " O'Reilly Media, Inc.".

[17] Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C. L., & Grundmann, M. (2020). Mediapipe hands: On-device real-time hand tracking. arXiv preprint arXiv:2006.10214.
[18] Zhang, Y., Pu, X., Wang, X., Guo, H., Liu, K., Yang, Q., & Wang, L. (2022, November). Design concept of sign language recognition translation and gesture recognition control system based on deep learning and machine vision. In Second International Conference on Optics and Communication Technology (ICOCT 2022) (Vol. 12473, pp. 494-498). SPIE.
[19] Liu, Z., Pan, C., & Wang, H. (2022). Continuous Gesture Sequences Recognition Based on Few-Shot Learning. International Journal of Aerospace Engineering, 2022.

BIOGRAPHY OF AUTHORS

Muneera Altayeb earned her BSc in Computer Engineering in 2007, and MSc in Communications Engineering from the University of Jordan in 2010. She is currently Assistant Dean of the Faculty of Engineering at Al-Ahliyya Amman University (AAU) and she has been a lecturer in the Department of Communications and Computer Engineering at (AAU) since 2015. Her research interests are focused on the following areas: Digital Signals & Image Processing and Machine Learning.