Robotic Arm Control Using Computer Vision and Machine Learning

Kennymar Bezerra de Oliveira[1], Geraldo Almeida Neto[1], Franklin Thiago Pereira Correia [1], Richardson Menezes[1], Orivaldo Vieira de Santana Junior[1], Helton Maia*[1]

[1]Federal University of Rio Grande do Norte, Natal, Brazil (helton.maia@ufrn.br)

Abstract: In this project, computer vision, 3D printing, and Arduino were integrated to control a robotic arm, replicating the movements of the human arm. Using OpenCV, we capture real-time images from a webcam or smartphone and identify key points, such as hand and arm, through the MediaPipe framework. Control of the robotic arm, manufactured by 3D printing, is carried out by Arduino, based on an intelligent system that associates the user's movements to the movements of the robotic arm.

Keywords: Computer vision; 3D printing; Robotic arm; Real-time; Intelligent system.

INTRODUCTION

Robotic arms have gained increasing acceptance across a variety of industries, including manufacturing, healthcare, and home automation. These versatile machines are capable of performing complex tasks with precision and efficiency, which makes them invaluable in modern technological advancements. It remains, however, a significant challenge to control robotic arms to mimic human arm movements. (Santello et al., 2016).

Combining computer vision, machine learning, and 3D printing technologies presents a promising solution for achieving more intuitive and natural control of robotic arms. By leveraging computer vision techniques, such as real-time image capture and object detection, the movements of the human arm can be accurately tracked and analyzed. Machine learning algorithms can then interpret these movements and generate corresponding control commands for the robotic arm (Asadi-Aghbolaghi et al., 2017; De Menezes et al., 2019).

This paper presents a novel approach to controlling a robotic arm using computer vision and machine learning techniques. To enable seamless interaction between human and robotic arms, we propose an integrated system that combines the power of the Open-Source Computer Vision Library (OpenCV), MediaPipe, and Arduino (OpenCV, 2023; MediaPipe, 2023; Arduino, 2023). Real-time images are captured using a webcam or smartphone, and key points of interest, such as the hand and arm, are detected using OpenCV and the MediaPipe framework. An Arduino board is used to control the robotic arm, which is fabricated using 3D printed parts. The board processes the input from the computer vision system to generate precise motor commands.

This research aims to develop a robust and intuitive control system that enables the robotic arm to reproduce the movements of the human arm accurately. We aim to improve the usability and versatility of robotic arms in various applications by bridging the gap between human motion and robotic motion.

MATERIAL AND METHODS

Based on the basic patterns of human hand movements, it was possible to create a logic that connects these actions to the robotic arm, which consists of articulated parts capable of moving through servo motors that control each part separately. The robotic arms structure includes a gripper at the end, which is used for object manipulation. There are also two vertical bars responsible for vertical movement, a rotating base, and a fixed wood base. To assist the development of robotics and automation research in our laboratory, all components of the robotic arm were 3D printed. The printer was constructed using a tinybee Mks controller board, which facilitates the control and usage of such printers.

Initially, we chose Jupyter Notebooks because they provided an extensive environment for the development of tools based on the Python programming language. A virtual environment was created and OpenCV, Mediapipe, and PySerial libraries were installed and imported. This enabled the creation of a script that utilized specific hand reference points provided by Mediapipe and drew them in real-time on each frame of the webcam video captured using OpenCV. Figure 1 provides an example of this process of capturing the hand in a delimited area and displaying the points.

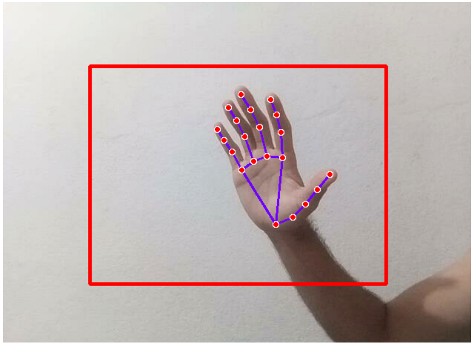

Figure 1. Example of detecting the open right hand and its points of interest using MediaPipe.

The implemented algorithm considered landmarks provided by MediaPipe, including the index finger, thumb, and wrist. The algorithm determines whether the hand is in a closed state by defining a minimum distance between the thumb and index finger points, as shown in Figure 2.

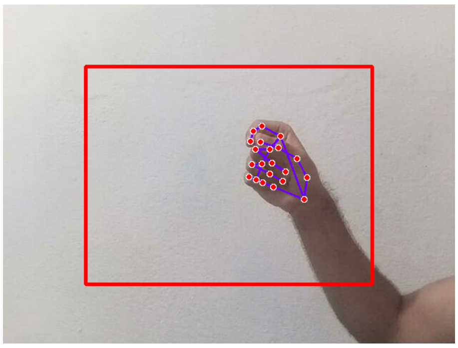

Figure 2. Closed right-hand detection for robotic arm simulation.

If the horizontal component of the distance between these fingers exceeds the minimum distance, the hand is considered to be in an open state, as shown in Figure 3. The behavior of the gripper at the end of the robotic is defined by these open and closed states.

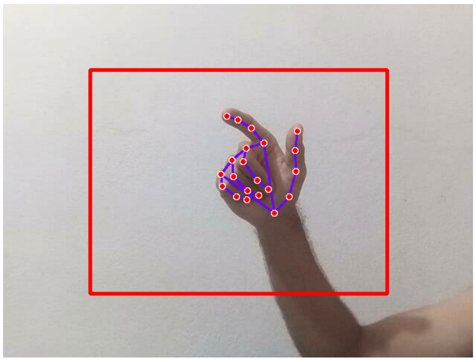

Figure 3. Open right-hand detection for robotic arm simulation.

By considering the wrist point coordinates and using imaginary lines to divide the captured frames in half, movements in the horizontal and vertical directions were defined and linked to the motors located in the vertical bars and rotating base. The algorithm uses a function from the OpenCV library to work within a rectangular area, shown in red in the processed image. The instructions for each movement and action were created, and these commands were sent to the serial port with the PySerial library.

The initial step in the Arduino development process involved the use of the online platform Wokwi, as illustrated in Figure 4, to simulate the movements of the servo motors and ensure the correct assembly of the circuit. The Servo library, which implements C++ code for controlling servo motors, was used to implement the Arduino's control logic. The library provides tools for manipulating the servos attached to the board. The servos were configured and tested by moving them to specific angles using the write method.

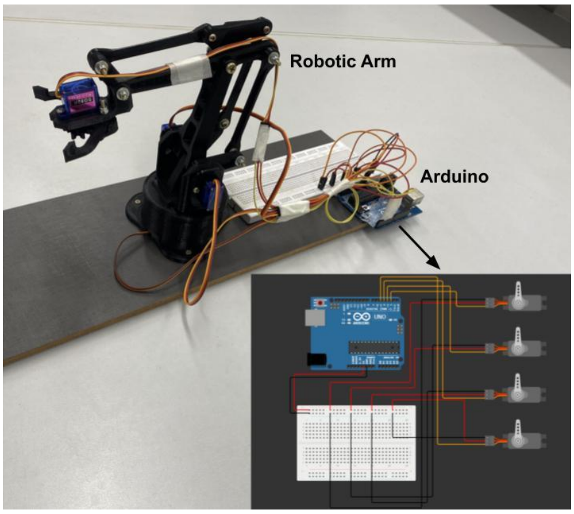

Figure 4. Robotic arm connected to arduino and simulated circuit on Wokwi platform.

As part of the development process, we created a repository on GitHub: https://github.com/heltonmaia/ECT-proj-roboticArm. This repository contains all scripts created from tests to final versions. Initial codes were implemented to detect the right hand and perform simple movements using the MediaPipe, OpenCV, and PyAutoGUI libraries, which automated the tasks.The application of PyAutoGUI was pertinent in capturing in a text file the manner in which the fingers moved, which would serve as the instructions for the serial interface in the future. Thus, improvements were achieved, and the result was satisfactory.

RESULTS AND DISCUSSION

After the system was fully implemented, tests were performed to verify the robotic arm's ability to replicate hand movements. The results were satisfactory, as the robotic arm was capable of reproducing the desired movements. The testing encountered some obstacles that were resolved throughout the process.

One of the challenges encountered was related to the connectivity of the wires used in the robotic arm circuit. Due to blockages in the operating system used, difficulties were found in properly transmitting commands through the serial port. The problem was identified and resolved, ensuring the correct transmission of commands to the Arduino.

Another area that was improved was the definition of the limit angles for the servomotor movements. A series of tests was required to determine the optimal values for achieving the desired movements without exceeding the engine speed limits. By carrying out these tests, it was possible to determine the correct angles for each movement of the robotic arm.

Even with the obstacles encountered, this project is still in progress, and it's important to emphasize the possibilities for enhancements in the prototype that's currently being developed. To increase accuracy and minimize flickering observed during repetitive movements, one could consider using motors with greater precision and mechanical stability, as well as improving the control and calibration systems.

CONCLUSION

The study demonstrated that it was possible to control a robotic arm with hand movements without the use of expensive hardware. The replication of movements was successfully achieved, despite the obstacles encountered. Continuous improvement will allow us to obtain an even more precise and functional robotic arm, which will expand its application possibilities in several areas.

ACKNOWLEDGMENTS

This study is carried out with the support of the Laboratory of Automation and Robotics (LAR) of the School of Science and Technology (ECT) at the Federal University of Rio Grande do Norte (UFRN).

REFERENCES

Santello, Marco, et al. "Hand synergies: Integration of robotics and neuroscience for understanding the control of biological and artificial hands." *Physics of life reviews* 17 (2016): 1-23.

Asadi-Aghbolaghi, Maryam, et al. "A survey on deep learning based approaches for action and gesture recognition in image sequences." *2017 12th IEEE international conference on automatic face & gesture recognition (FG 2017)*.

De Menezes, Richardson Santiago Teles, Rafael Marrocos Magalhaes, and Helton Maia. "Object recognition using convolutional neural networks." *Recent Trends in Artificial Neural Networks-from Training to Prediction* (2019).

OpenCV: Open Source Computer Vision Library. Available at <https://opencv.org/>. Accessed on 21/05/2023.

MediaPipe: Cross-Platform Framework for Perception AI. Available at <https://mediapipe.dev/>. Accessed on 21/05/2023.

Arduino: An Open-Source Electronics Platform. Available at <https://www.arduino.cc/>. Accessed on 21/05/2023.