Embedded systems and artificial intelligence for enhanced humanoid robotics applications

Jalal Hdid[1], Oussama Lamsellak[2], Ahmad Benlghazi[3], Abdelhamid Benali[3], Ouafae El Melhaoui[4]
[1]Laboratory of Electronics and Systems, Faculty of Science, Mohamed 1 University, Oujda, Morocco
[2]Laboratory of Applied Mathematics of the Oriental, Faculty of Science, Mohamed 1 University, Oujda, Morocco
[3]Department of Electronic Computer Sciences and Telecommunication, National School of Applied Sciences, Mohamed 1 University, Oujda, Morocco
[4]Faculty of Science, Mohamed 1 University, Oujda, Morocco

Article Info

Article history:

Received Apr 30, 2024
Revised Nov 29, 2024
Accepted Dec 2, 2024

Keywords:

Data acquisition
Embedded artificial intelligence
Embedded systems
Feature extraction
Hand gesture recognition
Humanoid robotics
Supervised machine learning

ABSTRACT

This paper presents a method for collecting precise hand gesture (HG) data using a low-cost embedded device for an embedded artificial intelligence (EAI)-based humanoid robotics (HR) application. Despite advancements in the field, challenges remain in deploying cost-effective methods for accurately capturing and recognizing body gesture data. The ultimate objective is to develop humanoid robots (HRS) capable of better understanding human activities and providing optimal daily life support. In this regard, our approach utilizes a Raspberry Pi Pico microcontroller with a 3-axis accelerometer and a 3-axis gyroscope motion sensor to capture real-time HG data, describing ten distinct real-world tasks performed by the hand in experimental scenarios. Collected data is stored on a personal computer (PC) via a micro-python program, forming a dataset where tasks are classified using ten supervised machine learning (SML) models. Two classification experiments were conducted: the first involved predicting raw data, and the second applied normalization and feature extraction (FE) techniques to improve prediction performance. The results showed promising accuracy in the first phase (89% max), with further improvements achieved in the second phase (94% max). Finally, by employing similar methods, we can integrate highly trained machine learning (ML) models into embedded humanoid robotic systems, enabling real-time human assistance.

Corresponding Author:

Jalal Hdid
Laboratory of Electronics and Systems, Faculty of Science, Mohamed 1 University
Mohammed VI Road - P.O. Box 717, Oujda 60000, Morocco
Email: j.hdid@ump.ac.ma

1. INTRODUCTION

The field of humanoid robotics (HR) [1], [2] is rapidly advancing on a global scale, greatly influencing various aspects of human life. Humanoid robots (HRS) powered by artificial intelligence (AI) [3] are increasingly being tested and used across multiple industries, such as hospitality, education, healthcare, and the service industry. However, despite these technological advancements, real-world HR applications [4], [5] remain limited, partly due to high development costs. Their deployment in practical scenarios demands the association of high-performance motion acquisition systems [6] and machine learning (ML) techniques [7] to accurately collect and recognize well-expressive human body gesture data [8]. The captured data must be meticulously structured into well-processed datasets to be effectively adopted and interpreted by ML algorithms. One of the main challenges is to develop cost-effective, real-time data acquisition (DA) methods that capture highly accurate gesture data for efficient ML training. The ultimate goal is to develop HRS that can improve recognition and reproduction of real-life human activities [9], thereby offering better assistance in daily tasks for an enhanced user experience.

Various studies and related works have been undertaken in an effort to contribute to the development of HRS for real-world jobs. These include hospitality services such as serving food, acting as hotel concierges or in other customer-facing roles [10], bionic applications [11], and becoming senior companions [12]. Humanoid robot prototypes are already in existence today, all around the world. Many of them have advanced to the ground phase or have even touched down in the real world. Nevertheless, Table 1 lists a few examples of the top HRS [13]–[16] that are currently employed in the sectors of deep-sea exploration, industry, sociology, and education.

Table 1. Review of existing humanoid robot systems across various sectors

Sector	Name	Description	Tasks
Industry	"ARMAR-6" [13]	- A high-performance humanoid robot for human-robot collaboration in real-world scenarios. - It recognizes humans' needs and offers help in a proactive way.	- Grasping and manipulating tools, - Perception and recognition, - Manual collaboration with humans, - Compliant-motion execution, - Natural language understanding, - Industrial maintenance assistance.
Education	"Pepper" [14]	- Humanoid robot deployed for educational purposes. - Equipped with features and high-level interfaces for multimodal communication with the humans around it.	- Exhibiting body language, - Perceiving and interacting with its surroundings, - Moving around, - Analyzing people's expressions and voice tones, - Creating content and teaching programming.
Deep-sea exploration	"The ocean one hands" [15]	- An adaptive deep-sea diver design for robust marine manipulation in unstructured environments. - Utilizing two hands with elastic finger joints and a spring transmission to achieve a variety of pinch and wrap grasps.	- Acquiring and manipulating small and large objects with one or two hands, - Grasping and handling delicate and stiff items, - Diving in the Mediterranean for its first mission, investigating a shipwreck off the coast of Toulon, France.
Sociology	"Advanced step in innovative mobility (ASIMO)" [16]	- Being the latest biped Honda's humanoid robot with its ability to have five-finger arms as well as its walking function. - It is a social robot that is designed to operate freely in a human living space to assist and improve people's lives.	- Walking over uneven surfaces and stairs in a dynamic and stable way, - Recognizes and interacts with people, - Carrying a payload of 70 kg, - Ability to compensate against external forces, - Collision avoidance, - Access things like doorknobs and light switches, - Waving and shaking hands.

For HR applications, the acquisition and recognition of gesture data can be efficiently managed using embedded systems (ES) [17] equipped with motion sensors and actuators and powered by EAI [18]. Indeed, these devices offer cost-effective and low-power solutions for real-time, dynamic DA, capturing expressive human gesture data with precision through affordable motion sensors [19]. Involving performant ML methods, well-preprocessed and structured datasets can be gathered, and enhanced AI training, prediction, and decision-making capabilities can be achieved. Moreover, thanks to the internet of things (IoT) [20] and connected objects (CO) [21] industries, real-time and autonomous ES based AI [22] can be created for acquisition, recognition, and reactive tasks using robust sensors and actuators. As a result, ES are playing a critical role in building efficient, operational EAI-based HRS [23], [24] capable of providing continuous and optimal support in daily human activities. Industries have already achieved significant innovation in computational physics [25] and engineering by leveraging these technologies [26]. Thus, microcontrollers [27], [28] like the Arduino Nano 33 BLE sense [29], [30] and Raspberry Pi Pico [31], [32] are particularly well-suited for cost-effective HR projects due to their low cost, small size, and ability to run low-power embedded ML programs [18].

In this context, this paper presents an approach utilizing a low-cost Raspberry Pi Pico microcontroller paired with a motion sensor to capture precise, real-time hand gesture (HG) data. These gestures represent ten simple, daily human tasks, which will be classified using SML models [7] for potential adoption in embedded HR applications. The methodology involves collecting multiple examples of each task's hand gestures (HGs) via the microcontroller and sensor and classifying them with various ML models. The motion sensor is a 6-axis device that records 3-axis linear accelerations and 3-axis rotational velocities across the X, Y, and Z axes [31], [33]. The data collection process consists of attaching the motion sensor to a human hand and executing the tasks in real-world scenarios by performing controlled movements while capturing real-time measurements. The captured measures are stored on a PC using the Raspberry Pi Pico running a micro-python program. The collected data is then saved in an Excel file, forming a dataset that includes multiple instances of each hand task, all of which have been performed several times.

The primary objective of this research is to develop a cost-effective method for generating a well-labeled and representative HG dataset being effectively trained and interpreted by ML algorithms. The results aim to demonstrate how well the collected data describe the ten tasks and how accurately the ML models can predict them for future usage. Two experimental phases of classification were conducted. The first experiment applied classification to raw data, minimizing resource use, to evaluate how effectively the raw gesture data represents the activities. The second experiment employed data preprocessing techniques [34], including normalization and feature extraction (FE) [35], [36] to enhance classification accuracy. In the initial experiment, five models achieved accuracies between 80%–90%, while the second experiment showed enhanced results, with six models achieving accuracies between 87%–94%. For simplicity, the Raspberry Pi Pico microcontroller will be referred to as "Pico" throughout the remainder of this article.

The article is structured as follows: section 2 describes the ten hand tasks adopted for the study. Section 3 details the methodology and materials used for data capture, with a presentation of ML models and techniques applied for classification. Section 4 shows the results achieved with the collected dataset and the ML prediction performances, along with an analysis and discussion. Finally, a conclusion summarizes the key findings.

2. ADOPTED HAND TASKS FOR THE STUDY

In this study, ten human hand tasks were designed to simulate real-world HG exercises that represent simple daily activities. Each task consists of multiple cycles of repeated HGs, with movements detected by the motion sensor and captured by the Pico. To optimize the use of a single sensor, the selected tasks are simple, one-handed actions illustrated in Figure 1. During the exercises, the fist and forearm are the only moving parts, while the fingers remain stationary. In this approach, the fist is the primary portion of the hand used for recording motions with the sensor.

A task is defined as starting with the hand in a fixed posture (fingers and fist immobile), followed by continuous movement until the cycle completes, at which point the hand returns back to its initial position before starting the next cycle. Each elementary movement or action that constitutes a task is carefully tracked, generating a real-time gesture dataset with detailed features. The data capture is processed using a PC and a micro-python program, with the results stored in an Excel file for further analysis. Figures 1(a) to 1(j) illustrates the 10 hand tasks, which are described as follows:

a. Opening/closing a door: a hand reaches for the door handle, turns it, and either pushes or pulls the door to open or close it.
b. Locking/unlocking a lock key: fingers manipulate a key, inserting it into a lock to turn and secure the door or unlock it, and pull the key after that.
c. Tapping keys: delicate index finger movements pressing keys buttons (of an electronic lock, for example, entering a code or sequence to secure or gain access).
d. Pouring a cup with coffee: a hand takes up a thermos, holds it, and carefully pours coffee into a cup, controlling the flow to prevent spills, and puts the thermos down.
e. Arranging glass cups: fingers or a hand arrange glass cups in a specific order or pattern, demonstrating dexterity and precision.
f. Cleaning a window: a hand wields a cleaning tool (squeegee or cloth) to clean the window surface sprayed with a cleaner liquid following systematic and controlled movements.
g. Panting a wall: a hand dips a paintbrush into a paint bucket and applies paint to a wall with steady and intentional strokes.
h. Hammering a nail: a hand grasps a hammer and drives a nail into a surface by delivering controlled strikes.
i. Sawing a tube: a hand guides a saw through a tube, demonstrating a precise cutting motion.
j. Tightening/loosening a bolt: fingers or a hand grip a wrench, either tightening or loosening a bolt with calculated turns.

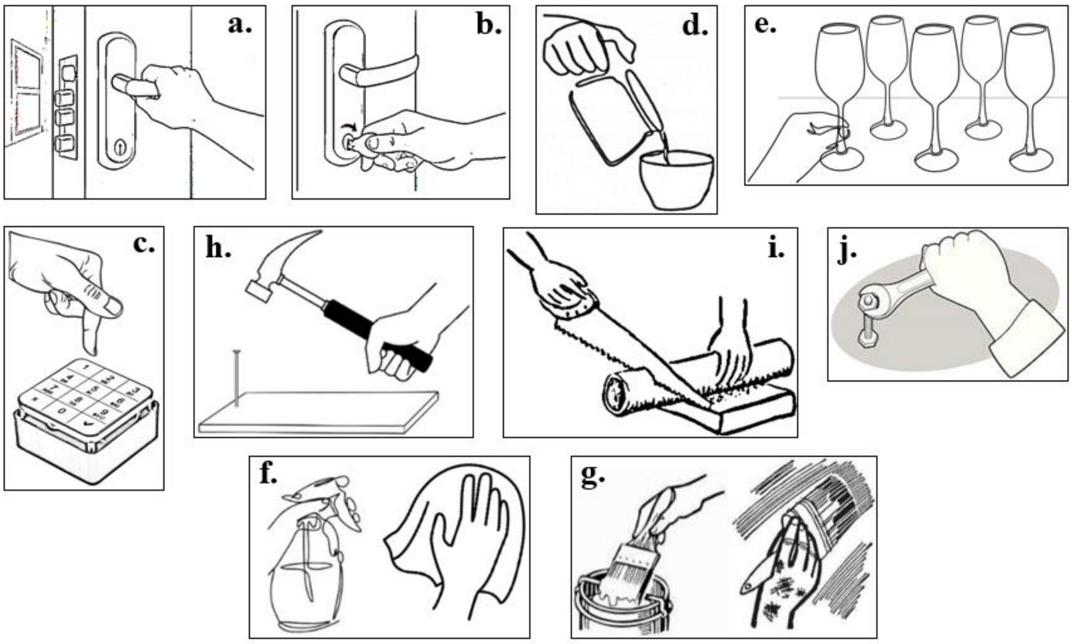


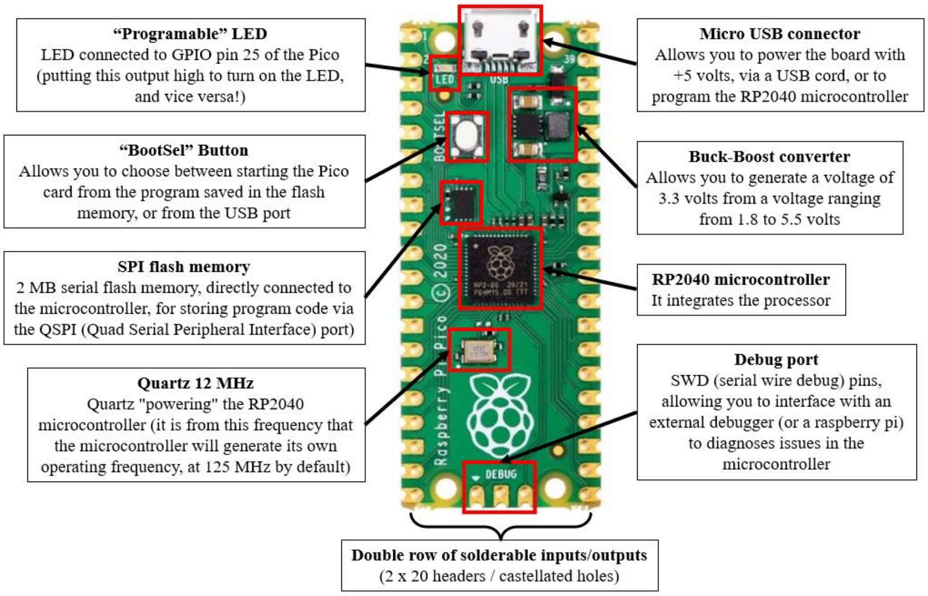
Figure 1. Illustrations representing the 10 selected hand tasks: (a) opening/closing a door, (b) locking or unlocking a lock key, (c) tapping keys, (d) pouring a cup with coffee, (e) arranging glass cups, (f) cleaning a window, (g) panting a wall, (h) hammering a nail, (i) sawing a tube, and (j) tightening/loosening a bolt

3. METHOD

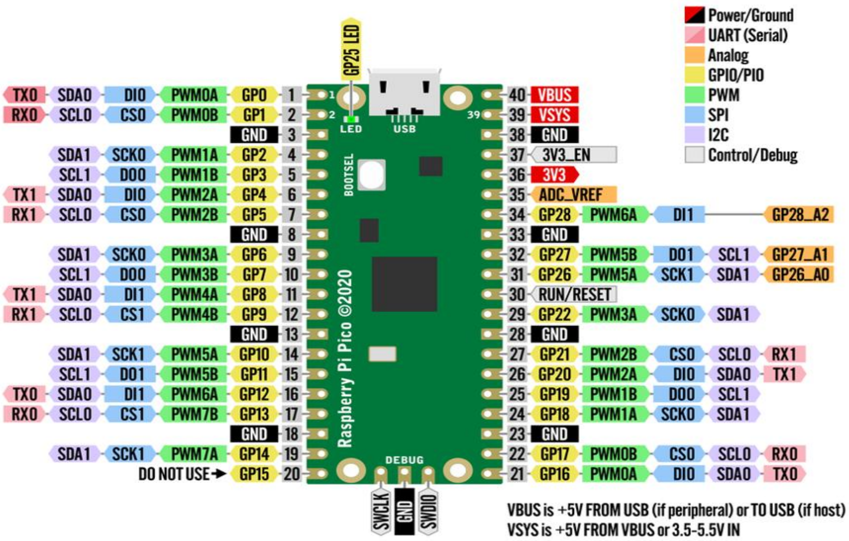
This work employs the Pico [31], [32] as the primary electronic component, integrated with a micro electro mechanical systems processor unit (MPU6050) inertial measurement unit (IMU) module [31], [33] containing accelerometer and gyroscope sensors to detect HGs. The Pico is typically part of an Experimenting Kit, which features various onboard components and modules useful for developing hardware-based connected electronic projects [21]. The kit generally consists of a development board, relay, servo motor, IMU module, potentiometer, ultrasonic sensor, Wi-Fi module, and grove-compatible jumper cables [31]. However, for this study, only the development board, IMU module, and necessary wiring are used alongside a PC and micro-python programming language. The goal is to create a connected system in which the Pico functions as the master and the IMU module as the slave, utilizing an inter-integrated circuit (I2C) communication protocol [37]. Upon running the micro-python program, sensor data is collected and stored to be preprocessed, trained, and tested using ML techniques for classification.

3.1. Raspberry Pi Pico microcontroller

The Pico is a high-performance microcontroller designed especially for computational physics [25] dealing with electronic systems that include IoT [20] projects. The Pico has its biggest advantages of being a very low-cost $4 microcontroller with more digital input-output pins, large memory, and accurate timing modules. Additionally, it has the ability to implement and run ML algorithms for low-power EAI applications [38], [39]. It is an integrated single-small board based on the RP2040 microcontroller chip with a 32-bit dual-core Cortex processor running at up to 133 MHz, 264 kB of static random-access memory (SRAM), 2 MB of on-board flash memory for storing program codes, a timer, and a real-time counter. With a micro-USB type B port for providing 5 V power to and programming the board, it can be programmed using C, C++, or micro-python languages to run a single task and can be used in fast real-time control and monitoring applications. Besides power and ground pins, it has 26 multiple-function general purpose input/output (GPIO) pins. These include 2× serial peripheral interface (SPI) [40], 2× inter integrated circuit (I2C) [37], and 2× universal asynchronous receiver transmitter (UART) [41] interfaces allowing connection and communication with other devices. The pins also include 3× 12-bit analog to digital converter (ADC) and 16× controllable pulse width modulation (PWM) channels. Figure 2 shows the Pico with its internal components in Figure 2(a) and pin interfaces in Figure 2(b). For more detailed information, refer to [31], [32].



(a)



(b)

Figure 2. Schematics of the Pico board: (a) internal components and (b) pinout interfaces

3.2. MPU6040 IMU module

The MPU6050 IMU module [31], [33] as shown in Figure 3 is a 6-axis motion detector unit consisting of a 3-axis accelerometer, a 3-axis gyroscope, and a temperature sensor. Being integrated into a single chip, the module is interfaced to and communicates with the Pico through the I2C bus [37]. The gyroscope measures the rate of rotational velocity, capturing the angular speed over time across the X, Y, and Z axes using micro electro-mechanical systems (MEMS) technology and leveraging the Coriolis effect [42]. The accelerometer, on the other hand, measures gravitational acceleration along the same three axes, and through trigonometric calculations, it can determine the angle of the sensor's orientation. Acceleration values are recorded as ax, ay, and az, while rotational velocity values are represented as gx, gy, and gz, where clockwise motion produces negative readings. As shown in Figure 3, the Z-axis is oriented perpendicular to the sensor module, while the Y- and X-axes are aligned perpendicular to the sides of the chip. Although the range of measures is from 0 to 3, acceleration data is reported in units of "g" while rotational velocity is measured in degrees per second as shown in Table 2. The fusion of accelerometer and gyroscope data enables precise tracking of the sensor's movement and orientation, which constitutes the strength of the data collection approach employed in this article.

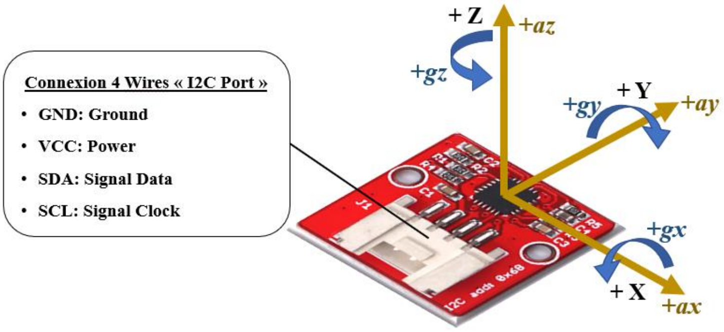


Figure 3. IMU module with its motion orientations and connection bus

Table 2. Acceleration and velocity values in units and range

Range	Acceleration value (+- g)	Rotation value (+- degs/sec)
0	2	250
1	4	500
2	8	1000
3	16	2000

3.3. Development board

The development board is an integrated platform designed to facilitate practical and efficient connection and communication between the Pico and various external devices, such as sensors and motors. As shown in Figure 4, the board includes a thin-film-transistor liquid crystal display (TFT LCD), six grove-interface compatible network sockets (labeled CN1 to CN6) [31], and a pin interface for plugging the Pico, ensuring easy access to the GPIO pins. Additionally, it features three light-emitting diodes (LEDs), three buttons, and a buzzer. External components can be connected to the board through the six sockets via jumper wires, allowing interaction with Pico's GPIO pins. For more detailed information, refer to [31].

3.4. Measurement programming

To develop a program in the Pico that measures and stores acceleration and velocity data from the IMU sensor, a few steps must be followed. First, the Pico is plugged into the development board by aligning it with its GPIO pin holes, and the IMU sensor is connected to the development board through the compatible network (CN3) socket using jumper wires as shown in Figure 4. Second, via its micro-USB-B port, the Pico is connected to a PC using a micro-USB-B/USB cable, providing 5 V power and enabling data transmission as shown in Figure 4. Finally, by downloading the Thonny environment software [31], [32], the program can be developed by creating a new file and using the micro-python program language. Upon running the program, the 6-axis values: ax, ay, az, gx, gy, and gz start being measured by the IMU sensor and stored in an Excel file on the PC.

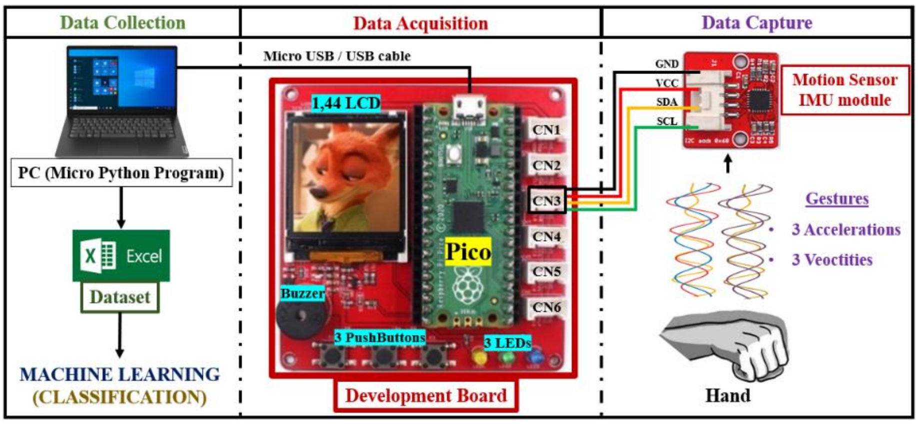


Figure 4. Schema describing the proposed data collection method

3.5. Data collection methodology

All tasks covered in this paper are one-hand actions (right or left hand) that have been performed in a caring manner using my hand. The IMU sensor was securely attached to the back of my hand and connected to the Pico, which was affixed to my forearm and linked to a PC via a micro-USB/USB cable to provide both power and data transmission as shown in Figure 5. Once the acquisition system was set up, I launched the program and began to carefully perform, for every single task, the required hand movements, including directional shifts and rotations, that define the task. While doing that repetitively in cycles, acceleration and gyroscope measurements were being recorded in real-time by the Pico from the IMU sensor and were automatically being stored and saved in an Excel file. The micro-python program operated in a continuous "while" loop, capable of recording up to 999 rows of data per session. Each row corresponded to a sample consisting of ax, ay, az, gx, gy, and gz measures. For each task, the program ran twice, collecting data in two sessions, resulting in 1,998 total samples per task. The data were collected at a frequency of 100 Hz with a time interval of 0.01 seconds. For tasks involving the tipping pad and unlocking/locking door actions, the IMU sensor was placed on the index finger, while for the rest of the tasks, it was positioned near the middle knuckle of the hand, as shown in Figure 5.

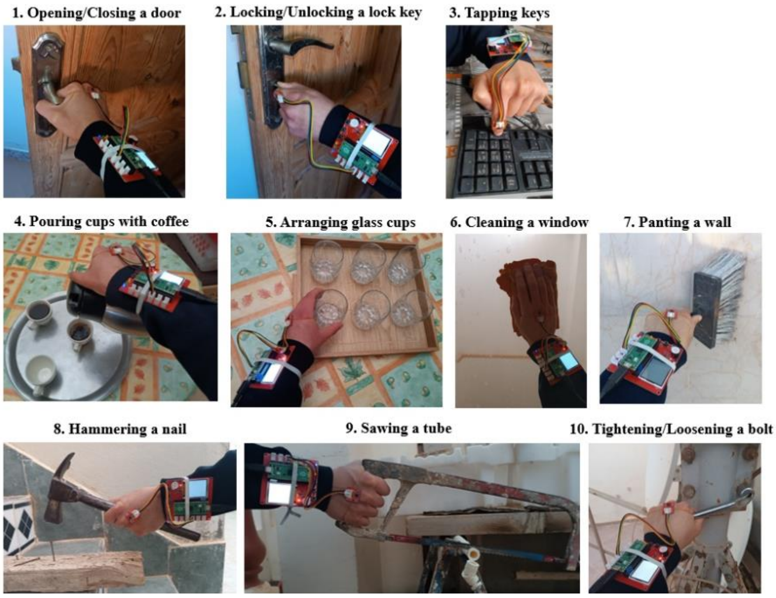


Figure 5. Pictures showing the hand performing the 10 hand tasks into real-scenarios experiments

3.6. ML applied in classification

To evaluate the accuracy of the acquired dataset in representing our ten activities, SML classification has been employed through two classification experimental phases. The initial step involves forecasting the tasks based on raw data to assess the accuracy of the real-time data capture in depicting the HGs. The second phase preprocesses the dataset [34], applying normalization and FE techniques [35], [36] to enhance our prediction performance. This entails evenly dividing the normalized dataset's axis signals over a time interval and then using FE methods on each segment. This allows us to extract a new, reduced, and more informative dataset. The signals were segmented into two-second intervals, each comprising 20 discrete samples with a 10-sample overlap, acquired at a rate of 0.01 seconds, corresponding to a sampling frequency of 10 Hz.

3.6.1. Feature extraction

FE [35], [36] refers to the process of identifying and extracting relevant features from raw data to create a more informative and simpler dataset. It aims to reduce data complexity (often known as "data dimensionality") using statistical methods while retaining as much relevant information as possible in the original dataset. This helps to improve the performance and efficiency of ML algorithms and simplify the analysis process.

In our study, four statistical FE techniques were used in classification: arithmetic mean, max–min values, root mean square (RMS), and principal component analysis (PCA) [34]–[36]. These techniques, when applied together, help to form a robust feature set that accurately represents the underlying patterns of hand movements for the classification. Utilizing these features ensures that ML models can more effectively interpret the data and deliver better performance in gesture recognition. The significance of these features and their role in this research are described as follows:

a. Arithmetic mean: it is the average value over each time interval, providing insight into the central tendency of each gesture. It gives a clear indication of the overall movement trend of the hand.
b. RMS: it provides insight into the overall magnitude and energy of the motion in each data segment, identifying its most recurrent form. It is critical for gesture recognition tasks where variations in movement intensity are significant.
c. PCA: it observes correlations between movement patterns in each axis bin and reduces information complexity while preserving the most significant patterns and enhancing computational efficiency.
d. Max–Min values: they represent the highest and lowest values of the motion within each window. They offer insights into the spread of the movement, which can be important for boundary detection in tasks.

3.6.2. ML models

Through Python programming, ten ML models were trained on the dataset for both classification experiments using an 80% training set and tested on a 20% testing set. Yielding varying accuracy scores, the results will be interpreted by comparing the performance of the two experimental classification phases. For more in-depth evaluation and analysis, a confusion matrix [43] was utilized to gain detailed insights into the highest-performing model. Additionally, cross-validation [44] was applied during the training process to mitigate any potential overfitting [45] in prediction. The ten ML classifiers employed include naïve Bayes (NB), AdaBoost (AB), multi-layer perceptron (MLP), random forest (RF), support vector machine (SVM), decision tree (DT), logistic regression (LR), XGBoost (XGB), k-nearest neighbors (KNN), and gradient boosting (GB) [46]. These classifiers were selected for their suitability in multi-label classification tasks, particularly those dealing with datasets that contain high-dimensional clusters or patterns, such as the 6-axis HG data used in this study to classify 10 hand tasks.

4. RESULTS AND DISCUSSION

4.1. The obtained dataset

The dataset comprises 3-axis accelerations (ax, ay, az) and 3-axis velocities (gx, gy, gz) as input variables and the class of activities serving as the outcome variable. With 1,998 samples allocated for each task, the raw dataset comprises 19,980 samples, resulting in an input matrix X of dimensions (19,980, 6) and an output vector Y of dimensions (19,980, 1). Post-preprocessing, the dataset was reduced to 31 rows or windows and 30 axis inputs (Mean, RMS, PCA, Max and Min per 6-axis), including 9 classes with 3 bins each and one class with 4 bins. In Table 3, the 10 tasks are designated T0, T1, …, T9, and for ML training purposes, they are encoded and categorized by numerical values ranging from 0 to 9.

Table 3. The 10 hand tasks with their names and encoded classes

Task	Designation	Classes
Opening/Closing a door	T0	0
Locking/Unlocking a lock key	T1	1
Tapping keys	T2	2
Pouring cups with coffee	T3	3
Arranging glass cups	T4	4
Cleaning a window	T5	5
Panting a wall	T6	6
Hammering a nail	T7	7
Sawing a tube	T8	8
Tightening/Loosening a bolt	T9	9

4.2. Classification performances

In the first experiment, classification on the raw dataset exhibited promising performance for half of the ten classifiers as shown in Figure 6. Prediction results show high accuracy scores (80%–90%) obtained by the RF, MLP, DT, XGB, and GB classifiers, with two of them nearing 90%. Medium scores (53%–62%) were obtained by the NB and LR classifiers, and low scores (33%–47%) by AB, SVM, and KNN.

In the second classification trial concerning dataset preprocessing as shown in Figure 7, KNN improved significantly from 46.7% to 92.63%, joining five other top models. Those include the XGB, MLP, DT, RF, KNN, and GB classifiers, achieving high accuracy scores ranging from 87.41% to 94.11%. The NB, SVM, and LR classifiers attained medium scores between 60.81% and 70.30%, while AB rose modestly to 49.57%.

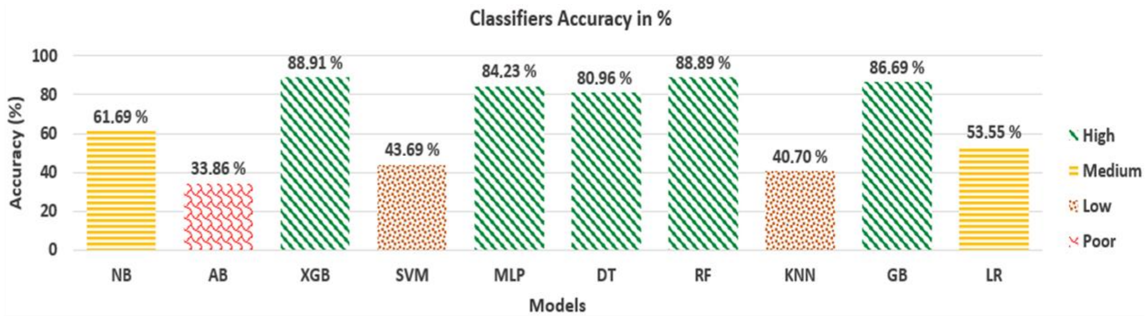


Figure 6. Classification performance on the raw dataset

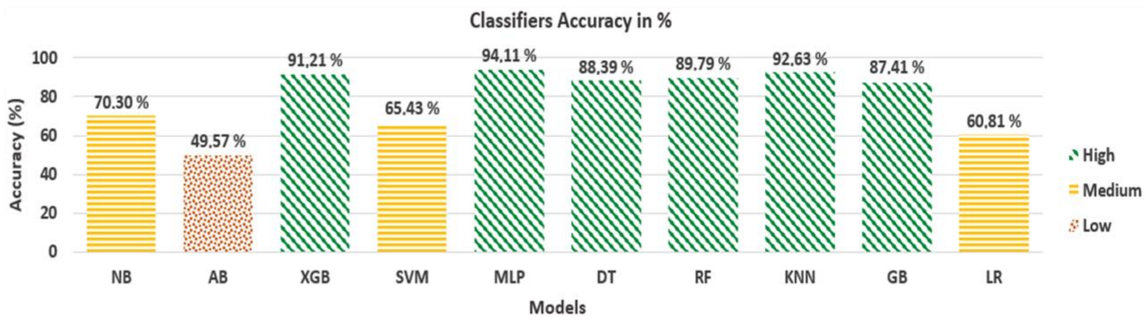


Figure 7. Classification performance on the preprocessed dataset

The prediction results demonstrate that most of the models performed well in identifying our 10 classes, where each model had a unique interpretation of the gathered dataset. Five models (XGB, MLP, DT, RF, and GB) have already shown promising results on the raw dataset as shown in Figure 6, while four other models (SVM, KNN, LR, and NB) have shown middling performances. Following data preprocessing as shown in Figure 7, six models (XGB, MLP, DT, RF, KNN, and GB) provided high performances (87.41%–94.11%), while three models (NB, SVM, and LR) had performances above 60%. Additionally, the dataset has a certain structure that works well for most classifiers but not so well for others due to the low accuracy ratings. All of this shows how a precise dataset was gathered and trained by ML classifiers to accurately depict and predict the gestures used in 10 different human tasks without either overfitting or underfitting [45].

4.3. MLP confusion matrix

While all six classifiers demonstrated highly promising performances, it is more practical to select the best-performing model to be implemented on the Pico and used for future predictions. In this case, the highest accuracy is 94.11%, which is achieved by the MLP classifier. Its confusion matrix [43] in Table 4 offers an overview of correct and incorrect predictions for each class by exposing potential confusion between classes and the prediction sensitivity (known as TPR: true positive rate or recall) [43] of each class.

Table 4. Confusion matrix for the MLP classifier

		Predicted classes										
		T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	TPR (%)
True classes	T0	**360**	4	4	4	9	2	12	2	4	3	90.60
	T1	4	**380**	7	0	0	1	5	1	1	1	95.72
	T2	2	1	**376**	3	3	2	3	4	5	0	94.34
	T3	0	0	1	**395**	0	0	0	0	0	5	**98.55**
	T4	0	2	2	0	**385**	0	0	0	6	5	96.45
	T5	1	3	0	0	0	**386**	0	0	4	5	96.84
	T6	3	2	8	0	3	0	**365**	2	3	1	94.69
	T7	2	0	5	0	0	0	3	**364**	3	12	93.74
	T8	2	1	1	1	3	2	1	2	**381**	3	95.97
	T9	15	6	2	0	21	1	7	0	10	**356**	**85.16**

Although some confounding points between certain classes were observed, the MLP confusion matrix in Table 4 demonstrates a satisfactory identification of the majority of points across the ten task classes. While accounting for non-linear trends, the matrix also highlights a few problematic points within the dataset. These issues arose during the execution of tasks like T0, T4, and T9, where hand movements were occasionally unsteady or subject to vibrations at specific instances. Nonetheless, the TPR values for all classes indicate that the MLP model predicted each task individually with a high precision rate. Nine tasks were predicted at precision rates ranging from 90.60% to 98.55%, with the exception of T9, which was predicted at a slightly lower precision rate of 85.16%. Overall, these results underscore the MLP's ability to distinguish effectively between the ten activities, achieving a performance of 94.11% as the TPR mean.

4.4. Outcomes and analysis

The results demonstrate that the collected dataset effectively represents our ten distinct activities that were accurately classified and recognized by ML models. On the raw dataset, prediction exhibited strong performance with a maximum accuracy of 89%, while preprocessing techniques further enhanced the performance to a maximum score of 94.11%. The achievement of high outcomes can be attributed to the careful selection of tasks and controlled execution of hand movements, with the use of suitable embedded tools alongside performant ML methods. This highlights the success of a low-cost data collection methodology in accurately capturing and predicting HG data for HR embedded ML-based applications.

5. CONCLUSION

In conclusion, the application of ML techniques in our low-cost data collection strategy yielded favorable results in identifying the ten task classes. These outcomes can be further leveraged by training on new data and deploying the best-performing models on the Pico as embedded ML algorithms for real-world applications. This could serve as a follow-up to this work for a new project, demonstrating the importance of affordable embedded devices for efficient data collection and reliable ML performance. Future work could involve expanding the dataset, exploring additional AI features, and testing different ML models across various real-world scenarios to improve generalizability. Overall, this research lays a strong foundation for applying similar methods in HR, incorporating advanced ML techniques using additional tools. The ability to accurately recognize specific HGs creates opportunities to integrate such of our ten activities into robotic systems, advancing automation in human-assistive tasks. A practical example is underwater welding, which is performed under high pressure in oceans. Well-trained HRS could efficiently execute this job, mitigating risks to humans such as electric shock, decompression sickness, drowning, hearing loss, and hypothermia.

ACKNOWLEDGEMENTS

We would like to express our appreciation to all individuals and organizations who contributed to this project. Special thanks to the professors and supervisors at Mohamed 1 University for their invaluable support and guidance. We also extend our gratitude to the Electronics and Instrumentation Team from the Laboratory of Electronics and Systems for their collaboration and input. Additionally, we acknowledge the efforts of the Operational Research and Applied Statistics Team from the Laboratory of Applied Mathematics of the Oriental for their dedication. Finally, we thank our families, friends, and colleagues for their unwavering support.

REFERENCES

[1] S. Saeedvand, M. Jafari, H. S. Aghdasi, and J. Baltes, "A comprehensive survey on humanoid robot development," *The Knowledge Engineering Review*, vol. 34, 2019, doi: 10.1017/S0269888919000158.
[2] Y. Tong, H. Liu, and Z. Zhang, "Advancements in humanoid robots: a comprehensive review and future prospects," *IEEE/CAA Journal of Automatica Sinica*, vol. 11, no. 2, pp. 301–328, Feb. 2024, doi: 10.1109/JAS.2023.124140.
[3] S. A. Oke, "A literature review on artificial intelligence," *International Journal of Information and Management Sciences*, vol. 19, no. 4, pp. 535–570, 2008.
[4] A. Joseph, B. Christian, A. A. Abiodun, and F. Oyawale, "A review on humanoid robotics in healthcare," *MATEC Web of Conferences*, vol. 153, p. 2004, 2018, doi: 10.1051/matecconf/201815302004.
[5] J. Fox and A. Gambino, "Relationship development with humanoid social robots: Applying interpersonal theories to human-robot interaction," *Cyberpsychology, Behavior, and Social Networking*, vol. 24, no. 5, pp. 294–299, 2021, doi: 10.1089/cyber.2020.0181.
[6] K. Y. Fok, N. Ganganath, C. T. Cheng, and C. K. Tse, "A real-time ASL recognition system using leap motion sensors," in *Proceedings - 2015 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, CyberC 2015*, Sep. 2015, pp. 411–414, doi: 10.1109/CyberC.2015.81.
[7] V. Nasteski, "An overview of the supervised machine learning methods," *Horizons.B*, vol. 4, pp. 51–62, Dec. 2017, doi: 10.20544/horizons.b.04.1.17.p05.
[8] C. C. Hsieh, D. H. Liou, and D. Lee, "A real time hand gesture recognition system using motion history image," in *ICSPS 2010 - Proceedings of the 2010 2nd International Conference on Signal Processing Systems*, Jul. 2010, vol. 2, doi: 10.1109/ICSPS.2010.5555462.

[9] R. Chalodhorn, K. F. MacDorman, and M. Asada, “Humanoid robot motion recognition and reproduction,” *Advanced Robotics*, vol. 23, no. 3, pp. 349–366, Jan. 2009, doi: 10.1163/156855308X397569.
[10] J. Yang and E. Chew, “A systematic review for service humanoid robotics model in hospitality,” *International Journal of Social Robotics*, vol. 13, no. 6, pp. 1397–1410, Nov. 2021, doi: 10.1007/s12369-020-00724-y.
[11] O. Lamsellak, A. Benlghazi, A. Chetouani, and A. Benali, “Hand gesture recognition using machine learning for bionic applications: Forearm case study,” in *Lecture Notes in Networks and Systems*, vol. 668, Springer Nature Switzerland, 2023, pp. 153–163, doi: 10.1007/978-3-031-29857-8_16.
[12] E. Torta *et al.*, “Evaluation of a small socially-assistive humanoid robot in intelligent homes for the care of the elderly,” *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 76, no. 1, pp. 57–71, Feb. 2014, doi: 10.1007/s10846-013-0019-0.
[13] T. Asfour *et al.*, “ARMAR-6: a high-performance humanoid for human-robot collaboration in real-world scenarios,” *IEEE Robotics and Automation Magazine*, vol. 26, no. 4, pp. 108–121, Dec. 2019, doi: 10.1109/MRA.2019.2941246.
[14] A. K. Pandey and R. Gelin, “A mass-produced sociable humanoid robot: Pepper: the first machine of its kind,” *IEEE Robotics and Automation Magazine*, vol. 25, no. 3, pp. 40–48, Sep. 2018, doi: 10.1109/MRA.2018.2833157.
[15] H. Stuart, S. Wang, O. Khatib, and M. R. Cutkosky, “The ocean one hands: an adaptive design for robust marine manipulation,” *International Journal of Robotics Research*, vol. 36, no. 2, pp. 150–166, Feb. 2017, doi: 10.1177/0278364917694723.
[16] M. Hirose and K. Ogawa, “Honda humanoid robots development,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 365, no. 1850, pp. 11–19, Nov. 2007, doi: 10.1098/rsta.2006.1917.
[17] L. P. Kaelbling, *Learning in embedded systems*. Cambridge, 1993.
[18] Z. Zhang and J. Li, “A review of artificial intelligence in embedded systems,” *Micromachines*, vol. 14, no. 5, Apr. 2023, doi: 10.3390/mi14050897.
[19] P. Sarma, H. K. Singh, and T. Bezboruah, “A real-time data acquisition system for monitoring sensor data,” *International Journal of Computer Sciences and Engineering*, vol. 6, no. 6, pp. 539–542, Jun. 2018, doi: 10.26438/ijcse/v6i6.539542.
[20] S. Li, L. Da Xu, and S. Zhao, “The internet of things: a survey,” *Information Systems Frontiers*, vol. 17, no. 2, pp. 243–259, Apr. 2015, doi: 10.1007/s10796-014-9492-7.
[21] S. L. Mondragón-González, E. Burguière, and K. N’diaye, “Mobile devices, connected objects, and sensors,” in *Neuromethods*, vol. 197, Springer US, 2023, pp. 355–388, doi: 10.1007/978-1-0716-3195-9_12.
[22] M. S. Diab and E. Rodriguez-Villegas, “Embedded machine learning using microcontrollers in wearable and ambulatory systems for health and care applications: a review,” *IEEE Access*, vol. 10, pp. 98450–98474, 2022, doi: 10.1109/ACCESS.2022.3206782.
[23] M. Wang *et al.*, “Embedded system design and implementation of a humanoid service robot,” in *2024 IEEE International Conference on Mechatronics and Automation, ICMA 2024*, Aug. 2024, pp. 1297–1302, doi: 10.1109/ICMA61710.2024.10633172.
[24] J. Baltes and J. Anderson, “Complex AI on small embedded systems: Humanoid robotics using mobile phones,” *AAAI Spring Symposium - Technical Report*, vol. SS-10-04, pp. 2–9, 2010.
[25] R. H. Landau, M. J. Páez Mejía, and C. C. Bordeianu, *Computational physics: problem solving with Python*, John Wiley & Sons. Ed., 4th ed. Weinheim, Germany: Wiley-VCH, 2024.
[26] V. Vasek, P. Dostálek, J. Dolinay, D. Janáčová, V. Vašek, and K. Kolomazník, “Microcontrollers and modern control methods,” *in Proceedings of the 2nd WSEAS International Conference on Computer Engineering and Applications: Modern Topics of Computer Science. World Scientific and Engineering Academy and Society (WSEAS),* Acapulco, Mexico, 2008.
[27] R. Bannatyne and G. Viot, “Introduction to microcontrollers,” in *WESCON/97 Conference Proceedings,* Santa Clara, CA, USA, 1997, pp. 564-574, doi: 10.1109/WESCON.1997.632384.
[28] M. R. Idris, M. N. Irdayanti, and M. Ramlee, “A review of microcontroller design in future industrial revolution,” in *AIP Conference Proceedings*, 2024, vol. 2750, no. 1, doi: 10.1063/5.0149679.
[29] P. Patel, N. Gupta, and S. Gajjar, “Real time voice recognition system using tiny ML on Arduino Nano 33 BLE,” in *Proceedings - 2023 IEEE International Symposium on Smart Electronic Systems, iSES 2023*, Dec. 2023, pp. 385–388, doi: 10.1109/iSES58672.2023.00085.
[30] D. M. Waqar, T. S. Gunawan, M. A. Morshidi, and M. Kartiwi, “Design of a speech anger recognition system on Arduino Nano 33 BLE Sense,” in *2021 IEEE 7th International Conference on Smart Instrumentation, Measurement and Applications, ICSIMA 2021*, Aug. 2021, pp. 64–69, doi: 10.1109/ICSIMA50015.2021.9526323.
[31] D. Ibrahim, *Raspberry Pi Pico experimenting kit, programming*. Susteren, The Netherlands: Elector, 2022.
[32] D. R. Loker, “Raspberry Pi Pico as an IoT device,” in *ASEE Annual Conference & Exposition. Conference Proceedings*, Baltimore, Maryland, 2023, doi: 10.18260/1-2--44016.
[33] I. Arun Faisal, T. Waluyo Purboyo, and A. Siswo Raharjo Ansori, “A review of accelerometer sensor and gyroscope sensor in IMU sensors on motion capture,” *Journal of Engineering and Applied Sciences*, vol. 15, no. 3, pp. 826–829, Nov. 2019, doi: 10.36478/jeasci.2020.826.829.
[34] C. Fan, M. Chen, X. Wang, J. Wang, and B. Huang, “A review on data preprocessing techniques toward efficient and reliable knowledge discovery from building operational data,” *Frontiers in Energy Research*, vol. 9, 2021, doi: 10.3389/fenrg.2021.652801.
[35] T. Buckley, B. Ghosh, and V. Pakrashi, “A feature extraction & selection benchmark for structural health monitoring,” *Structural Health Monitoring*, vol. 22, no. 3, pp. 2082–2127, Sep. 2023, doi: 10.1177/14759217221111141.
[36] N. E. Rasan and D. K. Mani, “A survey on feature extraction techniques,” *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 03, no. 01, pp. 52–55, Jan. 2015, doi: 10.15680/ijircce.2015.0301009.
[37] J. Mankar, C. Darode, K. Trivedi, M. Kanoje, and P. Shahare, “Review of I2C protocol,” *International Journal of Research in Advent Technology*, vol. 2, no. 1, 2014.
[38] D. Rossos, A. Mihailidis, and B. Laschowski, “AI-powered smart glasses for sensing and recognition of human-robot walking environments,” *bioRxiv*, pp. 2023--10, Oct. 2023, doi: 10.1101/2023.10.24.563804.
[39] H. Soy, “Edge AI-assisted IoV application for aggressive driver monitoring: A case study on public transport buses,” *International Journal of Automotive Science and Technology*, vol. 7, no. 3, pp. 213–222, Sep. 2023, doi: 10.30939/ijastech..1335390.
[40] M. Kundu and A. Kumar, “A review on low power SPI protocol,” *International Journal of VLSI System and Communication Systems*, vol. 2, no. 3, pp. 193–195, Apr. 2014.
[41] Laddha, R. Neha, and A. P. Thakare, “A review on serial communication by UART,” *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 1, pp. 366–369, 2013.
[42] D. Ünsal Öztürk and A. M. Erkmen, “Coriolis vibratory MEMS gyro drive axis control with proxy-based sliding mode controller,” *Micromachines*, vol. 13, no. 3, p. 446, Mar. 2022, doi: 10.3390/mi13030446.
[43] M. Heydarian, T. E. Doyle, and R. Samavi, “MLCM: multi-label confusion Matrix,” *IEEE Access*, vol. 10, pp. 19083–19095, 2022, doi: 10.1109/ACCESS.2022.3151048.

[44] D. Berrar, "Cross-validation," in *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics*, vol. 1–3, Elsevier, 2018, pp. 542–545, doi: 10.1016/B978-0-12-809633-8.20349-X.
[45] C. Aliferis and G. Simon, "Overfitting, underfitting and general model overconfidence and under-performance pitfalls and best practices in machine learning and AI," in *Artificial Intelligence and Machine Learning in Health Care and Medical Sciences*, Springer International Publishing, 2024, pp. 477–524, doi: 10.1007/978-3-031-39355-6_10.
[46] M. J. Uddin *et al.*, "An integrated statistical and clinically applicable machine learning framework for the detection of autism spectrum disorder," *Computers*, vol. 12, no. 5, Apr. 2023, doi: 10.3390/computers12050092.

BIOGRAPHIES OF AUTHORS

Jalal Hdid received a specialized master degree in electronics, electrical engineering, automation, and industrial computing from the Faculty of Science Aïn Chock, Hassan II University, Casablanca, Morocco, in 2019. Currently, he is a Ph.D. student at the Laboratory of Electronics and Systems, Mohamed 1 University, Oujda, Morocco. His thesis subject is the supervision of industrial installations using artificial intelligence. He can be contacted at email: j.hdid@ump.ac.ma.

Oussama Lamsellak received a specialized master degree in information systems engineering from the Faculty of Science Semlalia, Cadi Ayyad University, Marrakesh, Morocco, in 2016. Currently, he is a dedicated researcher in the field of human gesture recognition with artificial intelligence at Mohamed 1 University, Oujda, Morocco. His research interests lie in software development and programming, focusing on innovative approaches to enhance gesture recognition systems. He can be contacted at email: oussama.lamsellak@ump.ac.ma.

Ahmad Benlghazi received a PhD degree in electronic and microprogramming from Mohammed First University, Morocco. Currently, he is a professor and researcher at the National School of Applied Sciences, Mohamed 1 University, Oujda, Morocco. His specialization is electronics. He can be contacted at email: a.benlghazi@ump.ac.ma.

Abdelhamid Benali is a doctor of engineering and head of the Electronic Computer Sciences and Telecommunication Department. Currently, he is a professor and researcher at the National School of Applied Sciences, Mohamed 1 University, Oujda, Morocco. He can be contacted at email: a.benali@ump.ac.ma.

Ouafae El Melhaoui is a doctor in physics/classification of data with new signal processing techniques, concept and applications from the Faculty of Sciences in Mohamed First University, Oujda, Morocco, in 2013. She received the bachelor and the master degrees in electronics and informatics from Mohamed First University, Oujda, Morocco, in 2004 and 2006 respectively. She can be contacted at email: wafa19819@gmail.com.