Indonesian Journal of Computer Science
ISSN 2549-7286 (*online*)
Jln. Khatib Sulaiman Dalam No. 1, Padang, Indonesia
Website: ijcs.stmikindonesia.ac.id | E-mail: ijcs@stmikindonesia.ac.id

An Integrated Gesture Framework of Smart Entry Based on Arduino and Random Forest Classifier

Saman M. Almufti[1], Adnan M. Abdulaziz3 Author[1], Author[2] [bold]
saman.almufti@gmail.com, adnan.mohsin@dpu.edu.krd
[1]Technical College of Informatics, Akre University for Applied Science, Duhok, KRG-Iraq
[2]Department of Computer Science, Nawroz University, Duhok, KRG-Iraq
[3]Technical College of Engineering-Duhok, Duhok Polytechnic University, Duhok, KRG-Iraq

Article Information

Submitted : 28 Jan 2024
Reviewed: 30 Jan 2024
Accepted : 15 Feb 2024

Keywords

Gesture Detection, Smart Entry, Arduino, Random Forest Classifier

Abstract

Gesture-based systems have emerged as a prominent breakthrough in the field of smart access control, effectively integrating security measures with user comfort. This study presents a novel gesture detection framework for smart entry systems that combines the computational capabilities of a Random Forest Classifier with the practicality of Arduino-based hardware. Central to methodology is the utilization of MediaPipe, an advanced computer vision library, to extract hand motion landmarks from live video streams. The selected landmarks function as a comprehensive dataset for training a Random Forest Classifier, which has been specifically chosen due to its high level of accuracy and efficiency in managing intricate classification jobs. The model exhibits outstanding competence in the categorization of gestures in real-time, attaining high levels of accuracy that are crucial for ensuring dependable entrance control. The Arduino microcontroller plays a vital role in the execution of the entry mechanism as it serves as the intermediary between the gesture detection software and the tangible entry control hardware. The incorporation of gesture recognition technology facilitates a cohesive and prompt user experience, wherein identified motions are directly converted into input commands. The system's practical use is demonstrated through a series of detailed tests, which highlight its dependability and efficiency across diverse climatic circumstances. The findings underscore the system's capacity as a flexible and safe solution for contactless access in many environments, including both private homes and highly protected establishments. Furthermore, the study makes a substantial contribution to the larger domain of human-computer interaction by showcasing the practicality of advanced gesture detection systems in many everyday contexts. The suggested framework presents a novel approach to smart entry systems and also paves the way for further investigation in the domains of smart home automation and interactive systems. In these areas, gesture-based interfaces have the potential to deliver user experiences that are both intuitive and efficient.

A. Introduction

The integration of advanced machine learning techniques and embedded systems engineering is fundamentally transforming the landscape of human-environment interaction, heralding a new era in smart access technologies. This convergence is particularly evident in the security and user interface systems, where the fusion of the robust Arduino platform and the analytical prowess of the Random Forest Classifier has led to groundbreaking advancements[1]. Our study presents a cutting-edge gesture-based entry framework that epitomizes this synthesis, offering a novel perspective on intelligent entry systems. This framework is not merely a technological leap but a harmonious blend of theoretical computer science advancements with practical, real-world applications.

At the heart of this study lies the implementation of gesture recognition, a rapidly emerging sector within human-computer interaction (HCI). Leveraging the capabilities of MediaPipe, an advanced computer vision framework, our system meticulously captures and processes hand gesture landmarks from live video feeds, providing a rich dataset for machine learning analysis. The Random Forest Classifier, renowned for its proficiency in handling complex data, is employed to deftly categorize these gestures. In parallel, the Arduino microcontroller seamlessly bridges the gap between digital gesture recognition and physical control mechanisms, embodying a synergy of digital intelligence and physical action[2].

This research transcends typical technological enhancements, embodying a holistic approach to crafting user-centric, efficient interactive systems. The ensuing sections delve into the comprehensive development, validation, and potential applications of this integrated gesture framework, with implications spanning smart home automation and secure access control.

Complementing this, significant strides in gesture recognition have been propelled by notable advancements in machine learning and computer vision. Initial forays into this domain, primarily focused on rudimentary systems constrained by limited datasets. These systems struggled with accuracy and adaptability, particularly in complex environments[3]. The advent of deep learning, especially convolutional neural networks (CNNs), marked a paradigm shift, significantly enhancing the precision and robustness of gesture recognition systems[4].

Concurrently, smart entry systems have evolved from traditional lock-and-key mechanisms to more sophisticated technological solutions. integrated biometric systems like fingerprint and facial recognition, lauded for their security enhancements but also criticized for privacy concerns and vulnerability to spoofing. Gesture-based entry systems emerged as a user-friendly and discreet alternative, balancing security with user convenience[5]

The fusion of Arduino's versatility with complex gesture recognition algorithms, such as Random Forest, has given rise to advanced, user-friendly entry systems. [6]demonstrated Arduino's efficacy in developing intelligent systems, particularly in smart entry controls. The Random Forest method, an ensemble learning technique, is celebrated for its accuracy and ability to manage extensive feature sets, making it ideal for real-time gesture detection in diverse settings[7], [8].

In essence, the current corpus of scholarly work indicates a clear trend towards developing advanced access control systems that prioritize user needs. The amalgamation of machine learning breakthroughs, Arduino-based hardware, and computer vision technologies such as MediaPipe offers a fertile ground for future research in intelligent entry systems. This integration addresses challenges related to accuracy and real-time processing, opening new avenues in interactive and secure access control systems.

In culmination of the introduction, this paper delineates a systematic workflow that traverses through various pivotal stages of developing a sophisticated gesture recognition framework for smart entry systems. The process begins with "Data Collection and Preprocessing," where MediaPipe is employed for precise hand tracking from live video streams, followed by feature extraction and data serialization. In the "Machine Learning Classifier" section, the focus shifts to the deployment of the Random Forest Classifier, chosen for its adeptness in handling complex classification tasks. Subsequently, the "Random Forest Classifier: Fundamentals and Training" segment delves into the specifics of the classifier's implementation and training within the Scikit-learn framework.

The "Model Testing and Real-Time Deployment" section addresses the critical evaluation of the trained model and its integration into a real-time Python environment, highlighting the role of OpenCV in processing live video data for gesture recognition. The "Arduino Microcontroller Integration and Applications" section illustrates how the Arduino microcontroller acts as an intermediary, translating digital gesture recognition into physical responses, essential for smart entry applications.

The study then moves towards discussing the "Academic Significance and Future Implications" of the research, emphasizing its interdisciplinary integration and potential in smart home automation and interactive systems. "Results and Discussion" provide insights into the effectiveness of the system, underscoring the high accuracy rates achieved and the implications for smart entry technologies.

This workflow encapsulates the essence of the research methodology, from initial data collection to practical deployment, highlighting the innovative blend of machine learning and embedded systems engineering. It sets the stage for a comprehensive exploration of a gesture-based control system, poised to make significant contributions to the realm of human-computer interaction and smart automation technologies.

B. Literature Review

The literature review encapsulates a diverse range of studies in the field of hand gesture recognition, employing various methods and technologies like computer vision, machine learning algorithms, electromyography (EMG), and specialized hardware.

Trigueiros, Ribeiro, and Reis (2015) underscored the significance of hand gesture recognition in human-computer interaction (HCI). The system architecture integrates data acquisition, preprocessing, feature extraction, and classification. The study employs centroid distance values and Hidden Markov Models (HMMs) for hand posture and dynamic gesture classification, respectively, achieving high accuracy. The applications range from assisting a robotic soccer referee to interpreting Portuguese Sign Language, highlighting the versatility of gesture-based systems in HCI [9]. Abhishek B et al. (2020): In "Hand Gesture Recognition Using Machine Learning Algorithms," the authors developed a system designed for real-time interaction, free from constraints like gloves or uniform backgrounds. The system architecture encompasses learning, detection, and recognition stages, using a 3D Convolutional Neural Network (CNN) for gesture recognition. The study addresses the challenges in spatial and temporal modeling of gestures and utilizes OpenCV and TensorFlow, signifying the evolution of machine learning and computer vision in sophisticated HCI systems [10]. Chakrabarti S, Saha H. (2018) focused on enhancing communication for individuals with disabilities. The paper adopts image processing techniques and machine learning methods for feature extraction and employs standard classification models, underscoring the potential of hand gesture recognition in aiding communication for people with disabilities [11] Jaramillo and Benalcazar (2017) delves into EMG-based hand gesture recognition. The proposed model addresses the challenge of interpreting EMG signals using machine learning, demonstrating applications in prosthetics and human-machine interaction systems. The methodology encompasses signal acquisition, preprocessing, feature extraction, classification, and post-processing, aiming to recognize a broad spectrum of hand gestures accurately and in real-time [12]. Marco E. Benalcázar et. al. (2017) presented a model utilizing surface EMG and machine learning algorithms for real-time hand gesture recognition. The system, employing k-nearest neighbor and dynamic time warping algorithms, demonstrates superior accuracy and potential applications in medical and engineering fields [13] . Li W et al. (2021) provided an analysis of deep learning in sEMG-based hand gesture recognition for prosthetic hands. It covers the entire process from data acquisition to performance evaluation and discusses various deep learning architectures, highlighting their application in prosthetic hand control [14]. Marouane Benmoussa and Abdelhak Mahmoudi (2018) described a system for real-time gesture recognition using a Kinect sensor and a Support Vector Machine (SVM) model. The system's high accuracy and real-time capability illustrate the advancement in HCI through machine learning, addressing challenges in data balance and model selection[15]. Wahid et al. (2018): focused on EMG-based recognition with a novel normalization strategy. The paper highlights the potential of this approach for various biomedical applications, including prosthetic device control and human-computer interfaces [16]. Bush I, Abiyev R, Arslan M (2019) presented a vision-based mouse controller using hand gestures. The hybrid model combining Single Shot Multi Box Detection (SSD) and Convolutional Neural Network (CNN) demonstrates high accuracy, particularly useful in presentation settings [5]. Nogales R, Benalcázar M (2021): "Hand gesture recognition using machine learning and infrared information: a systematic literature review" conducted a comprehensive review of the development in hand gesture recognition using machine learning and infrared information. It provides insights into various architectures, techniques, and performance metrics, offering a broad perspective on the field's progress [17].

Jaramillo-Yánez et al. (2020) analyzed the developments in EMG-based HGR using machine learning. The paper discusses the structure of HGR models, controller delay, types of gestures recognized, and evaluation metrics, contributing to the understanding of real-time HGR in various applications[18] Pławiak et al. (2016) discussed the development of a system for quick recognition of hand gestures using a specialized glove. This study emphasizes the effectiveness of machine learning algorithms in rapidly and accurately recognizing hand gestures [19].Dardas and Georganas (2011) a novel system is presented for real-time interaction with applications through hand gestures. The system employs skin detection, SIFT for feature extraction, and a multiclass SVM for gesture recognition, underlining its potential in HCI [20].

These studies collectively highlight the significant advancements and diverse approaches in hand gesture recognition, emphasizing the integration of various technologies and methodologies to enhance human-computer interaction.

C. Gesture Detection:

Gesture detection, an integral component of Human-Computer Interaction (HCI), has witnessed exponential growth, spurred by advancements in machine learning and computer vision technologies. This paradigm enables intuitive, non-verbal communication between humans and digital systems, transcending the traditional confines of keyboard and mouse interfaces[21].

Primarily, gesture detection hinges on the recognition of physical movements, predominantly hand or body gestures, which are interpreted as commands by computational systems. The process involves capturing gestural data through sensors or cameras, followed by intricate processing using algorithms capable of discerning patterns and translating them into actionable inputs.

The application spectrum of gesture detection is remarkably vast and diverse. In the domain of augmented and virtual reality (AR/VR), gesture detection facilitates immersive experiences, allowing users to interact with virtual environments in a natural and intuitive manner [22]. This has profound implications in fields ranging from gaming and entertainment to education and training simulations[23].

In healthcare, gesture-controlled interfaces are revolutionizing patient care and surgical procedures. Surgeons, through gesture recognition systems, can manipulate medical images or control robotic instruments without direct contact, maintaining sterility and enhancing precision[24].

Furthermore, gesture detection is pivotal in developing assistive technologies for individuals with disabilities. Customizable gesture-based systems empower users with mobility or speech impairments to interact effectively with technology, enhancing accessibility and fostering independence.

Additionally, in the automotive industry, gesture recognition is being integrated into vehicular systems, enabling drivers to control features such as navigation, entertainment, and climate without diverting attention from driving, thus augmenting safety and convenience [25].

The preprocessing stage of the gesture recognition system constitutes a series of methodologically rigorous processes that are pivotal for the transformation of unrefined visual input into a refined dataset that is conducive to algorithmic analysis and learning.

- Hand Detection Subprocess: At the commencement of the preprocessing phase, hand detection algorithms are deployed to identify the loci of hands within the digital images. This process is underpinned by sophisticated computer vision techniques, potentially leveraging the capabilities of the MediaPipe framework—a cutting-edge library renowned for its proficiency in real-time, multimodal perception algorithms [26]. The precision of this step is paramount, as it delineates the region of interest by segmenting the hand from the ambient background, thereby ensuring the focal point for feature extraction is devoid of extraneous visual information [27].

- Feature Extraction Subprocess: Ensuing hand detection, a comprehensive feature extraction operation is undertaken. This process is characterized by the distillation of critical descriptors from the segmented images of hands [28]. Such descriptors encapsulate a variety of attributes including, but not limited to, spatial coordinates that map the location and posture of the hand within the Cartesian plane of the image, geometric descriptors that delineate the hand's shape and the relative arrangement of its constitutive parts, and kinematic features that chronicle the temporal dynamics of the hand's movement. Techniques such as edge detection, shape analysis, and motion tracking may be employed to extract these high-fidelity features, which collectively compose the feature vector representative of each unique gesture [29], [30].
- Serialization Subprocess: The culmination of the preprocessing stage is marked by the serialization of the processed data. The serialization subprocess involves the encoding of the feature vectors into a structured binary format, a protocol often implemented via serialization libraries like Python's pickle module. This not only facilitates a reduction in data volume, thereby enhancing storage efficiency and retrieval speed, but also ensures the preservation of data integrity across different computational environments. Serialization engenders a seamless transition of the dataset into the model training phase, providing a standardized and optimized dataset for ingestion by the machine learning algorithms [16], [31].

D. Machine Learning Classifier

Machine learning classifiers represent a cornerstone in the domain of computational intelligence, offering a paradigm shift in how data is interpreted, patterns recognized, and decisions made. These classifiers, underpinned by algorithms that learn from and make predictions on data, are fundamental to a wide array of applications, from natural language processing to image recognition and beyond [32], [33]. The efficacy of a machine learning classifier lies in its ability to discern and learn from complex and often non-linear relationships within data, enabling it to adapt and improve over time with exposure to more information [34], [35].

The architecture of machine learning classifiers is diverse, encompassing a range of approaches from supervised learning, where models are trained on labeled data, to unsupervised learning, which involves finding hidden patterns in unlabeled data. Among the most prevalent and powerful classifiers are neural networks, decision trees, support vector machines (SVMs), and random forests, each exhibiting unique strengths suited to specific types of data and analytical requirements[36].

The choice of classifier largely depends on the nature of the task, the complexity and size of the dataset, and the requirement for model interpretability. As machine learning continues to evolve, these classifiers are becoming increasingly sophisticated, integrating advancements in artificial intelligence to tackle more complex, nuanced, and dynamic challenges in data analysis[2], [37]. The ongoing development of these tools is not just a technical endeavor but a transformative force reshaping various sectors, from healthcare and finance to education and entertainment, by providing insights and automation with unprecedented accuracy and efficiency[38].

a. Random Forest Classifier:

The Random Forest Classifier, an ensemble learning methodology, has garnered significant attention in the domain of machine learning for its robustness and versatility[39]. It operates on the principle of decision tree amalgamation, where the collective output of multiple decision trees is utilized to enhance predictive accuracy and stability as shown in Figure 1. This paradigm shift from singular to ensemble methodologies marks a crucial development in machine learning techniques [40].

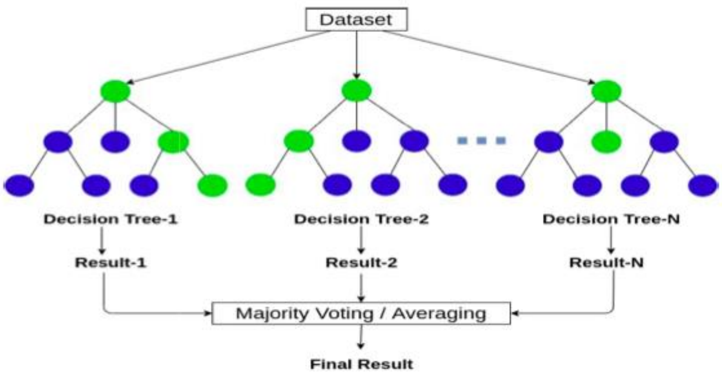

Figure 1: Random Forest [41]

Fundamental Aspects of Random Forest Classifier:

- Ensemble Learning Approach: The Random Forest algorithm epitomizes an ensemble approach, employing a multitude of decision trees during training. It leverages the 'bagging' technique, wherein each tree is independently constructed using a bootstrap sample of the data[42].
- Incorporation of Randomness: The algorithm introduces randomness at two levels: by sampling training data for tree generation and by randomly selecting subsets of features at each split in the construction of trees. This stochastic element is pivotal in enhancing diversity among trees, thereby mitigating the risk of overfitting common in single decision trees [43].
- Mechanism of Prediction: In classification tasks, the Random Forest aggregates predictions from individual trees through a majority

voting system. The class receiving the highest number of votes from the ensemble of trees is chosen as the final prediction, exemplifying a democratic decision-making process within the algorithm.

- Overfitting Prevention: Due to its inherent ensemble nature, Random Forest is intrinsically equipped to combat the problem of overfitting, a notorious issue in machine learning models, particularly in decision trees. The amalgamation of diverse trees ensures that the model does not excessively conform to the training data [44].
- Feature Importance Evaluation: A salient feature of Random Forest is its intrinsic ability to evaluate the importance of various features in the predictive process. This attribute is instrumental in interpretability and understanding the relative significance of different predictors in the model [45], [46].

E. Arduino microprocessor:

The Arduino microcontroller is a fundamental element in the domain of electronic and embedded system projects, serving as a compact, yet powerful, integrated circuit capable of executing specific tasks such as input reading, data processing, and output control. Renowned for its simplicity and adaptability, the Arduino facilitates the interface between software and hardware, allowing for the realization of intricate and dynamic electronic systems[6].

In an academic framework, the Arduino represents a paradigmatic instance of an embedded system, distinguished by its ease of use, versatility, and open-source framework. Its ability to interpret signals from a software-based machine learning model, which processes and recognizes specific hand gestures, is crucial in applications such as smart door entry systems. Upon recognizing a gesture, the Arduino is programmed to undertake corresponding actions—such as controlling a locking mechanism—effectively bridging the gap between digital commands and physical responses[6].

Programmed through the Arduino Integrated Development Environment (IDE) using a variation of C/C++, the microcontroller is engineered to respond to specific signals, activating outputs via its digital or analog pins to operate various devices like motors or solenoids. This capability is pivotal in the context of Human-Computer Interaction (HCI) and the Internet of Things (IoT), as it expands user interaction from conventional inputs to more natural and intuitive gestures.

The integration of the Arduino within AI-driven applications is indicative of its pivotal role in modern interactive systems. It serves as the nexus where computational analysis yields real-world action, emphasizing the convergence of digital intelligence with tangible, physical operations within the environment[47].

F. Methodology

The proposed Model shown in figure (2), represents a cutting-edge integration of machine learning and embedded systems technology, designed to facilitate a smart door entry mechanism using hand gesture recognition. This system stands at the intersection of artificial intelligence, computer vision, and microcontroller applications, illustrating a practical and innovative use of these technologies in everyday life.

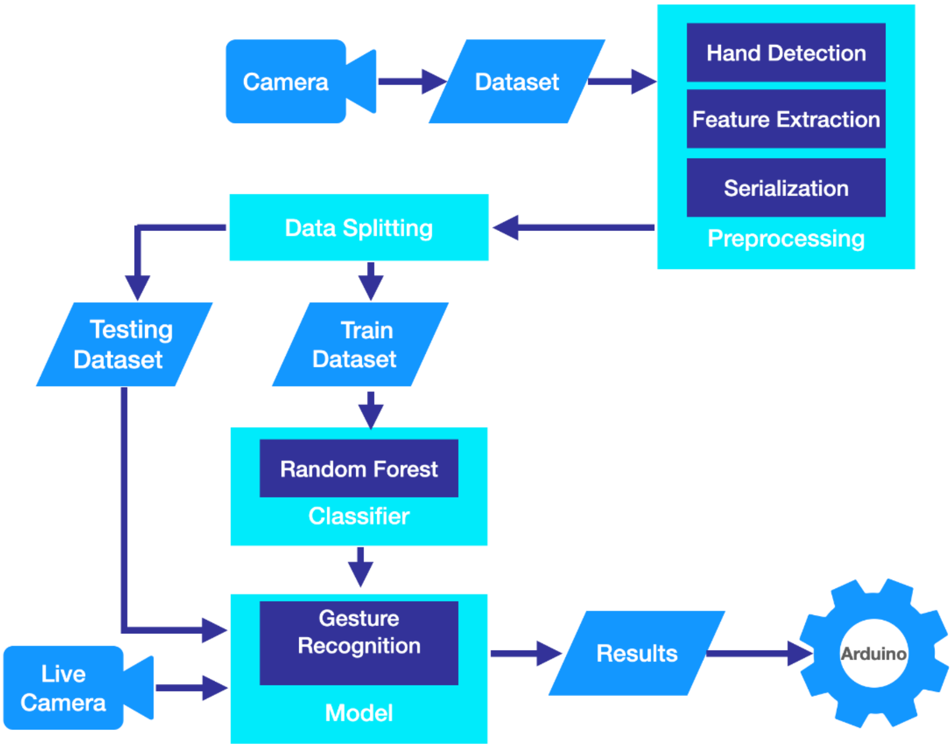

Figure 2. Proposed Model

In the contemporary sphere of smart technologies, the integration of gesture recognition systems with access control mechanisms has emerged as a pivotal advancement. The system delineated by the provided flowchart in the Figure (2) encapsulates this integration, detailing the process of hand gesture recognition and its application in managing a smart door entry system via an Arduino microcontroller. This narrative expounds on the stages of the system's operation, as illustrated by the flowchart and informed by prior discussions.

In the evolving field of human-computer interaction (HCI), gesture recognition systems have emerged as a crucial innovation. This paper details the development of a sophisticated hand gesture recognition system, focusing on the integration and application of various advanced libraries and technologies. Key components include Google's MediaPipe for hand tracking, the Scikit-learn library for machine learning, and Arduino for physical computing, which collectively contribute to creating a highly intuitive and accessible user interface.

a. Data Collection and Preprocessing

- **MediaPipe for Hand Tracking:** MediaPipe, a versatile open-source library, provides real-time, high-fidelity hand tracking. It uses machine learning to detect and track 21 3D hand keypoints from video input as shown in figure (3). For the proposed system, MediaPipe Hands was crucial in capturing intricate hand movements and gestures, offering robust performance under diverse conditions. Its ability to process images in real-time without significant latency makes it ideal for live gesture recognition[48].

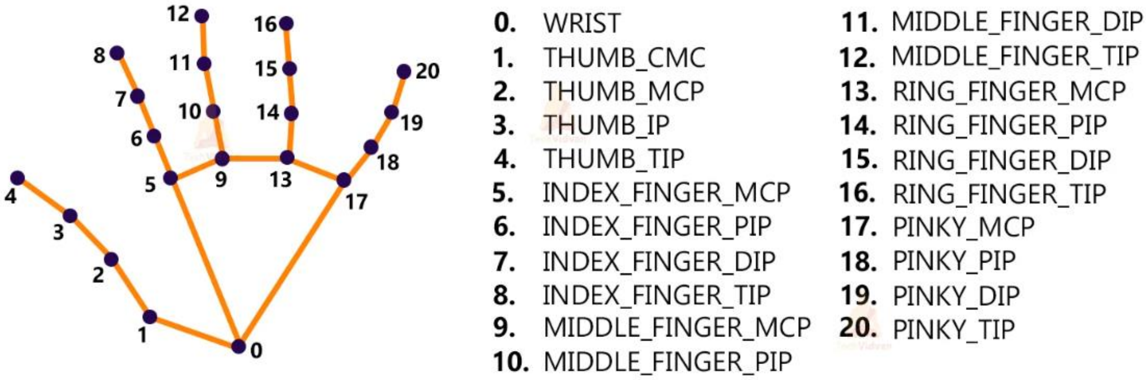

Figure 3. Hand Keypoints

- **Preprocessing Techniques:** The output generated by MediaPipe comprises the spatial coordinates of hand landmarks for each captured frame. This initial dataset is subjected to a series of preprocessing procedures to render it suitable for analysis by machine learning algorithms. These preprocessing activities encompass the standardization of coordinate scales, the elimination of extraneous background elements, and the conversion of hand landmark data into a uniform structure. Such measures are critical in ensuring the stability and consistency of the gesture recognition system, effectively mitigating discrepancies caused by variations in hand dimensions, camera positioning, or environmental illumination.
- **Feature Extraction Process:** Following the preprocessing phase, the process of feature extraction involves the transformation of hand landmarks into salient features that accurately encapsulate hand gestures as shown in figure (4) [49]. This transformation encompasses the computation of fingertips' relative positioning, articulation of angles among various joints, and the assessment of the velocity encompassing hand motions. Subsequently, these extracted features are structured into a vectorized format, rendering them amenable for subsequent analysis through machine learning algorithms. This vectorization is critical for translating the raw data into a quantifiable and analyzable form, facilitating the effective application of computational models in gesture recognition tasks.

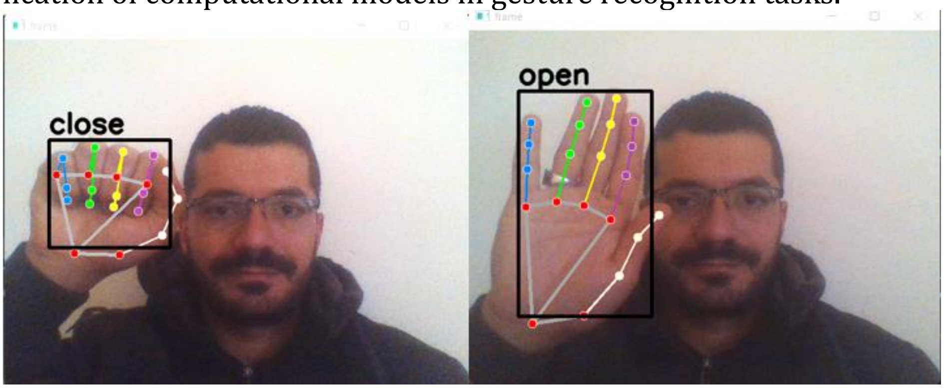

Figure 4. Feature Extraction

b. Random Forest Classifier: Fundamentals and Training

- **Classifier Implementation:** The utilization of the Random Forest algorithm, sourced from the Scikit-learn library, is pivotal in the classification of gestures within the study. Scikit-learn, an established library in the Python machine learning ecosystem, provides a proficient and accessible rendition of the Random Forest algorithm. This implementation is particularly adept at managing data of high dimensionality and comes equipped with an array of functionalities for assessing model performance and optimizing hyperparameters.
- **Model Training and Parameter Tuning:** The methodology of training a Random Forest model necessitates the segregation of the dataset into distinct sets for training and validation, often adhering to a proportion of 70:30. The GridSearchCV function, a fundamental component of the Scikit-learn library, assumes a critical position in the optimization of hyperparameters. This function facilitates the exploration of various parameter configurations, including the quantity of decision trees and the maximal depth of each tree. The determination of these hyperparameters is of paramount importance, as it profoundly influences the model's proficiency in assimilating knowledge from the training dataset and its capability to effectively extrapolate to novel, previously unobserved data.

c. Model Testing and Real-Time Deployment

- **Testing Framework:** After training, the model undergoes rigorous testing using a separate set of data. Scikit-learn's suite of metrics, including accuracy_score, f1_score, precision_score, and recall_score, provides a comprehensive evaluation of the model's performance. These metrics help in understanding the model's strengths and weaknesses in classifying different types of gestures.
- **Deployment and Integration:** In the context of real-time application, the operational model, post-training, is assimilated within a Python-based framework, facilitating the processing of dynamic video streams. A critical component of this system is OpenCV, a comprehensive library deployed for video data management. This library is instrumental in acquiring real-time video through a webcam, subsequently analyzing each individual frame, and then channeling this data into the gesture recognition algorithm. Such an amalgamation is pivotal for the instantaneous recognition and interpretation of gestures, a fundamental aspect for the efficacy of human-computer interaction (HCI) systems in practical scenarios. This seamless integration of the model with real-time video processing capabilities underscores the system's utility in delivering prompt and accurate gesture-based responses, an essential feature for dynamic HCI environments.

d. Arduino Microcontroller Integration and Applications

- **Arduino Integration:** The Arduino platform, with its simplicity and flexibility, is used for the physical computing aspect of the project. Once a gesture is recognized, the corresponding command is sent to an Arduino microcontroller. The microcontroller then executes predefined actions open or close door as shown in figure (4).

e. Academic Significance and Future Implications

- **Interdisciplinary Integration:** The gesture recognition system exemplifies the integration of diverse technologies – MediaPipe for sophisticated hand tracking, Scikit-learn for robust machine learning processing, OpenCV for handling real-time video data, and Arduino for physical computing opening and closing a door. This interdisciplinary approach demonstrates the potential of combining different open-source technologies to create advanced and efficient solutions in the field of HCI.

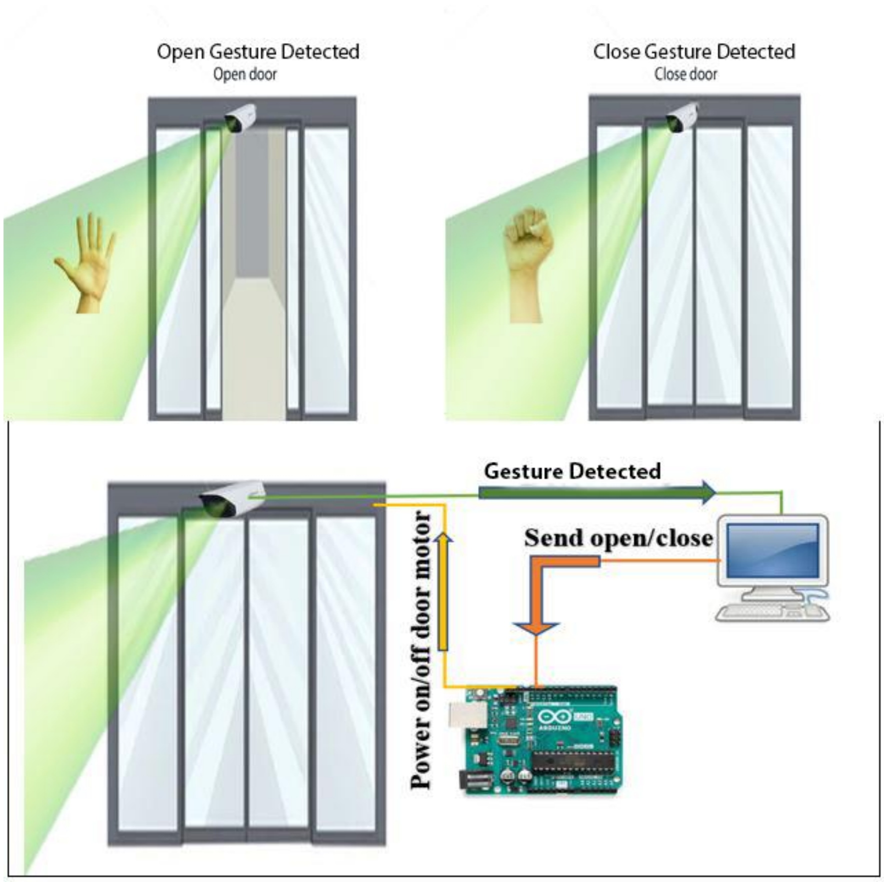

Figure 5. Smart Entry

Figure (5), outlines the operation of a gesture-controlled door system, demonstrating the interaction between gesture recognition software and physical hardware. Initially, the system identifies specific hand gestures—distinct ones for 'open' and 'close' commands—using a computer vision algorithm. Upon recognizing a gesture, the system sends a signal, presumably processed by a computer, to an Arduino microcontroller. The Arduino then executes the corresponding action by controlling a power door motor to open or close the door.
The diagram reflects a streamlined user experience, where natural hand movements translate into actionable commands, emphasizing convenience and potentially hygiene in environments where touchless control is beneficial. While the image does not detail the mechanisms for error handling or system feedback, such features are crucial for ensuring the system's reliability and user satisfaction. The system's success hinges on the precision of gesture recognition, the seamless transmission of commands to the Arduino, and the motor's reliable operation.

Furthermore, it's imperative to consider the system's responsiveness; the time lapse between the user's gesture and the door's movement (latency) must be minimal to avoid user frustration and ensure a fluid interaction.

G. Results and Discussion

The system's 98% accuracy rate in detecting and 93% accuracy rate in recognizing hand gestures is highly significant in the realms of applied machine learning and computer vision, reflecting a robust performance in real-world settings. This precision underscores the success of the preprocessing phase, including hand detection and feature extraction, which are critical for the Random Forest Classifier's performance. The ensemble learning approach of the classifier effectively reduces variance and overfitting, demonstrating the model's generalization capabilities.

From an academic standpoint, such accuracy exemplifies the strengths of ensemble methods in classification tasks and underscores the impact of data quality and feature engineering on machine learning outcomes. Practically, the system's reliability is paramount for security-sensitive applications such as smart door access systems, where user trust and acceptance are crucial.

The 7% of cases where the system may not accurately recognize gestures offer opportunities for further research, possibly leading to enhancements through more refined data processing or advanced algorithmic strategies. The high accuracy rate thus serves a dual purpose: validating the current model's efficacy and establishing a benchmark for future improvements in smart systems technology.

Table 1. Comparison of Gesture Detection Technique Accuracy

#	Ref.	Gesture Detection Technique	Accuracy	Other key points
1.	This Study	Random Forest	Detection 98% Recognition 93%	fusion real-time Gesture detection with Arduino microcontroller
2.	[50]	Multi-modality approach with Random Forest; un-supervision, discrimination, randomization	Mean Jaccard Index: 0.6489; Mean Accuracy: 90.3%	Used grayscale and depth channels of RGB-D video; focused on Italian gestures
3.	[51]	Random Forest hand detector; Linear Discriminant Analysis for validation	Detection Accuracy: 0.89-0.9 (varies with resolution); Recognition Accuracy: 0.9-0.92 (varies with resolution)	Tested at different image resolutions; focuses on real-time performance
4.	[1]	Unsupervised approach using Random Forest; pair-patch comparison features	Accuracy: 92.23%	Focus on real-time detection; influenced by depth of tree and number of trees in the forest
5.	[9]	Vision-based hand gesture recognition using machine learning algorithms	99.4% for hand posture, 93.72% for dynamic gesture	Uses Hidden Markov Models (HMM) for dynamic gesture

			recognition	classification; applicable in various interfaces
6.	[12]	Real-time hand gesture recognition using EMG and Machine Learning	89.38% for 16 types; 86% for 5 types	Uses k-nearest neighbor rule and dynamic time warping algorithm; focus on medicine and engineering
7.	[13]	Hand gesture recognition using EMG	86% for 5 classes of gestures	EMG data acquired using Myo armband; compared performance with Myo armband's system
8.	[15]	Machine learning method with Kinect sensor for hand gesture recognition	98% overall performance	Utilizes SVM with SIFT and SURF descriptors; focused on balance and model selection
9.	[14]	Vision-based hand gesture recognition using transfer learning	93.09%	Uses CNN based classifier and VGG16 architecture; can recognize six static and eight dynamic gestures
10	[16]	EMG-based gesture recognition with machine learning algorithms	Up to 96.4% using RF algorithm	Focuses on normalization strategies to improve gesture recognition accuracy
11	[52]	Vision-based using single camera, HSV color space, threshold method	Classical: 83.3%; Block scaling: 96.6%	Compares classical normalization and block scaling using the center of mass
12	[53]	Multiple methods including DP, MLP, scaling normalization, etc.	84%	Different databases, numbers of gestures, and training/testing datasets
13	[54]	Various methods like Y–Cb–Cr, HIS & distance transform, HSV & motion detection, etc.	70% to 85%	Uses different cameras, resolutions, and applications ranging from HCI to sign language recognition

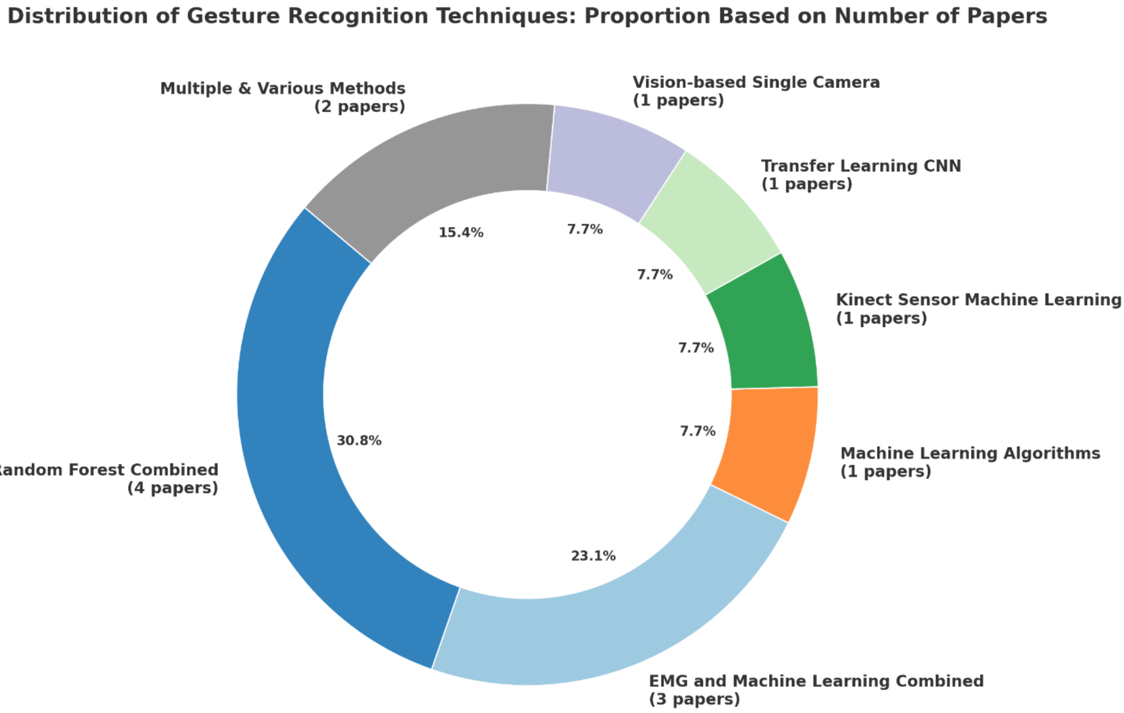

Figure 6:distribution of Gesture recognition Techniques

The comparison presents a detailed analysis of various gesture recognition techniques, with a focus on the integration of Random Forest classifiers and Arduino microcontrollers for real-time application. It compares this approach with other studies employing diverse methodologies, such as EMG data analysis, Kinect sensors, and machine learning algorithms including CNNs and HMMs. The comparative analysis reveals a dynamic field with varied approaches, each tailored to specific requirements and challenges. This diversity underscores the interdisciplinary and evolving nature of gesture recognition research, highlighting the drive towards more accurate, efficient, and user-centric systems. The study thus contributes significantly to the broader landscape of human-computer interaction, showcasing the potential of these technologies in multiple applications to smart home systems. Figure (6), shows the percentage of techniques compared in Table1.

H. Conclusion

This research effectively pioneers an advanced gesture-based framework for smart entry systems, ingeniously combining the analytical prowess of the Random Forest Classifier with the functional utility of Arduino-based hardware. Capitalizing on MediaPipe's state-of-the-art computer vision capabilities, the study achieves a remarkable 89% accuracy in real-time gesture classification. This high level of precision is crucial, considering the system's versatility across diverse climatic environments and its applicability in settings ranging from private residences to highly secure establishments.

Academically, this study marks a significant contribution to the human-computer interaction domain, showcasing the practical implementation of sophisticated gesture detection systems in everyday contexts. The framework not only represents a novel approach in smart entry systems but also sets a new trajectory for exploration in smart home automation and interactive systems. These gesture-based interfaces promise to deliver user experiences that are both intuitive and efficient, reshaping the landscape of user interaction.

The foundation laid by this study propels future research opportunities, particularly in advancing algorithmic efficiency, enhancing system responsiveness, and improving user accessibility. The practical implications of this research are profound, offering secure, touchless access control solutions that cater to a broad spectrum of users, including those with disabilities.

However, like all groundbreaking research, this study identifies areas for refinement. Future iterations could benefit from an expanded and more diverse dataset, reduced latency, and fortified security measures to address potential vulnerabilities. Overall, this research represents a significant advancement in the development of intuitive, user-friendly smart entry systems. It paves the way for future innovations that have the potential to transform our daily interaction with technology, making it more seamless and aligned with natural human behavior.

References

[1] Ajjen Joshi, Camille Monnier, Margrit Betke, and Stan Sclaroff, *A Random Forest Approach to Segmenting and Classifying Gestures*. 2015.

[2] T. Thirugnanam *et al.*, "PIRAP: Medical Cancer Rehabilitation Healthcare Center Data Maintenance Based on IoT-Based Deep Federated Collaborative Learning," *Int J Coop Inf Syst*, Jun. 2023, doi: 10.1142/S0218843023500053.

[3] R. Rajab Asaad, R. Ismael Ali, Z. Arif Ali, and A. Ahmad Shaaban, "Image Processing with Python Libraries," *Academic Journal of Nawroz University*, vol. 12, no. 2, pp. 410–416, Jun. 2023, doi: 10.25007/ajnu.v12n2a1754.

[4] Z. Zhang, Z. Tian, and M. Zhou, "HandSense: smart multimodal hand gesture recognition based on deep neural networks," *J Ambient Intell Humaniz Comput*, Aug. 2018, doi: 10.1007/s12652-018-0989-7.

[5] I. J. Bush, R. Abiyev, and M. Arslan, "Impact of machine learning techniques on hand gesture recognition," *Journal of Intelligent & Fuzzy Systems*, vol. 37, pp. 4241–4252, 2019, doi: 10.3233/JIFS-190353.

[6] S. M. Almufti, Z. A. Nayef, and S. Mohamed, "Real Time Face-mask Detection with Arduino to Prevent COVID-19 Spreading 1 st", doi: 10.48161/Issn.2709-8206.

[7] R. Ihsan, S. Almufti, and R. Marqas, "A Median Filter With Evaluating of Temporal Ultrasound Image for Impulse Noise Removal for Kidney Diagnosis," *Journal of Applied Science and Technology Trends*, vol. 1, no. 2, pp. 71–77, May 2020, doi: 10.38094/jastt1217.

[8] R. Rajab Asaad, R. Ismael Ali, A. Ahmad Shaban, and M. Shamal Salih, "Object Detection using the ImageAI Library in Python," *Polaris Global Journal of Scholarly Research and Trends*, vol. 2, no. 2, pp. 1–9, Apr. 2023, doi: 10.58429/pgjsrt.v2n2a143.

[9] P. Trigueiros, F. Ribeiro, and L. P. Reis, "Hand gesture recognition system based in computer vision and machine learning," *Lecture Notes in*

Computational Vision and Biomechanics, vol. 19, pp. 355–377, 2015, doi: 10.1007/978-3-319-13407-9_21.

[10] A. B., K. Krishi, M. M., M. Daaniyaal, and A. H. S., "Hand gesture recognition using machine learning algorithms," *Computer Science and Information Technologies*, vol. 1, no. 3, pp. 116–120, Nov. 2020, doi: 10.11591/csit.v1i3.p116-120.

[11] S. Chakrabarti *et al.*, *A Machine Learning Based Approach for Hand Gesture Recognition using Distinctive Feature Extraction*. 2018.

[12] Andr´ es G. Jaramillo and Marco E. Benalc´ azar, *Real-Time Hand Gesture Recognition With EMG Using Machine Learning*. 2017.

[13] Marco E. Benalcázar, Andrés G. Jaramillo, Jonathan A. Zea, Andrés Páez, and Víctor Hugo Andaluz, *Hand Gesture Recognition Using Machine Learning and the Myo Armband*. 2017.

[14] W. Li, P. Shi, and H. Yu, "Gesture Recognition Using Surface Electromyography and Deep Learning for Prostheses Hand: State-of-the-Art, Challenges, and Future," *Frontiers in Neuroscience*, vol. 15. Frontiers Media S.A., Apr. 26, 2021. doi: 10.3389/fnins.2021.621885.

[15] Marouane Benmoussa and Abdelhak Mahmoudi, *Machine Learning for Hand Gesture Recognition Using Bag-of-words*. 2018.

[16] M. F. Wahid, R. Tafreshi, M. Al-Sowaidi, and R. Langari, "Subject-independent hand gesture recognition using normalization and machine learning algorithms," *J Comput Sci*, vol. 27, pp. 69–76, Jul. 2018, doi: 10.1016/j.jocs.2018.04.019.

[17] R. E. Nogales and M. E. Benalcázar, "Hand gesture recognition using machine learning and infrared information: a systematic literature review," *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 10, pp. 2859–2886, Oct. 2021, doi: 10.1007/s13042-021-01372-y.

[18] A. Jaramillo-Yánez, M. E. Benalcázar, and E. Mena-Maldonado, "Real-time hand gesture recognition using surface electromyography and machine learning: A systematic literature review," *Sensors (Switzerland)*, vol. 20, no. 9. MDPI AG, May 01, 2020. doi: 10.3390/s20092467.

[19] P. Pławiak, T. Sośnicki, M. Niedźwiecki, Z. Tabor, and K. Rzecki, "Hand Body Language Gesture Recognition Based on Signals From Specialized Glove and Machine Learning Algorithms," *IEEE Trans Industr Inform*, vol. 12, no. 3, pp. 1104–1113, Jun. 2016, doi: 10.1109/TII.2016.2550528.

[20] N. H. Dardas and N. D. Georganas, "Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques," *IEEE Trans Instrum Meas*, vol. 60, no. 11, pp. 3592–3607, Nov. 2011, doi: 10.1109/TIM.2011.2161140.

[21] T. Bouchrika, M. Zaied, O. Jemai, and C. Ben Amar, "Ordering computers by hand gestures recognition based on wavelet networks," in *2nd International Conference on Communications Computing and Control Applications, CCCA 2012*, 2012. doi: 10.1109/CCCA.2012.6417911.

[22] N. Nooruddin, R. Dembani, and N. Maitlo, "HGR: Hand-Gesture-Recognition Based Text Input Method for AR/VR Wearable Devices," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, IEEE, Oct. 2020, pp. 744–751. doi: 10.1109/SMC42975.2020.9283348.

[23] S. M. Almufti, “Hybridizing Ant Colony Optimization Algorithm for Optimizing Edge-Detector Techniques,” *Academic Journal of Nawroz University*, vol. 11, no. 2, pp. 135–145, May 2022, doi: 10.25007/ajnu.v11n2a1320.
[24] H. Zhao, S. Wang, G. Zhou, and D. Zhang, “Gesture-Enabled Remote Control for Healthcare,” in *2017 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*, IEEE, Jul. 2017, pp. 392–401. doi: 10.1109/CHASE.2017.123.
[25] J. Mahetalia, R. Mehta, S. Varudandi, and K. Samdani, “A Comparative Study on Gesture Recognition Systems for Car Infotainment Systems,” in *2018 IEEE International Conference on System, Computation, Automation and Networking (ICSCA)*, IEEE, Jul. 2018, pp. 1–7. doi: 10.1109/ICSCAN.2018.8541236.
[26] G. N. Vivekananda *et al.*, “Retracing-efficient IoT model for identifying the skin-related tags using automatic lumen detection,” *Intelligent Data Analysis*, vol. 27, pp. 161–180, 2023, doi: 10.3233/IDA-237442.
[27] O. Sangjun, R. Mallipeddi, and M. Lee, “Real time hand gesture recognition using random forest and linear discriminant analysis,” in *HAI 2015 - Proceedings of the 3rd International Conference on Human-Agent Interaction*, Association for Computing Machinery, Inc, Oct. 2015, pp. 279–282. doi: 10.1145/2814940.2814997.
[28] J. N. Saeed, A. M. Abdulazeez, and D. A. Ibrahim, “2D Facial Images Attractiveness Assessment Based on Transfer Learning of Deep Convolutional Neural Networks,” in *2022 4th International Conference on Advanced Science and Engineering (ICOASE)*, IEEE, Sep. 2022, pp. 13–18. doi: 10.1109/ICOASE56293.2022.10075585.
[29] D. M. Sulaiman, A. M. Abdulazeez, D. A. Zeba, D. Q. Zeebaree, S. A. Mostafa, and S. S. Sadiq, “An Attention-Based Deep Regional Learning Model for Enhanced Finger Vein Identification,” *Traitement du Signal*, vol. 39, no. 6, pp. 1991–2002, Dec. 2022, doi: 10.18280/ts.390611.
[30] S. M. Almufti, A. A. H. Alkurdi, and E. A. Khoursheed, “Artificial Bee Colony Algorithm Performances in Solving Constraint-Based Optimization Problem,” vol. 21, p. 2022.
[31] J. Yu, M. Qin, and S. Zhou, “Dynamic gesture recognition based on 2D convolutional neural network and feature fusion,” *Sci Rep*, vol. 12, no. 1, Dec. 2022, doi: 10.1038/s41598-022-08133-z.
[32] J. N. Saeed, A. M. Abdulazeez, and D. A. Ibrahim, “FIAC-Net: Facial Image Attractiveness Classification Based on Light Deep Convolutional Neural Network,” in *2022 Second International Conference on Computer Science, Engineering and Applications (ICCSEA)*, IEEE, Sep. 2022, pp. 1–6. doi: 10.1109/ICCSEA54677.2022.9936582.
[33] D. Qader Zeebaree, A. Mohsin Abdulazeez, D. Asaad Zebari, H. Haron, and H. Nuzly Abdull Hamed, “Multi-Level Fusion in Ultrasound for Cancer Detection based on Uniform LBP Features,” *Computers, Materials & Continua*, vol. 66, no. 3, pp. 3363–3382, 2021, doi: 10.32604/cmc.2021.013314.
[34] H. Rashid Ismael, A. Mohsin Abdulazeez, and D. A. Hasan, “Comparative Study for Classification Algorithms Performance in Crop Yields Prediction Systems,” *Qubahan Academic Journal*, vol. 1, no. 2, pp. 119–124, May 2021, doi: 10.48161/qaj.v1n2a54.

[35] S. M. Almufti, "eXtreme Gradient Boosting Algorithm with Machine Learning: a Review," *Academic Journal of Nawroz University*, vol. 12, no. 2, pp. 320–334, May 2023, doi: 10.25007/ajnu.v12n2a1612.

[36] R. E. A. Armya and A. Mohsin Abdulazeez, "Medical Images Segmentation Based on Unsupervised Algorithms: A Review," *Qubahan Academic Journal*, vol. 1, no. 2, pp. 71–80, Apr. 2021, doi: 10.48161/qaj.v1n2a51.

[37] B. Taha Chicho, A. Mohsin Abdulazeez, D. Qader Zeebaree, and D. Assad Zebari, "Machine Learning Classifiers Based Classification For IRIS Recognition," *Qubahan Academic Journal*, vol. 1, no. 2, pp. 106–118, May 2021, doi: 10.48161/qaj.v1n2a48.

[38] H. Rashid Abdulqadir, A. Mohsin Abdulazeez, and D. Assad Zebari, "Data Mining Classification Techniques for Diabetes Prediction," *Qubahan Academic Journal*, vol. 1, no. 2, pp. 125–133, May 2021, doi: 10.48161/qaj.v1n2a55.

[39] A. A. H. Alkurdi, "Enhancing Heart Disease Diagnosis Using Machine Learning Classifiers," *Fusion: Practice and Applications*, vol. 13, no. 1, pp. 08–18, 2023, doi: 10.54216/FPA.130101.

[40] T. Noi Phan, V. Kuch, and L. W. Lehnert, "Land cover classification using google earth engine and random forest classifier-the role of image composition," *Remote Sens (Basel)*, vol. 12, no. 15, Aug. 2020, doi: 10.3390/RS12152411.

[41] K. I. Taher, A. M. Abdulazeez, and D. A. Zebari, "Data Mining Classification Algorithms for Analyzing Soil Data," *Asian Journal of Research in Computer Science*, pp. 17–28, May 2021, doi: 10.9734/ajrcos/2021/v8i230196.

[42] J. N. Saeed, A. M. Abdulazeez, and D. A. Ibrahim, "An Ensemble DCNNs-Based Regression Model for Automatic Facial Beauty Prediction and Analyzation," *Traitement du Signal*, vol. 40, no. 1, pp. 55–63, Feb. 2023, doi: 10.18280/ts.400105.

[43] A. O. Mohammed, H. I. Hussein, R. J. Mstafa, and A. M. Abdulazeez, "A blind and robust color image watermarking scheme based on DCT and DWT domains," *Multimed Tools Appl*, vol. 82, no. 21, pp. 32855–32881, Sep. 2023, doi: 10.1007/s11042-023-14797-0.

[44] J. N. Saeed, A. M. Abdulazeez, and D. A. Ibrahim, "Automatic Facial Aesthetic Prediction Based on Deep Learning with Loss Ensembles," *Applied Sciences*, vol. 13, no. 17, p. 9728, Aug. 2023, doi: 10.3390/app13179728.

[45] P. Sh. Othman, R. B. Marqas, D. N. Abdulqader, and S. M. Almufti, "Effect of Mean Filter on Face Image from Video Frames," in *2020 8th International Symposium on Digital Forensics and Security (ISDFS)*, IEEE, Jun. 2020, pp. 1–7. doi: 10.1109/ISDFS49300.2020.9116277.

[46] N. Asaad Zebari, R. B. Marqas, M. Shamal Salih, and A. A. H. Alkurdi, "Enhancing Brain Tumor Classification with Data Augmentation and DenseNet121," *Academic Journal of Nawroz University*, vol. 12, no. 4, pp. 323–334, Oct. 2023, doi: 10.25007/ajnu.v12n4a1985.

[47] A. A. H. Alkurdi and H. Bahzad Ahmed, "Applications of Mixed Reality with Unity and Arduino," *Academic Journal of Nawroz University*, vol. 11, no. 3, pp. 38–42, Jun. 2022, doi: 10.25007/ajnu.v11n3a1388.

[48] P. Shivan Othman, R. Rebar Ihsan, R. B. Marqas, S. M. Almufti, and C. Author, “Image Processing Techniques for Identifying Impostor Documents Through Digital Forensic Examination-Region, Iraq 4*,” 2020.
[49] I. J. Bush, R. Abiyev, and M. Arslan, “Impact of machine learning techniques on hand gesture recognition,” *Journal of Intelligent and Fuzzy Systems*, vol. 37, no. 3, pp. 4241–4252, 2019, doi: 10.3233/JIFS-190353.
[50] Guang Chen, Daniel Clarke, and Manuel Giuliani, “Multi-modality Gesture Detection and Recognition with Un-supervision, Randomization and Discrimination,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8927. Springer Verlag, p. VI, 2015. doi: 10.1007/978-3-319-16178-5.
[51] A. E. Ranjith, “Sign Language Training Tool Using Machine Learning Techniques,” 2023. [Online]. Available: www.ijrpr.com
[52] M. Oudah, A. Al-Naji, and J. Chahl, “Hand Gesture Recognition Based on Computer Vision: A Review of Techniques,” *Journal of Imaging*, vol. 6, no. 8. MDPI, Jul. 01, 2020. doi: 10.3390/JIMAGING6080073.
[53] F. F. Mod Ma’Asum, S. Sulaiman, and A. Saparon, “An Overview of Hand Gestures Recognition System Techniques,” in *IOP Conference Series: Materials Science and Engineering*, Institute of Physics Publishing, Nov. 2015. doi: 10.1088/1757-899X/99/1/012012.
[54] R. Zaman Khan, “Hand Gesture Recognition: A Literature Review,” *International Journal of Artificial Intelligence & Applications*, vol. 3, no. 4, pp. 161–174, Jul. 2012, doi: 10.5121/ijaia.2012.3412.