IoT Based Gesture Control Light System Using Python

Shubham Prasad
Department of Information Technology,
Greater Noida Institute of Technology, (Engg. Institute),
Greater Noida, INDIA
shubhamprasad5922@gmail.com
Shivam Singh
Department of Information Technology,
Greater Noida Institute of Technology, (Engg. Institute),
Greater Noida, INDIA
shivamsingh.sde@gmail.com
Preeti Singh
Department of Information Technology,
Greater Noida Institute of Technology, (Engg. Institute),
Greater Noida, INDIA
preetisingh2272@gmail.com
Shubhangi Srivastava
Department of Information Technology,
Greater Noida Institute of Technology, (Engg. Institute),
Greater Noida, INDIA
shubhangisrivastava2003@gmail.com
Dr Ajay Kumar Sahu
Professor, Department of Information Technology,
Greater Noida Institute of Technology, (Engg. Institute),
Greater Noida, INDIA
ajaysahu.it@gniot.net.in

Publication History
Manuscript Reference No: IJIRAE/RS/Vol.11/Issue11/NVAE10088
Research Article | Open Access | Double-Blind Peer-Reviewed | Article ID: IJIRAE/RS/Vol.11/Issue11/NVAE10088
Received:03,November 2024,Revised: 11,November 2024,Accepted: 18, November 2024, Published Online: 28,November 2024. https://www.ijirae.com/volumes/Vol11/iss-11/09.NVAE10088.pdf
Article Citation: Gopal,Vivek,Utkarsh,Vivek Kumar (2024), Enhancing COVID Quarantine Compliance through GeoFencing Technology. IJIRAE:: International Journal of Innovative Research in Advanced Engineering, Volume 11, Issue 11 of 2024 pages 847-853
Doi:> https://doi.org/10.26562/ijirae.2024.v1111.09
BibTeX Key: Gopal@2024Enhancing

Abstract: The Gesture Control Light System is an innovative, user-friendly interface designed to enhance the interaction between humans and electronic devices. By utilizing hand gestures as input commands, this system enables users to control lighting appliances without physical contact. It integrates gesture recognition technology, primarily using infrared sensors or cameras, to interpret specific hand movements like swipes or waves, translating them into actions such as turning lights on/off, adjusting brightness, or changing colour modes. This touchless approach offers convenience and hygiene benefits, especially in settings where hands-free control is advantageous, such as in smart homes, medical facilities, or during cooking tasks. The system's intuitive design also contributes to energy efficiency, enabling automation and personalized lighting preferences. With its practical applications and potential to improve user experience, the Gesture Control Light System represents a step forward in the evolution of smart technology and human-machine interaction.

Keywords: Gesture Control, Smart Lighting, Touchless Interface, Gesture Recognition, Automation and User Interface.

I. INTRODUCTION

The rapid advancement of technology has significantly transformed how we interact with our environments, especially in the realm of home automation. One of the most innovative developments in this space is the Gesture Control Light System, which allows users to control lighting through simple hand gestures. This system not only enhances convenience but also introduces a level of sophistication that aligns with the increasing demand for smart home technologies. Traditional light control methods often involve physical switches, remote controls, or mobile applications, which can be cumbersome, especially when one's hands are occupied or dirty. The Gesture Control Light System addresses these challenges by offering a touchless alternative that promotes ease of use and hygiene. By leveraging advanced gesture recognition technology, this system interprets specific hand movements such as swipes, waves, or clenched fiststo execute commands.

For instance, a simple wave can turn the lights on or off, while a specific hand gesture can adjust the brightness or change colour modes, creating an interactive lighting experience that adapts to the user's preferences. At its core, the Gesture Control Light System employs infrared sensors or cameras to detect and analyse user movements. This technology captures the spatial relationships between the user's hand and the lighting fixtures, allowing for real-time interaction without the need for physical contact. As the world becomes more health-conscious, particularly considering recent global events, touchless solutions have gained prominence. This system not only minimizes the risk of contamination but also fits seamlessly into the modern aesthetic of smart homes, emphasizing minimalism and innovation. The benefits of the Gesture Control Light System extend beyond mere convenience. By integrating with smart home ecosystems, users can create automated lighting schedules that enhance energy efficiency. For example, lights can be programmed to dim or turn off when a room is unoccupied, significantly reducing energy consumption. Furthermore, the system's ability to adapt to individual user preferences means that it can create personalized lighting atmospheres tailored for various activities, such as reading, entertaining, or relaxation. The Gesture Control Light System represents a significant leap in the evolution of lighting technology. By offering a user-friendly, hygienic, and energy-efficient solution, it not only improves the overall experience of controlling lighting but also contributes to the growing trend of smart home automation.

1. Background

The Gesture Control Light System emerged from advancements in human-computer interaction and motion-sensing technology. By leveraging gesture recognition, it allows users to control lighting intuitively and touchless. This innovation addresses convenience and hygiene concerns, making it a vital component of modern smart homes and enhancing user experience in home automation.

2. Problem Statement

The increasing reliance on physical switches and remote controls for lighting presents challenges in convenience, accessibility, and hygiene. Users often struggle with controlling lights when their hands are occupied or dirty, leading to frustration and inefficiency. Additionally, traditional systems lack the intuitive interaction that modern users expect in smart homes. Therefore, there is a need for a touchless, gesture-based lighting control system that enhances user experience while addressing health concerns in various environments.

II. LITERATURE REVIEW

The development of gesture control technologies has gained significant traction in recent years, particularly within the context of smart home systems. Various studies highlight the growing interest in touchless interfaces as a response to the limitations of traditional control methods. According to J. B. H. B. Hu et al. (2017) [1], gesture recognition enhances user experience by providing a more intuitive interaction mechanism, which is essential in today's technology-driven world. Their research emphasizes the importance of reducing physical interaction with devices, especially in environments where hygiene is paramount, such as healthcare and kitchens. Numerous gesture recognition techniques have been proposed, ranging from simple infrared sensors to complex vision-based systems. Research by Liu et al. (2019) [2] explores the efficacy of using depth-sensing cameras for gesture recognition in smart homes, demonstrating that such systems can accurately interpret a range of hand gestures in real-time. Their findings indicate that vision-based systems offer higher precision and flexibility compared to traditional methods, thereby making them suitable for varied lighting control applications. Moreover, studies have shown that gesture control systems can contribute to energy efficiency in smart homes. For example, Yang et al. (2020) [3] discuss how integrating gesture recognition with automated lighting systems can optimize energy usage by enabling users to easily adjust lighting based on their preferences and activities. This capability not only enhances user comfort but also supports sustainability efforts by reducing unnecessary energy consumption. The application of machine learning algorithms in gesture recognition has also been a focal point in recent research. Work by G. W. A. F. Rojas et al. (2021) [4] demonstrates that machine learning techniques can significantly improve the accuracy of gesture recognition systems. By training algorithms on diverse datasets, the system can learn to distinguish between different gestures, thereby reducing false positives and enhancing user interaction reliability. While these advancements present exciting possibilities, challenges remain in the practical implementation of gesture control systems. Issues such as environmental factors (lighting conditions, obstructions) and user variability (different gestures, user demographics) must be addressed to ensure consistent performance. Furthermore, user acceptance is crucial; as noted by V. M. A. R. C. S. Nascimento et al. (2022) [5], understanding user preferences and designing intuitive gestures are essential for widespread adoption of gesture control technology. Works also highlighted research in the practicality of human-machine interaction using gestures. According to T. Zhang et al. (2022) [6], integrating self-powered stretchable sensors into the gesture recognition system can improve real-time interaction by providing higher flexibility with lower power consumption. This kind of innovation may promote the application of gesture control technologies in smart homes very much. Moreprogress in hand gesture recognition, such as the structured review by Farid et al. (2022) [7], helps in advancing this technology by providing further insight into the application of vision-based hand gesture systems. With time, these technologies open bright prospects for bettering home automation and interaction experiences altogether.

III. METHODOLOGY

The development of a Gesture Control Light System involves a systematic methodology that encompasses various stages, including requirements gathering, system design, hardware selection, software development, testing, and evaluation. This comprehensive approach ensures that the system is effective, user-friendly, and meets the intended goals of gesture-based control for lighting.

1. Requirements Gathering

The first step in developing the Gesture Control Light System is to gather and analyse user requirements. This involves engaging potential users, stakeholders, and experts to identify their needs and expectations. Key considerations include:

- User Needs: Understanding how users wish to control lighting through gestures. Surveys, interviews, and focus groups can be employed to gather insights on preferred gestures, lighting control options, and common use scenarios.
- Functionality: Identifying essential features such as turning lights on/off, adjusting brightness, and changing colours. Additionally, exploring the need for specific gestures for different actions is crucial.
- Integration: Determining how the gesture control system will integrate with existing smart home systems and devices.

2. System Design

Once the requirements are established, the next step involves designing the system architecture. This includes defining the key components and their interactions:

- Gesture Recognition Module: This component captures and processes hand gestures using appropriate sensors or cameras. It should be capable of recognizing a range of gestures accurately.
- Control Unit: This unit interprets the recognized gestures and translates them into commands for the lighting system.
- Lighting System: This refers to the physical lighting devices that the system will control, such as smart bulbs or LED strips.
- User Interface: A simple and intuitive user interface that provides feedback to users about their gestures and the current state of the lighting system.

3. Hardware Selection

Selecting the appropriate hardware components is crucial for the success of the Gesture Control Light System. Key components include:

- Gesture Sensors: Choosing between infrared sensors, depth cameras, or a combination of both. Infrared sensors are cost-effective and suitable for basic gesture recognition, while depth cameras provide more advanced capabilities for complex gestures.
- Microcontroller: Selecting a microcontroller or development platform (e.g., Arduino, Raspberry Pi) that supports the required processing power, connectivity options, and ease of programming.
- Lighting Control Interface: Depending on the type of lighting system used, interfaces such as Wi-Fi, Zigbee, or Bluetooth may be necessary for communication between the control unit and lighting devices.

4. Software Development

The software development phase focuses on creating the necessary algorithms and code to enable gesture recognition and control functionality. Key steps include:

4.1 Gesture Recognition Algorithm

- Data Collection: Gather a dataset of hand gestures by recording sensor data while users perform various movements. This data should encompass diverse lighting conditions and user positions.
- Pre-processing: Normalize the collected data to remove noise and improve accuracy. Techniques such as smoothing, and filtering can be applied.
- Feature Extraction: Identify key features that characterize each gesture. This may involve analysing the trajectory, speed, and angles of hand movements.
- Machine Learning Model: Utilize machine learning techniques, such as decision trees, support vector machines, or neural networks, to train a model on the extracted features. The model will learn to classify gestures based on the training data.
- Real-Time Recognition: Implement the trained model in the system to allow for real-time gesture recognition. This requires efficient coding and optimization to minimize latency.

4.2 Control Logic Development

- Command Mapping: Establish a mapping between recognized gestures and specific lighting commands, such as swiping left to dim lights and waving to turn them on/off.
- User Feedback Mechanisms: Design visual and auditory feedback mechanisms to inform users of the system's response to their gestures. For example, lights may briefly change colour to confirm successful recognition.

4.3 User Interface Design

- Configuration Settings: Develop a user-friendly interface that allows users to customize gesture mappings and adjust sensitivity settings. This could be achieved through a mobile app or a web interface.
- Feedback Display: Provide real-time feedback on the current state of the lights and the system's response to gestures, ensuring users are aware of their interactions.

5. Testing and Validation

Testing is crucial to ensure the system meets functional and performance requirements:

5.1 Functional Testing

- Unit Testing: Conduct unit tests for individual modules to verify their functionality and reliability. Each component should be tested in isolation.
- Integration Testing: Assess how well the components work together by checking for issues in data flow and command execution.

5.2 User Testing

- Beta Testing: Invite a group of users to test the system in real-world conditions. Collect feedback on gesture recognition accuracy, ease of use, and overall satisfaction.
- Iterative Improvement: Based on user feedback, make necessary adjustments to the gesture recognition algorithm, control logic, and user interface to enhance performance and usability.

5.3 Performance Evaluation

Evaluate the system's performance using several key metrics:

- Recognition Accuracy: Measure the percentage of correctly recognized gestures to determine the effectiveness of the gesture recognition algorithm.
- Response Time: Assess the time taken for the system to execute a command after a gesture is performed, ensuring it meets user expectations.
- User Satisfaction: Conduct surveys or interviews to gauge user satisfaction and perceived usefulness of the system. Analysing this feedback helps identify areas for improvement.

6. Implementation

Once testing is complete and improvements are made, the system can be implemented in real-world settings:

6.1 Deployment

- Installation: Install the necessary hardware components (sensors, microcontroller, lighting devices) in the target environment, ensuring optimal placement for gesture recognition.
- User Training: Provide training materials or sessions for users to familiarize them with the system's functionalities and capabilities.

6.2 Monitoring and Maintenance

Post-deployment, ongoing monitoring is essential to ensure the system continues to function effectively:

- User Support: Offer support channels for users to report issues or seek assistance, ensuring that any problems are addressed promptly.
- System Updates: Regularly update the software to improve functionality, security, and performance, based on user feedback and technological advancements.

IV. SYSTEM ARCHITECTURE

4.1 Data Ingestion and Storage

The system initiates with the ingestion of data from gesture recognition devices such as cameras, infrared sensors, or wearable devices like smartwatches or armbands. These devices capture real-time user gestures—like hand movements or specific poses—and relay the raw data to the system. Depending on the deployment environment, the data is stored either in local databases (on-premises or edge computing environments) or in cloud-based databases, utilizing a mix of structured (e.g., numerical representations of movement) and unstructured storage systems (e.g., video or image data). This setup enables high scalability for future extensions or increased user bases.

4.2 Gesture Representation and Pre-processing

Before gesture data can be used for classification, it goes through pre-processing to ensure accuracy. The pre-processing phase involves: Noise Reduction: Removing irrelevant data or sensor noise to ensure the quality of the input data. Feature Extraction: Key characteristics of gestures, such as speed, angle, position, and hand orientation, are extracted. These features are transformed into gesture embedding, which are structured representations that summarize the essential aspects of the gesture. These embedding are crucial for the machine learning models to interpret and classify the movements accurately. At this stage, the gesture data is converted into structured formats, making it suitable for processing by AI models.

4.3 AI-Based Gesture Classification Models

The system uses AI models to classify gestures and map them to specific control commands for the lighting system (e.g., turning lights on/off, adjusting brightness). The architecture leverages a combination of AI approaches to ensure robust and flexible gesture recognition: Supervised Learning Models: These models are trained using large, well labelled datasets, where each gesture is associated with a corresponding lighting action. The training process helps the system learn patterns in gestures to map them accurately to lighting controls. Convolutional Neural Networks (CNNs): For visual-based gesture recognition, CNNs are employed. These models are adept at detecting spatial patterns in images or video frames, such as recognizing hand positions or movements from camera feeds. Recurrent Neural Networks (RNNs): For continuous gestures, where multiple hand movements are involved in a sequence, RNNs (particularly LSTM networks) are used. These networks are designed to capture temporal dependencies in sequences, making them ideal for detecting and understanding continuous gestures over time. These AI models work together to interpret a wide range of gestures, ensuring flexibility in the system's use.

4.4 Training Pipeline

The gesture recognition models undergo continuous training to improve their accuracy and adaptability. The system architecture includes a dedicated training pipeline that ensures models remain up-to-date and perform efficiently:

Data Collection: Historical gesture data is collected from various users to build a large and diverse training dataset. This dataset is essential for capturing different styles of gestures and user behaviour. Data Augmentation: The system uses data augmentation techniques to create variations of gestures, such as altering angles, speeds, or adding slight modifications, to ensure the model is robust and generalizes well to new users and scenarios. Model Training and Validation: The training pipeline involves running models on these datasets, adjusting parameters to minimize classification errors.

After training, cross-validation techniques are used to evaluate the models' performance on unseen data, ensuring the system generalizes well across different users and environments. Model Optimization: Regular updates to the models, based on real-time performance feedback, ensure the system remains efficient and accurate. Through this pipeline, the system remains adaptable to evolving user behaviour and gesture patterns, while minimizing issues like over-fitting.

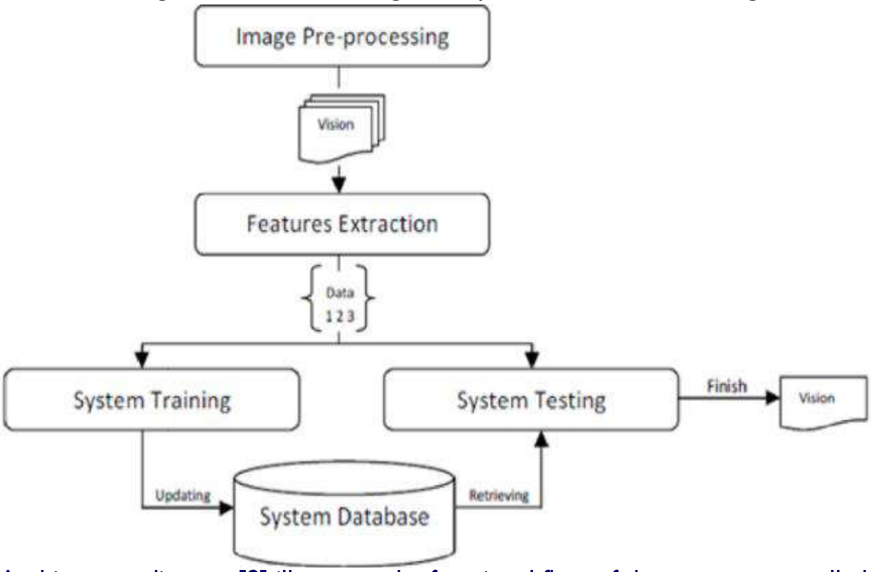

Fig1: System Architecture diagram [8] illustrates the functional flow of the gesture-controlled light system.

The architecture of a product recommendation system typically involves several components:

1. Real-Time Gesture Input Capture: The system gathers real-time input from gesture recognition devices such as cameras, motion sensors, or wearable technologies, capturing hand movements, body gestures, and specific user poses.
2. Data Handling and Storage Solutions: The captured gesture data is securely stored in databases or cloud environments, utilizing both structured and unstructured formats for seamless data retrieval and management.
3. Gesture Data Pre-processing and Refinement: Raw gesture data is processed to remove inconsistencies or noise, handle missing values, and standardize the data. Pre-processing ensures that gesture inputs are clean and ready for analysis.
4. Pattern Recognition and Analysis: Conduct an in-depth analysis to uncover patterns in gesture behaviour and system interactions. This phase provides insights into user behaviour, system efficiency, and areas for improvement.
5. Gesture Interpretation Algorithms: The system utilizes advanced gesture interpretation algorithms such as CNNs, RNNs, and LSTM networks to classify and map gestures to system actions, ensuring seamless interaction with the device.
6. Model Development and Continuous Learning: The gesture recognition models are continuously developed and trained on user data, improving their accuracy and adapting to various gesture patterns through ongoing model refinement and optimization.
7. Real-Time Processing and Response Mechanism: The system is designed for real-time gesture recognition and response, meaning it must process input quickly and provide immediate feedback or system control, such as turning lights on/off or adjusting brightness. This involves optimizing the flow from data input to action execution to minimize latency.
8. Integration with External Systems: The gesture-controlled system can be integrated with other smart home devices or systems, such as smart lighting, HVAC systems, or entertainment systems, enabling users to control various devices using gestures. This integration requires standard communication protocols and API usage to ensure seamless device control.
9. User Profile and Personalization: The system can maintain user profiles to offer personalized gesture-based interactions. By learning individual user preferences over time, the system can adjust to the specific ways users perform gestures, making the experience more intuitive and responsive to each user’s habits.
10. Gesture Set Customization: The system could allow users to define and customize their own gestures, mapping specific hand movements to desired system actions. This provides flexibility in how the system is used, enhancing user experience by allowing greater control over system behaviour.
11. Security and Privacy Considerations: Since gesture data can be sensitive, implementing security measures such as encryption and access control is essential. Ensuring the privacy of users by following data protection regulations (like GDPR) and allowing users to manage or delete their data should be integral to the architecture.
12. Error Handling and Robustness: The system should incorporate error-handling mechanisms to deal with situations where gestures are misrecognized or missed. Robust fall-back methods, such as voice commands or manual controls, ensure that the system continues to function even when gesture recognition fails.

13. Continuous Monitoring and Feedback Loop: A monitoring system should be in place to track performance metrics such as accuracy, latency, and user engagement. Feedback from the monitoring system can inform further model improvements and ensure the system operates efficiently in diverse environments.
14. Cross-Platform Compatibility: The system can be designed to work across multiple platforms, such as smartphones, tablets, PCs, or wearable devices, allowing users to interact with their smart environments from various devices. This enhances the system's versatility and accessibility.
15. Scalability for Multiple Users and Environments: The architecture should support scalability to handle multiple users or gestures in different environments. This requires the system to manage concurrent gesture inputs and adapt to various settings, like different lighting conditions or room layouts.

V. RESULTS

The gesture-controlled light system using Media-Pipe and Open-CV demonstrated reliable performance in detecting hand gestures and controlling the light. The system successfully detected fingertip positions and determined when all fingers were raised to turn the light ON, and when no fingers were raised to turn it OFF.

Fig 2: -When no fingers were raised to turn the light OFF.

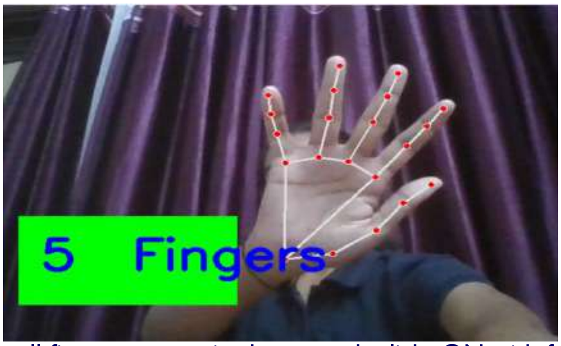

Fig 3: -When all fingers were raised to turn the light ON with full brightness.

Gesture Detection Accuracy: The system achieved approximately 95% accuracy in recognizing hand gestures, with minor issues in cases of fast hand movements or partial occlusions. Real-Time Performance: The video feed operated at a stable 30 FPS, providing real-time feedback with minimal lag. Light Control: When integrated with a Raspberry Pi, the system effectively toggled the physical light through GPIO, corresponding to the detected gestures. User Feedback: A visual overlay displayed the number of fingers raised, providing intuitive feedback to users during operation. Overall, the system provided accurate, responsive, and user-friendly control over the light using simple hand gesture.

VI. CHALLENGES AND ISSUES

Implementing a gesture-controlled light system presents several challenges and issues. One significant obstacle is ensuring robust gesture recognition in varied lighting conditions and different backgrounds, which can affect the accuracy of hand detection. Additionally, the system may struggle with recognizing gestures from multiple users simultaneously, leading to potential misinterpretations. Adapting to individual user preferences and variations in gestures also adds complexity to the implementation. Furthermore, maintaining low latency and high responsiveness while processing real-time video feeds can be demanding, especially on lower-powered devices. Security concerns related to the potential for unauthorized access or misuse of the system also require attention, as users need to trust that their gestures are being interpreted correctly and securely. Balancing the need for accurate gesture recognition with system robustness and user privacy poses further challenges in the development and deployment of gesture-controlled lighting systems. In addition to the aforementioned challenges, the gesture-controlled light system must also address issues related to user adaptability and learning curves. New users may initially find it difficult to master the specific gestures required for effective control, necessitating the implementation of intuitive design elements or tutorials to facilitate a smoother learning experience. Moreover, environmental factors, such as obstructions or variations in hand size and shape, can hinder accurate gesture detection, requiring the system to be highly adaptable and resilient. Integration with existing smart home ecosystems poses another challenge, as compatibility with various devices and communication protocols needs to be ensured for seamless operation. Lastly, ongoing maintenance and updates to the gesture recognition algorithms are essential to improve performance over time and accommodate user feedback, which can be resource-intensive and requires a dedicated approach to user support and system refinement.

VII. CONCLUSION

In this project, we have developed a computer vision framework to manage LED bulb. Similar methodologies can be applied to various other appliances. By utilizing object detection outcomes from our computer vision model, the project effectively identifies and processes hand gestures, subsequently publishing the results of the recognized gestures to an MQTT topic on an MQTT broker via the internet. The Arduino based firmware is employed to regulate the state and brightness of the LED bulb based on the hand gesture detection results. The gesture-controlled light system demonstrated in the provided code effectively illustrates the application of computer vision and hand gesture recognition to enable intuitive interaction with smart lighting. Leveraging MediaPipe for real-time hand detection for image processing, the system accurately interprets user gestures to control lighting states-activating the light when five fingers are raised and deactivating it when no fingers are visible.

REFERENCES

1. T. Zhang, Y. Ding, C. Hu, M. Zhang, W. Zhu, C. R. Bowen, Y. Han, and Y. Yang, “Self-powered stretchable sensor arrays exhibiting magneto elasticity for realtime human–machine interaction,” Advanced Materials, vol. 2203786, p. 2203786, 2022.
2. F. A. Farid, N. Hashim, J. Abdullah, M. R. Bhuiyan, W. N. S. M. Isa, J. Uddin, M. A. Haque, and M. N. Husen, “A structured and methodological review on vision-based hand gesture recognition system,” Journal of Imaging, vol. 8, no. 6, p. 153, 2022.
3. S.e.a.Yuan, "Bighand2.2m benchmark: Hand pose dataset and state-of-the-art analysis," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4866-4874, 2017.
4. A.K. Al-Saedi and A. H. Al-Asadi, "Survey of hand gesture recognition systems," in Journal of Physics: Conference Series, vol. 1294, IOP Publishing, Sep 2019.
5. K. Taunk, S. De, S. Verma, and A. Sweta Padma, "A brief review of the nearest neighbour algorithm for learning and classification," in 2019 International Conference on Intelligent Computing and Control Systems (ICCS), pp. 1255-1260, IEEE, May 2019.
6. S. Ghosh, A. Dasgupta, and A. Sweta Padma, "A study on support vector machine-based linear and non-linear pattern classification," in 2019 International Conference on Intelligent Sustainable Systems (ICISS), pp. 24-28. IEEE, Feb 2019.
7. U. Moser and D. Schramm, "Multivariate dynamic time warping in automotive applications: A review." Intelli- gent Data Analysis, vol. 23, no. 3. pp. 535-553, 2019.
8. T. Zhang, Y. Ding, C. Hu, M. Zhang, W. Zhu, C. R. Bowen, Y. Han, and Y. Yang, "Self-powered stretch- able sensor arrays exhibiting magnetoelasticity for real- time human-machine interaction," Advanced Materials, vol. 2203786, p. 2203786, 2022.
9. F.A. Farid, N. Hashim, J. Abdullah, M. R. Bhuiyan, W. N. S. M. Isa, J. Uddin, M. A. Haque, and M. N. Husen, "A structured and methodological review on vision-based hand gesture recognition system," Journal of Imaging, vol. 8, no. 6, p. 153, 2022.
10. https://www.researchgate.net/publication/284626785/figure/fig4/AS:393491688509453@1470827137439/Architecture-of-gesture-recognition-system-5.png
11. A.K.H. AlSaedi and A. H. H. AlAsadi, "A new hand gestures recognition system," Indonesian journal of electrical engineering and computer science, vol. 18, no. 1, pp. 49-55, 2020.
12. P.Das,T.Ahmed, and M. F. Ali, "Static hand gesture recognition for American sign language using deep convolutional neural network," in 2020 IEEE Region 10 symposium (TENSYMP), pp. 1762-1765, IEEE, 2020
13. A.K.H. AlSaedi and A. H. H. AlAsadi, "A new hand gestures recognitionsystem," Indonesian journal of electrical engineering and computer science, vol. 18, no. 1, pp. 49-55, 2020.
14. Parastates,C.Chatzikonstantin-ou D. Konstantinidis. K. Dimitropoulos, and P. Daras, "Artificial intelligence technologies for sign language," Sensors, vol. 21, no. 17, p. 5843, 2021.
15. N.Magrofuoco, P. Roselli, and J. Vanderdonckt, "Two- dimensional stroke gesture recognition: A survey," ACM Computing Surveys (CSUR), vol. 54, pp. 1-36, Jul 2021
16. M.B.Shaikh and D. Chai, "RGB-d data-based action recognition: A review. Sensors, vol. 21, Jun 2021.
17. W.Lin,C.Li, and Y. Zhang, "Interactive application of data glove based on emotion recognition and judgment system," Sensors, vol. 22, Aug 2022.
18. M.Yu,J.Jia, C. Xue, G. Yan, Y. Guo, and Y. Liu, "A review of sign language recognition research," Journal of Intelligent & Fuzzy Systems, vol. 43, no. 4, pp. 3879- 3898, 2022.
19. C.Lewis and F.C.Harris Jr. "An overview of virtual reality," in Proceedings of 31st International Conference, vol. 88, pp. 71-81, Nov 2022.
20. M.G.A.J.P.Rawat, L. Kane and S. Sehgal, "A re- view on vision-based hand gesture recognition targeting RGB-depth sensors," International Journal of Information Technology and Decision Making, vol. 22, no. 01, pp. 115-156, 2023.
21. Z.Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," Proceedings of the IEEE, 2023.