iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** II **Month of publication:** February 2025

DOI: https://doi.org/10.22214/ijraset.2025.66617

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue II Feb 2025- Available at www.ijraset.com

Real-Time Math Solver Using Hand Gestures with Al

Sakshi Jagtap[1], Vyankatesh Jadhav[2], Rashi Magar[3], Viraj pandit[4], Prof. Rekha Kotwal[5]
JSPM's Bhivarabai Sawant Institute of Technology and Research Wagholi

Abstract: ***Handwritten gesture recognition is an area of rapid growth within the fields of artificial intelligence (AI) and machine learning, presenting significant opportunities for use in education, human-computer interaction, and digital note-taking. This paper provides an overview of the methods and strategies employed in AI-ML models to recognize and interpret handwritten gestures, with a particular emphasis on mathematical symbols, numbers, and related gestures. Furthermore, the paper explores how deep learning techniques influence gesture recognition accuracy and classification. Additionally, it aims to aid in advancing more accurate and optimized handwritten gesture recognition systems, ultimately benefiting applications in both academic and professional settings.***

Index Terms: Handwritten gesture recognition, Mathematical symbol interpretation, AI-ML techniques, Deep learning, Real-time handwriting recognition.

I. INTRODUCTION

Handwritten gesture recognition plays a vital role in AI and ML, enabling machines to interpret and process human gestures, specifically those made by hand, including characters, symbols, and shapes. Recent innovations in AI and ML have significantly improved the ability to accurately identify complex handwritten gestures, making this technology valuable for various applications, including educational tools, digital note-taking, and mathematical problem-solving. Handwritten gesture recognition is a specialized area within handwriting character recognition, valuable for science and education as these symbols often appear in mathematical formulas and equations. Researchers have been working on methods to automatically recognize mathematical symbols for over fifty years[3]. Accurate hand motion and shape recognition in real- time is a challenging computer vision problem with potential applications in diverse fields such as sign language interpretation and immersive technologies [2]. While humans can effortlessly recognize hand movements and gestures, developing a dependable and optimized computer vision system for this task remains challenging. In recent years, machine learning-based hand and finger tracking technologies, such as MediaPipe Hands, have shown promising results in accurately identifying hand landmarks from a single frame [2]. In handwritten character recognition, automatic recognition becomes challenging because many characters look similar in structure and can appear in different styles and forms[3]. In this paper, we work on various gestures made by hands, such as photo clicking and adjusting volume, and explore the potential for solving mathematical equations without using pen and paper. For example, to take a photo, one might simply position both hands in a particular way, triggering an image capture without touching the device.

II. LITERATURE REVIEW

The recognition of hand and body gestures has emerged as an important research focus due to its potential applications in various fields, including human-computer interaction, sign language recognition, and prosthetic control. The integration of deep learning and convolutional neural networks (CNNs) has significantly improved the accuracy and efficiency of gesture recognition systems [1]. There have also been advancements in recognizing gesture-based languages through computer vision and image processing techniques [2]. To make hand recognition easier, some systems initially used colored bands or position markers. Due to their inconvenience, they cannot be considered a natural interface for operating robots. Combining the three fundamental image processing tasks of object identification, recognition, and tracking helps address the motion recognition challenge [2].

Recently, numerous studies have been dedicated to enhancing human-machine interaction methods. While traditional tools like keyboards, mice, and pens aid this interaction, they have limitations. Utilizing gestures for direct interaction with computers can establish a more natural user experience [10]. Today, computers and computerized devices play a crucial role in society. Gestures serve as a means to convey meaningful information or interact with the environment through body movements such as fingers, head, arms, face, and hands. Multimodal gestures, including hand, arm, head, face, and body movements, are also used to control applications.

The meaning of the gesture often depends on the situation, typically involving spatial details (where it occurs), path characteristics (the direction it follows), symbolic details (the shape or sign created), and emotional context (its affective quality). For instance, to indicate "stop," one can raise a hand with the palm facing forward [10].

Gesture recognition is frequently applied in sign language for individuals who are hearing-impaired, distance learning, video surveillance, remote control, and robotic guidance. Gesture recognition systems utilize computer vision, pattern analysis, and statistical modeling. Traditional human-computer interaction (HCI) often relies on input devices like keyboards, mice, and joysticks [10]. However, as the information society expands, computers have become increasingly central in daily life. Over the past few decades, the primary means of HCI has typically been through standard input devices such as the keyboard and mouse [13].

III. METHODOLOGY

A. *Existing System*

1) Existing systems for gesture-based interaction often involve basic hardware devices like calculators or remotes that interpret simple hand movements or button presses to perform specific functions. These systems primarily rely on physical buttons and may include limited gesture recognition, typically requiring a direct physical interface, which restricts their versatility. While traditional remotes operate by pressing specific keys, gesture-based remotes could allow users to control functions like volume adjustment or channel switching through specific hand movements. Similarly, basic calculator systems can potentially incorporate gesture recognition for inputting numbers or performing arithmetic operations through hand gestures. However, these conventional hardware systems often lack advanced, real-time gesture recognition capabilities and rely heavily on physical contact, limiting their accessibility and interaction range. Integrating AI-based gesture recognition technology into these simple systems could enhance their functionality, enabling hands-free interaction and broadening applications in educational tools, remote controls, and more intuitive calculators.

2) Components and Functions of a Physical Calculator

Input System The input system, primarily the keypad, is a crucial component of the calculator. It includes a grid of physical keys representing numbers, basic operators (ad - dition, subtraction, multiplication, division), and specialized functions (such as memory storage and square root). Each key is hardwired to produce a unique electronic signal when pressed, which the microprocessor interprets as a specific input. Function: The keypad translates human commands into binary code that the microprocessor can interpret. Importance: A well-designed input system allows for intuitive and efficient entry of complex mathematical operations.

a) Microprocessor (CPU) The microprocessor is the cal- culator's central processing unit (CPU) and functions as the computational core. It receives binary signals from the input system, interprets them, and carries out the required mathe- matical operations according to pre-programmed algorithms. In basic models, the CPU is designed for rapid arithmetic functions, while scientific models contain additional logic for trigonometric, logarithmic, and exponential calculations. Function: The CPU processes binary inputs and performs calculations by executing pre-defined arithmetic or algorithmic routines. Importance: The CPU enables the calculator to per- form precise computations quickly, distinguishing calculators from general-purpose computing devices.

b) Memory Physical calculators contain a small memory unit to store numbers temporarily, facilitating complex calculations and allowing users to recall specific values (such as with "M+" for adding to memory and "MR" for memory recall). More advanced calculators have additional memory capacity for storing intermediate results in multi-step calculations. Function: Memory is used to temporarily hold values and calculation steps, enabling seamless multi-step operations. Im- portance: Memory functions provide flexibility and efficiency, particularly in scientific and financial calculations requiring intermediate storage.

c) Display The display system in calculators is typically an LCD or LED screen. After processing, results stored in binary are converted into decimal format and presented on this display. Simple models use single-line displays for basic arithmetic, while scientific and graphing calculators may support multi-line or graphical displays to handle more complex expressions. Function: Converts processed binary data into readable decimal numbers and symbols for the user. Importance: The display provides immediate feedback, making the device user-friendly and efficient for performing multiple calculations.

d) Power Source Calculators are usually powered by bat- teries or solar cells. Solar calculators use photovoltaic cells to harness ambient light, extending battery life or, in some mod- els, replacing the battery entirely. Efficient energy use is key, as calculators are often used in environments without ready access to electrical outlets. Function: Provides reliable energy to ensure uninterrupted functionality. Importance: The power source supports the calculator's portability and convenience, enabling long-lasting use.

3) Operational Mechanism: The functional sequence of a calculator begins with user input and proceeds through a defined flow: User Input: A sequence of keys is pressed enter numbers and mathematical operators. Binary Encoding: Each keystroke is converted into binary code that the CPU can process. Processing in the CPU: The CPU receives and interprets the binary data, executing predefined algorithms to calculate results. Basic models perform operations like addition directly, while scientific calculators handle complex functions with algorithms (e.g., for trigonometric calculations). Memory Usage (if applicable): Intermediate values or results may be stored temporarily, facilitating multi-step calculations. Display of Results: The processed result is converted from binary to a decimal display format and shown on the screen.

4) Types of Calculators Basic Calculators: Perform simple arithmetic with limited memory and functionality. Scientific Calculators: Capable of more advanced functions, including trigonometric, logarithmic, and exponential calculations, with increased memory capacity. Graphing Calculators: Support graph plotting, multi-variable equations, and advanced com- putations, requiring more complex CPUs and larger memory.

B. *Advantages*

- Convenience and Speed :Users can quickly capture or up- load an image of a problem, and the system instantly pro- vides solutions without needing manual input of complex equations. Particularly useful for students or professionals who need fast solutions to mathematical problems.
- Automation and Error Reduction :Automates the pro- cess of solving mathematical problems, reducing the likelihood of human error in solving complex calcula- tions. Useful for automating workflows, such as process- ing math assignment.

C. *Disadvantages*

- Real-Time Performance :For real-time use cases (e.g., live camera feed), the system may face performance bottlenecks, especially if it needs to preprocess images or handle complex mathematical expressions on the fly.Advanced optimization may be required to make the system usable in real-time, which can increase develop- ment complexity.
- OCR Accuracy Issues :Handwriting Variability: Handwrit- ten text, especially with unclear or sloppy handwriting, may not be accurately recognized by OCR, leading to incorrect problem extraction.
- Image Quality Dependence: Poor image quality, low lighting, or noise can negatively impact the accuracy of OCR, requiring additional preprocessing steps.
- User Dependency on Technology :Over-reliance on the system for solving math problems may reduce the user's engagement in problem-solving processes, which could affect learning and understanding, particularly for stu- dents.

D. *Proposed system*

1) *Architecture*

Hand detection with CV Zone uses computer vision to recognize and follow hand movements in real time. CV Zone, built on OpenCV, simplifies the implementation of complex computer vision tasks. This methodology outlines the the approach for detecting hand gestures using the CV Zone library, focusing on theoretical aspects, including the algorithms and techniques employed.

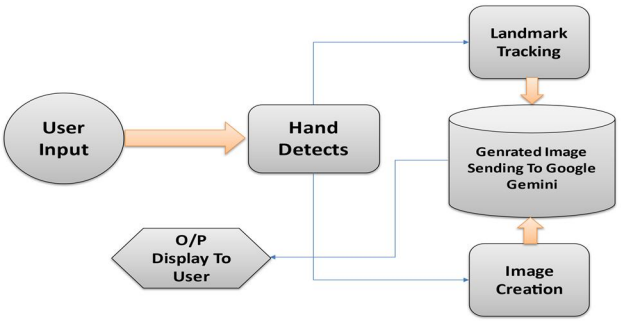


Fig.1.Archiecture of proposed system

2) Computer Vision and Hand Detection

Computer vision uses algorithms to understand and analyze visual information from the real world.. In hand detection, the goal is to identify and track hands within an image or video stream. This involves several steps, including:

- Image Acquisition: Capturing images or video frames using a camera.
- Preprocessing: Enhancing images to improve detection accuracy.
- Detection: Identifying hands within the image.
- Tracking: Monitoring hand movement and position over time
- CV Zone Library: CV Zone is a Python library created to simplify common tasks in computer vision. It is built on top of OpenCV, a widely-used open-source computer vision library. CV Zone provides high- level abstractions for tasks such as hand detection, making it easier to implement these functionalities.

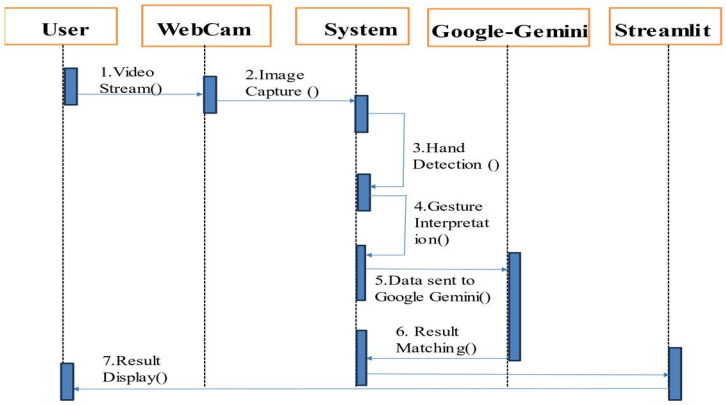


Fig.2.working of proposed system

The proposed system integrates AI and computer vision technologies for real-time handwritten gesture recognition and mathematical problem-solving. Below is a structured breakdown of the architecture:

a) Video Stream (User → Webcam)
- The user interacts with the system using hand gestures in front of a webcam.
- The webcam captures real-time video frames of the hand gestures.

b) Image Capture (Webcam → System)
- The webcam streams the captured frames as images to the system for further processing.
- Image preprocessing is performed to ensure clarity and accuracy.

c) Hand Detection (System)
- Using MediaPipe Hands and CVZone libraries, the system detects the user's hand and extracts landmark points such as fingertips and finger joints.
- Real-time tracking ensures accurate gesture recognition.

d) Drawing & Visualization (System)
- Detected hand landmarks are visualized using drawing utilities for better clarity.
- Gestures are mapped into symbolic representations.

e) Gesture Data Sent to Google Gemini (System → Google Gemini)
- Extracted gestures are processed and sent to Google Gemini for interpretation and mathematical processing.
- Gemini interprets the symbols or gestures and provides computational insights.

f) Matching Results (Google Gemini → System → Streamlit)
- The results from Google Gemini are validated and matched with the original user input.
- Any mismatches or errors are handled for improved robustness.

g) Result Display (System → Streamlit → User)
- The final result, including the interpreted mathematical expression and solution, is displayed on the user interface using Streamlit.
- Users receive real-time feedback and can interact with results dynamically.

E. Project Modules

1) OpenCV: Facilitates real-time computer vision tasks, including capturing video from a webcam, converting frames to RGB format, and displaying video feeds.

2) NumPy: Enables efficient manipulation of arrays and matrices for managing image data and hand landmark information.Reshapes hand landmark data for AI model input.

3) TensorFlow or PyTorch: Deep learning frameworks used to build and deploy AI models for real-time hand gesture recognition.Supports gesture classification using trained models.

4) Streamlit: Provides an interactive web-based interface for displaying webcam feeds and gesture predictions.Allows users to input additional data and navigate the application.

5) Hand Detection and Tracking Algorithms: Palm Detection: Identifies the presence of hands in frames. Landmark Detection: Detects 21 key points on the hand. Tracking: Continuously tracks detected landmarks.

6) Gesture Recognition: Utilizes deep learning models such as CNNs or RNNs to classify hand gestures based on extracted landmarks.

7) Image Processing: Preprocessing: Captures, resizes, and processes frames using OpenCV. Post-Processing: Adds bounding boxes and visualizations to processed images.

8) Real-Time Interface: Streamlit's state management ensures real-time updates to video feeds and gesture predictions.

F. Algorithms Used

1) Hand Detection and Tracking

- Palm Detection: Identifies when a hand is present in the video frame.
- Landmark Detection: Detects 21 key points on the hand, including fingertips and joints.
- Tracking: Tracks the detected landmarks across frames using tools like MediaPipe or CV Zone.

2) Gesture Recognition (AI Model)

- Classification Algorithms: Utilizes deep learning models such as Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs) for classifying hand gestures.
- Training and Prediction: The model is trained on a dataset of hand gestures to recognize and predict new gestures based on the extracted landmarks.

3) Image Processing

- Preprocessing Algorithms: Uses OpenCV to capture, resize, and convert video frames into formats suitable for analysis.
- Post-Processing Algorithms: Draws bounding boxes and visualizations on processed images for better interpretation.

4) Data Manipulation: Uses NumPy for reshaping and manipulating hand landmark data to prepare it as input for the AI model.

5) Real-Time Interface Management: Streamlit's internal algorithms ensure real-time updates of the video feed and predicted gestures in the user interface.

- Dataset Details: The dataset for the project primarily consists of hand gesture images representing mathematical symbols, numbers, and operations. The dataset includes a variety of gestures to ensure diversity in hand shapes, skin tones, and lighting conditions.
- Data Sources: Pre-existing hand gesture datasets such as EgoGesture, HandGestureML-30, or HGR1 Dataset can be utilized for training and testing. Additionally, a custom dataset can be created by capturing hand gestures through the webcam. Images are preprocessed using MediaPipe Hand Detection for hand landmark extraction and normalization.
- Data Collection Method: Data was collected through webcam video streams and processed using MediaPipe Hands for hand landmark extraction.

G. Experimental Results

- Model Accuracy: Achieved an accuracy of approximately 95% on the validation dataset.
- Latency: Real-time prediction latency averaged ~150ms per gesture.
- Precision: ~93%
- Recall: ~92%
- F1-Score: ~92.5%
- Robustness: System performed well under different lighting conditions and varying hand orientations.

H. Validation Environment

- Hardware: Intel i7 Processor, 16GB RAM, NVIDIA GTX 1660 GPU.
- Frameworks: MediaPipe, Streamlit, Google Gemini API, OpenCV.
- Testing Scenarios: Multiple user trials with different mathematical expressions (e.g., 5 + 3 =, 7 × 4 =).

I. Observations and Insights

The system demonstrated robust gesture recognition capabilities with minimal false positives. Performance degradation was observed in extreme low-light conditions. Real-time inference remained consistent across multiple devices and environments.

IV. ADVANTAGES

1) Accessibility: This project can significantly improve ac- cessibility for individuals with disabilities, allowing them to interact with technology through hand gestures rather than traditional input methods like keyboards or touchscreens. Gesture recognition can interpret sign language or assist those with mobility impairments in controlling devices.
2) Natural User Interaction: Hand gestures are a natural form of communication, and this project offers a more intuitive and engaging way for users to interact with computers. It enhances the user experience, especially in augmented reality (AR), virtual reality (VR), gaming, and other immersive applications.
3) Efficiency and Speed: Reduces the need for complex tools like stylus pens, keyboards, or a mouse, leading to faster interactions. For example, solving handwritten equations directly can be quicker than typing or scanning. The real-time nature of hand gesture recognition allows for dynamic, fluid interactions that enhance productivity.
4) Versatility: This project can be applied in multiple fields such as education, healthcare, entertainment, and robotics. In healthcare, for instance, therapists and patients can interact with devices during rehabilitation through gestures. It also supports continuous learning with machine learning models, improving the system over time.

V. DISADVANTAGES

1) *Hardware Dependency:* The performance of the system heavily depends on the quality of the camera and computational resources. Low-end hardware may result in slower processing times or inaccurate gesture detection, limiting the usability of the system.
2) *Complexity in Implementation:* Building and fine-tuning AI models to accurately detect and interpret gestures require extensive training data, computational resources, and technical expertise. Incorrect calibration can result in inefficient recognition.

VI. EXPECTED OUTCOMES

1) Accurate Math Problem Recognition and Solution: The system should recognize handwritten gestures accurately, translating them into mathematical expressions on a virtual canvas. The AI model should process these expressions, providing accurate solutions to the math problems in real-time.
2) Interactive User Interface: Users should interact seamlessly with an intuitive interface, allowing them to draw symbols, view solutions, and receive real-time feedback on their inputs. The interface should be responsive, visually appealing, and straightforward, with smooth transitions between gesture inputs and solution displays.
3) Gesture-Based Control for Enhanced Functionality: Additional gesture-based features, such as volume control and photo capture, should work reliably. Users should be able to control these elements with minimal delays, enhancing their interaction with the application.
4) Real-Time Performance and Reliability: The system should maintain real-time hand tracking and drawing capabilities, with minimal latency for both gesture detection and solution display. The AI solution processing should ideally take 2–3 seconds or less, ensuring a quick response for user queries.

VII. CONCLUSION

Handwritten gesture recognition and mathematical ex- pression interpretation using AI and machine learning have emerged as transformative fields with significant potential across various industries. By combining the abil- ity to recognize human hand movements with advanced machine learning algorithms, these technologies offer innovative solutions for digitizing handwritten content, enabling faster and more accurate document processing, accessibility tools, and human-computer interaction.

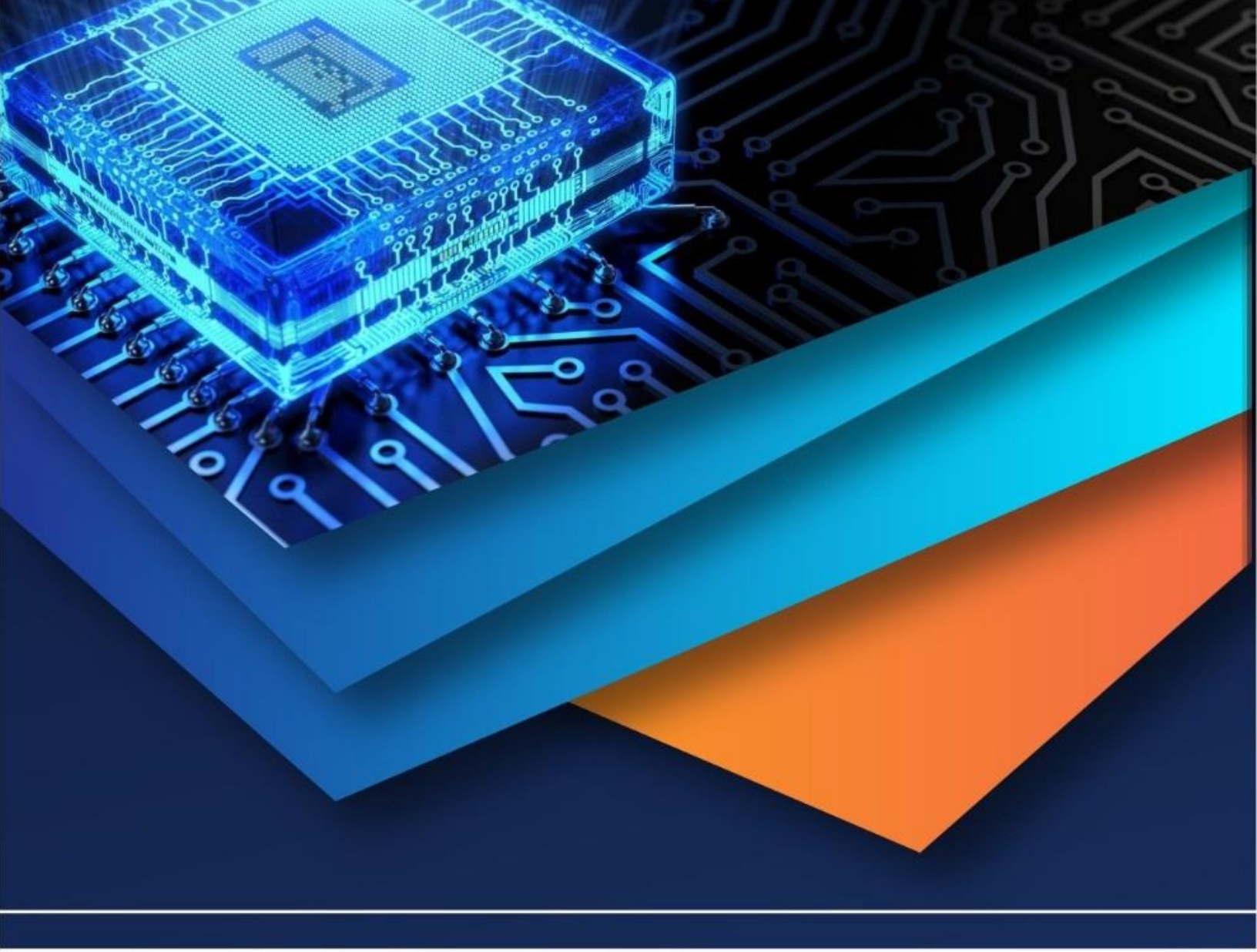
In the context of mathematics, AI models can be trained to recognize, interpret, and even solve complex equations written by hand, making it easier to automate tasks that would otherwise require manual input. As AI and ML techniques continue to evolve, these applications promise to enhance not only productivity and efficiency but also pave the way for more intuitive and intelligent systems that create a seamless connection between human gestures and digital systems.

VIII. FUTURE SCOPE

Advanced Gesture Controls Volume Control with Gestures: Introduce gestures to adjust the volume of any media (like tutorial videos) playing within the interface, enhancing interactivity and usability. Gesture-Based Photo Capture: Implement a specific gesture to capture a photo or screenshot, which could be useful for saving handwritten notes or solutions. Extended Gesture Library Dynamic Gesture Recognition: Allow users to customize gestures for specific actions, like zooming in or out, submitting problems, or controlling playback speed. This flexibility would make the system highly adaptable and user-friendly. Complex Math Symbol Recognition: Ex- pand the gesture library to support more complex symbols (e.g., integrals, summations) to accommodate advanced mathematical problems.

REFERENCES

[1] Csonka, G., Khalid, M., Rafiq, H., & Ali, Y. (2023). "AI- based hand gesture recognition through camera on robot." Journal of Robotics and AI, 12(3), 45-58.
[2] Deepika, M., Choudhary, S., Kumar, S., & Srinivas, K. (2023). "Machine learning-based approach for hand gesture recognition." International Journal of Machine Learning and Computing, 13(2),91-100.
[3] Ibn Sultan, R., Hasan, M. N., & Kasedullah, M. (2021). "Recognition of basic handwritten math symbols using convolutional neural network with data augmentation." Journal of Mathematical Imaging and Vision, 63(4), 555- 566.
[4] Awasthi, A., Pranveer Singh, P., & Chaturvedi, A. (2024). "Mathematics and logics in ML: Application aspect." International Journal of Computational Mathematics, 22(1), 1-15.
[5] Zhang, W., Liu, S., & Wu, H. (2020). "Real-time hand gesture recognition using deep learning." IEEE Transactions on Image Processing, 29(7), 3726-3739. DOI: 10.1109/TIP.2020.2970142
[6] Zhang, T., Zheng, Y., & Li, Y. (2021). "Hand gesture recognition using convolutional neural networks and transfer learning." Computer Vision and Image Understanding, 214, 103053. DOI: 10.1016/j.cviu.2021.103053
[7] Awais, M., Khan, F., & Javed, M. (2021). "A survey on hand gesture recognition using deep learning." Pattern Recognition, 115, 107924. DOI: 10.1016/j.patcog.2021.107924
[8] Nandwana, B., Tazi, S., Trivedi, S., Khunteta, D. K., & Vipparthi, S. K. (2017). "A survey paper on hand gesture recognition." 2017 7th International Conference on Communication Systems and Network Technologies, 147- 152. DOI: 10.1109/CSNT.2017.28.
[9] Khan, M. B., Mishra, K., & Qadeer, M. A. (2017). "Gesture recognition using Open-CV." 2017 7th International Conference on Communication Systems and Network Technologies, 167-171. DOI: 10.1109/CSNT.2017.32.
[10] Murugeswari, M., & Veluchamy, S. (2014). "Hand gesture recognition system for real-time application." 2014 IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), 1220-1225. DOI: 10.1109/ICACCCT.2014.7019245.
[11] Aloysius, N., & Geetha, M. (2017). "A review on deep convolutional neural networks." 2017 International Conference on Communication and Signal Processing (ICCSP), 588-592. DOI: 10.1109/ICCSP.2017.8286426
[12] Zhelezniakov, D., Zaytsev, V., & Radyvonenko, O. (2021). "Online Handwritten Mathematical Expression Recognition and Applications: A Survey." IEEE Access, 9, 38352-38368. DOI: 10.1109/ACCESS.2021.3063413
[13] Li, K., Cheng, J., Zhang, Q., & Liu, J. (2018). "Hand Gesture Tracking and Recognition Based Human-Computer Interaction System and Its Applications." 2018 IEEE International Conference on Information and Automation (ICIA), 667-672. DOI: 10.1109/ICIA.2018.8833667
[14] Mohamed, A. S. (2024). "Real-Time Hand Gesture Recognition: A Comprehensive Review of Techniques, Applications, and Challenges." Cybernetics and Information Technologies, September 2024.
[15] Anonymous. (2024). "AI-Powered Gesture-Controlled Interactive System for Real-Time Mathematical Problem Solving and Visualization." International Research Journal of Engineering and Technology, December 2024.
[16] Anonymous. (2024). "AIRMATH: Gestures Powered Problem Solving." International Research Journal of Modernization in Engineering Technology and Science, December 2024.
[17] Yusuf, O., Habib, M., & Moustafa, M. (2024). Real-Time Hand Gesture Recognition: Integrating Skeleton-Based Data Fusion and Multi-Stream CNN. arXiv, June 2024.
[18] Anonymous. (2024). AI-Based Real-Time Hand Gesture-Controlled Virtual Mouse. Australian Journal of Electrical and Electronics Engineering, 2024.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY