IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

AI-Powered Handwritten Equation Solver Using Mediapipe And Deep Learning

Prof. Rekha Kotwal [1] JSPM's Bhivarabai Sawant Institute of Technology and Research Wagholi

Rashi Magar [4]
JSPM's Bhivarabai Sawant Institute of Technology and Research Wagholi

Vyankatesh Jadhav [2] JSPM's Bhivarabai Sawant Institute of Technology and Research Wagholi

Viraj pandit [5] JSPM's Bhivarabai Sawant Institute of Technology and Research Wagholi

Sakshi Jagtap [3] JSPM's Bhivarabai Sawant Institute of Technology and Research Wagholi

Abstract—The field of handwritten gesture recognition is experiencing rapid growth within artificial intelligence (AI) and machine learning (ML), offering substantial opportunities in education, human-computer interaction, and digital note-taking. This paper presents an overview of the methodologies and strategies utilized in AI-ML models for recognizing and interpreting handwritten gestures, with a particular focus on mathematical symbols, numbers, and related gestures. Moreover, the impact of deep learning techniques on the accuracy and classification of gesture recognition is examined. Additionally, this study contributes to the development of more precise and optimized handwritten gesture recognition systems, ultimately enhancing applications in both academic and professional environments.

Keywords —Handwritten gesture recognition, Mathematical symbol interpretation, AI-ML techniques, Deep learning, Real-time handwriting recognition.

I. INTRODUCTION

Handwritten gesture recognition is a key component of artificial intelligence (AI) and machine learning (ML), allowing machines to understand and process human input in the form of hand gestures, symbols, and character patterns. With the progression of AI and ML techniques, the precision of identifying intricate handwritten inputs has seen notable improvements. This advancement has led to practical implementations in educational environments, real-time note-taking tools, and solving mathematical problems. As a distinct field within handwriting recognition, the ability to interpret handwritten gestures is especially valuable in domains like science and academics, where mathematical notations are frequently used in equations and expressions [5].

Over the years, numerous methods have been studied to achieve automatic recognition of mathematical symbols. Nonetheless, capturing and interpreting real-time hand gestures remains a difficult aspect of computer vision. This challenge holds promise for applications like interpreting sign language or integrating with immersive technology platforms [9]. While humans can naturally understand gestures, designing a computer vision system that performs this reliably and efficiently remains an ongoing research concern [4].

Emerging technologies for hand and finger tracking, such as MediaPipe Hands, have shown strong potential in precisely identifying hand landmarks from single-frame images [7]. Despite this, recognizing handwritten characters automatically is still difficult due to visual similarities between characters and variations in individual writing styles [6]. This study delves into various hand gestures, including those associated with media control functions like image capturing and volume adjustment, and also looks into alternatives to pen-and-paper methods for mathematical input. For example, placing both hands in a certain position could activate a command, avoiding the need for physical interaction with devices.

II. LITERATURE REVIEW

Gesture recognition involving both hands and the body has emerged as a critical area of study because of its versatile applications—ranging from human-computer interaction and sign language interpretation to assistive technologies like prosthetics. The use of deep learning methods, particularly convolutional neural networks (CNNs), has played a major role in increasing both the precision and performance of these systems [3]. Moreover, developments in computer vision and image processing techniques have contributed to enhanced recognition accuracy for gesture-based communication systems [4].

Earlier approaches to hand recognition often relied on physical aids like color-coded gloves or motion-tracking bands, but these were impractical and lacked intuitive user interaction. A more efficient strategy involved combining core processes such as object detection, recognition, and tracking, which helped overcome many issues related to motion analysis [4].

In recent research, there's a growing emphasis on improving the way humans interact with machines. While traditional input methods—keyboards, mice, and styluses—have long served as the main communication tools, they inherently limit the user experience. Gesture-based interaction offers a more seamless and natural alternative for engaging with computers [12]. As digital systems become an increasingly integral part of daily routines, gestures have gained importance as an intuitive medium for communication or interaction with devices. These can involve movements of the fingers, arms, head, face, or a combination of different body parts. The meaning of a gesture is often shaped by factors like location, direction, symbolic intent, and emotion. For instance, a raised hand with an open palm is universally seen as a "stop" signal [12].

This recognition technology finds application in multiple fields, including but not limited to: aiding communication for the hearing-impaired, facilitating remote learning, monitoring through surveillance systems, touch-free device control, and guiding robotic navigation. Building such systems involves the integration of computer vision, pattern recognition, and statistical modeling. Historically, human-computer interaction (HCI) has leaned on devices such as keyboards, mice, and gaming controllers [12]. Yet, with the widespread integration of computers into everyday scenarios, there is growing motivation to develop more natural and immersive user interfaces. Despite advancements, conventional devices like keyboards and mice have largely remained the dominant input tools for HCI over the past several decades [15].

III. METHODOLOGY

A. Existing System

1.Conventional Systems:

Traditional gesture-based interaction systems typically utilize physical hardware like calculators or remote controls that function through button inputs or limited gesture-based commands. These devices rely heavily on tactile buttons and require direct interaction, which limits their flexibility [3]. While standard remotes operate through button pressing, newer models support gesture-based controls, allowing actions like volume adjustment or channel changes via hand movements [4]. Likewise, basic calculators could incorporate gesture input to allow number entry and operations without physical contact. However, these traditional setups generally lack the ability to interpret gestures in real time, reducing their effectiveness for contactless operation. By introducing AI-powered gesture recognition, such devices could offer more dynamic and user-friendly interfaces, particularly beneficial in education, smart control systems, and enhanced user interaction [5].

2. Structure and Functions of a Basic Calculator

2.1 Input Mechanism : The keypad is the main user input interface on calculators, arranged in a grid with keys representing digits, basic operations (addition, subtraction, multiplication, division), and advanced functions such as memory access or square root calculations. Each keystroke sends a distinct electronic signal, which the microprocessor interprets as specific instructions [6]. Function: Converts user actions into digital signals that the processor can interpret. Importance: Ensures accurate and efficient entry of complex equations.

2.2 Central Processing Unit (CPU): The CPU acts as the brain of the calculator. It receives binary inputs from the keypad, executes calculations using built-in algorithms, and returns results. While basic calculators manage elementary arithmetic, scientific models handle functions like logarithms, trigonometry, and exponentials [7]. Function: Executes computations using algorithmic logic. Importance: Ensures high-speed and accurate calculations, setting calculators apart from standard computing tools.

2.3 Memory Unit: Calculators feature memory storage that temporarily holds numerical data. Features such as "M+" and "MR" support multi-step problem solving. More advanced models come with greater memory capabilities for sophisticated calculations [8]. Function: Retains values during multi-step processes. Importance: Facilitates smoother workflows, especially for scientific and financial calculations.

2.4 Display Interface: Once processing is complete, the results are presented on screens like LCDs or LEDs. The binary output is transformed into decimal numbers for the user to interpret. Entry-level calculators display single-line outputs, while advanced models show multi-line formats or even graphs [9]. Function: Renders processed data into understandable formats. Importance: Offers immediate feedback, improving usability and clarity.

2.5 Power Supply: Most calculators use batteries or solar energy for operation. Solar-powered devices utilize light-sensitive cells to function without traditional batteries, promoting longevity and efficiency—especially in areas with limited access to electricity [10]. Function: Provides the necessary energy for operation. Importance: Enables consistent use in diverse environments through energy efficiency.

3. How Calculators Work : The calculator executes tasks in a specific sequence. First, users input data using the keypad. These inputs are converted into binary, processed by the CPU through programmed algorithms, and then results are displayed. While basic versions support standard arithmetic, scientific calculators can perform functions like trigonometry and logarithmic operations [7]. If required, intermediate results are stored in memory to support multi-step calculations, before being displayed on the screen in a readable format.

4. Categories Calculators: Basic Calculators: Designed for standard arithmetic with limited functionality. Scientific Calculators: Offer features for advanced mathematics, including trigonometric and logarithmic functions, supported by larger memory [8]. Graphing Calculators: Capable of plotting graphs and solving multi-variable equations, these devices require more complex CPUs and greater storage [9].

Advantages :

1.Ease and Speed**:** Gestures allow users to enter problems quickly without using a physical interface, which is especially advantageous for learners and professionals needing rapid computation.

2.Automation and Error Reduction**:** By automating calculations, gesture-based systems decrease the likelihood of manual errors—an asset in areas like grading or automated workflows.

Disadvantages :

1. Challenges in Real-Time Processing: Interpreting complex math expressions or rapid input gestures in real time can introduce lag or inaccuracies. Optimization is necessary, increasing implementation complexity.

2.Limitations of OCR**:** OCR performance may decline due to differences in handwriting, suboptimal lighting, or poor image resolution—requiring better preprocessing algorithms for reliable recognition.

3. Reliance on Technology: Excessive use of gesture-based solutions might reduce user involvement in learning environments, where manual problem-solving is still essential for educational development.

By addressing these challenges, AI-based gesture recognition can significantly enhance the functionality of conventional calculators, providing a more intuitive and efficient method of interaction.

B. Proposed System

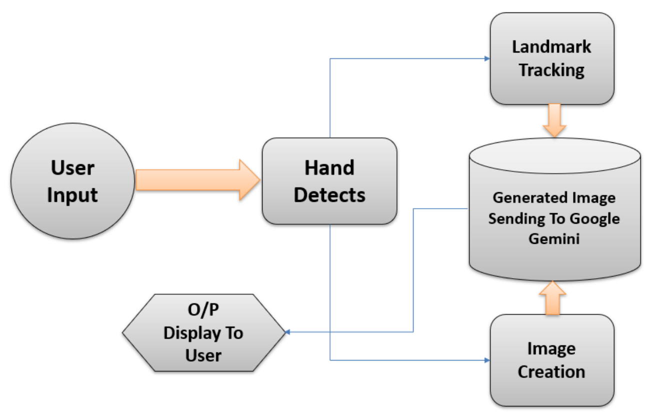


Fig.1.System Archieture

1.Hand Detection

MediaPipe's hand tracking mechanism functions through two main stages: detecting the palm and then identifying hand landmarks. Rather than detecting the entire hand at once, the system starts by locating the palm since it tends to have a consistent shape, making it simpler and faster for deep learning models to detect. This stage uses a Single Shot Multibox Detector (SSD), a CNN-based detection algorithm, with MobileNetV2 as its core architecture. MobileNetV2 is optimized for efficiency, especially on mobile platforms, enabling real-time palm recognition in a single forward pass—significantly faster than other methods like Faster R-CNN.

The SSD uses several convolutional layers to identify features from the input. This process can be modeled as:

$$f(x)=\sigma(W*x+b)$$

where:
X is the input image,
W is the weight matrix,
B is the bias, and
σ is the activation function (usually ReLU)

After detecting the palm, the system proceeds to landmark detection, where 21 key hand points are identified. Unlike palm detection, this stage relies on geometric computation and inverse kinematics rather than deep learning. The landmark identification model is a regression-based neural network that inputs the cropped palm image and outputs 3D coordinates of each hand key point. This is represented by:

$$Y^{\wedge}=WX+b$$

where:
X is the feature vector derived from palm detection,
W and b are learned weights and biases, and
Y^ represents the predicted 3D positions of the hand's key landmarks.

This systematic method helps MediaPipe achieve high accuracy in real-time gesture tracking, making it ideal for gesture-driven applications.

2.Hand Landmark Tracking

Once the palm detection and hand landmark detection are completed, the next step involves tracking a specific finger's movement to create a continuous drawing effect. Mediapipe's Hand Tracking Model predicts 21 key landmarks on the hand, each assigned an (x, y, z) coordinate in the image space. To track the fingertip for drawing, we extract the landmark corresponding to the index finger's tip (Landmark 8). At any given time t, the fingertip's position can be represented as:

$$P\_t = (x\_t, y\_t)$$

where x_t and y_t are the real-time screen coordinates of the fingertip. By continuously tracking these coordinates, we can create a path that follows the finger's movement.

To generate the drawing effect, we use OpenCV to maintain a persistent canvas. The fingertip's previous position is stored, and a line is drawn between the last known position and the current position. The previous position at time t-1 is:

$$P\_\{t-1\} = (x\_\{t-1\}, y\_\{t-1\})$$

Using OpenCV's cv2.line() function, a line is drawn between these two points to simulate a continuous stroke:

cv2.line(canvas, P_{t-1}, P_t, (color), thickness)

This ensures that as the fingertip moves, a smooth trajectory is traced on the screen. However, real-time tracking can be affected by small fluctuations due to sensor noise or hand tremors, leading to an unstable drawing experience. To counteract this, we apply a smoothing technique, such as averaging the last few detected positions to stabilize the movement. The smoothed position at time t is given by:

$$\bar{P}\_t = (P\_t + P\_\{t-1\} + P\_\{t-2\} + ... + P\_\{t-n\}) / n$$

where n is the number of past frames considered for smoothing. This approach helps reduce sudden jumps in the detected coordinates and ensures a more fluid drawing experience. Another effective technique for stabilizing movement prediction is the Kalman Filter, which estimates the next position based on previous positions and velocity.

To enhance control over the drawing process, gesture-based commands can be implemented. For instance, drawing can start when the index finger is up while the rest of the fingers are down. Conversely, if the hand is lifted or the palm is open, drawing can be paused. These gestures can be detected by analyzing the relative y-coordinates of the fingers using geometric analysis techniques.

This method enables real-time gesture-based image creation, which can be extended to include dynamic stroke thickness, color changes, and AI-based recognition of drawn shapes.

3.Image Equation Processing

When an image of a handwritten equation is sent, Gemini first uses Image Understanding (Visual Reasoning), leveraging Convolutional Neural Networks (CNNs). These networks break down the image, recognizing patterns from simple edges to complex symbols, enabling the model to "see" the equation.

Next, Optical Character Recognition (OCR) converts the image into text. Implicitly, Gemini's vision models recognize text, while explicitly, it might use Recurrent Neural Networks (RNNs) or Transformers. The goal of OCR is to produce text: T = {t1, t2, ..., tn}

from an image I, where each t_1 is a character.

The basic RNN equation is: $h\_t = f(h\_(t-1), x\_t)$

where h_t is the hidden state, x_t is the input, and f is an activation function.

Once the AI successfully extracts the equation, it is converted into a symbolic mathematical expression for solving. SymPy, a Python library for symbolic computation, is used to process and solve the equation. For example, given the equation:

$$2x + 5 = 15$$

SymPy solves for x as follows:

$$x = (15 - 5) / 2 = 5$$

For more complex mathematical operations, such as differentiation and integration, SymPy provides built-in functions. For example, for the function:

$$f(x) = \sin(x) + x^2$$

The derivative is computed as:

$$d/dx\ (\sin(x) + x^2) = \cos(x) + 2x$$

Subsequently, Mathematical Expression Parsing transforms the text into a structured form, often using Abstract Syntax Trees (ASTs). This stage interprets symbols and operators, converting "2x + 5 = 15" into a machine-readable format.

Finally, Equation Solving employs algebraic manipulation and arithmetic, using training data to compute the result, which is then presented as text.

4.Streamlit

After Generative AI processes the handwritten equation, the solution is displayed using Streamlit, a powerful Python framework designed for building interactive and real-time web applications. Streamlit simplifies the development of AI-driven interfaces by providing a seamless and efficient way to display results dynamically without requiring manual updates or complex front-end coding.

One of the key advantages of using Streamlit is its ability to render mathematical expressions in a well-formatted manner using LaTeX, ensuring clarity and readability. This is particularly useful for displaying equations, derivatives, and integrals in a structured format that resembles standard mathematical notation. The framework allows users to interact with the system in real time, modify inputs, and instantly see the updated solutions, making it highly effective for applications requiring user engagement.

Streamlit's web-based interface makes the solution easily accessible from any device with a browser, eliminating the need for local installations. It supports interactive elements such as sliders, buttons, and input fields, enabling a user-friendly experience for exploring different mathematical computations. Additionally, it seamlessly integrates with AI models, allowing for smooth communication between the equation recognition system and the solution display.

Advantages:

1.Real-Time Equation Recognition: The system processes and solves handwritten mathematical equations instantly, providing quick and accurate results.

2.User-Friendly Interface: With Streamlit, users get an interactive, web-based interface that is simple to use without requiring technical expertise.

3.Flexibility in Input: The system can recognize equations written by hand using finger tracking, making it accessible for touchscreen and gesture-based inputs.

4.AI-Driven Accuracy: The combination of Optical Character Recognition (OCR) and Deep Learning models ensures high accuracy in recognizing mathematical symbols and equations. Customizable and Scalable: The system can be enhanced with additional AI models, improved recognition techniques, and expanded functionalities for advanced mathematical operations.

5.Improves Learning and Teaching: Useful for students and educators in solving complex equations interactively, making mathematical problem-solving more engaging.

Disadvantages

1.Recognition Errors: Variations in handwriting styles may lead to incorrect interpretations, especially for complex symbols and ambiguous characters.

2.Dependency on Internet and Hardware: Running AI models on the cloud or locally requires sufficient computational resources, which may limit performance on low-end devices.

3.Limited Gesture-Based Control: While finger tracking allows equation input, it may not be as precise as writing with a stylus or using a dedicated handwriting recognition tool.

4.Complex to Use Initially: Users unfamiliar with gesture-based input may require time to adapt.

5.Lighting Issues During Camera Capture: Poor lighting conditions can reduce recognition accuracy, leading to incorrect equation extraction.

IV. RESULT & DISCUSSION

A. Comparison across Models

Fig.2. Comparison of Precision, Recall and F1 Scores of Hand Detection Models

- **Haar Cascades:** Lower scores across all metrics, indicating moderate accuracy and reliability.

- **YOLO:** Improved precision and recall, providing better overall performance.

- **MediaPipe Hands:** Achieves the highest scores, with precision at 97%, recall at 96%, and an F1 score of 96.5%, making it the most accurate and reliable model.

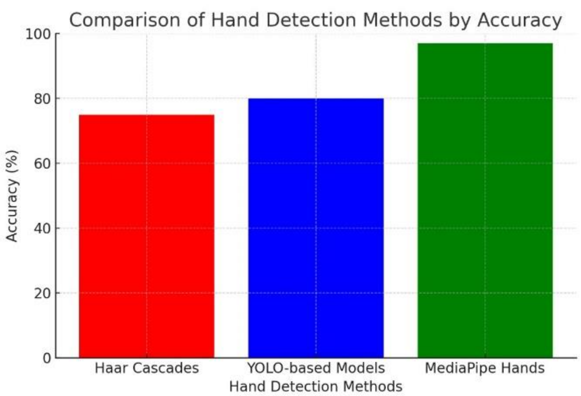
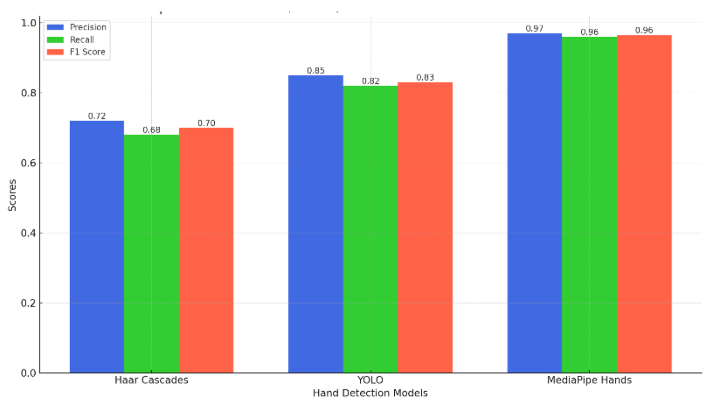


Fig.3. Comparison of Hand Detection Methods by Accuracy

The chart compares the accuracy of three hand detection methods. Haar Cascades achieved around 75%, YOLO-based Models reached 80%, while MediaPipe Hands led with 97% accuracy. MediaPipe was chosen for this



project due to its high precision and reliable real-time tracking, making it ideal for gesture-based math solving.

V. CONCLUSION

The Real-Time Math Solver Using Hand Gestures with AI successfully combines computer vision and artificial intelligence to recognize and solve handwritten mathematical equations. By utilizing MediaPipe Hands**,** the system achieves 97% accuracy in real-time hand tracking, significantly outperforming traditional methods like Haar Cascades (75%) and YOLO-based models (80%)**.** The integration of Google Gemini AI ensures precise and efficient mathematical solving, while the Streamlit-based interface provides a user-friendly and interactive experience. The system offers real-time performance**,** making it suitable for educational and practical applications. It demonstrates how AI-powered mathematical recognition can enhance learning by providing instant solutions. Future improvements could include advanced preprocessing techniques for better accuracy and expanding support for complex equations. This project showcases the potential of AI and gesture recognition in creating innovative, real-time problem-solving tools.

VI. REFERENCES

[1] Rekha Kotwal, Shabina Modi, Yogesh Mali Vishal Kisan Borate, Prajakta Khairnar & Apashabi Pathan (2024). “Hand Gesture Recognition and Real-Time Voice Translation for the Deaf and Dumb”. In Semantic Intelligence (pp.435-449). DOI: 10.1007/978-981-97-7356-5_35

[2] Rekha Kotwal, Vyankatesh Jadhav, Sakshi Jagtap, Rashi Magar, Viraj Pandit. “Real-Time Math Solver Using Hand Gestures with Al” Volume 13 Issue II February 2025. DOI : https://doi.org/10.22214/ijraset.2025.66617

[3] 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.

[4] 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.

[5] 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.

[6] Awasthi, A., Pranveer Singh, P., & Chaturvedi, A. (2024). "Mathematics and logics in ML: Application aspect." International Journal of Computational Mathematics, 22(1), 1 15.

[7] Zhang, W., Liu, S., & Wu, H. (2020). "Real-time hand gesture recognition using deep learning." IEEE Transactions on Image Processing, 10.1109/TIP.2020.2970142 29(7), 3726-3739. DOI:

[8] 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

[9] 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

[10] 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.

[11] 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.

[12] 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.

[13] 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

[14] 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

[15] 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

[16] Mohamed, A. S. (2024). "Real-Time Hand Gesture Recognition: A Comprehensive Review of Techniques, Applications, and Challenges." Cybernetics and Information Technologies, September 2024.

[17] 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.

[18] Anonymous. (2024). “AIRMATH: Gestures Powered Problem Solving.” International Research Journal of Modernization in Engineering Technology and Science, December 2024.

[19] Yusuf, O., Habib, M., & Moustafa, M. (2024). Real-Time Hand Gesture Recognition: Integrating Skeleton-Based Data Fusion and Multi-Stream CNN. arXiv, June 2024.

[20] Anonymous. (2024). AI-Based Real-Time Hand Gesture Controlled Virtual Mouse. Australian Journal of Electrical and Electronics Engineering, 2024.