



EDUDRAW: Problem Solver Using AI

Ayush Thakur¹, Param², Rajat Pahal³, Revant Rana⁴, Mr. Harkesh Kumar⁵

Department of CSE-AI&DS,

Panipat Institute of Engineering & Technology, Samalkha, India^{1,2,3,4,5}

⁵Mail ID: harkesh.cse.ai-ds@piet.co.in

Abstract— EDUDRAW is an AI-powered educational problem solver designed to provide an interactive and natural method for solving mathematical problems. The system combines computer vision, hand-gesture recognition, virtual drawing, and generative artificial intelligence to allow users to enter mathematical expressions through hand gestures instead of conventional keyboards, mice, or touchscreens. A webcam captures the user's hand movements, while MediaPipe detects hand landmarks and identifies gesture-based operations. The fingertip movement is converted into virtual ink on a digital canvas using computer vision and numerical processing techniques. Once a mathematical expression is drawn, the processed image is provided to a generative AI model for interpretation, calculation, and step-by-step explanation. The project integrates technologies including Python, OpenCV, MediaPipe, Google Gemini AI, NumPy, FastAPI, LangChain, Torch, React, TypeScript, and Vite. The system was evaluated using arithmetic expressions, algebraic and calculus expressions, and diagram-based geometry problems. The reported tests successfully recognized mathematical components and generated solutions for the tested cases. The project also identifies practical challenges involving gesture variability, lighting, camera quality, complex mathematical symbols, network dependency, privacy, user fatigue, and device scalability. EDUDRAW demonstrates how artificial intelligence and gesture-based human-computer interaction can be combined to create an accessible and interactive educational assistant.

Keywords: Artificial intelligence, computer vision, gesture recognition, mathematical problem solving, MediaPipe, Gemini AI, human-computer interaction, educational technology.

INTRODUCTION

Human-computer interaction has undergone significant development over the last several decades. Conventional computing systems have primarily depended on keyboards, mice, and touchscreens. Modern systems, however, increasingly explore natural and immersive methods of interaction, including gesture recognition, voice interfaces, augmented reality, and computer vision. Gesture recognition provides an opportunity to bridge the gap between physical human movement and digital interaction. It has been applied in gaming, robotics, virtual reality, augmented reality, and other interactive systems. However, its application to everyday educational tools such as calculators and mathematical problem solvers remains relatively limited. The EDUDRAW project addresses this opportunity by allowing users to perform mathematical operations through hand gestures and air-drawn expressions.



Mathematics is an important component of education and everyday problem solving. Conventional calculators require users to enter mathematical expressions using physical buttons or digital keyboards. Although such systems are efficient for numerical computation, they generally do not provide a natural visual method of expressing a problem or explaining the reasoning behind the answer.

The project therefore proposes a system in which a user can draw a mathematical expression in the air using a finger. A webcam captures the hand movement, computer vision techniques identify the user's fingertip position, and the movement is converted into a virtual drawing. The resulting image is then interpreted by an AI model, which identifies the mathematical expression, solves the problem, and provides an explanation.

Another motivation for EDUDRAW is contactless interaction. Touch-based systems may be inconvenient in shared environments such as classrooms, libraries, laboratories, and other public spaces. Gesture-based interaction reduces the requirement for physical contact with an input device.

The project also has potential accessibility benefits. Users who experience difficulty with conventional input devices may benefit from a gesture-based interface. The report therefore positions EDUDRAW not only as a mathematical calculator but also as an interactive and inclusive educational technology.

The system combines computer vision and generative AI. MediaPipe is used for hand tracking and landmark detection, while OpenCV handles video and image processing. Google Gemini AI [5] provides multimodal interpretation of mathematical input. The interface and backend components provide communication between the user, computer vision system, and AI model. The project demonstrates how these technologies can be integrated into a single educational application.

RELATED WORK

A. Gesture Recognition

Gesture recognition is a computer vision technique through which human movements are interpreted by computers. Modern camera-based systems have reduced the need for specialized hardware by allowing standard RGB cameras to detect hand positions.

MediaPipe provides real-time hand tracking and identifies 21 three-dimensional hand landmarks from RGB images. In EDUDRAW, these landmarks are used to determine finger positions and recognize predefined gestures.

The project uses gesture recognition as a natural user interface. Instead of pressing buttons, the user performs predefined gestures to draw, erase, reset, or submit a mathematical expression for AI analysis.

B. Computer Vision

Computer vision enables computers to process and interpret images and video. In EDUDRAW, it is responsible for capturing webcam frames, processing the images, tracking hand movements, and generating the virtual drawing canvas.

OpenCV is used for webcam capture, frame processing, image transformations, thresholding, resizing, and drawing operations. The system combines the processed hand information with a virtual canvas to reproduce the user's movement as digital writing.

C. Artificial Intelligence and Machine Learning



Artificial intelligence enables systems to perform tasks involving reasoning, problem solving, recognition, and decision making. EDUDRAW uses AI in two major areas: interpreting mathematical expressions and generating solutions.

Google Gemini AI is used as a multimodal generative model. The project sends the processed mathematical drawing to the AI model, which identifies the expression and generates a solution with detailed explanation. Modern generative AI systems, including large multimodal models, are commonly built upon transformer-based architectures [9] and have been demonstrated at scale in models such as GPT-3 [10] and GPT-4 [14].

D. Human-Computer Interaction

Human-computer interaction focuses on designing computer systems that are usable, accessible, and responsive to human behavior. EDUDRAW uses hand gestures as a natural user interface, replacing conventional keyboard and touchscreen input.

The touchless interface can provide advantages in environments where physical contact is undesirable. Real-time visual feedback also allows users to observe their interaction while drawing and submitting mathematical problems.

E. Educational Technology

Educational technology has increasingly incorporated artificial intelligence, interactive interfaces, adaptive learning, and virtual tutoring [13]. EDUDRAW applies these concepts specifically to mathematical problem solving.

The project can potentially be used in smart classrooms, online learning environments, home-based learning, and educational applications where students require immediate assistance with mathematical problems.

PROBLEM FORMULATION AND PROPOSED SOLUTION

A. Problem Statement

Traditional computing systems primarily depend on keyboards, mice, and touchscreens. These input methods may not always provide an intuitive or accessible experience. Traditional calculators also generally focus on obtaining a numerical result rather than explaining the steps required to reach the result.

The specific problem addressed by EDUDRAW is the absence of an intelligent, real-time, gesture-based mathematical problem solver that can: 1) track hand gestures using computer vision; 2) allow users to draw mathematical expressions in real time; 3) convert the drawings into interpretable visual data; 4) use generative AI to solve and explain the mathematical problem; and 5) provide a touch-free and interactive educational experience.

The report identifies the lack of integrated systems that combine gesture-driven input, mathematical interpretation, AI problem solving, and educational explanation.

B. Proposed Solution

EDUDRAW addresses the problem through a web-based problem-solving system. The user interacts with the application through a webcam and uses hand gestures to create mathematical expressions.



The proposed workflow consists of the following stages: 1) webcam video capture; 2) hand landmark detection; 3) gesture recognition; 4) virtual canvas drawing; 5) image preprocessing; 6) AI-based mathematical interpretation; 7) solution generation; and 8) display of the result and explanation.

The approach removes dependence on conventional physical input devices and provides a more visual and interactive method for mathematical problem solving.

SYSTEM ARCHITECTURE AND METHODOLOGY

A. System Architecture

The system is organized into multiple logical components: video input module, hand tracking module, gesture recognition module, virtual drawing canvas, image preprocessing module, AI communication module, mathematical solution module, and user interface.

The modular architecture allows individual components to be modified or improved without requiring major changes to the complete application. Fig. 1 illustrates the overall workflow of the system, showing the sequence of modules from video capture to solution display.

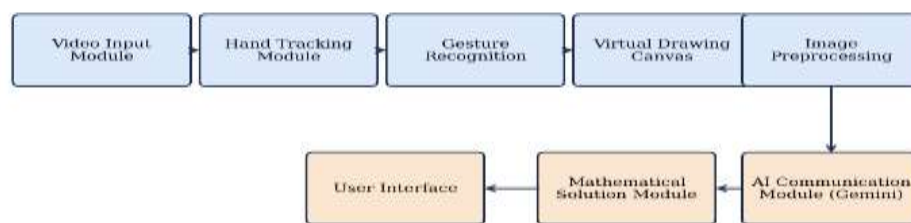


Fig. 1. Overall workflow of the EDUDRAW system.

The development methodology follows planning, modular design, incremental development, real-time gesture tracking, AI integration, testing, debugging, documentation, and deployment.

B. Video Processing

The webcam captures the user's hand movement continuously. Each frame is horizontally flipped to provide a mirror-like view and is converted from BGR to RGB format for compatibility with MediaPipe.

MediaPipe Hands identifies 21 hand landmarks in each frame. These landmarks include fingertips and knuckle positions. The system continuously tracks these landmarks to determine the current gesture.

C. Gesture Recognition

Gesture recognition is based on the combination of raised fingers. Different gestures correspond to different application modes, including drawing, erasing, resetting the canvas, and triggering AI analysis.

When drawing mode is activated, the coordinates of the index fingertip are recorded across successive frames. These coordinates are used to create lines on the virtual canvas.

D. Virtual Canvas



The virtual canvas is implemented using NumPy. The fingertip coordinates obtained from MediaPipe are mapped to the canvas and rendered as virtual ink. Rendering of the drawn strokes in the web interface follows conventions similar to browser-based drawing surfaces such as the HTML Canvas API [12] and vector graphics standards such as SVG [11].

This allows users to create mathematical expressions in the air. The virtual drawing behaves similarly to writing on a conventional digital board.

E. Image Preprocessing

After the mathematical expression has been completed, an AI trigger gesture initiates processing. The canvas image undergoes several preprocessing operations: 1) conversion to grayscale; 2) binary thresholding; 3) color inversion; 4) image resizing; and 5) conversion into a format suitable for AI interpretation.

These preprocessing operations improve the visual representation of the handwritten expression before it is supplied to the AI model. Fig. 2 illustrates the complete data flow from webcam input to the AI-generated mathematical solution.

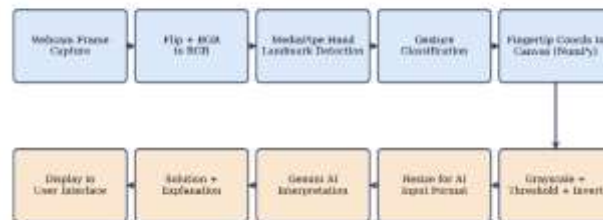


Fig. 2. Data flow from webcam input to AI-generated mathematical solution.

F. AI Interpretation

The processed image is provided to Google Gemini AI through an API. A prompt instructs the model to identify the mathematical expression, solve it, and provide a detailed explanation.

The AI response is captured and presented to the user through the application interface.

TECHNOLOGY STACK

The project uses several technologies to implement real-time gesture recognition, image processing, AI integration, and interactive user interfaces. Table I lists the technology stack used in EDUDRAW.

TABLE I
TECHNOLOGY STACK USED IN EDUDRAW

Component	Technology	Purpose
Programming	Python	Main development language
Computer Vision	OpenCV	Webcam and image processing



Hand Tracking	MediaPipe	Hand landmark detection
AI	Google Gemini AI [5]	Mathematical interpretation and solving
Numerical Processing	NumPy	Virtual canvas processing
Backend	FastAPI [1]	API and backend services
AI Workflow	LangChain [6]	AI application workflow
ML	Torch [8]	Machine-learning support
Frontend	React [2]	User interface
Frontend Language	TypeScript [3]	Typed frontend development
Build Tool	Vite [4]	Frontend development and build

The project report specifically describes Python as the primary programming language because of its machine-learning, computer-vision, and web-development ecosystem. OpenCV handles webcam video and image operations, MediaPipe provides 21 hand landmarks, Gemini performs multimodal interpretation, and NumPy supports the virtual canvas. The backend uses FastAPI [1] to expose interpretation and solution-generation endpoints, with LangChain [6] coordinating the AI workflow. The frontend is implemented using React [2] and TypeScript [3], built and served with Vite [4]. Model experimentation additionally draws on machine-learning frameworks such as PyTorch [8] (Torch) and, more broadly in the field, TensorFlow [7].

VI. IMPLEMENTATION

A. Hardware Requirements

The reported development system contains the hardware summarized in Table II.

TABLE II: HARDWARE SPECIFICATIONS

Component	Specification
Processor	AMD Ryzen 7 6800H
Clock Speed	3.20 GHz
RAM	16 GB
GPU	NVIDIA GeForce RTX 3050
GPU Memory	4096 MB GDDR6
CUDA Cores	2048
Storage	1 TB SSD



The GPU is used to accelerate computationally intensive machine-learning operations, while the SSD provides storage for datasets, models, and application resources.

B. Software Implementation

The project uses Python for core processing and combines OpenCV, MediaPipe, Gemini AI, NumPy, and web technologies.

The interface provides the webcam feed, virtual drawing canvas, gesture interaction, and AI-generated output. The project report describes both Streamlit-based interface components and a React/TypeScript/Vite interface in its implementation and results sections.

C. Development Process

The project was developed incrementally. Initially, webcam capture and hand landmark detection were implemented. Basic finger-up/finger-down logic was then used for gesture detection. Drawing functionality was added through fingertip tracking, followed by erase and reset functionality.

After the gesture system became functional, Gemini AI was integrated. The canvas image was preprocessed and supplied to the model together with instructions for mathematical interpretation and explanation. Community references and technical discussions [15] were consulted while integrating the frontend, backend, and AI components.

Each module was tested separately before integration. The final system was then tested under different users, lighting conditions, camera resolutions, mathematical inputs, and possible edge cases.

VII. RESULTS AND DISCUSSION

The EDUDRAW system was implemented and tested as an AI-powered educational assistant. The application accepts handwritten mathematical expressions, diagrams, and symbolic queries.

The report states that the system successfully identifies important mathematical elements including numbers, operators, variables, and geometric figures. It also reports successful calculation of mathematical properties such as area and perimeter.

A. Mathematical Expression Testing

The system was evaluated using multiple types of mathematical problems, summarized in Table III.

TABLE III: REPORTED TEST CASES

Test Case	Input Type	Reported Result
1	Arithmetic expression	Expression parsed and solution generated
2	Algebra/calculus expression	Expression interpreted and solution generated
3	Geometry diagram	Shape and parameters interpreted and calculation generated

The tested geometry examples included circles, squares, and triangles. Parameters such as radius, side lengths, and area labels were identified and used in calculations. Figs. 3 and 4 show an illustrative arithmetic input and its generated output, respectively. Figs. 5 and 6 show an illustrative algebra/calculus input and output. Figs. 7 and 8 show an illustrative geometry diagram input and its computed output.

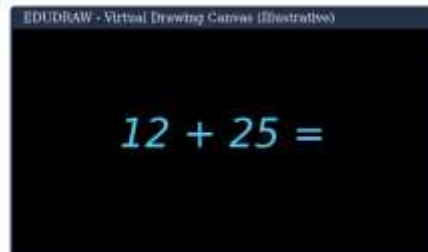


Fig. 3. Example Input 1 — mathematical expression.

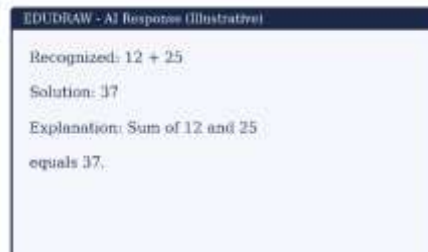


Fig. 4. Example Output 1 — generated mathematical solution.

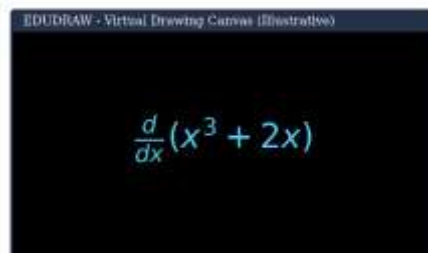


Fig. 5. Example Input 2 — mathematical/calculus expression.

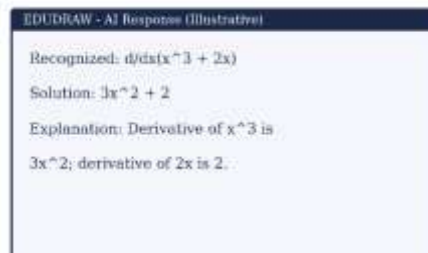


Fig. 6. Example Output 2 — generated solution.



Fig. 7. Example Input 3 — geometry diagram.

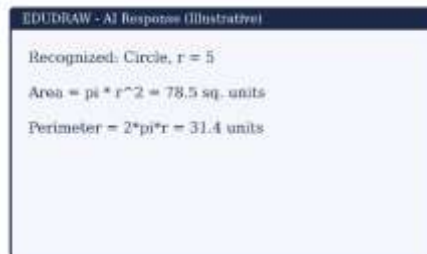


Fig. 8. Example Output 3 — computed geometry result.

B. Educational Benefits

EDUDRAW provides several potential educational benefits.

First, it provides immediate feedback. Students can draw a mathematical expression and receive an AI-generated response.

Second, it supports visual and experiential learning. Instead of entering a problem through a keyboard, the user interacts with the mathematical expression directly.

Third, the AI-generated explanation can support self-paced learning. The project report describes EDUDRAW as functioning similarly to a virtual tutor by providing immediate feedback.

Fourth, the modular architecture allows future expansion to subjects including physics, engineering, and data science.

C. Discussion of Results

The reported tests indicate that the system can successfully connect hand-based interaction with AI-based mathematical reasoning. The main contribution is therefore not a new mathematical algorithm but an integrated interaction model combining computer vision and generative AI.

The qualitative testing demonstrates the feasibility of the proposed approach. However, the project report does not provide a formal numerical accuracy benchmark, standardized test dataset, or statistical latency analysis. Consequently, this paper does not claim a specific recognition percentage or processing-time value.

LIMITATIONS

Although EDUDRAW successfully demonstrates its core functionality, several practical limitations remain.



A. Gesture Variability

Different users may draw the same digit or mathematical symbol differently. Variations in stroke thickness, orientation, movement speed, and size may affect recognition.

B. Lighting and Background

Poor lighting and cluttered backgrounds can reduce the reliability of hand landmark detection. Low contrast and visual interference may cause tracking errors.

C. Camera Quality

Low-resolution or low-frame-rate cameras may introduce delays or incomplete gesture tracking.

D. Complex Mathematical Symbols

Basic arithmetic expressions can be easier to recognize than complex mathematical notation. Symbols such as square roots, integrals, fractions, and exponents may require more advanced recognition capabilities.

E. Internet Dependence

Because the project uses a cloud-based generative AI service, an unstable internet connection may delay or prevent AI responses.

F. User Fatigue

Continuous air-drawing requires the user to maintain hand and arm positions. Long expressions may therefore cause fatigue and unstable input.

G. Security and Privacy

The system captures live webcam data and sends processed visual information to cloud-based AI services. Appropriate data-handling and privacy policies would therefore be important for deployment in schools and other sensitive environments.

H. Limited Gesture Vocabulary

The system relies on a limited set of gestures. Incorrect gesture interpretation may unintentionally activate drawing, erasing, or resetting operations.

I. Device Scalability

Performance may differ between devices depending on processing power, operating system, camera capability, and concurrent applications.

FUTURE SCOPE

The project report identifies several directions for future development.

A. Advanced Diagram Recognition

Future versions could recognize more complex diagrams from physics, chemistry, and engineering, including circuit diagrams, graphs, and molecular structures.

B. Voice and Gesture Integration

Voice commands could be combined with hand gestures to provide a more flexible interaction system.

C. Multilingual Support

Adding multiple languages could make the system accessible to students from diverse linguistic backgrounds.

D. Mobile Application



Android and iOS versions could extend accessibility to mobile users and students in remote or under-resourced areas.

E. Learning Management System Integration

EDUDRAW could be integrated with platforms such as Moodle, Google Classroom, or Microsoft Teams.

F. Progress Tracking

User-performance tracking and learning analytics could transform the system into a more comprehensive intelligent tutoring platform.

G. Technological Advancements

The report also identifies AI-powered vision systems, OCR, natural-language processing, real-time inference, modular architecture, and edge/cloud compatibility as important future directions.

CONCLUSION

EDUDRAW demonstrates how artificial intelligence, computer vision, and gesture-based human-computer interaction can be integrated to create an interactive mathematical problem solver.

The system enables students to solve mathematical problems through natural hand-drawn input rather than conventional text-based methods. It supports both equation solving and diagram-based problem solving, allowing users to draw shapes such as circles and rectangles and annotate them with relevant parameters.

The project combines multiple technologies, including Python, OpenCV, MediaPipe, Gemini AI, NumPy, FastAPI, LangChain, Torch, React, TypeScript, and Vite. The modular architecture makes it possible to extend the system with additional educational capabilities.

The reported test cases demonstrate successful interpretation of arithmetic, algebraic/calculus, and geometry inputs. The system also provides immediate AI-generated responses, supporting visual and experiential learning.

At the same time, the project identifies important limitations involving gesture variability, environmental conditions, camera quality, complex symbols, internet connectivity, privacy, user fatigue, and scalability. These limitations provide clear directions for further research and engineering.

Overall, EDUDRAW provides a foundation for future AI-enabled educational technologies. Its combination of touchless interaction and generative AI demonstrates how natural interfaces can make educational applications more interactive, accessible, and engaging.

REFERENCES

- [1] FastAPI Documentation, “FastAPI: The modern web framework for building APIs with Python 3.6+ based on standard Python type hints,” [Online]. Available: <https://fastapi.tiangolo.com>
- [2] React Official Documentation, “React — A JavaScript library for building user interfaces,” [Online]. Available: <https://react.dev>
- [3] TypeScript Documentation, “TypeScript: JavaScript with syntax for types,” [Online]. Available: <https://www.typescriptlang.org/docs>



-
- [4] Vite Documentation, “Vite — Next Generation Frontend Tooling,” [Online]. Available: <https://vitejs.dev/>
- [5] Google AI, “Gemini API: Multimodal large language model APIs,” [Online]. Available: <https://ai.google.dev>
- [6] LangChain Documentation, “LangChain: Building applications with language models,” [Online]. Available: <https://www.langchain.com>
- [7] M. Abadi et al., “TensorFlow: A System for Large-Scale Machine Learning,” in Proc. 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), 2016, pp. 265–283.
- [8] PyTorch Official Documentation, “PyTorch: An open-source machine learning library based on the Torch library,” [Online]. Available: <https://pytorch.org>
- [9] A. Vaswani et al., “Attention Is All You Need,” in Advances in Neural Information Processing Systems, vol. 30, 2017. [Online]. Available: <https://arxiv.org/abs/1706.03762>
- [10] T. Brown et al., “Language Models are Few-Shot Learners,” in Advances in Neural Information Processing Systems, vol. 33, 2020. [Online]. Available: <https://arxiv.org/abs/2005.14165>
- [11] World Wide Web Consortium, “Scalable Vector Graphics (SVG) 2,” 2023. [Online]. Available: <https://www.w3.org/TR/SVG2/>
- [12] MDN Web Docs, “Canvas API,” 2023. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API
- [13] A. Sharma and S. Kumar, “AI in Education: Role, Challenges, and Future Scope,” International Journal of Computer Applications, vol. 183, no. 45, 2021.
- [14] OpenAI, “GPT-4 Technical Report,” 2023. [Online]. Available: <https://openai.com/research/gpt-4>
- [15] Stack Overflow Discussions, “Various topics on integrating frontend and backend for AI applications, OCR, and interactive learning tools,” 2022–2024. [Online]. Available: <https://stackoverflow.com>
-