**GesturaMath: AI-Powered Hand Gesture Math Solver**

|  |
| --- |
| ***A Project Synopsis Submitted*** |
| ***In Partial Fulfilment*** |
| ***for award of Bachelor of Technology*** |
|  |
| **In** |
| **COMPUTER SCIENCE & ENGINEERING**  **(ARTIFICIAL INTELLIGENCE)** |
|  |
| **By** |
|  |
| **VAASU SRIVASTAVA (Roll No. 2201331520200)**  **VIBHAW KUMAR VERMA (Roll No. 2201331520208)**  **SANJAY PRAMANIK (Roll No. 2301331529010)** |
|  |
|  |
| **Under the Supervision of**  **Prof. RUCHIKA SHARMA**  **Assistant Professor, CSE(AI)** |
| Home |
|  |
| **Department of CSE (Artificial Intelligence)**  **School of Computer Science in Emerging Technologies**  **NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY, GREATER NOIDA**  **(An Autonomous Institute)**  **Affiliated to**  **DR. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY, LUCKNOW**  **May, 2025** |
|  |

1. **Title of the Project**

**GesturaMath: AI-Powered Hand Gesture Math Solver**

1. **Introduction / Background**

GesturaMath is an innovative application that combines computer vision and Artificial Intelligence to enable users to solve mathematical problems using hand gestures. The project addresses the need for more intuitive and interactive ways to interact with technology, particularly in educational and problem-solving contexts. Traditional methods of inputting mathematical expressions, such as typing or writing, can be cumbersome and inefficient, especially in dynamic learning environments or for users with specific accessibility needs. This project leverages the advancements in computer vision and AI, specifically deep learning, to provide a hands-free, natural, and efficient method for inputting and solving mathematical expressions. The system aims to bridge the gap between human intuition and machine computation by allowing users to express mathematical problems in a way that is both natural and familiar.

1. **Problem Statement**

The project aims to solve the problem of limited user interaction methods for mathematical problem solving. Traditional methods often involve typing or writing, which can be cumbersome and inefficient. This project explores a more natural and intuitive approach using hand gestures, but faces challenges including:

* Accurate recognition of complex mathematical symbols and notations through hand gestures.
* Robustness to variations in hand shapes, sizes, and writing styles.
* Real-time processing of gestures for a seamless user experience.
* Integration of a powerful AI model to accurately interpret and solve a wide range of mathematical problems.
* Designing an intuitive and user-friendly interface.

1. **Objectives**

* Develop a system that accurately recognizes hand gestures for drawing mathematical expressions, including complex symbols and notations.
* Integrate AI, specifically Google's Gemini 1.5 Flash, to process and solve the drawn mathematical problems with high accuracy and efficiency.
* Create a user-friendly interface that provides real-time feedback on gesture recognition and solution generation.
* Ensure the system's robustness to variations in hand shapes, sizes, and drawing styles.
* Evaluate the system's performance in terms of accuracy, speed, and usability.

1. **Scope of the Project**

The project scope includes:

* Real-time hand gesture recognition for drawing mathematical expressions, including numbers, operators (+, -, \*, /), and common mathematical symbols (e.g., square root, fractions).
* Integration with Google's Gemini AI for solving a range of mathematical problems, from basic arithmetic to more advanced algebra.
* Displaying the drawn expression and the AI-generated solution in a clear and understandable format.
* Implementation of drawing and erasing functionalities using hand gestures for input correction.
* Developing a user-friendly interface with visual feedback.
* Testing and evaluating the system's performance.

1. **Literature Review**

The literature review will cover existing research and tools in several key areas:

* Hand Gesture Recognition: Examination of various techniques for hand detection, tracking, and gesture classification, including classical computer vision methods and deep learning-based approaches.
* Mathematical Expression Recognition: Exploration of methods for recognizing and interpreting handwritten or drawn mathematical expressions, including optical character recognition (OCR) and specialized algorithms.
* AI for Mathematical Problem Solving: Review of the capabilities of large language models (LLMs) and other AI techniques in solving mathematical problems, with a focus on models like Google's Gemini.
* Human-Computer Interaction (HCI): Analysis of principles and best practices for designing intuitive and user-friendly interfaces for gesture-based interaction.

The review will highlight the novelty of combining these technologies for an intuitive math input and solving system, and identify any research gaps.

1. **Proposed System / Methodology**

The GesturaMath system will use the following methodology:

1. **Hand Gesture Recognition:**

* Utilize OpenCV to capture video frames from a webcam.
* Employ the cvzone library's HandTrackingModule for real-time hand detection and tracking.
* Implement algorithms to identify specific hand gestures for drawing, erasing, and submitting expressions. This will involve mapping hand landmark data to specific actions.
* Explore techniques to improve the robustness of gesture recognition under varying lighting conditions and hand orientations.

1. **Drawing and Erasing:**

* Develop functions to translate hand movements into drawn mathematical expressions on a virtual canvas.
* Implement an erasing function triggered by a specific hand gesture, allowing users to correct errors.
* Use computer graphics techniques to render the drawn expressions clearly and smoothly.

1. **AI Processing:**

* Capture the drawn expression from the virtual canvas as an image using the Pillow (PIL) library.
* Send the image to Google's Gemini 1.5 Flash API for processing.
* Structure the API request to include a prompt that instructs Gemini to interpret and solve the mathematical problem.
* Handle the API response, extracting the solution provided by Gemini.

1. **Output and Display:**

* Display the original drawn expression on the user interface in real-time.
* Present the AI-generated solution clearly and understandably, using appropriate formatting.
* Design the user interface to provide feedback on gesture recognition, drawing, and solution generation.

Technologies/tools to be used: Python, OpenCV, cvzone, Google Generative AI (Gemini 1.5 Flash), Pillow (PIL).

1. **Expected Outcome**

The expected outcome is a functional application that allows users to:

* Draw mathematical expressions, including a range of symbols and notations, using hand gestures.
* Erase parts of the drawn expression for correction.
* Submit the expression to Google's Gemini AI.
* Receive the solution to the mathematical problem in real-time.
* Interact with a system that is intuitive, user-friendly, and robust to variations in hand gestures.
* Demonstrate the potential of combining computer vision and AI for educational and problem-solving applications.

1. **Project Plan / Timeline**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Phase** | **Duration (Weeks)** | **Start Date** | **End Date** | **Tasks** |
| 1. Project Setup | 1 | 2025-07-01 | 2025-07-07 | Define project scope, gather resources, set up development environment (Python, OpenCV, cvzone, Google Cloud SDK), initial literature review. |
| 2. Hand Gesture Recognition Development | 3 | 2025-07-08 | 2025-07-28 | Implement hand detection and tracking using OpenCV and cvzone, develop gesture recognition algorithms for drawing, erasing, and submitting, test and refine gesture recognition accuracy. |
| 3. AI Integration | 2 | 2025-07-29 | 2025-08-11 | Set up Google Cloud project and API access, integrate Gemini 1.5 Flash for solving mathematical expressions, handle API requests and responses, and test AI integration. |
| 4. UI Development | 2 | 2025-08-12 | 2025-08-25 | Design and develop the user interface, implement real-time display of drawn expressions and AI solutions, incorporate user feedback mechanisms, and ensure a user-friendly design. |
| 5. Testing and Documentation | 1 | 2025-08-26 | 2025-09-01 | System testing (unit, integration, and user testing), performance evaluation, and create project documentation. |
| 6. Finalization | 1 | 2025-09-02 | 2025-09-08 | Finalize project report, prepare for presentation, and address any remaining issues. |

1. **References**

<https://medium.com/@odil.tokhirov/how-i-built-a-hand-gesture-recognition-model-in-python-part-1-db378cf196e6>