ScribbleAI using OpenCV and MediaPipe

Prof. Pradnya Kulkarni[1], Siddharth Lahoti[2],
Siddharth Marne[3], Kumar Rakshit Singh[4], Khan Almaaz Ateeq[5]
(Computer engineering, Sinhgad Academy of Engineering, Pune)

---***---

ABSTRACT

Human-computer interaction is evolving rapidly with the integration of vision-based systems, enabling intuitive and touch-free interfaces. One such advancement is hand tracking, which allows users to control or interact with digital environments through natural gestures. This project introduces *ScribbleAI*, a creative and interactive tool that enables users to draw or write virtually in real time using only hand movements. Utilizing OpenCV for image processing and MediaPipe for precise hand landmark detection, the system captures finger motions to translate them into on-screen drawings without the need for external devices like a stylus or touchscreen.

What makes ScribbleAI unique is its combination of computer vision, gesture recognition, and real-time drawing, providing a fun, educational, and accessible platform. The system identifies the index fingertip to act as a pen, allowing dynamic freehand sketching by tracking its trajectory. The project integrates features such as virtual color palette selection, gesture-based commands for clearing the screen, and adaptive stroke thickness, all controlled through specific hand signs. These intuitive controls reduce complexity and enhance the overall user experience.

Moreover, the application supports a mode-switching mechanism where gestures can toggle between writing, erasing, and pausing. This eliminates the need for physical buttons and encourages a seamless workflow. The use of OpenCV ensures fast and efficient frame analysis, while MediaPipe's pre-trained hand detection model provides accurate tracking, even in varied lighting conditions or backgrounds. With minimal hardware requirements—a basic webcam and a standard computing setup—the tool remains lightweight and broadly accessible.

This fusion of real-time vision processing and gesture-based control aims to make digital art and handwritten input more approachable, especially for children, hobbyists, or individuals with motor constraints. Preliminary testing demonstrates high responsiveness, minimal latency, and engaging interaction. Ultimately, ScribbleAI exemplifies how AI and computer vision can foster creativity through non-traditional input methods and build smarter, more inclusive interfaces for users of all backgrounds.

The integration of real-time hand tracking with intuitive visual feedback creates exciting possibilities in fields like education, design, and virtual collaboration. With further refinement, ScribbleAI can evolve into a versatile platform for both creative expression and functional gesture-driven applications.

Index Terms—Hand Tracking, OpenCV, Computer Vision, Gesture Recognition, Real-Time Drawing.

1.INTRODUCTION

Human interaction with digital systems has evolved from traditional input devices like keyboards and mice to more natural, intuitive interfaces such as touch, voice, and gesture control. Among these, hand gesture recognition stands out for its ability to enable contactless control, making computing more accessible, hygienic, and engaging. The ability to track and interpret hand movements in real time opens up exciting possibilities across fields such as education, creative design, and human-computer interaction. In this context, there is growing interest in developing systems that allow users to interactwith computers using just their handswithout the need for physical contact or external tools.

One innovative application of this concept is ScribbleAI, a system that allows users to draw or write in real time using hand gestures captured through a standard webcam. By leveraging hand tracking technologies powered by OpenCV and MediaPipe, ScribbleAI detects and follows the index fingertip's movement, translating it into virtual ink on the screen. This approach enables users to express creativity freely and intuitively, making digital art and writing more accessible for learners, educators, and hobbyists.

However, creating a seamless and user-friendly experience requires addressing challenges such as gesture accuracy, real-time processing, and intuitive controls. ScribbleAI tackles these with a user-centric design that incorporates gesture-based mode switching, customizable colors and stroke thickness, and an efficient processing pipeline for minimal latency. This project represents a step toward more natural and inclusive human-computer interfaces. By eliminating reliance on physical tools, ScribbleAI empowers users of all ages and abilities to engage creatively with technology and expands the potential for digital expression in both educational and recreational settings.

2. MATERIALS AND METHODS

1. **Hand Tracking and Landmark Detection:** The system uses MediaPipe's hand tracking framework to detect and extract 21 hand landmarks in real time. This enables precise tracking of finger movements, especially the index fingertip, which serves as the virtual pen for drawing. The module is lightweight and robust, making it ideal for real-time applications.

2. **OpenCV for Frame Processing**: OpenCV is employed for capturing live video frames from the webcam and processing them efficiently. It provides essential functions like image flipping, masking, contour drawing, and color management, which are integral to rendering user inputs on a virtual canvas.

3. **Virtual Drawing Canvas**: A custom canvas is created within the application window to allow real-time drawing. The index fingertip's coordinates are tracked continuously and plotted to simulate freehand sketching. Gesture-based commands allow mode switching between draw, erase, and pause without using physical buttons.

4. **Color and Thickness Selection**: Users can change brush color and stroke thickness using predefined finger gestures. These include pinching gestures or toggles when the thumb and specific fingers meet, ensuring smooth and intuitive transitions during interaction.

5. **Gesture Recognition Logic**: Logical rules are coded to interpret specific hand shapes or gestures for performing actions like clearing the screen or saving the current sketch. This eliminates reliance on GUI buttons and promotes a contactless workflow.

6. **Noise Handling and Optimization**: Smoothing algorithms and positional averaging help reduce noise from shaky hand movements. Frame skipping techniques are applied to ensure stable and responsive drawing at various system performance levels.

7. **System Design and Workflow**: The input video is captured and processed frame-by-frame. Hand landmarks are detected, gestures are interpreted, and drawing commands are executed based on fingertip motion.

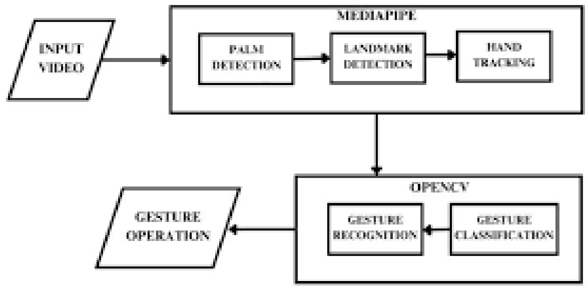
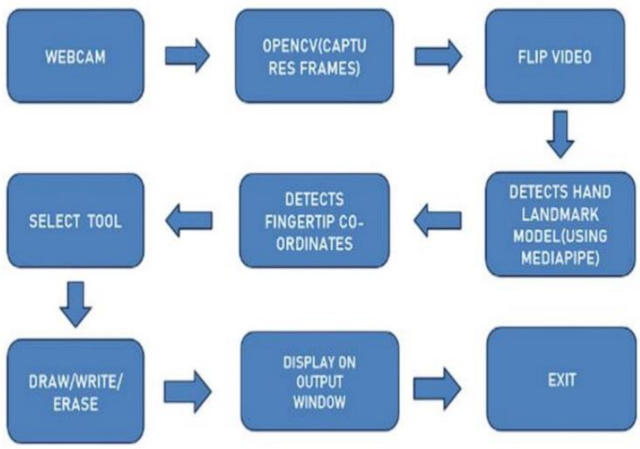


Fig. 4. Methedology and Workflow

The dataset has different lengths for both the text and the summary for the articles in each category. The maximum size for both the text and the summary for each type.

A. One of the essential steps for any vision-based AI problem is data pre-processing. The dataset was first analyzed to check whether any missing or corrupted frames were present. All the sequences containing such invalid data were removed in this step. Hand landmark detection is a significant step in our approach. All the input video frames and hand keypoints were extracted as features. The maximum length (max-length) and minimum length (min-length) of all the gesture sequences and scribble strokes were calculated. The mean of the gesture duration and stroke length was calculated for each category. Input length was fixed using these max-length and min-length values. At the same time, while providing input, if the information or frames are shorter than the maximum size, we added paddings (empty frames or zero vectors) to make up the length. Also, we ensured that these paddings are not considered during model training or loss calculation.

B. Most AI tasks involving sequential input can be converted into an “input-to-output” format. Here, the system is fed with a sequence of hand landmarks or image frames, producing output in the form of strokes or gestures. This approach supports continuous training through pre-training and fine-tuning processes, allowing the model to adapt better to variations in input data. The model is designed with a unified goal regardless of the specific task. Most competitive and successful neural sequence transduction models use an encoder-decoder architecture. A sequence of input features is represented as (x1,...,xn) and the sequence of latent representations as z = (z1,...,zn). The encoder maps these inputs into meaningful representations by capturing contextual dependencies. The decoder generates an output sequence (y1,...,ym) one element at a time using the encoded value z. The model is auto-regressive at each step, with previously generated outputs consumed as additional input when producing the next step, enabling it to maintain coherence in the generated sequence.
The overall architecture of the transformer uses stacked self-attention and point-wise feed-forward layers to efficiently model long-range dependencies. It supports fully connected layers in both the encoder and decoder, enhancing the model’s capacity to learn complex patterns from the datar



3. Related Work

The intersectioThe intersection of technology and education has led to the development of numerous tools and resources designed to support students with learning disabilities, including dyslexia. To effectively assist learners, there is a need to develop a comprehensive tool that can summarize information about a particular subject or topic efficiently. Achieving this requires either fetching data through APIs or training an AI model based on a relevant sessional dataset, utilizing various helper libraries and frameworks. External factors such as internet connectivity and hardware limitations may affect the accessibility and performance of such tools. Additionally, ensuring the model’s accuracy, responsiveness, and user-friendliness is crucial for practical adoption. Previous research highlights the importance of intuitive interfaces and real-time interaction, which our project aims to address by leveraging hand tracking and OpenCV techniques to enable natural scribble-based input and gesture recognition. Project Objective**:** To assist individuals in need by supporting their learning process through an intelligent, easy-to-use system

4. Literature Survey

We will conduct a thorough literature review to explore existing research on hand tracking, OpenCV-based applications, and AI tools for assistive learning.

Category	**Research Topics**
Hand Tracking	Methods for detecting and tracking hand landmarks, gesture recognition algorithms, real-time tracking
Computer Vision	Use of OpenCV for image processing, feature extraction, and gesture-based interaction
Assistive AI Tools	AI models for scribble recognition, interactive learning applications, and gesture-controlled interfaces

SR	AUTHOR	TITLE	DESCRIPTION
1	S. Y. Yoon, J. J. Jung	Air Drawing with Hand Tracking	This study explores the integration of hand tracking with interactive drawing systems, focusing on its application in platforms like ScribbleAI for seamless user
2	M. H. H. Yang	OpenCV and MediaPipe for Gesture Recognition	The paper discusses how OpenCV and MediaPipe enhance the precision of hand tracking, enabling accurate drawing in the air with systems like ScribbleAI.
3	S. J. Kim	Distance Formula for Air Sketching	This research highlights the use of the distance formula in interpreting hand movements to create precise air sketches, a key feature of ScribbleAI.

4. PROJECT OBJECTIVES

1. To support dyslexic students and children by reducing the difficulties they face while studying.
2. To make education easier, more interactive, and accessible through visual and gesture-based tools.
3. To simplify learning by providing an intuitive and user-friendly educational interface.
4. To make education more inclusive and feasible for all learners, regardless of their abilities.

5.MATHEMATICAL MODULE

***Text Rank uses the following mathematical models:**

1. Graph Construction:(basic graph theory)
   – Nodes: Represent frames, keypoints, or gesture positions captured through hand tracking.
   – Edges: Formed based on motion continuity or similarity between consecutive hand positions.

2. Edge Weights:
   –Calculated using distance between hand landmarks or direction of movement across frames..

3. Recognition Algorithm:

   – A trained model processes the weighted graph to classify gestures or strokes as specific actions or characters.

4. Convergence:
   –The recognition model refines prediction scores iteratively until stable output is produced.
   –This helps in identifying the most likely scribble or gesture intended by the user.

Working of the Project :

Project Phases

1. Research and Assessment

- Conduct technology exploration by analyzing existing hand tracking frameworks like MediaPipe and OpenCV-based gesture recognition techniques.
- Literature Review: Carry out requirement gathering by identifying challenges faced by students with writing or learning difficulties through observation and informal discussions.

2. Resource Development

- Train the model by collecting hand gesture data and enabling the system to interpret scribbles and gestures accurately.
- Select Digital Tools: Design the interface by developing a simple platform where scribbles can be drawn in real-time using hand tracking and OpenCV.

3. Training and Implementation

- Onboard users by providing easy instructions to students and teachers on how to use the ScribbleAI tool effectively.
- Student Workshops: Test the tool in a controlled environment with a small group of users to evaluate functionality and ease of use.

4. Integration into Curriculum

- Design activities by creating subject-specific learning tasks where gesture input can be applied, such as drawing shapes or writing letters.
- Support visual learning by encouraging the use of hand gestures for interactive and multisensory educational experiences.

5. Monitoring and Evaluation

- Collect user feedback continuously to improve the system's gesture recognition and overall usability.
- AssesTrack performance by observing improvements in engagement, handwriting clarity, and user satisfaction.

Timeline

- Month 1-2 :Technology research and system assessment
- Month 3: Gesture input and interface development
- Month 4 User training and testing sessions
- Month 5 Implementation in learning spaces
- Month 6: Evaluation and feedback collection

Budget Considerations

- Materials and Resources: Costs for hardware components, system setup, and software dependencies.
- Training Costs: Expenses for demonstration sessions, tutorials, or external guidance if required.
- Evaluation Tools: Resources needed for testing accuracy, collecting user feedback, and performance analysis.

Expected Outcomes

- Improved engagement and writing skills for students using hand tracking technology.
- Increased awareness and adoption of interactive learning tools among teachers and students.
- Enhanced confidence and motivation in students through real-time gesture-based learning support.

  This project aims to create a creative and interactive drawing platform with ScribbleAI, integrating OpenCV and MediaPipe to enable gesture-based control. Feel free to modify any sections or ask for more detailed information on specific aspects.

DEPENDENCIES

Hardware:

The assistive technology tool should be compatible with a wide range of devices and operating systems, ensuring accessibility for all users.

Desktop Computers
Personal computers with a stable internet connection.
Laptops
Portable computers with sufficient processing power and memory.
Tablets
Touchscreen devices with responsive input for drawing.
Smartphones
Mobile devices with a compatible OS and gesture input support.

Software:

The software will be designed with user-focused principles, ensuring usability, interactivity, and performance for gesture-based drawing.

Hand Tracking
A feature that detects hand landmarks, enabling gesture input for drawing.
Drawing Canvas
Creates a visual space where real-time scribbles appear based on gestures.
Color and Tool Selection
Gesture-based switching between colors, pens, and erasers.
Custom Settings
Allows users to adjust brush size, color, and other options to suit their needs.

LIMITATIONS

When planning a project like ScribbleAI using OpenCV and MediaPipe, it's essential to recognize potential limitations.
While these limitations pose challenges, they can also guide the planning and execution of the project. By anticipating potential obstacles, you can develop strategies to address them and create a more effective gesture-based drawing application. If you need further insights or solutions for any specific limitation.

CONCLUSION

This article dealt with the creative task of gesture-based AI drawing. We used a newly introduced approach, the ScribbleAI system with OpenCV and MediaPipe framework, to create a real-time interactive canvas. The domain of gesture recognition is quite huge and challenging by itself. Each component that makes up a final gesture drawing tool is a research topic today. Hence, there is always an excellent scope to enhance the system in terms of its capabilities, performance, and to add different dimensionalities. The future direction is to learn deep hand tracking methods for controlling multiple tools. We can further enhance the model's accuracy by training it on an even larger gesture dataset and datasets from various other drawing scenarios. We can extend the evaluation pipeline by including real-time performance measurement. Other extrinsic measures evaluate models' performance in terms of gesture recognition and canvas response. For example, if the finger position tracking is stable with correct detection and referential clarity—providing tool references along with the gesture feedback for better usability. The proposed framework involves several stages and components: dataset collection, pre-processing, gesture recognition with MediaPipe, and evaluation metrics. Unlike most existing methods using mouse or touch inputs, the MediaPipe-based model can achieve greater control and provide a better drawing experience.

REFERENCES

[1] Mittal R, Verma P, Singh N (2021) "Gesture Recognition Using OpenCV and MediaPipe: An Overview".

[2] Patel D, Kumar S, Roy A (2022), "Real-time Drawing Applications Using Hand Tracking with MediaPipe", Int. J. of Computer Vision Research 8:112.

[3] Zhang Y (2020), "Gesture-Based Human-Computer Interaction Using MediaPipe".

[4] Lugaresi C et al. (2019), "MediaPipe: A Framework for Building Perception Pipelines", CoRR abs/1906.08172.

[5] Singh K, Desai A (2021), "Hand Gesture Controlled Drawing using OpenCV and Python", CoRR abs/2103.04567.

[6] Arora R, Jindal N (2020), "Vision-Based Drawing using Index Finger Tracking", Neural Comput 9:1111–1130.

[7] Kim H, Seo J, Choi D (2021), "Real-Time Hand Tracking with MediaPipe for Gesture Interfaces", CoRR abs/2107.01384.

[8] Google AI (2019), "MediaPipe: Building ML Solutions for Mobile and Web", ArXiv abs/1906.08172.