International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 13 Issue: 1
Article Received: *25 November 2024* ***Revised:*** *12 December 2024* ***Accepted:*** *30 January 2025*

Smart Conferencing Rooms: A Comprehensive Approach to AI-Driven Gesture and Virtual Interaction

Rohan Yadav
Dr. Vishwanath Karad MIT World Peace University
Pune, Maharashtra, India
yadavrohan021@gmail.com

Shreya Verma
Dr. Vishwanath Karad MIT World Peace University
Pune, Maharashtra, India
verma.shreya.094@gmail.com

Megha Dalsaniya
Dr. Vishwanath Karad MIT World Peace University
Pune, Maharashtra, India
meghadalsaniya@gmail.com

Aryan Ghogare
Dr. Vishwanath Karad MIT World Peace University
Pune, Maharashtra, India
aryanghogare95@gmail.com

Dr. Dhanashree Wategaonkar
Dr. Vishwanath Karad MIT World Peace University
Pune, Maharashtra, India
dhanashri.wategaonkar@mitwpu.edu.in

Abstract— The “Smart Conferencing Rooms” project is aimed at changing the interaction of users in educational and healthcare facilities through the use of artificial intelligence in hand gesture recognition and virtual communication technologies. By embedding Artificial Intelligent Hand Gesture Recognition and Virtual Communication Technologies to enhance the user interaction in education and health care these are the objectives of the “Smart Conferencing Rooms”. The goal is to create new AI/ML models for hand detection, fingertip tracking and air writing allowing for such a realization of interactions as drawing on the midair, typing, etc. , or solving mathematical problems. Further, it aims at improving more real-time interaction through the virtual chat system with better face and gesture recognition than Google Meet but with more functions. This cross-platform solution is asserted to enhance the qualities of communications and enhance collaboration and connectivity, which will be highly beneficial to these sectors.

This work is inspired from progress made in the field of object tracking which is one of the core issues of computer vision and it entails the ability to recognize and find objects like hand gestures in successive frames. It is suitable for those applications such as automatic surveillance and video recognition. By analyzing gestures the system can translate them into text which will especially be so beneficial to the deaf group of people because it helps them communicate effectively using gestures.

Keywords- AI-Driven Gesture Recognition, Air Canvas, Cross-Platform Video Conferencing, Fingertip Tracking, Virtual Interaction

I. INTRODUCTION

The "Smart Conferencing Rooms" project aims to transform educational and healthcare interactions by leveraging AI-driven gesture recognition and virtual interaction tools. This initiative focuses on developing AI/ML models for hand and fingertip tracking, air writing, and real-time collaboration within a web application, ensuring cross-platform compatibility.

Background and motivation for the study was basically Advances in AI and machine learning present opportunities to enhance user experiences in virtual environments. Traditional conferencing tools lack intuitive gesture-based controls [1] and real-time collaboration features, limiting their effectiveness in educational and healthcare settings. This project seeks to bridge this gap by creating an interactive platform that makes virtual interactions more natural and efficient.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 13 Issue: 1
Article Received: *25 November 2024* ***Revised:*** *12 December 2024* ***Accepted:*** *30 January 2025*

Problem statement— Our project aims to develop a comprehensive system that revolutionizes virtual interactions [2] by implementing gesture recognition to detect and track hand movements, integrating AI-driven features like fingertip tracking and air writing into a web application, and enhancing real-time collaboration with face and fingertip detection in a chat room environment. The solution is designed to be cross-platform compatible, ensuring accessibility across various devices.

Scope and significance of the study is that the project focuses on developing and integrating gesture-based interactions within a web conferencing application, aiming to revolutionize virtual communication [3] in education and healthcare. Its success could lead to more intuitive, engaging, and accessible tools, significantly improving user experience in these sectors.

II. LITERATURE REVIEW

A. ***Lavania et al. (2022)*** present an AI-integrated real-time signal processing framework for air-writing, improving human-machine interaction (HMI) for individuals with disabilities. The system addresses challenges in gesture detection, particularly fingertip tracking, through computer vision and object tracking algorithms. [4]

B. ***Mehra et al. (2023)*** developed an Air Canvas using MediaPipe for hand tracking, allowing students to draw on a virtual canvas in underprivileged educational settings, addressing the lack of resources and enhancing HCI in educational applications. [5]

C. ***Chavali et al.*** introduced a gesture-based system for mouse operations using only a camera, achieving 82% accuracy. Their work builds on gesture recognition techniques and contributes to the development of contactless interfaces, especially relevant during the COVID-19 pandemic. [6]

D. ***Revathy et al. (2023)*** proposed an air-canvas system using OpenCV and Python for gesture recognition, improving focus and interaction during online learning. Their work emphasizes the importance of gesture-based interfaces in educational platforms. [7]

E. ***Lugaresi et al. (2019)*** introduced MediaPipe, a real-time perception pipeline framework. Its efficient streaming architecture and GPU support make it ideal for video and audio processing tasks, surpassing frameworks like TensorFlow in media analysis. [8]

F. ***Singh et al. (2017)*** combined chain codes and graph neural networks (GNNs) for handwriting recognition, outperforming traditional methods and offering a more efficient solution for handwriting trajectory processing and finger vein recognition. [9]

G. ***Köpüklü et al.*** discussed real-time hand gesture detection using convolutional neural networks (CNNs). Their hierarchical structure combining 3D CNNs for gesture detection and 2D CNNs for classification offers a balance between real-time performance and computational complexity. [10]

H. ***Karis et al.*** explored the role of video conferencing and portals in enhancing remote collaboration. Their study highlights the role of integrated tools like Google Docs in maintaining productivity and trust in remote working environments. [11]

Sr. No.	Name of Paper	Year Published	Observations
1.	Visual Interpretation of Hand Gestures - Human Computer Interaction: A Review [17]	1997	A vision-based system was developed that has RGB and descriptor functionality for hand gesture classification.
2.	Real-Time Fingertip Tracking and Gesture Recognition [16]	2002	It uses a filtering method, can track multiple fingertips reliably, and improves fingertip shadowing.
3.	Object Tracking: Survey. [13]	2006	LED inserted at the user's finger and webcam tracked it, returning an alphabet-like pattern drawn.
4.	Automatic Hand Pose Trajectory Tracking System Using Video Sequences [14]	2010	The video camera enabled fingerprint use; different hands performed different functions.
5.	Text Writing in Air [12]	2013	Hand shape was detected using a Kinect sensor, which works well to trace large objects.
6.	Hand Gesture Recognition In Real-Time For Automotive Interfaces [15]	2014	A vision-based system was created to classify hand motions using a combined

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 13 Issue: 1
Article Received: *25 November 2024* ***Revised:*** *12 December 2024* ***Accepted:*** *30 January 2025*

			RGB and depth descriptor.
7.	Facial Expression Recognition Based on a Hybrid Model Combining Deep and Shallow Features [18]	2019	A facial emotion detection system was developed using classic deep learning techniques, such as convolutional neural networks, to simplify the process.
8.	Static and Dynamic Human Arm/Hand Gesture Capturing and Recognition via Multi-Information Fusion of Flexible Strain Sensors [19]	2020	This paper addresses static and dynamic human arm/hand gesture capture and recognition.

TABLE 1: CHRONOLOGY OF GESTURE RECOGNITION APPROACHES

III. METHODOLOGY

This research, we are creating a Virtual AI Writer in our existing video conferencing application, users can join the meetings, log in and engage in a setting. The first interface enables them to draw on the screen through the hand detection feature, and it has options of drawing shapes (rectangle, line, etc.), switching color of the pen, eraser tool and the option of putting their work on a virtual black board. To do this, we have deployed computer vision models and protocols utilizing the powerful CVZone [20] and MediaPipe.

A. Description of AI models and algorithms used

Our research design focuses on utilizing hand gesture detection for an intuitive drawing experience. By recognizing fingertip gestures, we enable users to interact with the screen in real-time, turning physical gestures into virtual input. The AI models we employ are based on the **MediaPipe** framework, which facilitates accurate hand tracking [21], detecting keypoints such as fingertips and joints. These keypoints are then processed to define various actions, such as drawing with the index finger or using the thumb to move objects around. We also support a color-changing feature where specific gestures change the drawing tool's color to blue, yellow, and other colors. Additionally, there is an eraser functionality that allows for easy corrections.

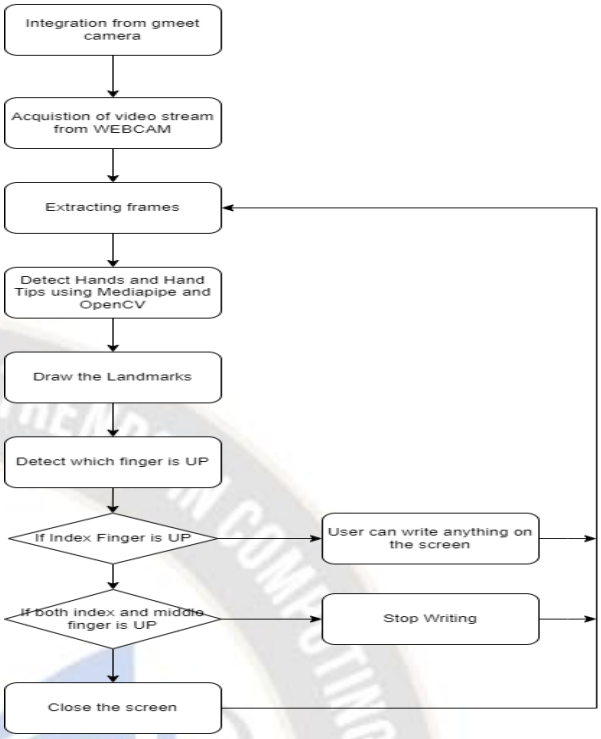

Fig 1: Flow Diagram of Gesture Recognition in Virtual AI Writer

B. Data collection methods

Our data collection process involved gathering hand gesture datasets and using pre-trained models for hand tracking and gesture recognition. MediaPipe's hand detection module, which we integrated into our system, already provides high accuracy for detecting finger positions, so we mainly focused on fine-tuning the system for drawing-related gestures. We tested the system with various user interactions to ensure the accuracy of fingertip detection and gesture responses in different lighting and user setups.

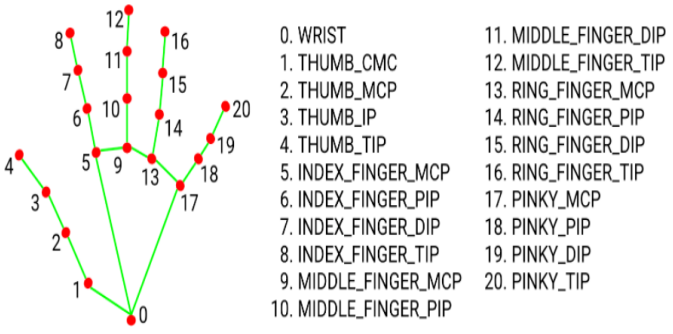

Fig 2: Hand Keypoint Mapping with Joint Labels

C. System architecture and components

The architecture of our system comprises both software and hardware components. On the software side, the system integrates **React** for the user interface, coupled with **OpenCV** for real-time video stream processing [22]. The CVZone library simplifies the interaction

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 13 Issue: 1
Article Received: *25 November 2024* ***Revised:*** *12 December 2024* ***Accepted:*** *30 January 2025*

between the OpenCV-based image processing pipeline and the machine learning models. On the hardware side, the system requires a standard webcam to capture hand movements, which are processed in real-time for hand gesture recognition and drawing on the virtual whiteboard.

D. *Implementation details and development environment*

In terms of implementation, we developed the project in a **JavaScript** and **Python**-based environment. The front-end of the system is built with **React.js**, while the backend integrates AI models for hand detection using **OpenCV** and **MediaPipe** in **Python**. To enable real-time drawing interactions, we use **WebRTC** [23] for video conferencing, ensuring that the drawing actions happen smoothly over video calls. The environment we used for development includes **Visual Studio Code** and **Node.js** for managing dependencies. We also rely on **CVZone** for its easy-to-use functions that help streamline the connection between hand gesture detection and drawing actions.

IV. SYSTEM DESIGN AND DEVELOPMENT

In creating our video conferencing system, our team made use of the Video SDK, which helped us simplify the process of building a dependable and scalable communication platform. The system's architecture is based on client-server communication, allowing users to join meetings by sending and receiving audio and video streams in real time [24]. Setting up the environment required configuring tokens for authentication, which are essential for starting meetings and ensuring that only authorized participants can access them.

Our development strategy also involved carefully installing dependencies to guarantee that the system would function seamlessly in both development and production settings. While a temporary token was adequate during development, we set up a dedicated authentication server for production, providing an additional layer of security. By taking advantage of the SDK's built-in features for media streaming and session management, we could concentrate more on customizations and enhancing user experience instead of rebuilding core components.

A. *Design principles and considerations*

As we developed the video conferencing system, we focused on several key principles that shaped our decisions, starting with the goal of ensuring seamless real-time communication. Each meeting session is treated
as a separate instance, allowing users to join and engage with different types of media streams, including video, audio, and screen sharing. The SDK provided essential components like Meeting, Participant, and Stream [25], which became the foundation of our system architecture. Security was a major concern throughout our design process, with token-based authentication being crucial for protecting our system. During development, we used

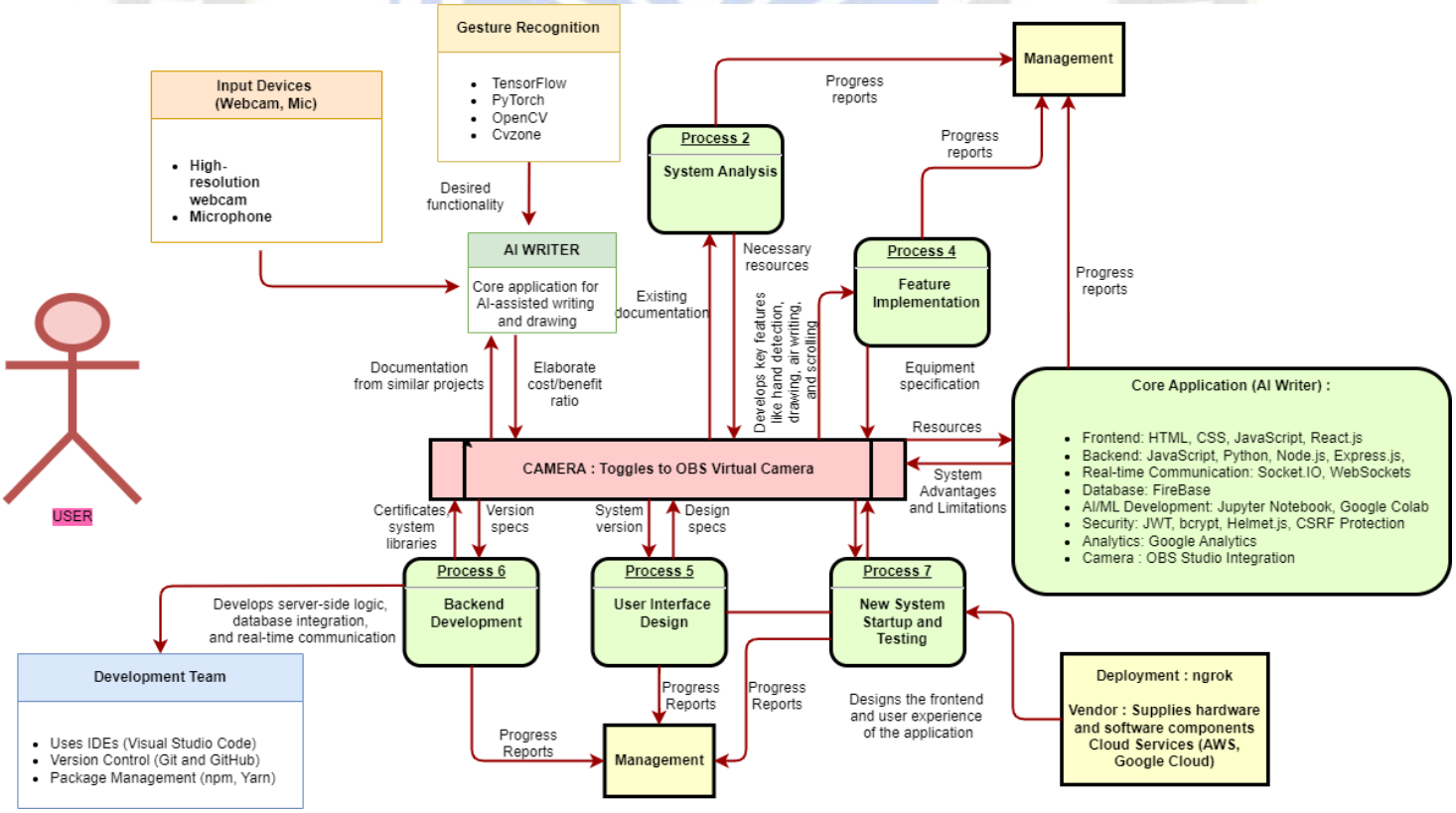

Fig. 3: System Architecture of Virtual AI Writer Project

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 13 Issue: 1
***Article Received:** 25 November 2024 **Revised:** 12 December 2024 **Accepted:** 30 January 2025*

temporary tokens for quick setup, but in the production phase, we will establish a strong authentication server to verify users. This guarantees that only authorized participants can access meetings. Performance was also a key focus—our system was crafted to reduce latency, optimize bandwidth usage, and deliver a consistent, high-quality user experience, no matter the number of participants or the type of media being shared.

B. Detailed description of the video conferencing room setup

For our video conferencing room setup, we focused on creating a user-friendly and functional space that allows participants to engage in real-time communication effortlessly. The layout features a dynamic grid that showcases each participant's video stream, ensuring everyone can see and interact with one another. This grid automatically adjusts to accommodate the number of attendees, providing an optimized view that remains responsive and intuitive throughout the session. Our system is also equipped to handle various types of media streams, including video, audio, and screen sharing. By using tokens for authentication and access control, we made sure that participants can join and leave sessions securely without causing interruptions. The seamless integration of these streams facilitates smooth interactions, whether for face-to-face discussions or sharing presentations during the meeting.

C. User interface and experience (UI/UX) design

The user interface and experience were crucial factors in our development process, as we aimed to create a platform that was both functional and user-friendly. The Video SDK [26] provided us with the flexibility to design a responsive UI, complete with a participant grid and interactive controls. Participants are displayed in a grid that automatically adjusts based on their number, ensuring a clear and easy-to-navigate layout. We included buttons and controls for actions such as muting, unmuting, screen sharing, and managing participants, all conveniently positioned for easy access.

We also emphasized real-time feedback to improve the overall experience. For instance, participants can see who is speaking through visual indicators, and they receive instant updates on their connection quality and device status. Our UI design simplifies the management of interactions, allowing users to concentrate on the meeting itself rather than getting bogged down by technical controls. We focused on creating simple, intuitive interactions to provide a seamless experience for all users.

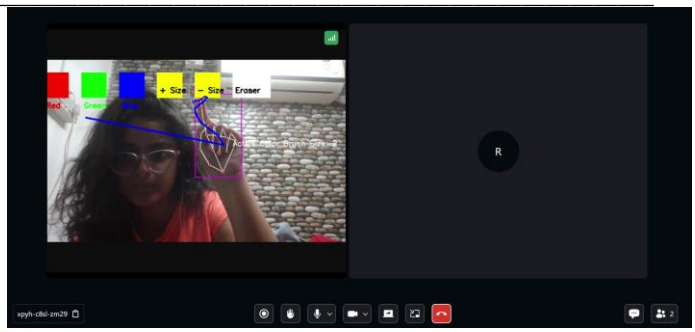

Fig 4 : Demonstration of Our Video Meet App Integrated with Virtual AI Writer

D. PSEUDO CODE

START

// Step 1: Create the Project
createProject()
- Initialize a new project for the video conferencing system.
- Set up the directory structure for the frontend and backend.
- Add necessary configuration files.
- Install Video SDK and other dependencies for real-time communication.

Command:
npm init
npm install videosdk-live --save

OUTPUT: Local project directory with required configurations and dependencies

// Step 2: Configure the Environment
configureEnvironment()
- Copy .env.example to .env
- Obtain VIDEO_SDK_TOKEN from dashboard
- Set REACT_APP_VIDEOSDK_TOKEN in .env

OUTPUT: Configured environment with authentication token

// Step 3: Install Dependencies
installDependencies()
- Run 'npm install' to install all required libraries

OUTPUT: All dependencies are installed

// Step 4: Run the Application
runApplication()
- Execute 'npm run start'

OUTPUT: Application launched and accessible via local server

// Step 5: Initialize a Meeting
initializeMeeting(token)
INPUT: token = REACT_APP_VIDEOSDK_TOKEN
- Authenticate user using token
- Create meeting session object

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 13 Issue: 1
***Article Received:** 25 November 2024 **Revised:** 12 December 2024 **Accepted:** 30 January 2025*

OUTPUT: New meeting session created

// Step 6: Manage Participants and Sessions
manageParticipants(meetingSession)
- Define local participant (self) and remote participants (others)
- Allow participants to join, leave, or rejoin meeting
FOR each participant:
IF participant joins:
- Initialize media stream (audio/video)
ELSE IF participant leaves:
- Terminate participant's media stream
OUTPUT: Active participants and media streams managed

// Step 7: Stream Media in Real-Time
streamMedia(participants)
FOR each participant in participants:
- Start video/audio stream
- Render participant's media in grid layout
- Synchronize streams in real time
OUTPUT: Real-time video/audio communication displayed

// Step 8: Token-Based Authentication
authenticateUser(token)
IF (in development mode):
- Use temporary token for quick setup
ELSE (in production):
- Set up authentication server to issue valid tokens
OUTPUT: Secure access to meeting

// Step 9: Integrate Virtual Drawing with OBS Virtual Camera
initializeCameraFeed()
- Capture webcam frames using OpenCV
- Flip the frame horizontally for mirror effect
- Initialize Hand Detector for gesture recognition
- Detect hand landmarks and determine finger positions

FOR each frame:
- Display color options (Red, Green, Blue) for drawing
- Detect if the user selects a color box via finger gestures
IF color selected:
Set current drawing color
- Detect gestures to increase or decrease brush size
IF brush_size_increase_gesture:
Increase brush size
IF brush_size_decrease_gesture:
Decrease brush size
- Toggle eraser mode when eraser button is selected
IF eraser_button_selected:
Set eraser mode to True
- Track finger positions to draw on the frame using selected color or eraser
- Detect gesture to clear screen (e.g., thumbs-down gesture)
IF clear_screen_gesture_detected:
Clear all drawing points
- Open MS Paint if a specific gesture (e.g., three fingers up) is detected

- Display the virtual drawing on the captured frame
- Output the video feed to a local URL (e.g., "http://localhost:5000/video_feed")

OUTPUT: Virtual drawing with hand gestures integrated into webcam feed

// Step 10: Use OBS Virtual Camera
setupOBSVirtualCamera()
- Input the local video feed URL into OBS
- Start OBS Virtual Camera
OUTPUT: OBS Virtual Camera feed with virtual drawing enabled

// Step 11: Switch to OBS Virtual Camera in the Video Conferencing App
switchCameraMode()
- In the conferencing app, switch input source to OBS Virtual Camera
- Allow participants to see virtual drawing and interactions in real time
OUTPUT: Real-time virtual drawing, object manipulation, and mouse integration in conferencing app

// Step 12: Customize the UI/UX
customizeUI()
- Design participant grid layout that dynamically adjusts to the number of participants
- Add buttons for mute/unmute, screen sharing, etc.
- Provide real-time feedback (e.g., active speaker indicator)
OUTPUT: User-friendly, responsive interface

// Step 13: Monitor and Manage System Performance
monitorPerformance()
- Track network latency, bandwidth usage, and connection quality
- Optimize performance as needed
OUTPUT: Smooth, high-quality user experience

END

V. AI FEATURES AND FUNCTIONALITIES

A. Gesture recognition system

The implemented gesture recognition system utilizes the **MediaPipe Hands** library to detect and track hand landmarks in real-time. This system supports the recognition of multiple hand gestures [27] through an intuitive user interface, enabling users to perform various actions without physical contact with a traditional input device. Key features of the gesture recognition system include:

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 13 Issue: 1
Article Received: *25 November 2024* ***Revised:*** *12 December 2024* ***Accepted:*** *30 January 2025*

- **Hand Landmark Detection**: The system detects 21 hand landmarks using a pre-trained model, providing precise tracking of finger movements. The configuration used allows for the recognition of hand positions with a detection confidence of up to **0.7**, ensuring robustness in various lighting conditions.
- **Finger State Monitoring**: The code includes logic to assess the position of fingers (index, middle, ring, and thumb) to determine if they are extended or retracted. This functionality [28] is critical for executing specific commands, such as:
    - **Drawing Shapes**: Triggered by the position of the index finger.
    - **Clearing the Canvas**: Activated by having the index, middle, and ring fingers up simultaneously.
    - **Color Selection**: Facilitated by moving the index finger over a designated color palette.
    - **Shape Selection**: Based on finger placement over shape buttons.

Dynamic Interaction: The application allows users to draw on a virtual canvas by tracking finger movements and translating them into real-time drawings, enhancing the interactive experience. The integration of functions for **lines**, **rectangles**, **circles**, and **freestyle drawing** [29] showcases the versatility of the system.

B. Air writing and virtual typing mechanisms

The air writing mechanism allows users to write in the air using hand gestures, which the system translates into visual representations on a digital canvas. This feature is particularly beneficial for—**Enhancing Collaboration**: In remote settings, users can visualize ideas quickly without physical materials [30], fostering a more collaborative environment during meetings or brainstorming sessions.

C. Real-time processing and interaction

The system operates at a high frame rate, capturing video input from the webcam and processing it in real-time. This efficiency is essential for maintaining an interactive user experience, allowing for:

- **Instantaneous Feedback**: Users see their actions immediately reflected on the screen, making the interaction more intuitive.
- **Fluid Transitions**: The code handles transitions between different drawing modes and gestures smoothly, enhancing usability.

D. Important code snippets

- *Clear Canvas Logic: This section demonstrates the interaction between finger states and canvas manipulation*

```
if fingers_up['index'] and fingers_up['middle'] and fingers_up['ring']:
    canvas = np.ones((screen_height, screen_width, 3), dtype=np.uint8) * 255
```

- *Color Selection Mechanism:*

```
# Check if index finger is inside the color palette
if palette_start_x + i * 80 < x < palette_start_x + i * 80 + 70 and 20 < y < 90:
    current_color = color
```

- *Drawing Logic Implementation:*

```
if current_shape == 'Line':
    cv2.line(canvas, (start_x, start_y), (x, y), current_color, brush_thickness)
```

VI. MEDIAPIPE vs. YOLO: AN ANALYSIS OF RELEVANCE AND PERFORMANCE IN AI-ENHANCED VIRTUAL INTERACTION

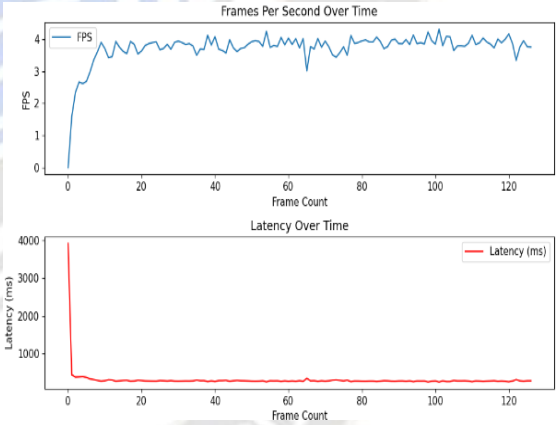

Fig 5: YOLO (FPS,LATENCY)

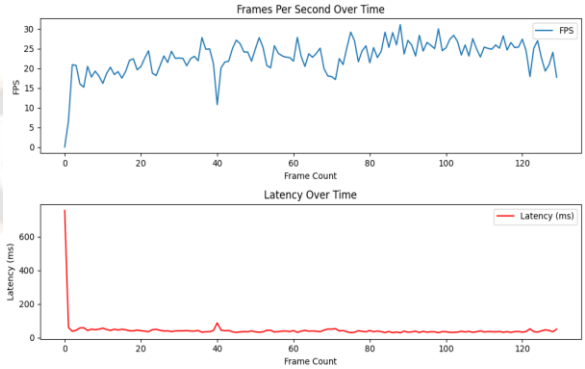

Fig 6: MEDIAPIPE (FPS,LATENCY)

MediaPipe achieves a relatively high frame rate, averaging around 25-30 FPS with some fluctuations. This is an indication that MediaPipe can provide a more real-time experience with faster frame processing.
Thus, they are more suitable for virtual reality, interactive environments like within chat rooms.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 13 Issue: 1
Article Received: *25 November 2024* ***Revised:*** *12 December 2024* ***Accepted:*** *30 January 2025*

YOLO's [31] frame rate is significantly lower, around 3-4 FPS, which suggests it requires more computational power and time to process each frame.

It shows lower latency, with an initial spike but stabilizing around 50-100 ms. This indicates an ephemeral response time, which is crucial for interactive tasks like real-time fingertip detection.

It exhibits a much higher initial latency (around 4000 ms) but then stabilizes to under 1000 ms. This high latency makes it less suitable for real-time applications, especially when low-latency response is needed.

CRITERIA	MediaPipe	YOLO
FPS	25-30 FPS (high real-time performance)	3-4 FPS (low performance, not real-time)
Latency	50-100 ms (low latency, highly responsive)	1000-4000 ms (high latency, non-responsive)
Development Time	Uses pre-trained model	Model tuning and training
Accuracy	Good for simple hand gestures and tracking	High accuracy, especially in complex environments
Computational Requirement	Low computational requirements, can run on mobile devices	High computational requirements, needs powerful hardware like GPUs
Processing Power	Low (can run on mobile devices)	High (requires GPU)
CRITERIA	MediaPipe	YOLO

TABLE 2: COMPARISON BETWEEN MEDIAPIPE V/S YOLO

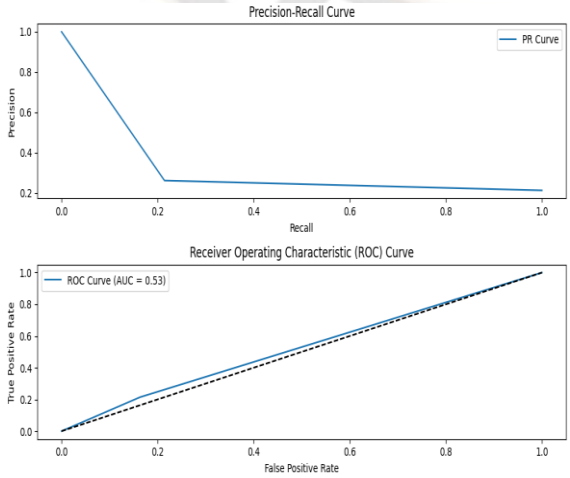

Fig 7: PR_CURVE(MEDIAPIPE)

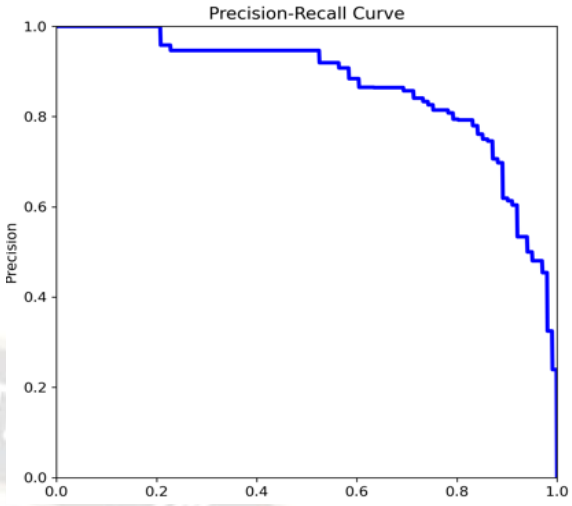

Fig 8: PR_CURVE(YOLOV5)

A. MediaPipe—

The PR [32] curve appears to drop steeply after a recall of 0.2, which suggests that this model has high precision when recall is very low but suffers from a rapid decrease in precision as recall increases.
High precision (1.0) at low recall indicates that the model is very selective about the points it predicts as positive but captures only a small portion of all positives.

As recall increases (more true positives are captured), precision drops to around 0.2.
This could mean that the model is good at identifying clear hand gestures (few false positives), but as it tries to predict more gestures, its confidence reduces, leading to more false positives.

B. YoloV5—

The curve shows high precision over a broad range of recall values (up to 0.8), followed by a gradual decrease.

The curve is nearly flat at the top-left, indicating that the model maintains high precision even as it increases recall (captures more true positives).

Precision only starts dropping significantly after recall reaches around 0.9, meaning the model can classify most true positives while maintaining accuracy in its predictions.
This suggests that **YOLOv5** [33] is able to detect hand gestures with much higher consistency compared to MediaPipe, achieving both high precision and high recall.

For our particular code, we tried to implement the FingerDetection System using **CVZone** [34]. The CVZone is build to abstract and simplify most of the repetitive tasks that are needed to be manually handled in MediaPipe. For Instance, **HandTrackingModule** in CVZone provides several useful functions out of the box.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 13 Issue: 1
Article Received: *25 November 2024* ***Revised:*** *12 December 2024* ***Accepted:*** *30 January 2025*

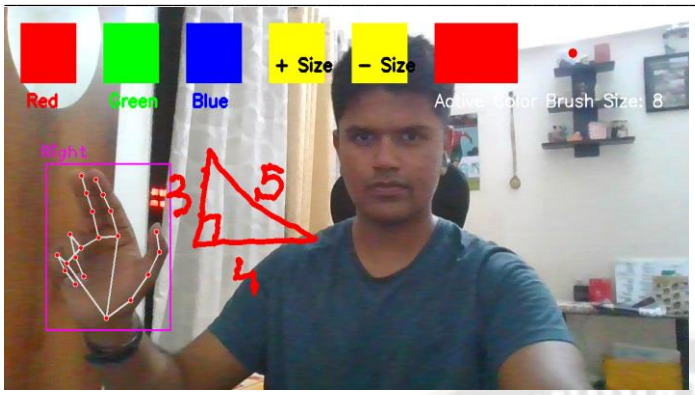

Fig 9: A demonstration using CVZone

C. YOLSE—

Another great approach, for finger detection is the Two way pipeline of YOLO+SOLO(YOLSE) [35] model which uses where Yolo is used for Hand Tracking and capture the frame of the hand, then utilize the SOLO model for tracking the fingertips.

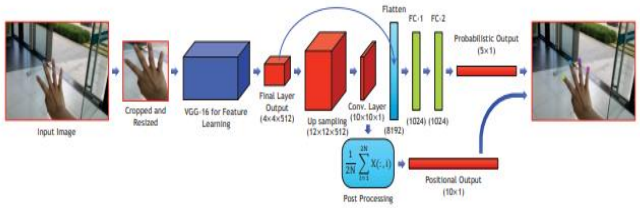

Fig 10: YOLSE MODEL ARCHITECTURE FLOW DIAGRAM

VII. TESTING AND DEPLOYMENT: EXPERIMENTAL SETUP

In our project, we created a thorough testing environment to ensure the video conferencing system's functionality and reliability. We began by setting up a local development environment using Flask for the backend and React for the frontend. This setup enabled us to develop a responsive user interface while effectively managing real-time audio and video streams with the Video SDK [36]. To improve the user experience, we incorporated a camera feed with OpenCV, which allowed for AI-driven virtual drawing and hand gestures.

After getting the local system up and running, we utilized Ngrok to make our application accessible to the public. Ngrok was essential in forwarding our local server to a publicly available URL, making it easy for external users to access the video conferencing platform. Participants could simply copy and paste the link generated by Ngrok into their browsers, leading them to a meeting site where they could enter their name and meeting ID to join the session. This straightforward access encouraged participation from various locations without requiring complicated setups.

Evaluation metrics and criteria and Case studies or scenarios tested— to assess the system's performance, we established several metrics and criteria. The primary focus was on latency [37], which measured the delay between the transmission of video and audio data and its reception by other participants. We used network latency trackers and performance logs to determine average latency under different participant loads. Additionally, we evaluated bandwidth usage to understand the system's data consumption during various scenarios, including video streams, screen sharing, and virtual drawing activities.

Video and audio quality were crucial to the user experience. We collected feedback from participants, especially when switching between the standard camera feed and the OBS Virtual Camera [38] that showcased virtual drawing. Key quality indicators included clarity, frame rate, and overall smoothness during virtual sessions. Furthermore, we monitored system resource usage.

VIII. RESULTS AND ANALYSIS

Implementation of the “Smart Conferencing Rooms” project has brought noteworthy progress in developing a working model of integrating an artificial intelligence gesture recognition system into a web application platform.

Performance evaluation of AI features— By using the CV Zone library we have integrated a few interesting objects in our React application which in turn has exposed the user experience. For example, drawing in midair has been enhanced by the development of features such as changing the size of the brush through simple hand gestures during virtual classes. Furthermore, the normal form of the system permits facile scrolling, hence offering time-efficient movement throughout the contents, an especially imperative factor in the application of the systems in education as well as the health sector. In addition to this, we have incorporated virtual mouse actions to make zoom-in/out gestures [39] and to click/ select objects by gestures of hand at the same level, making the interaction model slightly more kinetic. Our project also contains shape drawings in midair and can be useful in presentations or educational demonstrations. Finally, users can simply interchange the color of the drawing pen through some finger movements to make it friendly to their color preference during their interaction with the software. Combined, these features not only solve the problems of flaws in conventional conferencing and meeting equipment but also show how gesture recognition technologies can significantly change the dynamics of virtual communication patterns. It makes communication truly natural, intuitive, and efficient while preserving the possibility of using the application on different platforms due to compatibility with a wide range of devices.

Comparative analysis with existing systems—In comparison to existing systems such as the "Virtual Air Canvas" or webcam-based air-writing systems that rely heavily on frameworks like OpenCV and MediaPipe, our "Smart Conferencing Rooms" project introduces significant improvements by integrating AI-driven gesture recognition with a more sophisticated web application framework. While earlier systems typically focus on limited interaction models, such as simple virtual drawing or text input via gestures, our project enhances the user experience by adding features like dynamic color changes, shape drawing, and virtual mouse functionality [40]. This not only offers a more immersive and versatile tool but also addresses practical use cases in education and healthcare through gesture-based navigation and object manipulation. Furthermore, our system's integration within a cross-platform React environment ensures greater accessibility compared to standalone applications that might require specific hardware configurations.

***User feedback and usability studies*—** User feedback from early-stage usability testing indicates high satisfaction with the system's intuitive design and responsiveness. Participants found the gesture-based interactions, such as adjusting brush size and changing colors, to be both accurate and engaging [41], particularly during virtual conferencing sessions. Compared to traditional input methods, users noted that the gesture-based interface reduced the cognitive load and provided a more seamless interaction experience. Many users highlighted the system's potential for enhancing remote learning and virtual presentations, particularly the midair shape drawing and scrolling features, which were praised for improving content navigation in real-time. This positive reception suggests that our system effectively meets user needs and offers a practical alternative to conventional tools.

IX. FUTURE WORK

A. *Multi-Language Sign Language Interpretation*

In Future We aim to implement a feature that leverages fingertip detection to translate spoken languages into sign language [42] in real-time. This system will enhance accessibility for deaf or hard-of-hearing users by providing visual sign language translation during virtual meetings or conversations.

B. *Augmented Reality (AR) Annotations*

A proposed extension includes AR-based [43] annotations where users can mark their physical environment through their device's camera feed. These annotations would allow for enhanced collaborative discussions by providing interactive, location-based notes that can be viewed and modified in real-time.

C. *Character Enhancement for Readability*

To improve the readability of written or drawn content, we plan to integrate an Optical Character Recognition (OCR) [44] system that can convert handwritten or air-drawn text into clear, user-selected font types. This will provide more accessible content and customization options, enhancing the overall user experience.

X. CONCLUSION

One of the promising products is the "Smart Conferencing Rooms," which have been used to combine the application of artificial intelligence gesture recognition and create new virtual communication between clients and educators or healthcare providers. Besides these innovations, we created a separate application for video conferencing that includes the most relevant features like video off/on, microphone on/off, screen sharing, and in-meeting chat, allowing users to have complete and convenient communication. Our application has hand detection, fingertip tracking, and air writing with additional features such as midair drawing, creating shapes with fingers, virtual mouse interaction, and scrolling. These features respond to the problems of conventional conference solutions by providing improved and more effective means of collaborative communication interface. Our system has been designed to run on multiple platforms, thus making it versatile for use by any user. The combination of real-time collaboration, gesture-based controls, and basic conferencing tools significantly enhances virtual interactions, making them more engaging, accessible, and impactful, especially in education and healthcare.

REFERENCES

[1] J, S. R., A, S., & B, S. B. (2023). Air Canvas: Expressing Creativity in the Real World with OpenCV and Python. 2023 International Conference on Energy, Materials and Communication Engineering (ICEMCE), Madurai, India, 2023. DOI : https://doi.org/10.1109/icemce57940.2023.10434262

[2] Z. -Q. Zhao, P. Zheng, S. -T. Xu and X. Wu, "Object Detection With Deep Learning: A Review," in IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 11, pp. 3212-3232, Nov. 2019. DOI : https://doi.org/10.1109/tnnls.2018.2876865.

[3] O. Köpüklü, A. Gunduz, N. Kose and G. Rigoll, "Real-time Hand Gesture Detection and Classification Using Convolutional Neural Networks," 2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019), Lille, France, 2019, pp. 1-8, DOI : https://doi.org/10.1109/fg.2019.8756576

[4] G. Lavania, M. Rashid, V. Arya, S.V. Akram, and N. Sharma, "Real-Time Signal Processing using AI Integrated Framework for Color and Drawing in Gesture Recognition," 2022 5th International Conference on Contemporary Computing and Informatics (IC3I), 2022, DOI : https://doi.org/10.1109/ic3i56241.2022.10072448.

[5] A. Mehra, S. Singh, P. Singh, K. Gupta, and T. Agrawal, "Air Canvas for Educational Systems with Hand Tracking in Real Time using Mediapipe: A Computer Vision," 2023 International Conference on Communication, Security and Artificial Intelligence (ICCSAI), pp. 1-8, 2023, DOI : https://doi.org/10.1109/iccsai59793.2023.10421428.

[6] Chavali, E. S. (2023). Virtual mouse using hand gesture. International Journal of Scientific Research in Engineering and Management, 7(7), 1-5. DOI : https://doi.org/10.55041/IJSREM21501

[7] Q. Wang, J. Cheng, J. Pang and S. Shen, "Fingertip-based interactive projector-camera system," 2013 IEEE International Conference on Information and Automation (ICIA), Yinchuan, China, 2013, pp. 140-144, DOI : https://doi.org/10.1109/icinfa.2013.6720285.

[8] K. S. Abhishek, L. C. F. Qubeley, & D. Ho. (2016). Glove-based hand gesture recognition sign language translator using a capacitive touch sensor. In 2016 IEEE International Conference on Electron Devices and Solid-State Circuits (EDSSC) (pp. 334–337). IEEE. DOI : https://doi.org/10.1109/edssc.2016.7785276

[9] M. Wenzel and C. Meinel, "Full-body WebRTC video conferencing in a web-based real-time collaboration system," 2016 IEEE 20th International Conference on

Computer Supported Cooperative Work in Design (CSCWD), Nanchang, China, 2016, pp. 334-339,
DOI : https://doi.org/10.1109/cscwd.2016.7566010.

[10] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee, W.-T. Chang, W. Hua, M. Georg, and M. Grundmann, "MediaPipe: A framework for building perception pipelines," arXiv preprint, arXiv:1906.08172, 2019. [Online]. Available:
DOI : https://arxiv.org/abs/1906.08172.

[11] Sukhdeep Singh, Anuj Sharma, and Indu Chhabra. A dominant points-based feature extraction approach to recognize online handwritten strokes. International Journal of Document Analysis and Recognition, 20:37–58, 2017.
DOI : https://doi.org/10.1007/s10032-016-0279-x.

[12] Pavlovic, V., Sharma, R., & Huang, T. (1997). Visual interpretation of hand gestures for human-computer interaction: a review. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7), 677–695.
DOI : https://doi.org/10.1109/34.598226.

[13] Oka, K., Sato, Y., & Koike, H. (2002). Real-time fingertip tracking and gesture recognition. IEEE Computer Graphics and Applications, 22(6), 64–71.
DOI : https://doi.org/10.1109/mcg.2002.1046630.

[14] Yilmaz, A., Javed, O., & Shah, M. (2006). Object tracking. ACM Computing Surveys, 38(4), 13.
DOI : https://doi.org/10.1145/1177352.1177355

[15] Chang, Y., & Chang, C. (2010). Automatic Hand-Pose Trajectory Tracking System using video sequences. In InTech eBooks.
DOI : https://doi.org/10.5772/9492

[16] Baig, F., Khan, M. F., & Beg, S. (2013). Text writing in the air. Journal of Information Display, 14(4), 137–148.
DOI : https://doi.org/10.1080/15980316.2013.860928

[17] Ohn-Bar, E., & Trivedi, M. M. (2014). Hand gesture recognition in real time for automotive interfaces: a multimodal Vision-Based approach and evaluations. IEEE Transactions on Intelligent Transportation Systems, 15(6), 2368–2377.
DOI : https://doi.org/10.1109/tits.2014.2337331

[18] Sun, X., & Lv, M. (2019). Facial expression recognition based on a hybrid model combining deep and shallow features. Cognitive Computation, 11(4), 587–597.
DOI : https://doi.org/10.1007/s12559-019-09654-y.

[19] Zhang, Y., Huang, Y., Sun, X., Zhao, Y., Guo, X., Liu, P., Liu, C., & Zhang, Y. (2020). Static and dynamic human Arm/Hand gesture capturing and recognition via multiinformation fusion of flexible strain sensors. IEEE Sensors Journal, 20(12), 6450–6459.
DOI : https://doi.org/10.1109/jsen.2020.2965580

[20] Chandan, G., Jain, A., Jain, H., & Mohana, N. (2018). Real time object detection and tracking using deep learning and OpenCV. 2018 International Conference on Inventive Research in Computing Applications (ICIRCA).
DOI : https://doi.org/10.1109/icirca.2018.8597266

[21] Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C. L., & Grundmann, M., "Mediapipe hands: On-device real-time hand tracking," arXiv preprint arXiv:2006.10214, 2020.
DOI : https://doi.org/10.48550/arXiv.2006.10214.

[22] Jaimes, A., & Miyazaki, N. J. (2005). BUILDING A SMART MEETING ROOM: FROM INFRASTRUCTURE TO THE VIDEO GAP (RESEARCH AND OPEN ISSUES). Conference: Data Engineering Workshops, 2005. 21st International Conference On, 1, 1173.
DOI : https://doi.org/10.1109/icde.2005.202.

[23] Real-Time video conferencing application. (2022). International Journal of Creative Research Thoughts (IJCRT), 11, c212–c214.
DOI : https://ijcrt.org/papers/IJCRT2211254.pdf.

[24] Karis, D., Wildman, D., & Mané, A. (2015). Improving Remote Collaboration With Video Conferencing and Video Portals. Human-Computer Interaction, 31(1), 1–58.
DOI : https://doi.org/10.1080/07370024.2014.921506.

[25] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, & T. Darrell. (2015). Long-term recurrent convolutional networks for visual recognition and description. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2625–2634).
DOI : https://doi.org/10.1109/cvpr.2015.7298878.

[26] D. Pratiwi and I. J. Matheus Edward, "Analysis Efficiency Network Performance of 4G LTE in Video Conference Applications," 2022 16th International Conference on Telecommunication Systems, Services, and Applications (TSSA), Lombok, Indonesia, 2022, pp. 1-6,
DOI : https://doi.org/10.1109/tssa56819.2022.10063921.

[27] Khan, R. Z. (2012). Hand Gesture Recognition: A literature review. International Journal of Artificial Intelligence & Applications, 3(4), 161–174.
DOI : https://doi.org/10.5121/ijaia.2012.3412.

[28] Hong, W., Lee, J., & Lee, W. G. (2021). A finger-perimetric tactile sensor for analyzing the gripping force by chopsticks towards personalized dietary monitoring. Sensors and Actuators a Physical, 333, 113253.
DOI : https://doi.org/10.1016/j.sna.2021.113253.

[29] Agrawal, S. C., Tripathi, R. K., Bhardwaj, N., & Parashar, P. (2023). Virtual Drawing: An Air Paint Application. IEEE Conference, 971–975.
DOI : https://doi.org/10.1109/icecaa58104.2023.10212239.

[30] Khalifa, M., & Albadawy, M. (2024). Using Artificial intelligence in academic writing and research: an essential productivity tool. Computer Methods and Programs in Biomedicine Update, 5, 100145.
DOI : https://doi.org/10.1016/j.cmpbup.2024.100145.

[31] Joseph, R., Santosh, D., Ross, G., & Ali, F. (2015). You only look once: Unified, Real-Time Object Detection. arXiv (Cornell University).
DOI : https://doi.org/10.48550/arxiv.1506.02640.

[32] Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C., Yong, M. G., Lee, J., Chang, W., Hua, W., Georg, M., & Grundmann, M. (2019). MediaPipe: A framework for building perception Pipelines. arXiv (Cornell University).
DOI : https://doi.org/10.48550/arxiv.1906.08172.

[33] S, M., S, M. S., T, K., & P, S. (2023). Image Detection and Segmentation using YOLO v5 for surveillance. Applied and Computational Engineering, 8(1), 142–147.
DOI : https://doi.org/10.54254/2755-2721/8/20230109.

[34] Nguyen, H. A., Tran, T. T., Ho, H. Q., Ngo, T. D., Vu, K. N., & Huynh, V. L. T. (2024). Hand Gesture Recognition using CVZONE. ICIIT 2024: 2024 9th International Conference on Intelligent Information Technology.
DOI : https://doi.org/10.1145/3654522.3654540.

[35] Wu, T., & Dong, Y. (2023). YOLO-SE: Improved YOLOV8 for remote sensing object detection and recognition. Applied Sciences, 13(24), 12977.
DOI : https://doi.org/10.3390/app132412977.

[36] Labrie, A., Mok, T., Tang, A., Lui, M., Oehlberg, L., & Poretski, L. (2022). Toward Video-Conferencing tools for Hands-On activities in online teaching. Proceedings of the ACM on Human-Computer Interaction, 6(GROUP), 1–22.
DOI : https://doi.org/10.1145/3492829.

[37] Lyko, T., Broadbent, M., Race, N., Nilsson, M., Farrow, P., & Appleby, S. (2023). Improving quality of experience in adaptive low latency live streaming. Multimedia Tools and Applications, 83(6), 15957–15983.
DOI : https://doi.org/10.1007/s11042-023-15895-9.

[38] Kristandl, G. (2021). "All the world's a stage" – the Open Broadcaster Software (OBS) as enabling technology to overcome restrictions in online teaching. Compass Journal of Learning and Teaching, 14(2).
DOI : https://doi.org/10.21100/compass.v14i2.1241.

[39] Wang, L., Hossain, M. S., Pulfrey, J., & Lancor, L. (2021). The effectiveness of zoom touchscreen gestures for authentication and identification and its changes over time. Computers & Security, 111, 102462.
DOI : https://doi.org/10.1016/j.cose.2021.102462.

[40] Reddy, K. B., Fayazuddin, M., Manohar, M. J., & Tavanam, V. (2022). Artificial intelligence based virtual mouse. International Journal of Computer Science and Mobile Computing, 11(1), 25–35.
DOI : https://doi.org/10.47760/ijcsmc.2022.v11i01.005.

[41] Jabde, M. K., Patil, C. H., Vibhute, A. D., & Mali, S. (2024). A comprehensive literature review on air-written online handwritten recognition. International Journal of Computing and Digital Systems, 15(1), 307–322.
DOI : https://doi.org/10.12785/ijcds/150124.

[42] Kulkarni, A., Kariyal, A. V., Dhanush, V., & Singh, P. N. (2021). Speech to Indian sign language translator. Atlantis Highlights in Computer Sciences/Atlantis Highlights in Computer Sciences.
DOI : https://doi.org/10.2991/ahis.k.210913.035.

[43] Al-Ansi, A. M., Jaboob, M., Garad, A., & Al-Ansi, A. (2023). Analyzing augmented reality (AR) and virtual reality (VR) recent development in education. Social Sciences & Humanities Open, 8(1), 100532.
DOI : https://doi.org/10.1016/j.ssaho.2023.100532.

[44] Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR). (2020). IEEE Journals & Magazine | IEEE Xplore.
DOI : https://ieeexplore.ieee.org/document/9151144