

AirSwap: Touchless File Transfer Through Hand Gestures

MR. MOMIN UBED SALIM

*Ashokrao Mane Group Of institution,vathar
Department of Computer Science and Engineering
ubedmomin2017@gmail.com*

MR. SHAHA SUJAL SUNILKUMAR

*Ashokrao Mane Group Of institution,vathar
Department of Computer Science and Engineering
sujalshaha974@gmail.com*

MISS. MULLA NAZNEEN IBRAHIM

*Ashokrao Mane Group Of institution,vathar
Department of Computer Science and Engineering
nazninmulla21@gmail.com*

MISS. MAHAMUNI PRACHI PRAKASH

*Ashokrao Mane Group Of institution,vathar
Department of Computer Science and Engineering
prachimahamuni623@gmail.com*

MR. MOTKATTE SHRIYASH RAJENDRA

*Ashokrao Mane Group Of institution,vathar
Department of Computer Science and Engineering
shriyashmotkatte@gmail.com*

MR. ANIL B. DESAI

*Assitant Professor at Ashokrao Mane Group Of
institution,vathar
Department of Computer Science and Engineering
abd@amgoi.edu.in*

Abstract :

The increasing use of touchless technologies and intelligent human computer interaction systems has created a demand for seamless and contact free digital communication. However, traditional file-transfer methods still face issues such as hardware dependency, manual interaction, and complex workflows. This paper introduces AirSwap, an intelligent gesture-driven touchless file-transfer platform developed using computer vision, machine learning, and peer-to-peer communication technologies. The system uses a multi-layer architecture including user interfaces, real-time gesture recognition, socket-based communication, and synchronization mechanisms for efficient cross-device file sharing. AirSwap employs MediaPipe-based hand tracking and OpenCV preprocessing to recognize predefined hand gestures through landmark detection and motion analysis. To ensure reliable performance, the system integrates wireless peer-to-peer communication and low-latency transfer protocols for stable connectivity and reduced false gesture activation. Experimental results show that AirSwap achieves accurate gesture recognition, reliable low-latency communication, and improves accessibility, user convenience, and efficiency in touchless file sharing.

INDEX TERMS: Gesture Recognition, Computer Vision, MediaPipe, OpenCV, Touchless Interaction, Peer-to-Peer Communication, Socket Programming, Human-Computer Interaction, Wireless File Transfer, Machine Learning.

1.INTRODUCTION

AirSwap is a gesture-driven file-transfer platform designed to make it easy for users to share files between devices through simple hand movements without requiring physical contact. In many digital environments, users still face difficulties while transferring files quickly and efficiently. Traditional methods such as USB sharing, Bluetooth pairing, QR scanning, and cloud uploads are often slow, require multiple steps, and depend heavily on manual interaction or hardware availability. At the same time, users working in collaborative, fast-paced, or hygiene-sensitive environments require more seamless and touchless interaction systems. Existing file-sharing methods often introduce delays, pairing issues, and operational complexity, leading to reduced efficiency and poor user experience. AirSwap solves these challenges by connecting gesture recognition with wireless peer-to-peer communication into a single intelligent platform. Users can easily initiate file transfers using predefined hand gestures, monitor transfer status in real time, and complete sharing without manual device interaction. The platform also uses advanced technologies such as **computer vision, MediaPipe hand tracking, and OpenCV-based gesture processing** to recognize user gestures and trigger secure file exchange between devices. Additionally, the system ensures low-latency communication to improve transfer speed and reliability across connected devices. AirSwap also minimizes physical interaction, making file sharing more convenient and accessible in modern smart environments. Furthermore, the platform provides a seamless and efficient touchless experience by integrating real-time synchronization and secure transmission mechanisms.

1. LITERATURE REVIEW:

In recent years, gesture-recognition systems and touchless interaction technologies have seen major improvements through the use of **computer vision**, **machine learning**, and intelligent automation aimed at improving interaction efficiency and accessibility. Several studies have explored intelligent systems that enable users to interact with devices through gestures instead of traditional touch-based methods. Recent research found on platforms like IEEE Xplore, Springer, and ScienceDirect highlights the development of smart gesture-based systems that combine hand tracking, motion recognition, and real-time processing to improve interaction performance. These systems demonstrate the importance of vision-based interaction; however, most are designed to solve specific interaction problems rather than provide a complete touchless file-transfer solution.

Computer vision methods have also been widely applied in gesture-recognition platforms. Chen et al. proposed a gesture-based interaction system that uses hand-motion tracking and spatial movement analysis to enable device communication. While the system improves gesture-recognition accuracy, it mainly focuses on command interaction and does not include critical services such as seamless file transfer or real-time communication synchronization.

Environmental limitations and recognition inconsistency have emerged as important factors affecting touchless interaction performance. Researchers have studied gesture detection using machine-learning models under varying lighting conditions and camera constraints. Their work provides valuable insights into recognition reliability; however, it lacks real-time transfer execution and interactive communication features for users.

The adoption of gesture-driven technologies in daily digital workflows continues to face several challenges. Researchers discussed the gap between advanced gesture systems and their practical implementation, identifying limitations such as poor environmental adaptability, hardware dependency, and interface complexity. These challenges reduce the practical usability of touchless interaction systems for everyday users. In communication reliability and interaction validation, researchers developed intelligent frameworks using real-time landmark tracking and gesture prediction to improve recognition accuracy. Furthermore, recent advancements in gesture-based interaction increasingly rely on lightweight computer-vision algorithms to improve response speed and reduce operational latency, although many of these systems remain standalone and do not integrate complete file-sharing functionality for users. Market adoption and usability of touchless technologies have also been explored in recent studies. Researchers investigated automated interaction systems that improve user accessibility and communication efficiency, enabling users to experience more intuitive and seamless digital interaction.

However, most existing systems mainly focus on isolated functionalities such as basic gesture detection, simple device communication, or limited command execution and do not integrate intelligent interaction with seamless file-sharing workflows. From these studies, it is clear that although there has been progress in gesture recognition and touchless interaction systems, most current solutions focus on restricted device communication rather than providing a complete touchless file-transfer ecosystem.

2. PROPOSED SYSTEM:

2.1. System description:

The proposed system is **AirSwap**, a contactless, desktop-based peer-to-peer (P2P) file-transfer platform designed to maximize user interaction efficiency and device hygiene within local networks through a vision-driven, gesture-controlled ecosystem. AirSwap is engineered using advanced technologies including **computer vision pipelines (MediaPipe Hands and OpenCV)** for real-time 21-point geometric hand-landmark tracking, a modular **PyQt5 graphical user interface**, and multi-threaded, dedicated **TCP/IP socket communication channels** to deliver an intuitive, zero-touch data-sharing experience.

The AirSwap system fundamentally diverges from traditional local file-transfer paradigms which rely on fragmented physical tools (such as USB flash drives), internet-dependent services (like email attachments and third-party cloud storage), or unoptimized local wireless setups (such as cumbersome Bluetooth pairing menus). Instead of forcing users to navigate through manual UI hierarchies or manage tedious offline hardware dependencies, AirSwap introduces a unified, touchless interaction model that translates natural human kinesics directly into network commands.

2.1. Functional Modules:

1) Real-Time Hand Tracking Module

Uses **MediaPipe hand-tracking technology** and webcam-based input to continuously monitor user hand movements and detect landmark positions. The system processes live visual frames to identify finger coordinates and movement patterns, providing accurate hand-tracking information required for gesture recognition.

2) Gesture Recognition Module

Uses computer vision algorithms to identify predefined gestures based on hand movement and landmark positioning. **OpenCV** and gesture-processing techniques are used to analyze user actions and recognize commands such as *grab* and *throw* to initiate touchless file-transfer operations.

3) Peer-to-Peer Communication Module

Uses **socket communication protocols** to establish wireless connections between devices operating on the same network. The communication system manages device synchronization, transfer initiation, and stable connectivity to provide seamless and low-latency file sharing. The module also supports communication session management, transfer validation, and secure data exchange between sender and receiver devices.

4) Real-Time Transfer Monitoring Module

Uses an interactive feedback mechanism to monitor file-transfer progress and communication status. The module provides live updates regarding gesture detection, transfer initiation, completion notifications, and connection states to improve user interaction and operational transparency. Additionally, it helps users monitor transfer activity in real time through visual indicators and notification prompts, ensuring improved usability and better understanding of system operations throughout the communication process.

2.2. System Workflow

The AirSwap system operates under a lightweight **distributed client/server communication structure** designed to support seamless gesture-driven file transfer between connected devices. The workflow begins when the user launches the AirSwap application and activates the webcam interface to enable gesture-based interaction. The user initiates communication by performing predefined hand gestures in front of the camera, while the system continuously captures live visual frames through the webcam interface for processing.

Once the visual input is acquired, the captured frames are forwarded to the **computer vision processing layer**, where **OpenCV-based preprocessing techniques** are used to improve image clarity and frame consistency. The processed frames are then passed to the **MediaPipe hand-tracking engine**, which identifies hand landmarks and extracts movement coordinates required for gesture analysis. The system continuously monitors hand motion patterns to determine whether the performed gesture matches predefined interaction commands such as **grab** or **throw**.

The next stage of the workflow involves determining whether the detected gesture corresponds to valid transfer commands and processing the gesture through recognition and validation pipelines. The backend communication layer evaluates gesture accuracy and validates movement patterns before initiating further system operations. Simultaneously, the communication module checks device availability and establishes a **socket-based peer-to-peer connection** between sender and receiver devices operating within the same wireless network.

The backend also manages communication synchronization and transfer coordination by retrieving required file information and maintaining active session stability during transmission. Once all validation and communication steps are successfully completed, the **File Transfer Module** initiates low-latency wireless transmission between devices. Throughout the transfer process, live system updates such as gesture recognition status, transfer progress, communication stability, and completion confirmation are continuously displayed to users through the interactive monitoring interface, ensuring a smooth, transparent, and touchless file-sharing experience.

The next step in this workflow involves determining whether the detected gestures are intended for real-time transfer actions, such as communication initiation or file-sharing execution, and using gesture-classification

pipelines and movement-validation mechanisms to process these commands. The backend also manages active device connectivity and communication sessions through socket-based networking protocols. Once the processed gesture data has been validated, the system executes the transfer operation and presents real-time visual feedback regarding transfer progress and successful completion through the client interface.

3. SYSTEM ARCHITECTURE

The **AirSwap System**, with its lightweight and modular architecture, provides scalable, efficient, and reliable touchless file-transfer solutions. Utilizing a modular system architecture allows clear functional separation between different system components, enabling independent development of operational modules, simplified maintenance, and easier integration of future enhancements.

The complete architecture of the **AirSwap System** consists of four primary layers—the **Frontend Layer**, the **Backend Layer**, the **Gesture Recognition/Processing Layer**, and the **Communication Layer**. Each of these layers represents a core component of the AirSwap platform. Additionally, the system integrates external technologies such as **MediaPipe**, **OpenCV**, and **socket communication protocols** to support real-time gesture detection and wireless file transfer. Each layer performs specific responsibilities, while communication between components occurs through defined processing pipelines, ensuring smooth data flow and efficient system coordination throughout the AirSwap environment.

1. Front End:

The frontend provides the user with a platform through which to interact with the AirSwap system. The frontend contains three main interfaces:

- **Gesture Interaction Interface** – provides a way for the user to perform hand gestures, activate touchless communication, and initiate file-transfer operations through real-time webcam interaction.
 - **Transfer Monitoring Interface** – provides a way for the user to monitor transfer progress, communication status, gesture recognition responses, and completion notifications during file-sharing operations.
 - **System Control Interface** – provides a way for the user to manage file selection, connection setup, and system activity while also accessing transfer history and operational status information.
- The frontend layer is responsible for collecting input from the user through gesture interaction while also providing a visual method for interpreting the outputs generated by the AirSwap system.

2. Back End:

The backend layer is developed using **Python-based communication frameworks** and networking protocols. The backend is considered the “central processing unit” of the system. The backend serves as the primary means of communication between the frontend, gesture-recognition services, and wireless communication modules.

The backend provides many functionalities, including:

- Gesture-event processing and validation
- Communication management and routing
- Communication between sender and receiver devices
- Processing of gesture-recognition commands
- Handling requests for real-time transfer updates
- Processing of file-transfer and synchronization data
- Transfer status monitoring and report generation

The backend serves as a single point of entry for all requests to all components of the AirSwap system.

3. Gesture Recognition and Processing Layer

The Gesture Recognition and Processing Layer provides intelligent gesture analysis and real-time interaction handling by integrating computer vision techniques with deterministic gesture-validation mechanisms to deliver accurate touchless communication.

Key functionality that this layer provides includes:

- Hand landmark detection (MediaPipe Framework) – to provide real-time hand tracking and coordinate extraction
- Gesture classification and motion recognition – to identify gestures such as *grab* and *throw*
- Movement trajectory analysis – to calculate gesture direction and interaction intent
- Gesture validation and command processing – to verify predefined gesture accuracy
- Status verification engine (checking gesture validity and communication readiness)
- Real-time event synchronization handler for maintaining smooth interaction processing

The gesture recognition layer increases the system’s ability to provide accurate and reliable touchless interaction through intelligent gesture processing and real-time movement analysis.

3. Integrated External Services

Through the integration of multiple external technologies and frameworks, the AirSwap system can provide real-time gesture recognition as well as enhanced communication functionality. For instance, the following services/frameworks provide the required operations:

- MediaPipe Framework :
to provide real-time hand landmark detection
 - OpenCV Library :
to provide image processing and gesture recognition
 - Socket Communication Protocols :
to provide wireless file transfer
 - Webcam Interface :
to provide live gesture input detection
 - File Management Services :
to help process file transfer requests
- These services provide real-time interaction and efficient communication support for the system.

4. Communication and Data Layer

The communication layer of the AirSwap system is built using socket-based networking protocols alongside lightweight file-handling mechanisms and manages all communication processes used by the system. The system handles data in five categories:

- Gesture recognition and movement data
- Device connection and communication records
- File transfer and synchronization logs
- Transfer status and completion notifications
- Communication session and activity history

The communication layer ensures stable connectivity, fast processing, and efficient file transfer between devices.

5. System Workflow

The AirSwap system uses a structured workflow for both gesture input through the frontend and file-transfer output processed through the backend. Upon receiving user gestures through the webcam interface, the backend routes the request either to the **gesture-recognition layer** (MediaPipe/OpenCV processing) and/or to communication services based on the detected interaction. The processed output is then validated and transferred through socket communication protocols or generated dynamically during transfer operations and delivered back to the frontend for display. By configuring the AirSwap system with a layered architecture, it enables efficient data flow, modular development, and improved overall performance.

By configuring the AirSwap system with a layered architecture, it allows efficient data flow through the system, supports modular development, and provides improved performance and seamless touchless interaction.

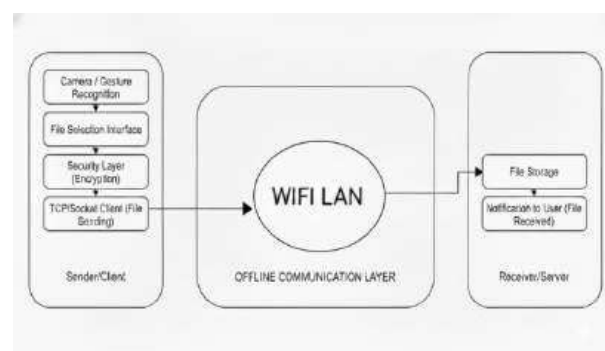


Fig 4.1 System Architecture

1.1 DFD Diagram:

1.1.1 Level 0 DFD:

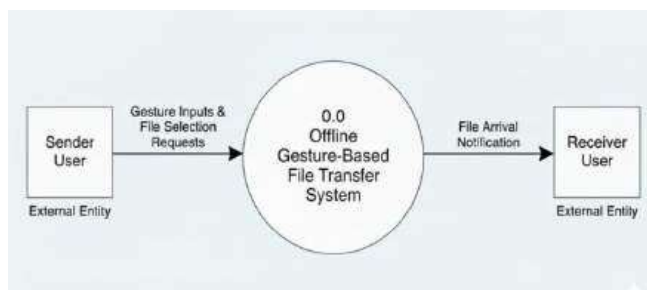


Fig 4.2 Level 0 Data Flow Diagram

The Level 0 DFD illustrates the general interactions between the external entities and the **WorkEase** system. Customers and workers provide initial inputs (e.g., service requests, location coordinates, and availability status) to the application. The system processes this information to coordinate matches, routing transaction data through the payment gateway while triggering the notification service. Ultimately, the system produces direct outputs (e.g., filtered provider listings, real-time tracking updates, and booking confirmations) to the respective stakeholders.

1.1.2 Level 1 DFD:

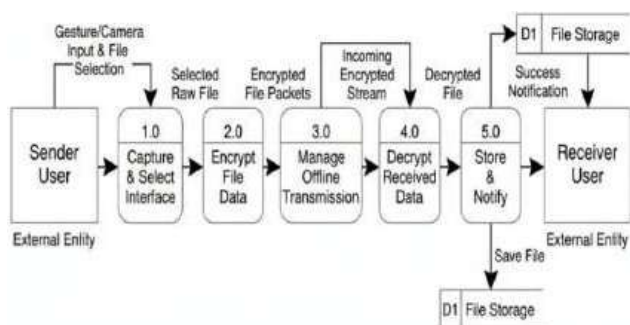


Fig 4.3 Level 1 Data Flow Diagram

The Level 1 DFD illustrates the internal sub-processes of the WorkEase system, breaking down data flows into service booking, worker matching, payment processing, and feedback collection. It tracks how customer requests and worker updates interact directly with the user. The Level 1 DFD illustrates the internal working and detailed sub-processes of the AirSwap system by breaking the overall file-transfer operation into multiple functional modules. It explains how data flows between the sender and receiver during the touchless file-sharing process. The system begins with the sender selecting a file through the capture and selection interface, where gesture commands and file inputs are received.

Level 2 DFD:

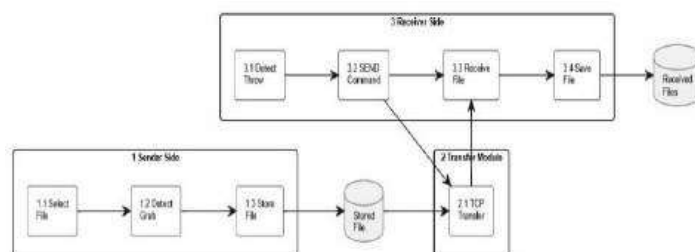


Fig 4.4 Level 2 Data Flow Diagram

The Level 2 DFD refines the system by decomposing the core sub-processes into detailed low-level functions such as file selection, gesture detection, file transmission, and file reception. It explains how the sender selects a file, performs gesture-based commands, and transfers the encrypted file through the transmission module. On the receiver side, the system processes gesture commands, receives the transferred file, and stores it successfully.

The diagram also highlights the interaction between sender and receiver modules to ensure smooth communication and reliable file transfer. Additionally, it demonstrates how real-time data flow and synchronization mechanisms improve transfer efficiency. The Level 2 DFD provides a clearer understanding of internal operations, making the system more structured, secure, and efficient for touchless file sharing.

4. METHODOLOGY

The proposed system is developed as a smart touchless platform that enables seamless wireless file transfer between devices through gesture-based interaction. The main objective of the system is to reduce the complexity users face while transferring files using traditional methods that depend on manual configuration, hardware devices, or multi-step communication procedures.

To achieve this, the project follows a modular approach where each feature is designed as an independent module, and all modules are connected through a centralized backend communication framework. The system works on a **client-server architecture**.

Users interact with the platform using a simple and easy-to-use interface supported by webcam-based gesture interaction. Through this interface, users can perform predefined gestures, initiate communication, monitor transfer progress, and interact with real-time transfer notifications. Once the gesture input is provided, the request is sent to the backend server for processing.

The **gesture-recognition and communication engine** helps users perform seamless file transfer by analyzing gesture patterns processed through the recognition controller. The detected hand movements are processed by the system, which examines visible parameters such as hand landmarks, movement direction, gesture validity, and communication readiness before initiating file-transfer operations. Along with gesture validation, the system also provides communication synchronization and connection management to ensure stable transfer execution between sender and receiver devices.

To provide better system control and operational reliability, a lightweight monitoring module is also included in the system. Gesture activity, communication sessions, and transfer logs are processed and managed to maintain smooth system operation and improve overall user interaction.

The **real-time feedback framework** acts as the primary communication interface of the system. It uses a combination of transfer notifications, communication status handlers, and gesture-event processing to understand application activity and generate visual updates in simple formats. The main focus of this module is to present transfer progress, communication status, and interaction feedback in a way that is easy for users to understand.

5. IMPLEMENTATION

The proposed system is implemented as a **gesture-driven touchless file-transfer application** by integrating computer vision technologies, communication protocols, gesture-recognition frameworks, and real-time synchronization features.

The frontend is developed with a user-friendly design so that users can easily perform gesture-based file-transfer operations. The interface includes options for gesture interaction, file selection, communication monitoring, transfer progress tracking, and real-time notification management.

The backend is responsible for handling all user requests and connecting different modules of the system. Whenever a user performs a gesture, the backend processes the request and forwards it to the required module.

A lightweight communication and data-management

framework is used to store gesture activity, communication sessions, transfer records, and system logs. This helps maintain operational consistency and improve response speed.

For gesture recognition and communication handling, the respective modules (**MediaPipe, OpenCV, and socket communication protocols**) are integrated with the backend server. The backend receives gesture input, processes movement data, and returns the output in a readable format.

For real-time gesture analysis, preprocessing steps such as **hand landmark detection, movement tracking, and gesture validation mechanisms** are applied before passing the interaction metrics to the communication state. This improves the overall performance and recognition accuracy of the system.

The wireless communication and file-transfer feature is implemented using **socket networking protocols** that provide stable peer-to-peer communication between devices. Similarly, gesture-recognition data is continuously processed and managed through computer-vision frameworks and lightweight backend controllers.

The gesture-event evaluation feature is connected with the processing layer, allowing users to perform touchless interaction through predefined gestures whenever seamless file transfer is required.

The complete system is tested using different file-sharing scenarios to ensure smooth performance, low communication latency, accurate gesture recognition, and stable transfer results.

AirSwap has been evaluated to support intelligent touchless file transfer between devices using gesture-based interaction. It contains various modules such as real-time hand tracking, gesture recognition, wireless communication management, file-transfer synchronization, and interactive transfer monitoring. The modules were evaluated by testing the performance of each individual component, analyzing gesture-recognition accuracy, communication stability, transfer response time, and user interface usability.

The Gesture Recognition and Communication Engine was assessed thoroughly using multiple sets of real-time gesture interactions under different environmental conditions. The performance of the system was rated positively for accurately identifying predefined gestures such as *grab* and *throw* using hand landmark detection and movement validation mechanisms. To achieve a smooth user interaction experience, the processing pipeline incorporates gesture validation logic, which handles real-time interaction variations and prevents false gesture activation during communication processes.

The Wireless File Transfer Module utilizes socket-based networking protocols alongside communication synchronization mechanisms to establish stable connections and process transfer operations. File-sharing sessions, communication states, and transfer milestones are processed to deliver low-latency wireless transmission directly between connected devices. The transfer monitoring dashboard and gesture-feedback modules appeared to function as expected under testing conditions, indicating low communication delay and reliable transfer performance, which improves user convenience and allows seamless touchless file sharing between devices.

6. FUTURE WORK

Improving accessibility and intelligence of the AirSwap platform can be achieved through:

- **Advanced Gesture Recognition:** Enhancing gesture detection accuracy by implementing advanced deep learning models to support more complex gesture interactions.
- **Cross-Platform Compatibility:** Expanding system accessibility by supporting multiple platforms such as Windows, Linux, Android, and iOS devices.
- **Multi-Gesture Interaction:** Introducing additional gesture commands to improve touchless control and enable more flexible file-transfer operations.
- **Enhanced Security Features:** Leveraging encryption mechanisms and authentication models to improve transfer privacy and secure communication between connected devices.
- **Cloud and Remote Connectivity:** Developing optional cloud-based communication support to enable file transfer across remote networks and long-distance environments.
- **Adaptive Environmental Optimization:** Improving gesture recognition performance under poor lighting conditions, complex backgrounds, and varying camera angles.

7. CONCLUSION

AirSwap is a gesture-driven touchless file-transfer platform designed to make it easy for users to share files between devices without requiring physical interaction or complicated communication methods. The project successfully connects gesture recognition and wireless communication through a simple and user-friendly system, offering features such as real-time hand tracking, gesture-based file transfer, wireless peer-to-peer communication, and interactive transfer monitoring. It helps users perform seamless file sharing with improved convenience, reduced operational complexity, and faster communication between devices. While the platform has certain limitations such as lighting dependency, webcam constraints, and network availability, it still provides a strong foundation for future improvements. With advanced gesture models, enhanced security, and wider platform compatibility, AirSwap can become an efficient, user-friendly, and highly impactful touchless communication platform for modern digital environments.

9. REFERENCES

1. K. Y. Chen, M. S. Tsai, D. S. Yang and H. Y. Chen, "AirLink: Sharing Files Between Multiple Devices Using In-Air Gestures," in *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*, pp. 565–569, 2014.
2. S. K. Vasudevan and C. Vivek, "An Intelligent Attempt to Export Files into Cloud in Handheld Devices through Gesture Recognition," *Indian Journal of Science and Technology*, vol. 8, no. 35, pp. 1–6, 2015.
3. X. Zhang, Y. Li and H. Wang, "Real-Time Hand Gesture Recognition Using Computer Vision Techniques," *IEEE Access*, vol. 10, pp. 11234–11246, 2022, doi: 10.1109/ACCESS.2022.3145678.
4. T. Kim and J. Lee, "Touchless Human–Computer Interaction Based on Hand Gesture Recognition," *International Journal of Human–Computer Studies*, vol. 145, p. 102512, 2021, doi: 10.1016/j.ijhcs.2020.102512.
5. Google Developers, "MediaPipe Hands Framework Documentation," Available: <https://developers.google.com/mediapipe>
6. OpenCV Organization, "OpenCV: Open Source Computer Vision Library," Available: <https://opencv.org/>
7. Python Software Foundation, "Socket Programming in Python," Available: <https://docs.python.org/3/library/socket.html>
8. H. Kukenovs, "HAGRID: Hand Gesture Recognition Dataset," GitHub Repository, Available: <https://github.com/hukenovs/hagrid>
9. S. Mitra and T. Acharya, "Gesture Recognition: A Survey," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 37, no. 3, pp. 311–324, 2007, doi: 10.1109/TSMCC.2007.893280.
10. IEEE Xplore, "Hand Gesture Recognition Systems for Human–Computer Interaction," Available: <https://ieeexplore.ieee.org/>

11. AUTHOR'S:

Mr.Ubed Salim Momin is currently pursuing a Bachelor's Degree in Computer Science & Engineering at Ashokrao Mane Group Of Institution,vathar. She is publishing this paper under the esteemed guidance of Mr.Anil B. Desai.



Mr. Sujal Sunilkumar Shaha is currently pursuing a Bachelor's Degree in Computer Science & Engineering at Ashokrao Mane Group Of Institution,vathar. he is publishing this paper under the esteemed guidance of Mr.Anil B. Desai.



Miss.Nazneen Ibrahim Mulla is currently pursuing a Bachelor's Degree in Computer Science & Engineering at Ashokrao Mane Group Of Institution,vathar. She is publishing this paper under the esteemed guidance of Mr.Anil B. Desai.



Miss.Prachi Prakash Mahamuni is currently pursuing a Bachelor's Degree in Computer Science & Engineering at Ashokrao Mane Group Of Institution,vathar. he is publishing this paper under the esteemed guidance of Mr.Anil B. Desai.



Mr.Shriyash Rajendra Motkatte is currently pursuing a Bachelor's Degree in Computer Science & Engineering at Ashokrao Mane Group Of Institution,vathar. he is publishing this paper under the esteemed guidance of Mr. Anil B. Desai.



Mr.Anil B. Desai is currently working as an Assistant Professor in Computer Science & Engineering at Ashokrao Mane Group Of Institution,vathar. he has been guiding and mentoring students in various academic and research activities. he has provided his valuable guidance for the successful completion and publication of this paper.

