Gestkey: A Real-Time Contactless Gesture Based Typing System using Mediapipe Hand Landmark

1st Mr.P.Rajapandian, 2nd P.vinoth kumar

1Professor, Department of Computer Applications, Sri Manakula Vinayagar Engineering College (Autonomous), Puducherry 605008, India
rajapandianmca@smvec.ac.in
2Post Graduate student, Department of computer Applications, Sri Manakula Vinayagar Engineering College (Autonomous), Puducherry 605008, India
Vinothkumar27062@gmail.com

Abstract: The growth of touchless technologies has created opportunities for clean communication between humans and computers. This paper introduces GestKey, a system that lets you type using hand gestures and MediaPipe Hand Landmark detection. With GestKey users can input text without touching anything. The system uses a webcam to capture hand movements and find finger positions.It does this by using a -trained model that tracks hands.

By linking gestures to keyboard inputs users can interact with a keyboard on the screen. GestKey aims to make things more accessible.

It reduces the need for keyboards. Makes it easier to use in situations where touch input is not convenient or safe. The system processes hand landmark coordinates over time. It uses algorithms to recognize gestures and convert them into keystrokes quickly. It also includes techniques to filter out noise and stabilize gestures for accuracy. This helps prevent inputs.

Experimental results show that GestKey works well in lighting conditions and backgrounds. This makes it suitable for uses like technologies, public interfaces and smart environments. Overall GestKey provides a cost and user-friendly way to type without contact. It supports the growing field of gesture-based interaction systems, like GestKey.

Objectives:

1. Building a Real-Time Hand Gesture Recognition System

The main goal of this research is to create a strong and fast hand gesture recognition system using MediaPipes 21-point hand landmark detection model. Unlike input methods that need physical contact GestKey aims to capture and understand finger movements through a regular webcam. This eliminates the need for hardware. The system targets landmark tracking in different lighting conditions hand positions and skin tones.

2. Designing a Virtual Keyboard Mapping System

Another goal is to design a system that maps hand gestures to characters. This involves creating a model that divides the cameras view into zones. The system uses dwell-time thresholds or pinch-gesture triggers to confirm character selection. The design aims to make it easy for users to learn by using hand postures.

3. Evaluating Performance and Usability

The third goal is to test GestKeys performance through experiments. These experiments measure character recognition accuracy, input speed, activation rate and system latency. The system is compared to existing gesture-based and gaze-based typing solutions. User studies are conducted to gather feedback on usability.

4. Improving Hygienic Human-Computer Interaction

The overall goal of GestKey is to improve accessibility and hygiene in computing. The system helps people with upper-limb motor impairments and those working in environments. By showing that a vision-based typing interface can be practical and reliable using hardware this research provides a blueprint, for future touchless HCI systems.

Keywords: Gesture Recognition, MediaPipe, Hand Landmarks, Contactless Typing, Computer Vision, Real-Time Interaction, Virtual Keyboard, OpenCV, Assistive Technology, Human-Computer Interaction, Deep Learning, Finger Tracking, Dwell-Time Input, Gesture-Based Interface, Touchless Computing

1. Literature Review

1. S. Mitra and T. Acharya (2007) - This is a study about recognizing gestures. The people who wrote this say that recognizing gestures is about finding the meaning in the way people move their hands, face or body to communicate with machines. They think this is really important for making systems that're easy for people to use with computers especially for things like understanding sign language, virtual reality keeping an eye on peoples health and making smart things that people can interact with. Gesture recognition is a part of this because it helps computers understand what people are trying to say through gestures. Recognizing gestures is crucial, for developing human-computer interaction systems that use gesture recognition.

2. N. Neverova, C. Wolf, G. Taylor, and F. Nebout (2014) - This paper is about using learning to recognize gestures. The authors have come up with a system that looks at lots of things like how your hands move how your body moves and where you are in a room. They look at all these things at times and from different distances. This way the model can see the movements of your hands and the bigger movements of your body. The model can understand gesture recognition well by doing this. Gesture recognition is what this system is, about.

3. J. Redmon and A. Farhadi(2018) - This paper is about YOLOv3, a real-time object detection algorithm.It uses one network to find objects in a picture.The picture is divided into grids. The network predicts boxes around objects. What class they belong to.YOLOv3 is better than versions.It uses a complex network called Darknet-53. The algorithm also makes predictions at different scales.

This helps it detect objects and large objects accurately.YOLOv3 and object detection are really important.the YOLOv3 model is very good at finding objects.It helps computers understand pictures.Object detection, with YOLOv3 is fast and accurate.The YOLOv3 algorithm is used in applications.

4. F. Zhang, X. Zhu, and Y. Wang(2019) -
presented a study on hand gesture recognition using cameras showing how it can help people interact with computers in a more natural way. The study looked at different methods, including old and new techniques like using images to find and understand hand movements.It also talked about problems like lighting,busy backgrounds and needing fast processing. These findings are a starting point for making new systems that use hand gestures, like typing without touching anything such as the GestKey system. The study covered a lot of ground on vision-based hand gesture recognition.

5. A. K. Sahoo and S. Ari(2020) - People want to find ways to work with computers. They have found a way to control the mouse and keyboard with movements of their hands. This system uses a camera to see how people move their hands and turns these movements into things like moving the cursor and clicking and typing on the keyboard. The study shows that people can do things on the computer in a flexible way without having to touch anything. This is really useful when people cannot touch devices or do not want to touch them.

The work shows that using hand movements to interact with computers is an idea. It also gives us ideas about how to design systems that're easy to use and do not need people to touch them. This is similar to the idea, behind the GestKey typing solution

2. Research Methodology

The research methodology for the proposed GestKey system is designed to systematically develop and evaluate a real-time, contactless gesture-based typing interface using MediaPipe Hand Landmark detection. The study begins with problem identification, focusing on the limitations of traditional input devices in terms of hygiene, accessibility, and flexibility. Based on this, a vision-based interaction model is conceptualized where human hand gestures are captured through a standard webcam and translated into virtual keystrokes. The system architecture is then designed to integrate computer vision, gesture recognition, and user interface modules into a cohesive framework.

The implementation phase involves utilizing the MediaPipe Hands module to detect and track 21 key landmarks of the human hand in real time. These landmarks provide precise spatial coordinates that are used to interpret finger movements and gestures. A mapping algorithm is developed to convert specific hand gestures into corresponding keyboard inputs. To ensure usability, a virtual keyboard layout is designed and overlaid on the screen, allowing users to interact without physical contact. The system is programmed using Python along with libraries such as OpenCV for image processing and real-time video capture.

For data collection and testing, multiple users interact with the system under varying environmental conditions, including different lighting levels and backgrounds. The performance of the system is evaluated based on metrics such as gesture recognition accuracy, response time, typing speed, and error rate. Experimental validation is carried out to assess robustness and reliability, ensuring that the system performs consistently in real-world scenarios. The results are analyzed to identify strengths and limitations, and necessary optimizations are applied to improve efficiency.

3. Current Scenario of Gestkey

The current scenario of human–computer interaction is rapidly evolving toward more natural, intuitive, and contactless methods, especially with the increasing demand for touch-free technologies in public and personal environments. Traditional input devices such as physical keyboards and touchscreens, while widely used, present limitations in terms of hygiene, accessibility, and flexibility. This became particularly evident during global health concerns, where minimizing physical contact with shared surfaces became essential. As a result, there has been a growing

ISSN: 2583-6129
DOI: 10.55041/ISJEM07870

interest in alternative input modalities that can offer safer and more seamless interaction experiences.

In recent years, gesture recognition technology has gained significant attention as a promising solution to these challenges. Advancements in computer vision and machine learning have enabled systems to accurately detect and interpret human hand movements in real time. Frameworks like MediaPipe have further accelerated this development by providing efficient and robust hand landmark detection models that can operate even on low-cost devices. These technologies allow users to interact with digital systems using simple hand gestures, eliminating the need for physical touch and making interaction more natural and intuitive.

Despite these advancements, existing gesture-based typing systems still face several challenges, including limited accuracy, high latency, and difficulty in adapting to different lighting conditions and backgrounds. Many current solutions also lack user-friendly interfaces or require complex hardware setups, which restricts their practical usability. Additionally, there is a need for systems that can support real-time typing with minimal errors while maintaining responsiveness and ease of use.

4. Challenges in Gestkey

In the development of GestKey, a real-time contactless gesture-based typing system built using MediaPipe Hand Landmark detection, several practical and technical challenges emerge that significantly influence system performance and usability. One of the primary concerns is the accuracy and consistency of hand landmark detection under varying environmental conditions. Changes in lighting, complex backgrounds, and camera quality can affect the robustness of hand tracking, leading to misinterpretation of gestures. Even though MediaPipe provides efficient real-time tracking, slight occlusions or overlapping fingers can cause incorrect landmark estimation, which directly impacts typing precision

.

Another critical challenge lies in gesture ambiguity and user variability. Different users naturally perform gestures in slightly different ways, including variations in hand size, finger movement speed, and orientation. Designing a system that generalizes well across such differences without extensive personalization is difficult. Moreover, certain gestures may appear similar to the system, especially when distinguishing between adjacent keys or subtle finger movements, increasing the likelihood of false positives or missed inputs. This becomes more problematic when aiming for fast typing speeds, as rapid gestures can reduce detection accuracy.

Latency and real-time responsiveness also pose significant constraints. For a typing system to feel natural, it must process gestures and produce outputs almost instantaneously. However, continuous video frame processing, landmark extraction, and gesture classification require computational resources, which can introduce delays, particularly on low-end devices. Balancing computational efficiency with high accuracy is therefore a key design challenge in GestKey systems.

User fatigue and ergonomics present another important limitation. Unlike physical keyboards, gesture-based typing requires users to hold their hands in mid-air for extended periods, which can lead to discomfort or fatigue, commonly referred to as the “gorilla arm” effect. This affects long-term usability and user acceptance of the system. Additionally, maintaining consistent hand positioning within the camera frame can be tiring and inconvenient in practical scenarios.

5. Existing System

The existing systems for gesture-based typing and virtual input primarily rely on traditional hardware devices such as physical keyboards, touchscreens, or stylus-based interfaces, which require direct physical interaction. In recent years, some vision-based approaches have been introduced, where hand gestures are detected using cameras; however, these systems often depend on complex setups, specialized sensors like depth cameras, or wearable devices such as data gloves. Such requirements increase cost and reduce accessibility for everyday users. Moreover, earlier gesture recognition systems frequently suffer from limited accuracy, especially under varying lighting conditions and complex backgrounds, making them less reliable for real-time applications.

In addition, many existing solutions lack real-time responsiveness and seamless user interaction. They often experience latency in gesture recognition, which affects typing speed and user experience. Some systems also require extensive calibration or predefined environments to function correctly, restricting their usability in dynamic or uncontrolled settings. Although frameworks like MediaPipe have improved hand tracking capabilities, earlier implementations did not fully utilize such advanced models, resulting in reduced precision in detecting finger landmarks and gestures.

Furthermore, current virtual keyboard systems are generally not optimized for contactless interaction, as they are designed for touch-based input. This makes them unsuitable for applications where hygiene, accessibility, or hands-free operation is important, such as in public interfaces or assistive technologies. Overall, the existing systems highlight the need for a more efficient, cost-effective, and accurate contactless gesture-based typing

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 05 Issue: 06 | June – 2026 DOI: 10.55041/ISJEM07870
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

solution that can operate reliably in real-time without requiring additional hardware or complex configurations.

6. Proposed system

The proposed system, GestKey: A Real-Time Contactless Gesture-Based Typing System, introduces an innovative approach to human–computer interaction by eliminating the need for physical keyboards and enabling text input through intuitive hand gestures. The system is built upon the capabilities of MediaPipe Hand Landmark detection, which allows accurate real-time tracking of finger positions using a standard camera without requiring specialized hardware.

In this system, a webcam continuously captures the user's hand movements, and the captured frames are processed to detect key hand landmarks such as fingertips, joints, and palm positions. These landmarks are then mapped into a virtual coordinate space where a digital keyboard layout is overlaid. When a user performs specific gestures—such as hovering, tapping, or pinching—the system interprets these movements as keystrokes. This interaction model ensures a natural and touch-free typing experience while maintaining efficiency and usability.

The proposed system incorporates a robust gesture recognition module that distinguishes between intentional typing gestures and unintended movements. By applying filtering techniques and threshold-based detection, the system reduces noise and enhances accuracy even under varying lighting conditions and complex backgrounds. Additionally, temporal smoothing algorithms are used to stabilize cursor movement and prevent false key activations, ensuring a seamless user experience.

To improve usability, the system provides real-time visual feedback by displaying the detected hand landmarks and highlighting selected keys on the virtual keyboard interface. This feedback loop helps users understand system responses instantly, thereby reducing the learning curve. The design also supports customizable keyboard layouts and gesture sensitivity settings, making it adaptable for different users, including individuals with physical disabilities.

7. Architectural Design

The architectural design of **GestKey: A Real-Time Contactless Gesture-Based Typing System Using MediaPipe Hand Landmark** is structured as a layered and modular framework that enables seamless interaction between hardware input, gesture recognition, and virtual typing output. At the foundational level, the system begins with a video acquisition module, where a standard webcam continuously captures real-time hand movements. This input stream acts as the primary data source, and its quality directly influences the overall system performance. The captured frames are then forwarded to the processing layer, where image preprocessing techniques such as resizing, normalization, and color space conversion are applied to ensure consistency and efficiency in further computations.

At the core of the architecture lies the hand tracking and landmark detection module powered by MediaPipe. This component is responsible for identifying the user's hand and extracting 21 key landmark points representing finger joints and palm positions. These landmarks form the basis for interpreting gestures. The system leverages spatial relationships between these points to understand finger positions, orientations, and movements in real time. By continuously tracking these landmarks across frames, the system maintains temporal consistency, which is crucial for accurate gesture recognition.

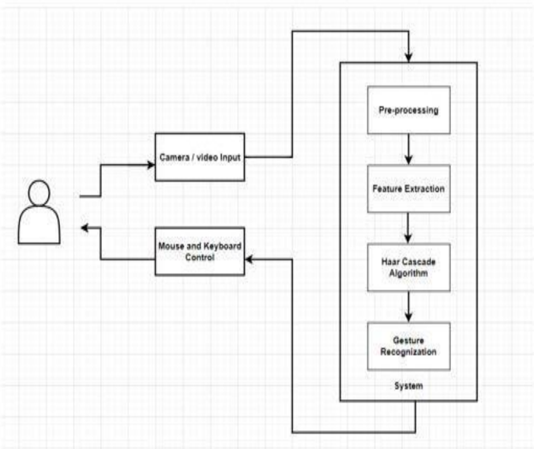

Fig. 1: Architecture Diagram

Front end (user interface)

The front-end of GestKey is designed to provide a simple, interactive, and real-time user experience where users can type using hand gestures instead of a physical keyboard. The UI connects with the
backend powered by MediaPipe for hand landmark detection.

The interface is mainly built using:

- Python GUI frameworks (Tkinter / PyQt) OR
- Web-based UI (HTML, CSS, JavaScript)

Backend (Server-side)

The backend of the GestKey system is responsible for processing hand gesture data captured from the webcam, interpreting the gestures, and converting them into corresponding keyboard inputs. It acts as the core logic layer that connects gesture recognition with text output. The backend is mainly implemented using Python, integrating libraries like OpenCV and MediaPipe for real-time hand tracking and gesture detection.

Database:

The GestKey system primarily focuses on real-time gesture recognition and typing; hence, it does not require a heavy traditional database. However, a lightweight and efficient data storage mechanism is used to manage user inputs, system configurations, and interaction logs.
The system can use:

- **SQLite** – for lightweight, local storage
- **Firebase Realtime Database / Firestore** – for cloud-based storage (optional)
- **MySQL** – if deployed in a large-scale environment

7. Modules

1. Video Capture Module

This module is responsible for capturing real-time video input from the system's webcam. It continuously reads frames and sends them for further processing. The quality and frame rate of the video directly affect system performance. Proper lighting and positioning of the camera are important to ensure accurate hand detection.

2. Hand Detection Module

This module detects the presence of a hand in the captured video frames using MediaPipe. It identifies the hand region and isolates it from the background. The module works efficiently even under different lighting conditions. It ensures that only valid hand inputs are passed to the next stage.

3. Hand Landmark Extraction Module

This module extracts key points (landmarks) of the detected hand using MediaPipe's hand tracking model. It identifies 21 important points such as fingertips, joints, and palm positions. These landmarks are used to understand the structure and movement of the hand. The extracted data is crucial for gesture recognition accuracy.

4. Gesture Recognition Module

This module interprets the hand landmarks to identify specific gestures. Different finger positions and movements correspond to different characters or commands. Algorithms are used to map these gestures to predefined actions. It plays a key role in converting physical gestures into meaningful inputs.

5. Virtual Keyboard Interface Module

This module displays a virtual keyboard layout on the screen. Users interact with the keyboard using hand gestures instead of physical touch. The interface is designed to be simple and user-friendly. It highlights keys when a gesture is detected, improving user interaction and feedback.

6. Gesture-to-Text Conversion Module

This module converts recognized gestures into corresponding text characters. Once a gesture is identified, it is mapped to a specific key or command. The system ensures minimal delay for real-time typing experience. It helps in forming words and sentences from continuous gestures.

7. Output Display Module

This module displays the typed text on the screen in real time. It acts like a text editor where users can see their input clearly. The output is updated continuously as gestures are detected. This module ensures proper formatting and readability of the typed content.

8. Error Handling and Noise Reduction Module

This module handles incorrect or unintended gestures during typing. It reduces noise caused by hand movement or background interference. Techniques like smoothing and thresholding are applied to improve accuracy. It ensures reliable and consistent system performance.

9. System Integration Module

This module integrates all the individual components into a single working system. It ensures smooth communication between video capture, processing, and output modules. Proper synchronization is maintained to avoid delays.

This module is essential for achieving real-time performance.

8. Screen Shots

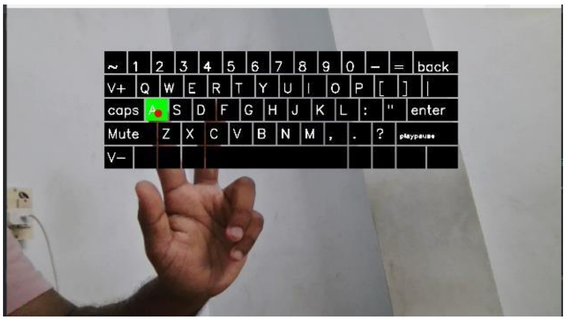

Fig:1 Gest key

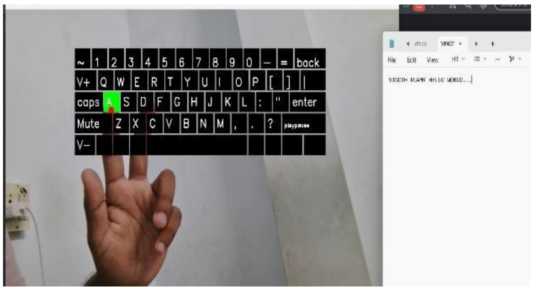

Fig:2 Gesture based typing system

10. key features and benefits of Gestkey

1. Real-Time Hand Gesture Recognition

GestKey uses advanced computer vision techniques to detect and track hand movements instantly. By leveraging MediaPipe's hand landmark model, it identifies finger positions with high accuracy. This allows the system to convert gestures into typing inputs without delay. As a result, users experience smooth and responsive interaction similar to a physical keyboard.

2. Contactless Typing Interface

The system eliminates the need for physical touch by enabling typing through mid-air gestures. Users can interact with a virtual keyboard simply by moving their fingers in front of a camera. This reduces dependency on hardware devices like keyboards and touchscreens. It is especially useful in environments where hygiene and cleanliness are critical.

3. Hand Landmark Detection Technology

GestKey uses 21 key hand landmarks to precisely detect finger joints and movements. This detailed tracking helps in recognizing complex gestures accurately. The system continuously monitors hand positions to ensure consistent performance. This feature forms the core intelligence behind gesture interpretation.

4. Virtual Keyboard Simulation

A virtual keyboard is displayed on the screen, allowing users to "press" keys using gestures. Each finger movement corresponds to a specific key input. The interface is designed to mimic a traditional keyboard layout for easy usability. This makes the transition from physical to virtual typing seamless.

5. Adaptive to Lighting and Background Conditions

GestKey is designed to work under different lighting conditions and backgrounds. It uses robust detection algorithms to maintain accuracy even in less-than-ideal environments. This adaptability ensures reliable performance in real-world situations. Users can operate the system without needing a controlled setup.

Benefits

1. Enhanced Hygiene and Safety

Since GestKey operates without physical contact, it significantly reduces the risk of spreading germs. This is highly beneficial in public places like hospitals, ATMs, and kiosks. Users can interact with systems without touching shared surfaces. It promotes a safer and cleaner environment.

2. Accessibility for Disabled Users

The system provides an alternative typing method for individuals with physical disabilities. Users who cannot use traditional keyboards can rely on gestures for communication. This improves inclusivity and accessibility in digital systems. It empowers users with more independence.

3. Increased Flexibility and Mobility

GestKey allows users to type from anywhere without needing a physical keyboard. It can be used in different environments such as offices, homes, or public spaces. The portability of the system makes it highly flexible. Users can work comfortably without being restricted by hardware.

4. Reduced Hardware Dependency

By replacing physical keyboards, GestKey minimizes the need for additional input devices. This reduces maintenance costs and hardware failures. It also simplifies system setup and reduces clutter. Organizations can benefit from cost-effective solutions.

5. Innovative Human-Computer Interaction

GestKey introduces a modern and futuristic way of interacting with computers. It enhances user engagement through natural hand movements. This makes computing more intuitive and interactive. It represents a step forward in touchless technology.

11. Opportunities Related to Gest key

The proposed GestKey system, a real-time contactless gesture-based typing solution using MediaPipe Hand Landmark detection, presents significant opportunities across multiple domains. It can be effectively applied in assistive technologies to support individuals with physical disabilities, enabling them to communicate without traditional input devices. In public environments such as kiosks, ATMs, and healthcare systems, it enhances hygiene by eliminating the need for physical contact, which is especially valuable in post-pandemic scenarios. The system also has potential in smart homes and IoT ecosystems, allowing users to control devices through intuitive hand gestures. In the field of virtual and augmented reality, GestKey can improve user interaction by providing a natural and immersive typing experience. Furthermore, it can be integrated into educational tools, gaming interfaces, and remote working platforms, making human-computer interaction more flexible, accessible, and efficient. Overall, the system opens pathways for innovation in touchless technology and next-generation user interfaces.

12. conclusion

The GestKey project represents a significant advancement in contactless, gesture-based human-computer interaction. By leveraging Google's MediaPipe Hand Landmark detection framework, the system achieves real-time, accurate recognition of hand gestures mapped to a full alphanumeric keyboard vocabulary, using only standard webcam hardware available on any modern laptop or desktop computer.

The hybrid recognition engine combining rule-based geometric analysis with machine learning classification provides both interpretability and robustness. The dwell-time confirmation mechanism effectively eliminates false keystrokes from natural hand movement, while the virtual keyboard overlay creates an intuitive and responsive typing interface that users can learn and operate effectively with modest practice.

GestKey is designed with accessibility at its core. By enabling text entry without physical contact, it opens computing to users who cannot use traditional keyboards due to motor impairments, and extends the utility of computers to environments where physical keyboards are impractical or unsafe. The system's on-device architecture ensures complete data privacy, while its open-source technology stack makes deployment cost-effective and widely accessible.

Looking forward, GestKey holds considerable potential for extension into dynamic gesture recognition for continuous typing, integration with augmented and virtual reality environments, multilingual gesture vocabularies, and AI-powered personalized gesture adaptation. The modular architecture of the system is specifically designed to accommodate these future developments with minimal re-engineering effort.

In summary, GestKey demonstrates that real-time, accurate, contactless gesture-based typing is achievable on commodity hardware using state-of-the-art hand landmark detection. It bridges the gap between research-grade gesture recognition and practical, deployable assistive and general-purpose input technology, establishing a strong foundation for next-generation touchless human-computer interaction.

References

References

1. S. Mitra and T. Acharya (2007). Gesture Recognition: A Survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C, 37(3), 311–324.

2. C. Keskin, F. Kirac, Y. E. Kara, and L. Akarun (2012). Hand Pose Estimation and Hand Shape Classification Using Multi-Layered Randomized Decision Forests. ECCV 2012.

3. F. Zhang, V. Bazarevsky, A. Vakunov, et al. (2020). MediaPipe Hands: On-device Real-time Hand Tracking. arXiv:2006.10214

4. A. Neverova, C. Wolf, G. W. Taylor, and F. Nebout (2016). ModDrop: Adaptive Multi-Modal Gesture Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence.

5. R. Jiang, A. Tanveer, and A. Roy Chowdhury (2021).Gesture Typing in Augmented Reality: Design and Evaluation. IEEE VR 2021.

6. P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz (2016). Online Detection and Classification of Dynamic Hand Gestures with Recurrent 3D Convolutional Neural Networks. CVPR 2016.

7. G. Rautaray and A. Agrawal (2015). Vision Based Hand Gesture Recognition for Human Computer Interaction: A Survey. Artificial Intelligence Review, 43(1), 1–54.

8. V. I. Pavlovic, R. Sharma, and T. S. Huang (1997). Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review. IEEE Transactions on Pattern Analysis and Machine Intelligence.

9. MediaPipe Documentation (2024). MediaPipe Hands Solution. Retrieved from https://mediapipe.dev/solutions/hands

10. OpenCV Documentation (2024). OpenCV 4.x Python API Reference. Retrieved from https://docs.opencv.org

11. Scikit-learn Documentation (2024). Scikit-learn Machine Learning in Python. Retrieved from https://scikit-learn.org

12. F. Chollet (2021). Deep Learning with Python (2nd ed.). Manning Publications.

13. S. Thrun, W. Burgard, and D. Fox (2005). Probabilistic Robotics. MIT Press.

14. W. T. Freeman and M. Roth (1995). Orientation Histograms for Hand Gesture Recognition. IEEE International Workshop on Automatic Face and Gesture Recognition.

15. D. Eck (2022). Gesture-Based Text Input: A Systematic Review. ACM Computing Surveys, 54(8).

16. A. Boyali and N. Kavakli (2013). Spectral Collaborative Representation-Based Classification for Hand Gestures Recognition on a Custom Set. Sensors.

17. Y. LeCun, Y. Bengio, and G. Hinton (2015). Deep Learning. Nature, 521, 436–444.

18. Google LLC (2023). MediaPipe Framework Documentation. Retrieved from https://google.github.io/mediapipe

19. Python Software Foundation (2024). Python 3.x Documentation. Retrieved from https://docs.python.org

20. K. He, X. Zhang, S. Ren, and J. Sun (2016). Deep Residual Learning for Image Recognition. CVPR 2016.