DOI 10.15622/ia.25.2.5

N.V. HUNG, P.D. HUYNH, M.V. TUNG, N.V. VU, N.P. DAT

CLVM: A HYBRID DEEP LEARNING FRAMEWORK FOR CONTACTLESS VIRTUAL MOUSE CONTROL

Nguyen Viet Hung, Phi Dinh Huynh, Ma Van Tung, Nguyen Van Vu, Nguyen Phu Dat. **CLVM: A Hybrid Deep Learning Framework for Contactless Virtual Mouse Control.**

Abstract. In the era of rapid digital transformation and the growing prevalence of artificial intelligence, enabling natural, seamless, and contactless human-computer interaction has become a critical priority across various domains. This paper presents a novel deep learning-based model for virtual mouse control using hand gestures, termed CLVM (CNN-LSTM Virtual Mouse). The proposed system introduces a hybrid architecture that integrates three powerful components: (1) MediaPipe for efficient and real-time hand landmark detection; (2) a Convolutional Neural Network (CNN) for spatial feature extraction; and (3) a Long Short-Term Memory (LSTM) network for temporal dynamics modeling, enhancing the system's ability to recognize gestures continuously and accurately over time. Unlike traditional models, CLVM is designed to maintain robust performance in real-world environments, particularly under conditions of inconsistent lighting and cluttered backgrounds. The system also provides low latency and high responsiveness and can be deployed effectively on resource-constrained devices, making it practical for widespread adoption. Experimental results demonstrate that CLVM achieves a high accuracy (99.88%) while reducing the loss to 0.38, significantly outperforming conventional gesture recognition methods. These findings highlight CLVM's potential to serve as a reliable, scalable, and efficient solution for natural gesture-based interaction. It offers a valuable step forward in the development of intelligent, user-friendly interfaces for contactless control applications.

Keywords: computer vision, contactless interface, hand landmarks, machine learning, MediaPipe, virtual mouse.

1. Introduction. A major area of research in fields such as artificial intelligence (AI), computer vision, and Human-Computer Interface (HCI) is the development of more intuitive and natural ways for people to interact with computer, as information technology and computer science continue to advance at a rapid pace [1-5]. Even though conventional peripherals such as physical keyboards and mice are still widely used, they have limitations in certain situations. Common scenarios include remote control in intelligent systems, non-contact settings such as industrial clean rooms, medical surgeries, or epidemic situations where contact must be minimized to reduce the risk of infection, and support systems for individuals with mobility impairments to enable easy interaction.

Therefore, gesture control systems – especially virtual mouse control systems that use hand gestures – have become a viable alternative to conventional peripherals [6-9]. However, there are still many significant technical obstacles to overcome before real-time hand gesture recognition systems can be implemented. These difficulties include maintaining high gesture recognition accuracy, reducing latency to meet real-time requirements

and improving the system's ability to adapt to a variety of environmental factors, such as changing lighting conditions, cluttered backgrounds, and variations of user hand sizes and shapes. Soving these issues requires not only strong algorithmic solutions but also close integration of hardware and software technologies, along with the development of deep learning architectures tailored for this purpose.

This paper proposes a new approach based on the integration of three key technological components to address the limitations of current virtual mouse control systems. The first element is the MediaPipe library [10, 11], a powerful and wildly-used computer vision framework that enables efficient and accurate real-time hand landmark detection Convolutional Neural Network (CNN) [12], which focuses on processing and interpreting spatial features of image data. The third element is a Long-Short-Term Memory network (LSTM) [13], which enables the system to recognize dynamic patterns in hand gestures by learning and modeling temporal sequences.

By integrating these three technologies into a single architecture, the system is able to provide users with a seamless and efficient interactive experience, achieving high gesture recognition accuracy and maintaining uninterrupted operation even in challenging real-world scenarios. To enhace practical applicability domains such as online education, remote healthcare, factory automation, and user support, the primary objective of the research is to create a virtual mouse control model that can be readily deployed on standard devices such as personal computers (PCs) or laptops with integrated webcams.

Experimental results show that the proposed model outperforms baseline methods in a number of aspects, including high-precision classification, inference time, and adaptability to a variety of usage scenarios.

The objectives of this research can be summarized in three main directions. First, we seek to place CLVM within the broader landscape of state-of-the-art (SOTA) gesture recognition systems. While many existing approaches rely on rule-based designs or use either CNNs or LSTMs in isolation, our framework combines the two to model both spatial and temporal aspects of hand gestures. This integration is intended to improve robustness and maintain continuity during real-time interaction. Second, beyond technical accuracy, we consider the practical aspects of user experience. Although this work primarily emphasizes performance metrics (accuracy, latency, stability), preliminary informal trials indicate that the defined gestures are intuitive and easy to perform, and we identify structured user evaluations as a critical next step. Third, we acknowledge that the current system supports only four fundamental gestures (Move, Scroll, Pause, Start). This deliberate choice allows us to

rigorously validate feasibility under real-world conditions. Simultaneously, it highlights the path for future work, where the gesture vocabulary will be expanded to include more complex actions such as clicking, dragging, and zooming, toward building a fully functional replacement for traditional mouse devices. Together, these goals establish CLVM not merely as a proof-of-concept, but as a robust and extensible framework that addresses both algorithmic and usability challenges in natural human-computer interaction.

The proposed system is illustrated in Figure 1. It shows the overall architecture of the contactless virtual mouse system.

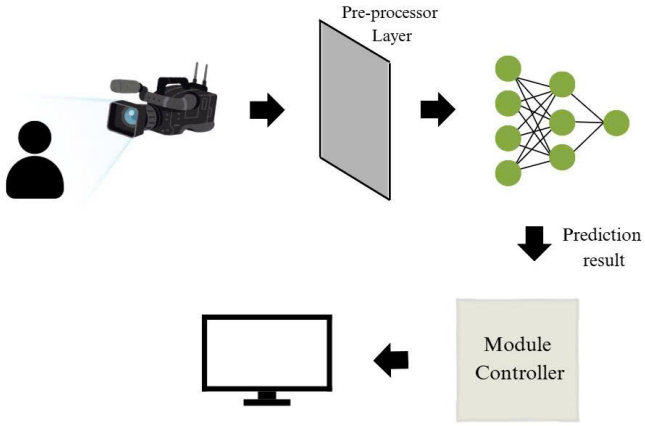


Fig. 1. System overview diagram

Initially, a camera device (such as a head-mounted camera or smart glasses) captures the user's hand movements or gestures. The captured data are then passed through a pre-processing layer, which is responsible for normalizing and extracting relevant features. This processed data is fed into a machine learning model or neural network that analyzes the input and produces a prediction representing the intended user action. The prediction is sent a module control, which interprets the output and converts it into corresponding mouse commands. Finally, these commands are executed on the computer system, enabling intuitive and touchless control of a virtual mouse cursor. Our target users include accessibility device users and operators in sterile or dusty workplaces, as well as public kiosk users and videoconferencing presenters. By eliminating contact with shared hardware, CLVM reduces the risk of cross-contamination and mechanical wear without sacrificing responsiveness.

Based on the proposed system architecture, this paper introduces the following key innovations:

– We propose CLVM (CNN-LSTM Virtual Mouse) – a novel optimization framework for gesture-based virtual mouse control – by integrating convolutional and recurrent neural networks. This hybrid design enables efficient extraction of both spatial features (via CNN) and temporal dynamics (via LSTM), overcoming the limitations of traditional single-model approaches.

– The proposed three-stage pipeline integrates: (1) MediaPipe to detect hand landmarks in real time, (2) a CNN to learn spatial gesture representations, and (3) an LSTM to model sequential patterns for robust, continuous recognition.

– Experimental results show that CLVM outperforms baseline models (CNN-only, LSTM-only, and conventional rule-based approaches) in accuracy and robustness. Specifically, CLVM achieves an average gesture recognition accuracy of 99.88% under diverse conditions, surpassing prior work that typically reports accuracies ranging from 99.08% to 99.80% in controlled environments. The system maintains performance even in low light and cluttered backgrounds, demonstrating strong generalization.

The remainder of this paper is organized as follows. Section 2 reviews related work to establish the context and highlight the gap our study addresses. In Section 3, we introduce our proposed method, CLVM (CNN-LSTM Virtual Mouse), including the overall system architecture and its key components. Specifically, we describe the use of 1D convolutional layers for local feature extraction, LSTM networks for temporal sequence modeling, and the final integration and output stage. Section 4 presents the performance evaluation of our method based on relevant benchmarks. Finally, Section 5 concludes the paper and outlines potential directions for future work.

2. Related work. This section provides a summary of previous and ongoing research projects that are directly relevant to our study. Within the field of HCI, these studies focus on developing touchless interaction solutions and virtual mouse control systems with the goal of improving the user experience.

One study [14] proposed a computer vision-based virtual mouse system that controls the cursor through hand movements instead of a conventional physical mouse. This system, implemented in Python, uses the OpenCV library to recognize and track the user's hand motions in real time via a camera [15-17]. Its primary features include double-click, right-click, and left-click functionalities, providing the user a variety of control options.

Another study [18] proposed a system that extends the virtual mouse control approach by combining voice commands with hand gestures to provide more convenient and natural interaction. The system uses the pyttsx3 package

for voice synthesis, along with MediaPipe and OpenCV for hand gesture recognition [11, 19, 20], allowing users to execute commands precisely via voice.

A further study [21] developed a real-time camera-based mouse control system that integrates voice assistance with hand gestures. The aim of this system is to enhance the efficiency, comfort, and naturalness of human-computer interaction. Its optimized algorithms, along with natural language control via an integrated voice assistant, significantly enhace the user experience.

Notably, recent studies on computer mouse control have focused on using artificial intelligence (AI) focused, leading to smoother and more natural mouse movements. Most earlier studies [9, 14] frequently employ the MediaPipe library for hand gesture recognition often under specific conditions.

Simultaneously, studies in related areas, such as virtual reality [22, 23] or frame estimation [24, 25], are also rapidly evolving with the aim of helping to accelerate processing speed and reduce system resource consumption. This trend indicates growing attention to spatial development.

Additionally, significant advancement has been made in this area through Virtual Whiteboard (VW) technology, which was introduced in [23, 26, 27]. VW technology translates hand motions into digital tasks such as writing, sketching, and form editing using computer vision and machine learning algorithms. The system efficiently meets the demands of note-taking and illustration in real time by supporting a variety of drawing tools, enabling fast PDF export, and automatically converting handwriting into typed text.

Another study [28] focuses on creating an interactive presentation. This system uses OpenCV and CvZone to recognize hand gestures and enables users to annotate content and control slides directly via a webcam [28-30]. In lectures or seminars, this approach enhances audience interaction and presenter flexibility by eliminating the need for keyboards or remote controls.

In a study [31], a virtual mouse system was proposed that uses a webcam and deep learning models to recognize hand movements and translate them into mouse commands without physical contact with the goal of increasing accessibility. By reducing contact, this method not only reduces the risk of infection but also assists people with disabilities who can use their hands and arms.

Further broadening the application range, a hands-free contactless virtual mouse system [32] combines facial gestures (e.g., mouth and eyes) to improve accuracy and expressiveness for users with limited hand function. The system fully supports keystrokes, mouse clicks, and page scrolling, thereby enhancing user independence. It employs advanced image processing techniques to mitigate errors caused by varying lighting.

To improve information security and prevent unauthorized access, research [33] combines virtual mouse technology with cutting-edge security strategies such as biometric authentication and encryption. This system guarantees the security of sensitive user data while also increasing accessibility in human-machine interaction.

The Enhanced Hand Tracking (EHT) model [34] utilizes the YOLOV8 architecture to improve the accuracy of real-time hand tracking and gesture recognition. This system outperforms MediaPipe, adapts to individual users' gesture patterns, supports multi-user interaction, and is tailored for virtual reality environments. Thus, EHT promises to provide a smooth, natural, and highly customized interaction experience in digital settings.

In addition to visual-only cursor control, previous works have explored bimodal interfaces that combine computer vision with automatic speech recognition to separate spatial pointing from command semantics. For example, Karpov et al. integrate head pointing for 2D cursor movements with voice commands and evaluate the interface using the ISO 9241-9 methodology, demonstrating the practicality of contactless pointing with speech [35]. Previous studies compare two speech and gesture systems (ICANDO, MOWGLI) and report Fitts' law experiments for multimodal pointing, highlighting the fusion time and user performance trade-offs [36]. More broadly, surveys on multimodal HCI promote combining modalities to reduce errors and improve robustness in real-world settings [37]. In our work, CLVM is positioned as a vision-centric module that leverages MediaPipe hand landmarks for real-time control and can be extended with voice commands for mode switching or click/drag actions in noisy or hands-busy situations [38].

3. Proposed Method – CLVM. This section presents the CLVM model, a contactless virtual mouse system that combines a Convolutional Neural Networks (CNN) for hand gesture recognition and Long Short-Term Memory (LSTM) network for temporal gesture tracking. The system enables real-time cursor control using hand movements, offering high accuracy and adaptability across various lighting conditions. CLVM provides a natural, intuitive interaction method, especially beneficial for touchless interfaces and accessibility applications.

3.1. System Model. Developing a hand gesture recognition system is a complex task that requires in-depth knowledge, system design abilities, and sophisticated data processing skills. To ensure precise, reliable, and consistent recognition of pre-defined key hand gestures, such as Move, Scroll, Pause, and Start, the system is meticulously designed, undergoes several implementation stages, and is carefully optimized.

Every phase of the development process, including preliminary data collection, data preprocessing, model design, training, parameter tuning, and final performance evaluation step, is crucial and interdependent. Creating a sufficiently rich dataset that reflects realistic variations in gestures, users, and environmental conditions is necessary for the data collection step. The suitable algorithm must be chosen and optimized during the model design and training phase to achieve the best accuracy and guarantee real-time performance. Finally, the evaluation process shows the system's efficacy and dependability in real-world scenarios, where a variety of intricate and unpredictable factors exist, in addition to testing the system's accuracy on the training data.

The proposed hand gesture recognition system achieves high effectiveness, accuracy, and robustness in real-world human-machine interaction through tight integration and coordinated operation across all core components. From the initial stages of data collection and preprocessing to feature extraction, gesture classification, and real-time deployment, each phase has been meticulously crafted, thoroughly refined, and experimentally assessed. This well-structured and unified framework enables the system to deliver consistent performance not only in laboratory settings but also in unpredictable, real-world environments, serving as a dependable and intuitive interface for natural human-computer interaction.

Each step of the system is illustrated in Figure 2, starting with raw video data collection, a fundamental step to the entire model-building and training process. The goal of this step is to collect a sufficiently large, rich, and diverse dataset to train a machine learning model that can reliably and accurately recognize common hand gestures.

Figure 3 visualizes the gestures – *Move*, *Scroll*, *Pause*, and *Start* – as a distribution of feature data points in a two-dimensional feature space. Each point represents a particular recording session from the processed video input, and the color indicates the associated behavior label. The inherent complexity of natural behavior recognition results in some overlap between gesture classes in the scatter plot underscoring the significance of deep learning models in identifying and extracting latent features.

To improve the model's generalizability, video data must be collected under a variety of conditions, including different lighting levels (natural, low, artificial) and camera angles (frontal, oblique, top-down).

This approach enables the model to better adapt to circumstances outside the training environment, reducing its dependency on specific conditions and thereby increasing the system's practicality and reliability in real-world scenarios. Furthermore, the model's capacity to process new data outside

of the training environment is enhanced when the training data is rich and representative of real-world variations.

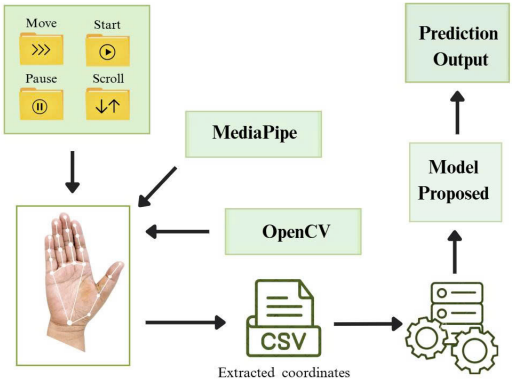


Fig. 2. Model construction diagram

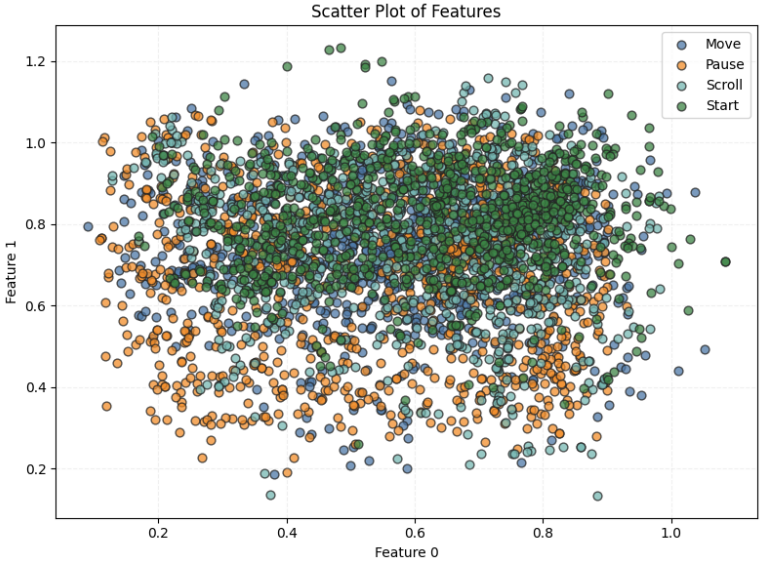
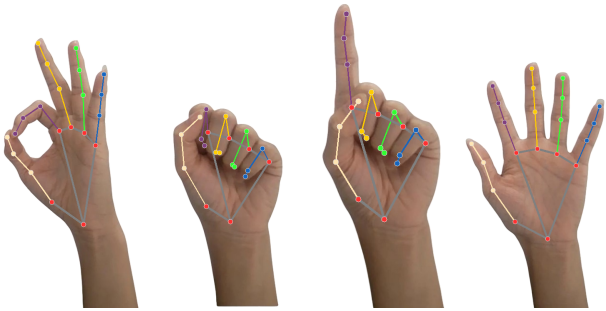


Fig. 3. Data Partition of Model

For this study, we constructed a dedicated dataset tailored to the requirements of the contactless virtual mouse application. The dataset comprises 521 short video clips, each corresponding to a single, well-defined gesture instance. Data were collected from four volunteers (three men and one woman), all right-handed. The participants exhibited natural variations in hand size, shape, and skin tone, thereby contributing to the diversity and robustness of the dataset. Each participant recorded at least 30 clips for each of the four gesture categories – Move, Scroll, Pause, and Start – resulting in a nearly uniform class distribution. The dataset was partitioned as follows: 365 clips for training, 104 for validation, and 52 for testing. The test subset remained strictly unseen during the training and validation phases. During evaluation, the gesture clips were processed using the same preprocessing pipeline adopted for training. A prediction was considered correct only when the inferred gesture label matched the corresponding ground-truth annotation. Evaluation is conducted at the window/sequence level (window length $T=10$ frames, stride $s=1$). Thus, the 52 test clips a larger number of test windows ($N_{\text{test}} \gg 52$). A prediction is counted as correct only when the inferred label matches the ground truth. To ensure consistency and reproducibility across all configurations, we use a fixed hold-out split (without k-fold cross-validation). Illustrations of the four gesture classes are provided in Figure 4.



a) Starting b) Pausing c) Moving d) Scrolling

Fig. 4. Gesture-based action descriptions (Start/Pause/Move/Scroll)

Move is a dynamic gesture. The cursor direction is determined from the frame-by-frame displacement vector of the index finger tip, which is smoothed using an Exponential Moving Average (EMA) and removing noise with a *dead-zone*; the mapping from the smoothed displacement to the cursor step follows (1)–(3). Left/Right/Up/Down correspond to the signs of the

displacement components, while diagonal directions follow the angle of the vector; **Start/Pause** is a static gesture to open/close the control loop, while **Scroll** is dynamic but only uses the vertical component.

$$\tilde{\mathbf{p}}_t = \alpha\,\mathbf{p}_t + (1-\alpha)\,\tilde{\mathbf{p}}_{t-1}, \qquad \alpha \in (0,1); \tag{1}$$

$$\Delta\tilde{\mathbf{p}}_t = \tilde{\mathbf{p}}_t - \tilde{\mathbf{p}}_{t-1}, \qquad r_t = \left\|\Delta\tilde{\mathbf{p}}_t\right\|_2; \tag{2}$$

$$\mathbf{c}_t = \gamma \max(0, r_t - \delta)\,\frac{\Delta\tilde{\mathbf{p}}_t}{\max(r_t, \varepsilon)}, \tag{3}$$

where $\mathbf{p}_t \in [0,1]^2$ are the normalized coordinates of index finger tip in frame t; $\tilde{\mathbf{p}}_t$ is the smoothed version; δ is *dead-zone*; γ is the amplification factor; $\varepsilon > 0$ prevents division by 0.

Recordings were performed under a range of conditions to test the model's ability to adapt beyond controlled laboratory settings. Lighting conditions included natural daylight (approximately 500 lux), typical indoor fluorescent lamps (around 300 lux), and a dim LED setup (close to 100 lux). Backgrounds alternated between a plain uniform wall and a cluttered office scene containing furniture and other objects. To add further variation, the camera was positioned in three different ways: directly in front of the participant (0°), at an oblique angle of about 45°, and from an overhead top-down view (90°).

All clips were recorded at a resolution of 640×480 pixels with a frame rate of 30 frames per second. The duration of each video ranged from two to five seconds. During preprocessing, horizontal flipping was applied to create both left- and right-hand variations, and all frames were converted from OpenCV's BGR format to the standard RGB format.

Unlike public gesture datasets such as SHREC, EgoHands, or NVGesture, which are oriented toward sign language or virtual reality scenarios, this dataset was specifically to reflect mouse-control actions. By tailoring the dataset to the target application, the training and evaluation of CLVM remained directly relevant to its use in real-world human-computer interaction.

Once a rich, representative, and extensive collection of videos covering the range of hand gestures in various real-world scenarios is gathered, the system proceeds to the preprocessing stage, where data are prepared for model training. This step is crucial to ensure that the input data are reliable, consistent, and suitable for the deep learning algorithms. This preprocessing phase consists of three primary tasks, each crucial for improving and normalizing the data:

– **Resizing:** data inconsistencies may arise from videos that were gathered from various sources and recorded with different resolutions and frame rates. To address this, all video frames are normalized to a fixed size of 640×480 pixels, ensuring a uniform input format. Consistent image sizes minimize computational expenses and prevent training errors, which not only guarantees uniformity, but also enhances the processing efficiency of deep learning models.

– **Image flipping:** the training dataset can be enhanced using this widely-used data augmentation method without requiring the collection of new data. The system can generate new variations of the same gesture by flipping video frames horizontally. For example, it can change the "Move" gesture from the right hand to the left. This technique helps models become less dependent on the specific hand used (right/left) and improves their generalization capability and performance on diverse real-world data.

– **Color format conversion:** ensuring a consistent color format for image data is a seemingly simple but crucial detail. While many deep learning frameworks, such as TensorFlow or PyTorch, require input data in the RGB (Red, Green, Blue) format, image processing libraries, such as OpenCV, frequently use the BGR (Blue, Green, Red) color format. To ensure compatibility and prevent errors in image data processing and analysis, all images are converted from BGR to RGB.

After preprocessing, when the input data conform to the required size, color format, and multidimensional calculation standards, the system moves on to the next stage, which is to identify and track the hand points in each video frame. For this task, we employ the **MediaPipe Hand Landmarker**, a powerful tool for precisely identifying the 3D structure of the hand. The wrist, matching fingers, and the first finger are among the 21 key points (hand landmarks) that represent each hand. A collection of 3D coordinates (x, y, z) corresponds to each of these points:

– The location of the landmark on the image plane (along the horizontal and vertical axes of the frame) is determined by x and y.

– z represents relative depth information (approximate distance along the camera axis), which helps describe the 3D hand pose in addition to the 2D image-plane coordinates (x, y).

The close coordination between landmark detection and preprocessing forms the foundation of the training process, ensuring that the system learns robust and discriminative features for reliable recognition.

The landmark data obtained from detection and tracking serve as the primary input to the deep learning model and form the basis for evaluating the overall system efficacy. This data provides rich spatio-temporal information

that captures the hand's movements, structure, and patterns of light over time. Each layer in the deep learning model's architecture is designed to contribute to the task of accurate hand gesture recognition.

The architecture includes the following key components:

– **Proposed Layer.** This layer is specifically designed to extract features from landmark data or processed images for gesture identification. Depending on the method, this layer can be implemented as either a Recurrent Neural Network (RNN) to process the motion of landmarks over time series or a Convolutional Neural Network (CNN) to take advantage of the hand's geometric and textural features in each frame. The complexity of the gestures to be classified and the type of data determine which architecture is best.

– **Dropout Layer.** This layer helps prevent overfitting. During training, it randomly deactivates a significant proportion of neurons in the layer, forcing the network to learn more robust and generalized features rather than relying on specific neuronal co-adaptations.

– **Dense Layer.** This layer combines the extracted features from previous layers to produce the final classification. The output dense layer consists of four neurons, each corresponding to one of the predefined gesture classes: Move, Scroll, Pause, and Start. The output probability for each gesture type represents the model's degree of confidence.

In addition to designing a single architecture, the system tests, evaluates, and compares numerous neural network architectures in order to determine which one best suits the data-response characteristics and real-world application requirements. Among the primary architectural elements taken into account are:

– **CLVM.** A hybrid architecture that combines CNNs to extract spatial features with LSTMs to process temporal information. This approach is effective for dynamic gestures where both motion and shape are crucial.

– **RNN.** Basic recurrent neural networks can process sequences but struggle with long-term dependencies, limiting their ability to recognize long or complex gestures.

– **LSTM.** An improved RNN that uses a unique memory mechanism to store and access information over long sequences. LSTMs are particularly useful for complex gestures with significant motion variations or long durations.

– **GRU.** A lightweight variant of LSTM that often achieves comparable performance with lower computational complexity and faster training.

In summary, the proposed hand gesture recognition system is a tightly integrated processing pipeline that incorporates advanced deep learning techniques, hand feature extraction, and image processing. The ultimate objective is to develop a solution that can meet the demands of contemporary

human-machine interaction applications by accurately, robustly, and efficiently recognizing hand gestures in real time from video data.

3.2. Method. The contactless virtual mouse system's training phase is a crucial phase that affects the system's overall performance in terms of accuracy and stability in real-world settings. Furthermore, the efficient Algorithm 1 is crucial for maintaining the system's capacity to recognize hand gestures accurately and function flawlessly in a variety of usage scenarios.

After evaluating and testing various neural network architectures, we decided to create a solution based on a combination of CNNs and LSTMs. The strengths of both model types are fully utilized in this hybrid architecture: CNNs extract spatial features from image data or landmark sequences, while LSTMs model temporal sequential relationships, a crucial component in obtaining dynamic hand gesture recognition. Figure 5 provides a detailed description of the system's interface training procedure and method, highlighting the steps involved in processing input data, architectural modeling, and training optimization.

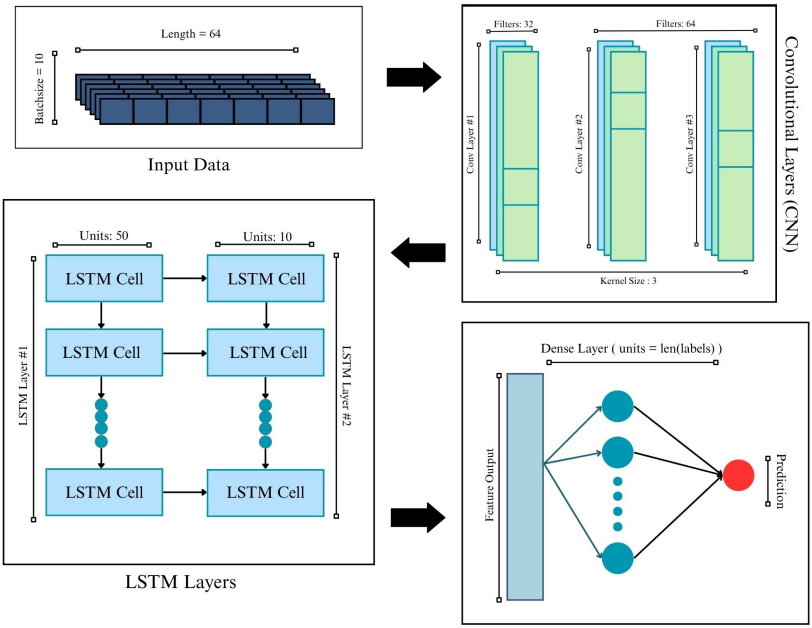


Fig. 5. System Model Method

Algorithm 1. Hand Gesture-based Mouse Control Algorithm

Input:
- Video frame F_t: Input frame from video at time t.
- Trained model M: Pre-trained model for hand gesture classification (Move/Start/Pause/Scroll).
- MediaPipe Hands mp: Hand landmark detector.
- Amplification factor α: Controls movement sensitivity.

Output: Mouse actions corresponding to detected gestures.
Initialize mouse position $(x_{prev}, y_{prev}) \leftarrow (None, None)$.
for each frame F_t from the video feed **do**
    Convert F_t to RGB image I_t.
    Detect hands H using mp on I_t.
    for each hand $h \in H$ **do**
        Extract hand landmarks L_h.
        Convert L_h to feature vector V_h.
        Predict gesture $G_h \leftarrow M(V_h)$.
        if $G_h =$ MoveMouse **then**
            Get fingertip coordinates (x, y) from L_h.
            Map (x, y) to screen coordinates and move mouse.
        else if $G_h =$ Start **then**
            Perform start virtual mouse.
        else if $G_h =$ Pause **then**
            Perform stop virtual mouse.
        else if $G_h =$ Scroll **then**
            Get index fingertip coordinates (x_{idx}, y_{idx}) from L_h.
            Map (x_{idx}, y_{idx}) to screen resolution (W, H).
            Compute scroll direction based on change in y_{idx} from previous frame.
            if $y_{prev} \neq None$ **then**
                if $y_{idx} < y_{prev}$ **then**
                    Scroll up.
                else if $y_{idx} > y_{prev}$ **then**
                    Scroll down.
                end if
            end if
            Update $y_{prev} \leftarrow y_{idx}$
        end if
    end for
    Visualize results: draw landmarks and show gesture label.
end for
return Real-time mouse control action.

To achieve robust hand gesture recognition from temporal data (such as landmark or sensor sequences), the system uses a hybrid deep learning architecture that combines LSTM units with a 1D-Dimensional Convolutional Neural Network (1D-CNN). By combining the temporal learning and memorization capabilities of LSTM with the feature extraction capabilities of CNN, the system is able to create a single end-to-end training pipeline that maximizes performance and reduces the accumulation process between intermediate stages.

1D Convolution for Local Feature Extraction. The 1D-CNN layers are designed to extract temporal feature sets from multivariate data series. Assuming the input to the network is of the form $x \in \mathbb{R}^{T \times C_{\text{in}}}$, where T is the number of time steps and C_{in} is the number of channels (or features) at each time step, the output at the time position corresponding to the k-th filter is calculated according to the formula:

$$y_t^{(k)} = \sum_{i=0}^{K-1} \sum_{j=0}^{C_{\text{in}}-1} x_{t+i,j} \cdot w_{i,j}^{(k)} + b^{(k)}, \tag{4}$$

where:

– $w_{i,j}^{(k)}$ represents the weight of the k-th kernel at position i and input channel j,

– $b^{(k)}$ is the bias term for the k-th filter,

– $x_{t+i,j}$ is the input feature value at time step $t+i$ and feature j.

Through this convolution, the network can identify subtle yet significant patterns by learning local dependencies in the input signal. The convolution's output is subsequently passed through a nonlinear activation function, usually a Rectified Linear Unit (ReLU), defined as:

$$\text{ReLU}(z) = \max(0, z). \tag{5}$$

Equation 5 introduces nonlinearity; the application of this activation function enables the network to model complex nonlinear relationships in the data, which is essential for accurate gesture classification.

To improve robustness to input variations and reduce spatial (or temporal) dimensionality, a max-pooling layer is typically applied after convulational layers. It down-samples the feature maps by selecting the maximum value in each pooling window, which provides translation invariance and helps highlight the most salient features.

The tight integration of CNN and LSTM in this framework creates a powerful and flexible deep learning model that meets the requirements for real-time hand gesture recognition with high accuracy and stable operation in large-scale human-computer interaction applications.

LSTM for Temporal Sequence Modeling. After the CNN layers extract salient local features, the LSTM layers model the temporal dependencies across successive time steps. LSTM units maintain an internal memory cell, allowing them to store and utilize information from earlier time steps in addition to processing the current input. Because of this mechanism, This mechanism enables LSTMs to learn patterns over long time horizons without being impacted by the vanishing gradient problem, a common limitation of basic RNNs on long sequences.

Given the input x_t at time step t, the hidden state h_{t-1}, and the cell state C_{t-1} from the previous time step, the LSTM updates are computed as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad \text{(Forget gate)}, \tag{6}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad \text{(Input gate)}, \tag{7}$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad \text{(Candidate cell state)}, \tag{8}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \quad \text{(Updated cell state)}, \tag{9}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad \text{(Output gate)}, \tag{10}$$

$$h_t = o_t \odot \tanh(C_t) \quad \text{(Updated hidden state)}, \tag{11}$$

where:

– $\sigma(\cdot)$ is the sigmoid activation function,

– $\tanh(\cdot)$ is the hyperbolic tangent function,

– $\odot$ denotes element-wise multiplication,

– W_f, W_i, W_C, W_o and b_f, b_i, b_C, b_o are trainable parameters for each gate.

Every gate and state in the LSTM architecture is crucial for managing information flow across time steps, enabling the learning of long-term dependencies without signal degradation.

– The **forget gate** f_t (Equation 6) determines which part of the previous memory C_{t-1} should discarded or retained. It acts as an intelligent filter that removes information no longer needed for subsequent steps, reducing memory load and preventing the accumulation of irrelevant information.

– The **input gate** i_t (Equation 7) controls what new information is added to the memory at the current time step. It works in conjunction with the candidate cell state to identify new features to learn and integrate into long-term memory.

– The **candidate cell state** $\tilde{C}_t$ (Equation 8) proposes a new memory update based on the current input and the previous hidden state. This represents the new information that the network considers incorporating into the memory.

– These three components interact in (Equation 9) to compute the new cell state C_t. Specifically, the new memory is formed by combining selected information from the old memory (via f_t) and new information added (via i_t and $\tilde{C}_t$).

– The **output gate** o_t (Equation 10) determines which part of the new cell state should be passed on as the hidden state. This serves as the summarized information that the network uses to make predictions at the current time step.

– Finally, the hidden state h_t is computed in (Equation 11) based on the output gate and the updated memory. This state is not only used to produce classification predictions at the current time step but is also passed to the next step, supporting the learning of motion sequences.

This mechanism helps the hand gesture recognition system function accurately and steadily, even with complex movement sequences, by preserving the memory required for long-term relationships and removing noise and redundant information.

Integration and Output. By integrating Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) layers into a unified architecture, the model is able to simultaneously capture both spatial (local) and temporal (sequential) features present in the input data. The CNN layers are responsible for extracting specific local features from individual frames, such as the shape, contour, or landmark configuration of the hand, which are key to distinguishing specific gestures. Meanwhile, the LSTM layers are responsible for modeling the dynamic evolution of these features over time, allowing the system to learn how gestures unfold as a sequence, rather than just considering static poses at each point in time. This combination significantly enhances the model's ability to distinguish between different dynamic gestures, particularly those that are similar in single frames but differ in time progression, which is crucial for the system to function correctly in a virtual mouse interface.

After processing by the CNN and LSTM layers, a dense layer with a softmax activation function receives the LSTM's final output. To translate the representation that the network has learned into probability values that correspond to each type of gesture, this layer must compute the probability

distribution over the specified gesture classes. The model is trained using the classification cross-entropy loss function, which guides the network to increase classification accuracy by measuring the degree of difference between the actual label and the predicted probability distribution. The model uses the Adam optimization algorithm, a widely-used optimization algorithm that combines advantages of AdaGrad and RMSProp, to increase training efficiency and stability and aid in faster model convergence.

Owing to its design, the system can effectively generalize to changes in user behavior, such as variations in speed, style, or consistency in gesture performance. The model is especially well-suited for interactive applications like virtual mouse control in human-machine interfaces, as it exhibits both high stability and responsiveness in real-time gesture recognition tasks.

4. Performance Evaluation. the proposed CLVM architecture was systematically evaluated using a dataset of time-series features derived from camera-captured hand gesture. These features capture the dynamics of hand movements during gesture execution, offering valuable insights into gesture patterns. Hand gestures in this study are categorized into four distinct classes: *Move*, *Pause*, *Scroll*, and *Start*. Each gesture class corresponds to a fundamental action in the virtual mouse interface, playing a critical role in enabling precise and efficient user interaction.

Accuracy, loss function, and F1-Score – three common metrics in machine learning – were used to thoroughly and impartially assess the model performance. Accuracy represents the percentage of correct predictions, Loss quantifies the discrepancy between predictions and ground truth, and the F1-Score provides a balanced measure between precision and recall, which is particularly useful for potentially imbalanced classes. To guarantee transparency and reproducibility of the results, Table 1 provides a detailed explanation of the formulas used to calculate these metrics, as well as an explanation of the relevant parameters, where:

– TP (True Positive): number of samples correctly classified into the intended gesture class.

– FN (False Negative): number of samples of that gesture class misclassified as another class.

– FP (False Positive): number of samples from other classes misclassified as that gesture class.

– TN (True Negative): number of samples from other classes correctly classified as not belonging to that gesture class.

In this part, the evaluation metrics are used to assess the models' performance on the four-class hand-gesture classification task (Move, Pause, Scroll, and Start) to ensure that the outcomes are thorough and impartial.

Table 1. Evaluation parameters

	Positive Prediction	**Negative Prediction**
Positive Action	TP	FN
Negative Action	FP	TN

Accuracy is one of the most fundamental and widely used metrics among them. We report standard classification metrics – Accuracy, Precision, Recall, and F_1-Score – computed per class and summarized by macro/micro averages [39-41]. The formulas are presented in Equations (12)-(15).

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN}, \tag{12}$$

$$\text{Precision} = \frac{TP}{TP+FP}, \tag{13}$$

$$\text{Recall} = \frac{TP}{TP+FN}, \tag{14}$$

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{15}$$

Table 2 summarizes the performance evaluation results, comparing the proposed model against several reference architectures. The compared models include CNN, LSTM, GRU, and RNN architectures. All these architectures are widely used in deep learning, particularly for time-series processing tasks such as hand gesture recognition.

To ensure the stability and objectivity of the outcomes, we deploy and assess each model through four separate test runs. For ease of visual comparison, accuracy and F1-Score are reported as percentages (%), while the loss is reported as the (unitless) cross-entropy value. In addition to increasing the reliability of the results obtained, performing multiple runs and averaging helps to reduce the impact of random factors in the training and evaluation process.

The results in Table 2 highlight the superior performance of the proposed CLVM architecture compared to individual LSTM, GRU, and RNN models.

The experimental results demonstrate that the proposed CLVM model outperforms the reference models. CLVM achieved an accuracy of 99.88% and

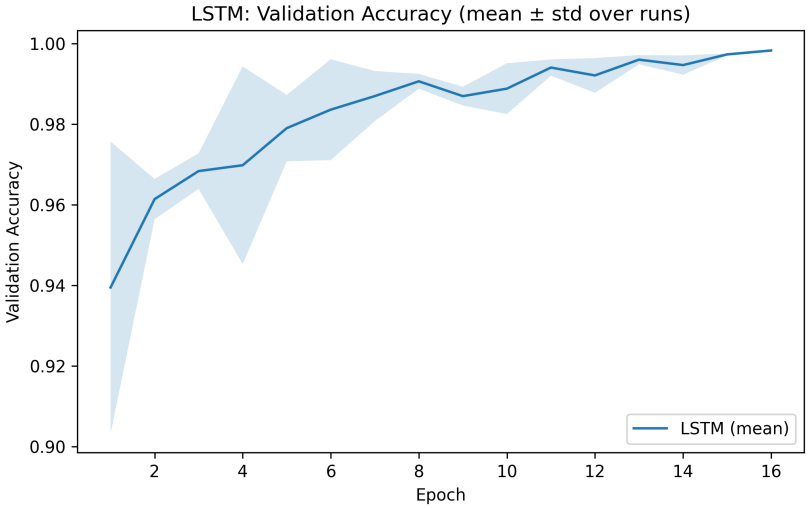
an F1-Score of 99.88%, demonstrating an excellent balance between precision and recall. Furthermore, the low loss of the model of only 0.38 indicated strong learning capacity and training stability.

Table 2. System evaluation results

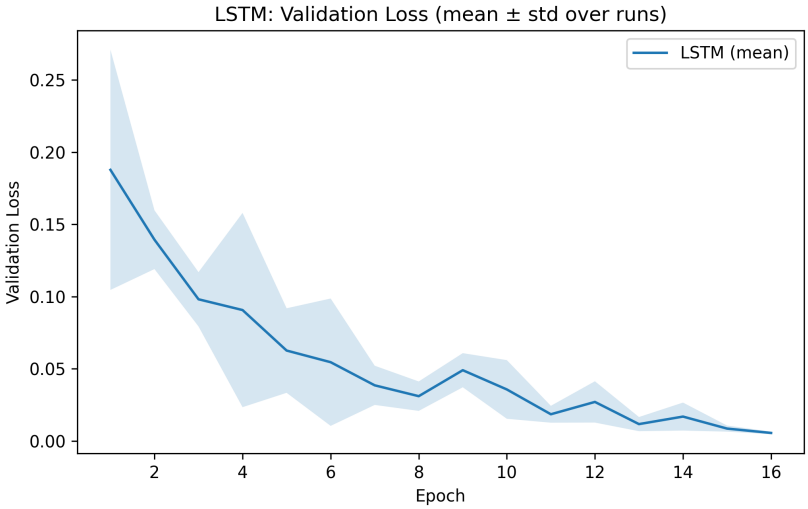
Model	Round	Accuracy (%)	Loss	F1
LSTM	Round 1	99.88	0.39	99.89
	Round 2	99.85	0.46	99.86
	Round 3	99.69	1.00	99.69
	Round 4	99.78	0.67	99.80
Average		**99.80**	**0.63**	**99.81**
GRU	Round 1	99.82	0.57	99.83
	Round 2	99.76	0.87	99.78
	Round 3	99.79	0.78	99.81
	Round 4	99.75	0.67	99.76
Average		**99.78**	**0.72**	**99.79**
RNN	Round 1	99.25	3.12	99.28
	Round 2	99.23	2.99	99.26
	Round 3	98.82	4.15	98.91
	Round 4	99.03	3.51	99.09
Average		**99.08**	**3.44**	**99.13**
CLVM	Round 1	99.93	0.26	99.93
	Round 2	99.88	0.32	99.87
	Round 3	99.82	0.57	99.83
	Round 4	99.90	0.39	99.90
Average		**99.88**	**0.38**	**99.88**

These results clearly demonstrate that the proposed CLVM system outperforms not only the individual models (LSTM, GRU, RNN) but also achieves top scores across all three evaluation metrics. This underscores the superiority of the hybrid CLVM architecture for processing and classifying gesture sequences in real-time virtual mouse interfaces.

Future work will involve evaluating CLVM on richer datasets with a wider variety of real-world conditions to further validate its generalization capability for complex contactless interaction scenarios. The performance difference between the CLVM hybrid architecture that we propose and conventional sequential models is evident from the obtained results. The training and testing curves for each model are depicted in Figures 6 – 9.

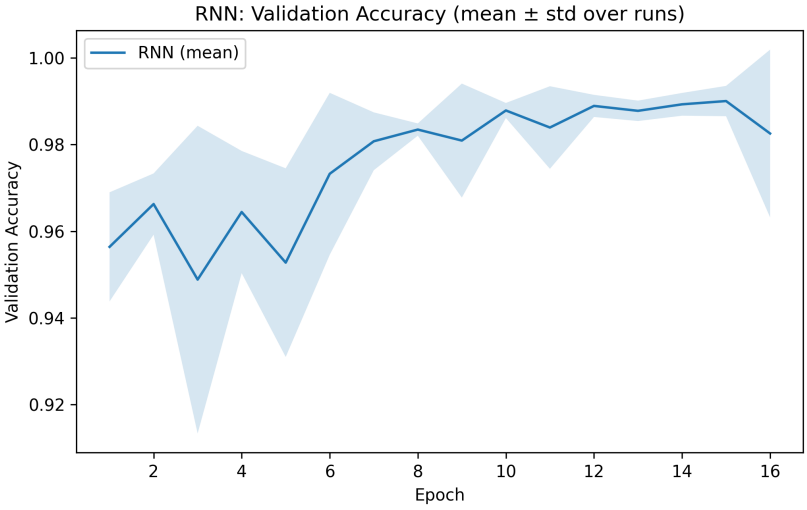


a)

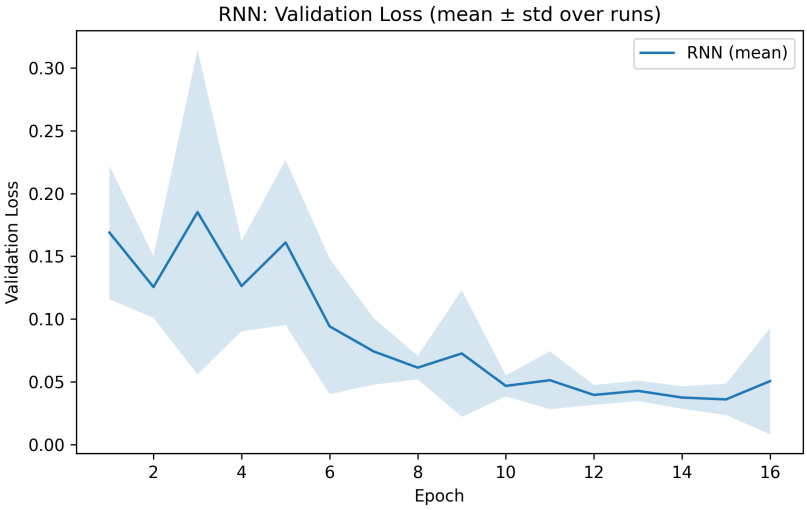


b)

Fig. 6. Accuracy and Loss mean value of LSTM model after 4 rounds

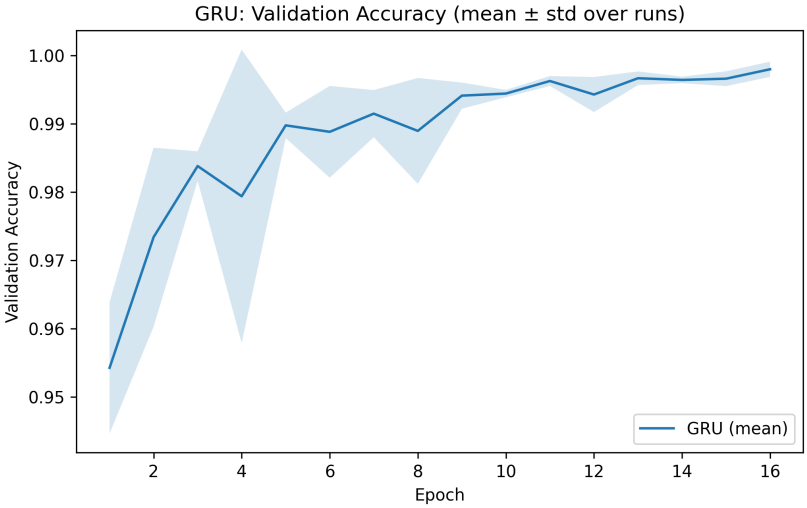


a)

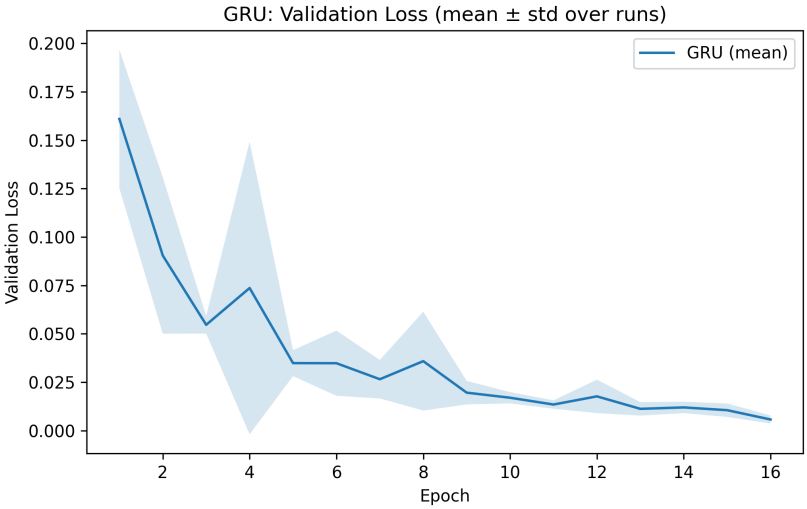


b)

Fig. 7. Accuracy and Loss mean value of RNN model after 4 rounds

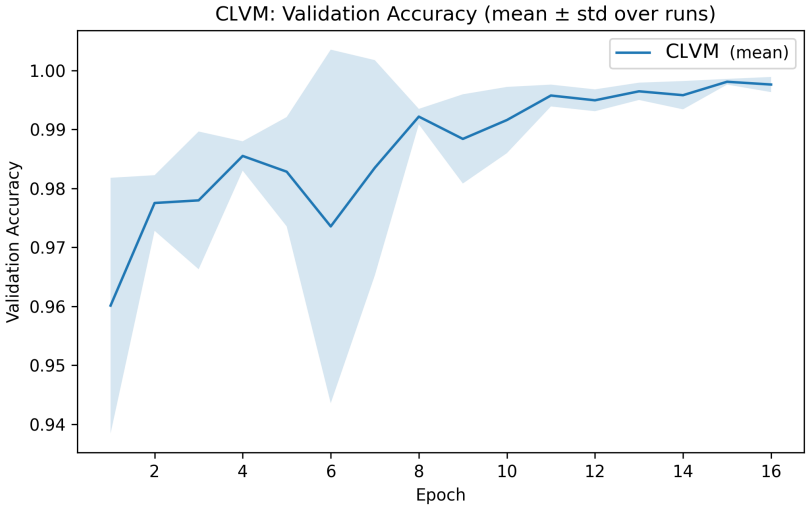


a)

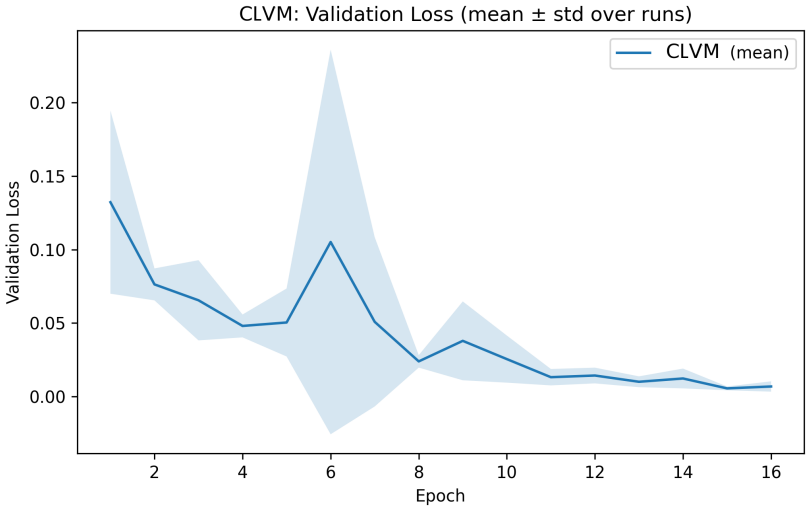


b)

Fig. 8. Accuracy and Loss mean value of GRU model after 4 rounds



a)



b)

Fig. 9. Accuracy and Loss mean value of CLVM model after 4 rounds

5. Conclusion. This paper presented a real-time virtual mouse control system based on hand gesture recognition using the CLVM deep learning architecture. The system uses MediaPipe for real-time hand landmark extraction, CNN to learn spatial features, and LSTM to process motion sequences. The development pipeline encompassed data collection, preprocessing, model design, and performance evaluation. Experimental results show that the proposed model achieved low loss of 0.38, accuracy of 99.88%, and an F1-Score of 99.88%. Compared to reference models (standalone LSTM, GRU, RNN) the system also demonstrates stable processing speed at a higher frame rate. These results demonstrate the effectiveness and reliability of CLVM for real-time hand gesture recognition, even under varying lighting conditions and cluttered backgrounds.

Although the results are promising, the current study is limited to only four basic gestures (Move, Scroll, Pause, Start), serving primarily as proof of concept. Future work will focus on the following directions:

– Expanding the gesture vocabulary to include more complex and practical mouse operations (e.g., single/double-clicking, dragging and dropping, zooming, and multi-finger interactions), thereby improving usability in real-world applications.

– Conducting structured user studies to evaluate experience factors such as intuitiveness, learnability, responsiveness, and potential fatigue during prolonged use, to complement technical performance results.

– Validating the system on standard benchmarks (e.g., SHREC, EgoHands, NVGesture) to enhance external generalizability and enable direct comparison with existing gesture recognition systems.

In summary, CLVM provides a powerful, scalable, and efficient platform for natural human-computer interaction. With planned future developments, it has the potential to evolve from a proof of concept into a versatile and practical alternative to conventional input devices.

References

1. Maslej N., Fattorini L., Perrault R., Gil Y., Parli V., Kariuki N., Capstick E., Reuel A., Brynjolfsson E., Etchemendy J. et al. Artificial intelligence index report. arXiv preprint arXiv:2504.07139. 2025.
2. Asgher U., Ayaz Y., Taiar R. Advances in artificial intelligence (AI) in brain computer interface (BCI) and industry 4.0 for human machine interaction (HMI). Frontiers in Human Neuroscience. 2023. vol. 17. DOI: 10.3389/fnhum.2023.1320536.
3. Sumak B., Brdnik S., Pusnik M. Sensors and artificial intelligence methods and algorithms for human–computer intelligent interaction: A systematic mapping study. Sensors. 2021. vol. 22. no. 1.
4. Mourtzis D., Angelopoulos J., Panopoulos N. The future of the human–machine interface (HMI) in society 5.0. Future Internet. 2023. vol. 15. no. 5.

5. Mukhtar H. Artificial intelligence techniques for human-machine interaction. Artificial Intelligence and Multimodal Signal Processing in Human-Machine Interaction. 2025. pp. 19–42.
6. Shibly K.H., Dey S.K., Islam M.A., Showrav S.I. Design and development of hand gesture based virtual mouse. 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT). IEEE, 2019. pp. 1–5. DOI: 10.1109/ICASERT.2019.8934612.
7. Shrivastava A., Pundir S., Sharma A., Srivastava A., Kumar R., Khan A.K. Control of a virtual system with hand gestures. 3rd International Conference on Pervasive Computing and Social Networking (ICPCSN). IEEE, 2023. pp. 1716–1721.
8. Reddy V.V., Dhyanchand T., Krishna G.V., Maheshwaram S. Virtual mouse control using colored finger tips and hand gesture recognition. IEEE-HYDCON. IEEE, 2020. pp. 1–5. DOI: 10.1109/HYDCON48903.2020.9242677.
9. Kasar M., Kavimandan P., Suryawanshi T., Abbad S. Ai-based real-time hand gesture-controlled virtual mouse. Australian Journal of Electrical and Electronics Engineering. 2024. vol. 21. no. 3. pp. 258–267.
10. Lugaresi C., Tang J., Nash H., McClanahan C., Uboweja E., Hays M., Zhang F., Chang C.-L., Yong M.G., Lee J., et al. Mediapipe: A framework for building perception pipelines. arXiv preprint arXiv:1906.08172. 2019.
11. Hung N.V., Loi T.Q., Binh N.H., Nga N.T.T., Huong T.T., Luu D.L. Building an online learning model through a dance recognition video based on deep learning. Informatics and Automation. 2024. vol. 23. no. 1. pp. 101–128.
12. Krizhevsky A., Sutskever I., Hinton G.E. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems. 2012. vol. 25.
13. Hochreiter S., Schmidhuber J. Long short-term memory. Neural Computation. 1997. vol. 9. no. 8. pp. 1735–1780.
14. Acharya K. Virtual mouse using hand gestures. Authorea. 2024. DOI: 10.22541/au.173161606.61659157/v1.
15. Beyeler M.. Machine Learning for OpenCV. Birmingham: Packt Publishing, 2017. 382 p.
16. Joshi P., Escriva D.M., Godoy V. OpenCV by example. Packt Publishing Ltd, 2016. 296 p.
17. Howse J. OpenCV computer vision with python. UK: Packt Publishing Birmingham, 2013. vol. 27.
18. Dharavath K., Kumar G.M., Reddy K.R., Reddy M.H. Gesture and voice controlled virtual mouse for elderly people. 2nd International Conference on Networking and Communications (ICNWC). IEEE, 2024. pp. 1–5.
19. Gupta A., Chawla N., Jain R., Thakur N., Devi A. Gesture-based touchless operations: leveraging mediapipe and OpenCV. NEU Journal for Artificial Intelligence and Internet of Things. 2023. vol. 2. no. 1.
20. Bansal B.S., Nailwal D., Bhatt G., Kumar A., Petwal H. Real-time video control via hand and eye movements using opencv and mediapipe. International Conference on Artificial Intelligence and Emerging Technology (Global AI Summit). IEEE, 2024. pp. 270–275.
21. Nandwalkar D.J., Mandal M., Khirari A., Bhalchim T. Control mouse using hand gesture and voice. International Journal for Research in Applied Science & Engineering Technology (IJRASET). 2023. vol. 11. pp. 3261–3268.
22. Hung N.V., Quan N.A., Tan N., Hai T.T., Trung D.K., Nam L.M., Loan B.T., Nga N.T.T. Building predictive smell models for virtual reality environments. Informatics and Automation. 2025. vol. 24. no. 2. pp. 556–582. DOI: 10.15622/ia.24.2.7.
23. Kruk A. The benefits of virtual learning environment (VLE) in teaching ESP. Current nutrition in the humanities. 2022.

24. Hung N., Dat P.T., Tan N., Quan N.A., Trang L., Nam L.M., et al. Heverl–viewport estimation using reinforcement learning for 360-degree video streaming. Informatics and Automation. 2025. vol. 24. no. 1. pp. 302–328.
25. Nguyen H., Dao T.N., Pham N.S., Dang T.L., Nguyen T.D., Truong T.H. An accurate viewport estimation method for 360 video streaming using deep learning. EAI Endorsed Trans. Ind. Networks Intell. Syst. 2022. vol. 9. no. 4.
26. Kumar M., Rathi G., Singh T., NL T. CNN based virtual whiteboard application. Second International Conference on Advances in Information Technology (ICAIT). IEEE, 2024. pp. 1–6.
27. Heemskerk I., Kuiper E., Meijer J. Interactive whiteboard and virtual learning environment combined: Effects on mathematics education. Journal of Computer Assisted Learning. 2014. vol. 30. no. 5. pp. 465–478.
28. Mundargi Z., Das S., Shinde A., Deokar O., Bahirat S., Shetiya D. Hand gesture desktop control with python. 2nd International Conference on Advances in Computation, Communication and Information Technology (ICAICCIT). IEEE, 2024. vol. 1. pp. 35–40.
29. Nguyen H.A., Tran T.T., Ho H.Q., Ngo T.D., Vu K.N., Huynh V.L.T. Hand gesture recognition using cvzone. Proceedings of the 9th International Conference on Intelligent Information Technology. 2024. pp. 108–113.
30. Uke S., Shaikh A., Rayate H., Kamble A., Rahane S. Towards touchless interaction: Implementing hand gesture recognition for presentation and media control. International Conference on Emerging Smart Computing and Informatics (ESCI). IEEE, 2025. pp. 1–6. DOI: 10.1109/ESCI63694.2025.10988099.
31. Vasanthagokul S., Kamakshi K.V.G., Mudbhari G., Chithrakumar T. Virtual mouse to enhance user experience and increase accessibility. 4th International Conference on Inventive Research in Computing Applications (ICIRCA). IEEE, 2022. pp. 1266–1271.
32. Naidu P., Muthukumaran N., Chandralekha S., Reddy K.T., Vaishnavi K.S. An analysis on virtual mouse control using human eye. 5th International Conference on Image Processing and Capsule Networks (ICIPCN). IEEE, 2024. pp. 233–237.
33. Praba B.V., Vinothini R., Jayarathna M., Subramani K., Sravanthi P. Virtual AI mouse with biometric authentication. International Conference on Knowledge Engineering and Communication Systems (ICKECS). IEEE, 2024. vol. 1. pp. 1–5. DOI: 10.1109/ICKECS61492.2024.10617264.
34. Karthick S., Dinesh M., Jeffery Dani Raj C., Jayapandian N. Artificial intelligence based enhanced virtual mouse hand gesture tracking using yolo algorithm. IEEE 2nd International Conference on Data, Decision and Systems (ICDDS). IEEE, 2023. pp. 1–6. DOI: 10.1109/ICDDS59137.2023.10434330.
35. Karpov A., Ronzhin A., Kipyatkova I. An assistive Bi-modal user interface integrating multi-channel speech recognition and computer vision. Human-Computer Interaction: Interaction Techniques and Environments (HCI 2011). Lecture Notes in Computer Science. 2011. vol. 6762. pp. 454–463. DOI: 10.1007/978-3-642-21605-3_50.
36. Karpov A., Carbini S., Ronzhin A., Viallet J.E. Comparison of two different similar speech and gestures multimodal interfaces. Proc. of the 16th European Signal Processing Conference (EUSIPCO). 2008. pp. 1–5.
37. Jaimes A., Sebe N. Multimodal human–computer interaction: A survey. Computer Vision and Image Understanding. 2007. vol. 108. no. 1–2. pp. 116–134.
38. Bazarevsky V., Zhang F. On-device, real-time hand tracking with mediapipe. Available at: https://research.google/blog/on-device-real-time-hand-tracking-with-mediapipe/ (accessed 15.01.2026).
39. Manning C.D., Raghavan P., Scutze H. Introduction to Information Retrieval. Cambridge University Press, 2008.

40. Sokolova M., Lapalme G. A systematic analysis of performance measures for classification tasks. Information Processing & Management. 2009. vol. 45. no. 4. pp. 427–437.

41. Powers D.M.W. Evaluation: From precision, recall and f-measure to roc, informedness, markedness & correlation. Journal of Machine Learning Technologies. 2011. vol. 2. no. 1. pp. 37–63.

Nguyen Viet Hung — Ph.D., Lecturer, International Training and Cooperation Institute, East Asia University of Technology. Research interests: multimedia communications, network security, artificial intelligence, traffic engineering in next-generation networks, QoE/QoS guarantee for network services, green networking, applications. The number of publications — 35. hungnv@eaut.edu.vn; Ky Anh, Ha Tinh, Viet Nam; office phone: +84(098)911-2079.

Phi Dinh Huynh — Research assistant, Faculty of Information Technology, East Asia University of Technology. Research interests: nature language processing, machine and deep learning, computer vision. The number of publications — 2. 20222072@eaut.edu.vn; Huong Ngai – Thach That, Ha Noi, Viet Nam; office phone: +84(037)395-0094.

Ma Van Tung — Research assistant, International Training and Cooperation Institute, East Asia University of Technology. Research interests: artificial intelligence, machine and deep learning, virtual reality. The number of publications — 1. 20230240@eaut.edu.vn; Minh Quang, Tuyen Quang, Viet Nam; office phone: +84(092)882-2756.

Nguyen Van Vu — Research assistant, Faculty of Information Technology, East Asia University of Technology. Research interests: machine and deep learning, computer vision. The number of publications — 1. 20222048@eaut.edu.vn; Quoc Oai, Ha Noi, Viet Nam; office phone: +84(038)379-0656.

Nguyen Phu Dat — Research assistant, Faculty of Information Technology, East Asia University of Technology. Research interests: nature language processing, artificial intelligence, computer network. The number of publications — 1. 20220344@eaut.edu.vn; Son Trung – Quoc Oai, Ha Noi, Viet Nam; office phone: +84(096)548-3341.

УДК 006.72 **DOI** 10.15622/ia.25.2.5

Н.В. Хунг, Ф.Д. Хуинь, М.В. Тунг, Н.В. Ву, Н.Ф. Дат

CLVM: ГИБРИДНАЯ МОДЕЛЬ ГЛУБОКОГО ОБУЧЕНИЯ ДЛЯ БЕСКОНТАКТНОГО УПРАВЛЕНИЯ ВИРТУАЛЬНОЙ МЫШЬЮ

Хунг Н.В., Хуинь Ф.Д., Тунг М.В., Ву Н.В., Дат Н.Ф. **CLVM: гибридная модель глубокого обучения для бесконтактного управления виртуальной мышью.**

Аннотация. В эпоху стремительной цифровой трансформации и растущего распространения искусственного интеллекта обеспечение естественного, непрерывного и бесконтактного человеко-компьютерного взаимодействия приобретает первостепенное значение для различных областей. Данная работа представляет новую модель на базе глубокого обучения для управления виртуальной мышью посредством жестов, получившая название CLVM (CNN-LSTM Virtual Mouse). Разработанная система основывается на гибридной архитектуре, интегрирующей три мощных компонента: (1) MediaPipe – для высокоэффективной детекции ключевых ориентиров кисти в режиме реального времени; (2) сверточную нейронную сеть (CNN) – для извлечения пространственных признаков; (3) сеть долгой краткосрочной памяти (LSTM) – для моделирования временной динамики, что существенно повышает точность и непрерывность распознавания жестов во временной последовательности. В отличие от традиционных подходов, модель CLVM разработана для сохранения высокой производительности в условиях реальной среды, особенно при неравномерном освещении и наличии загроможденного фона. Система характеризуется низкой задержкой и высокой скоростью отклика, а также возможностью эффективного функционирования на устройствах с ограниченными ресурсами, что обуславливает ее пригодность для широкого практического применения. Результаты экспериментов демонстрируют, что CLVM достигает высокой точности (99,88%) при снижении потерь до 0,38, значительно превосходя по эффективности традиционные методы распознавания жестов. Полученные данные подчеркивают потенциал CLVM как надежного, масштабируемого и эффективного решения для организации естественного взаимодействия на основе жестов, представляя собой важный шаг вперед в разработке интеллектуальных, удобных для пользователя интерфейсов для бесконтактного управления.

Ключевые слова: компьютерное зрение, бесконтактный интерфейс, ориентиры кисти, машинное обучение, MediaPipe, виртуальная мышь.

Литература

1. Maslej N., Fattorini L., Perrault R., Gil Y., Parli V., Kariuki N., Capstick E., Reuel A., Brynjolfsson E., Etchemendy J. et al. Artificial intelligence index report. arXiv preprint arXiv:2504.07139. 2025.
2. Asgher U., Ayaz Y., Taiar R. Advances in artificial intelligence (AI) in brain computer interface (BCI) and industry 4.0 for human machine interaction (HMI). Frontiers in Human Neuroscience. 2023. vol. 17. DOI: 10.3389/fnhum.2023.1320536.
3. Sumak B., Brdnik S., Pusnik M. Sensors and artificial intelligence methods and algorithms for human–computer intelligent interaction: A systematic mapping study. Sensors. 2021. vol. 22. no. 1.
4. Mourtzis D., Angelopoulos J., Panopoulos N. The future of the human–machine interface (HMI) in society 5.0. Future Internet. 2023. vol. 15. no. 5.

5. Mukhtar H. Artificial intelligence techniques for human-machine interaction. Artificial Intelligence and Multimodal Signal Processing in Human-Machine Interaction. 2025. pp. 19–42.
6. Shibly K.H., Dey S.K., Islam M.A., Showrav S.I. Design and development of hand gesture based virtual mouse. 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT). IEEE, 2019. pp. 1–5. DOI: 10.1109/ICASERT.2019.8934612.
7. Shrivastava A., Pundir S., Sharma A., Srivastava A., Kumar R., Khan A.K. Control of a virtual system with hand gestures. 3rd International Conference on Pervasive Computing and Social Networking (ICPCSN). IEEE, 2023. pp. 1716–1721.
8. Reddy V.V., Dhyanchand T., Krishna G.V., Maheshwaram S. Virtual mouse control using colored finger tips and hand gesture recognition. IEEE-HYDCON. IEEE, 2020. pp. 1–5. DOI: 10.1109/HYDCON48903.2020.9242677.
9. Kasar M., Kavimandan P., Suryawanshi T., Abbad S. Ai-based real-time hand gesture-controlled virtual mouse. Australian Journal of Electrical and Electronics Engineering. 2024. vol. 21. no. 3. pp. 258–267.
10. Lugaresi C., Tang J., Nash H., McClanahan C., Uboweja E., Hays M., Zhang F., Chang C.-L., Yong M.G., Lee J., et al. Mediapipe: A framework for building perception pipelines. arXiv preprint arXiv:1906.08172. 2019.
11. Hung N.V., Loi T.Q., Binh N.H., Nga N.T.T., Huong T.T., Luu D.L. Building an online learning model through a dance recognition video based on deep learning. Informatics and Automation. 2024. vol. 23. no. 1. pp. 101–128.
12. Krizhevsky A., Sutskever I., Hinton G.E. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems. 2012. vol. 25.
13. Hochreiter S., Schmidhuber J. Long short-term memory. Neural Computation. 1997. vol. 9. no. 8. pp. 1735–1780.
14. Acharya K. Virtual mouse using hand gestures. Authorea. 2024. DOI: 10.22541/au.173161606.61659157/v1.
15. Beyeler M.. Machine Learning for OpenCV. Birmingham: Packt Publishing, 2017. 382 p.
16. Joshi P., Escriva D.M., Godoy V. OpenCV by example. Packt Publishing Ltd, 2016. 296 p.
17. Howse J. OpenCV computer vision with python. UK: Packt Publishing Birmingham, 2013. vol. 27.
18. Dharavath K., Kumar G.M., Reddy K.R., Reddy M.H. Gesture and voice controlled virtual mouse for elderly people. 2nd International Conference on Networking and Communications (ICNWC). IEEE, 2024. pp. 1–5.
19. Gupta A., Chawla N., Jain R., Thakur N., Devi A. Gesture-based touchless operations: leveraging mediapipe and OpenCV. NEU Journal for Artificial Intelligence and Internet of Things. 2023. vol. 2. no. 1.
20. Bansal B.S., Nailwal D., Bhatt G., Kumar A., Petwal H. Real-time video control via hand and eye movements using opencv and mediapipe. International Conference on Artificial Intelligence and Emerging Technology (Global AI Summit). IEEE, 2024. pp. 270–275.
21. Nandwalkar D.J., Mandal M., Khirari A., Bhalchim T. Control mouse using hand gesture and voice. International Journal for Research in Applied Science & Engineering Technology (IJRASET). 2023. vol. 11. pp. 3261–3268.
22. Hung N.V., Quan N.A., Tan N., Hai T.T., Trung D.K., Nam L.M., Loan B.T., Nga N.T.T. Building predictive smell models for virtual reality environments. Informatics and Automation. 2025. vol. 24. no. 2. pp. 556–582. DOI: 10.15622/ia.24.2.7.
23. Kruk A. The benefits of virtual learning environment (VLE) in teaching ESP. Current nutrition in the humanities. 2022.

24. Hung N., Dat P.T., Tan N., Quan N.A., Trang L., Nam L.M., et al. Heverl–viewport estimation using reinforcement learning for 360-degree video streaming. Informatics and Automation. 2025. vol. 24. no. 1. pp. 302–328.
25. Nguyen H., Dao T.N., Pham N.S., Dang T.L., Nguyen T.D., Truong T.H. An accurate viewport estimation method for 360 video streaming using deep learning. EAI Endorsed Trans. Ind. Networks Intell. Syst. 2022. vol. 9. no. 4.
26. Kumar M., Rathi G., Singh T., NL T. CNN based virtual whiteboard application. Second International Conference on Advances in Information Technology (ICAIT). IEEE, 2024. pp. 1–6.
27. Heemskerk I., Kuiper E., Meijer J. Interactive whiteboard and virtual learning environment combined: Effects on mathematics education. Journal of Computer Assisted Learning. 2014. vol. 30. no. 5. pp. 465–478.
28. Mundargi Z., Das S., Shinde A., Deokar O., Bahirat S., Shetiya D. Hand gesture desktop control with python. 2nd International Conference on Advances in Computation, Communication and Information Technology (ICAICCIT). IEEE, 2024. vol. 1. pp. 35–40.
29. Nguyen H.A., Tran T.T., Ho H.Q., Ngo T.D., Vu K.N., Huynh V.L.T. Hand gesture recognition using cvzone. Proceedings of the 9th International Conference on Intelligent Information Technology. 2024. pp. 108–113.
30. Uke S., Shaikh A., Rayate H., Kamble A., Rahane S. Towards touchless interaction: Implementing hand gesture recognition for presentation and media control. International Conference on Emerging Smart Computing and Informatics (ESCI). IEEE, 2025. pp. 1–6. DOI: 10.1109/ESCI63694.2025.10988099.
31. Vasanthagokul S., Kamakshi K.V.G., Mudbhari G., Chithrakumar T. Virtual mouse to enhance user experience and increase accessibility. 4th International Conference on Inventive Research in Computing Applications (ICIRCA). IEEE, 2022. pp. 1266–1271.
32. Naidu P., Muthukumaran N., Chandralekha S., Reddy K.T., Vaishnavi K.S. An analysis on virtual mouse control using human eye. 5th International Conference on Image Processing and Capsule Networks (ICIPCN). IEEE, 2024. pp. 233–237.
33. Praba B.V., Vinothini R., Jayarathna M., Subramani K., Sravanthi P. Virtual AI mouse with biometric authentication. International Conference on Knowledge Engineering and Communication Systems (ICKECS). IEEE, 2024. vol. 1. pp. 1–5. DOI: 10.1109/ICKECS61492.2024.10617264.
34. Karthick S., Dinesh M., Jeffery Dani Raj C., Jayapandian N. Artificial intelligence based enhanced virtual mouse hand gesture tracking using yolo algorithm. IEEE 2nd International Conference on Data, Decision and Systems (ICDDS). IEEE, 2023. pp. 1–6. DOI: 10.1109/ICDDS59137.2023.10434330.
35. Karpov A., Ronzhin A., Kipyatkova I. An assistive Bi-modal user interface integrating multi-channel speech recognition and computer vision. Human-Computer Interaction: Interaction Techniques and Environments (HCI 2011). Lecture Notes in Computer Science. 2011. vol. 6762. pp. 454–463. DOI: 10.1007/978-3-642-21605-3_50.
36. Karpov A., Carbini S., Ronzhin A., Viallet J.E. Comparison of two different similar speech and gestures multimodal interfaces. Proc. of the 16th European Signal Processing Conference (EUSIPCO). 2008. pp. 1–5.
37. Jaimes A., Sebe N. Multimodal human–computer interaction: A survey. Computer Vision and Image Understanding. 2007. vol. 108. no. 1–2. pp. 116–134.
38. Bazarevsky V., Zhang F. On-device, real-time hand tracking with mediapipe. Available at: https://research.google/blog/on-device-real-time-hand-tracking-with-mediapipe/ (accessed 15.01.2026).
39. Manning C.D., Raghavan P., Scutze H. Introduction to Information Retrieval. Cambridge University Press, 2008.

40. Sokolova M., Lapalme G. A systematic analysis of performance measures for classification tasks. Information Processing & Management. 2009. vol. 45. no. 4. pp. 427–437.
41. Powers D.M.W. Evaluation: From precision, recall and f-measure to roc, informedness, markedness & correlation. Journal of Machine Learning Technologies. 2011. vol. 2. no. 1. pp. 37–63.

Хунг Нгуен Вьет — Ph.D., преподаватель, международный институт подготовки кадров и сотрудничества, Восточноазиатский технологический университет. Область научных интересов: мультимедийные коммуникации, сетевая безопасность, искусственный интеллект, управление трафиком в сетях следующего поколения, гарантия качества обслуживания (QoE/QoS) для сетевых сервисов, экологичные сети, приложения. Число научных публикаций — 35. hungnv@eaut.edu.vn; Ки Ань, Хатинь, Вьетнам; р.т.: +84(098)911-2079.

Хуинь Фи Динь — научный сотрудник, факультет информационных технологий, Восточноазиатский технологический университет. Область научных интересов: обработка естественного языка, машинное и глубокое обучение, компьютерное зрение. Число научных публикаций — 2. 20222072@eaut.edu.vn; Хуонг Нгай – Тхач Тхат, Ханой, Вьетнам; р.т.: +84(037)395-0094.

Тунг Ма Ван — научный сотрудник, международный институт обучения и сотрудничества, Восточноазиатский технологический университет. Область научных интересов: искусственный интеллект, машинное и глубокое обучение, виртуальная реальность. Число научных публикаций — 1. 20230240@eaut.edu.vn; Минь Куанг, Туенкуанг, Вьетнам; р.т.: +84(092)882-2756.

Ву Нгуен Ван — научный сотрудник, факультет информационных технологий, Восточноазиатский технологический университет. Область научных интересов: машинное и глубокое обучение, компьютерное зрение. Число научных публикаций — 1. 20222048@eaut.edu.vn; Куок Оай, Ханой, Вьетнам; р.т.: +84(038)379-0656.

Дат Нгуен Фу — научный сотрудник, факультет информационных технологий, Восточноазиатский технологический университет. Область научных интересов: обработка естественного языка, искусственный интеллект, компьютерные сети. Число научных публикаций — 1. 20220344@eaut.edu.vn; Сон Чунг – Куок Оай, Ханой, Вьетнам; р.т.: +84(096)548-3341.