Real Time Cursor Control Using Hand Gestures with Vision Transformer

Ketan Kaiwart[1], Pratik Manjarekar[2], Prasad Khanapure[3]
[1,2,3]*Computer Engineering, RMD Sinhgad School of Engineering®, Pune, India*

***Abstract*—This paper present real-time cursor control using hand gestures with vision transformer for contactless human computer interaction (HCI). The proposed system uses webcam for capturing live hand movement, while Media Pipe can efficient hand detect & region-of-interest (ROI) extraction. The extracted hand images are pre-processed and classified into gesture classes using a pretrained Vision Transformer model fine-tuned on a custom gesture dataset. The cursor actions such as movement, click, drag and scrolling are mapped using recognized gestures through PyAutoGUI. The system follows pipeline consist hand detection, preprocess the input data, gesture classification and cursor actions mapping. To improve performance in real time environment use smoothing and gesture debouncing techniques. The model is train using two phase transfer learning strategy, in that classification head is initially train with frozen backbone followed by full fine tuning of the model. Experimental evaluation Conduct on custom dataset content images across 7,248 for 7 gesture classes. The system achieves 99.91% classification accuracy on the static dataset and 93.52% accuracy during real-time, the accuracy in real time conditions get vary. The results demonstrate that Vision Transformers effectively capture global spatial relationships between hand features, providing a robust and efficient alternative to traditional input devices for contactless computer applications.**

***Index Terms*—Virtual Mouse, Gesture Recognition, Human-Computer Interaction (HCI), Computer Vision, CNN, Vision Transformer (ViT).**

I. INTRODUCTION

Human Computer Interaction systems traditionally depend on physical devices such as keyboard, touchpad, mouse,mic for communication between the users and the computer. These devices efficient interaction, but required direct physical communication, which is not suitable contactless operation. The increasing demand for natural and contactless interaction systems has accelerated research in gesture-based interfaces using computer vision and artificial intelligence techniques.

Users to interact with computers through movement without depend on hardware devices using hand gesture recognized. Early gesture-controlled systems used colour-based tracking, contour detection, and handcrafted feature extraction methods. While these approaches achieved acceptable performance in controlled environments, but they highly sensitive background complexity, different conditions. Later on deep learning approaches based on Convolutional Neural Networks (CNNs) improved gesture recognition accuracy by learning hierarchical visual features automatically. But cnn still face challenges in capturing long-range spatial dependencies & global relationships between hand features.

In computer vision applications, the vision transformers has powerful alternative for CNN architecture. Where the vision transformer use self-attention mechanism to model the global contextual relationship among the image patches, while CNN rely on the local convolution operations. ViTs highly suitable for gesture recognition tasks because of it have ability to capture spatial dependencies between different regions of the hand and relative positioning of fingers plays critical role in classification.

In this work, the real time cursor control using hand gestures uses Vision Transformer for gestures classification model. The system integrates with MediaPipe for hand detection & ROI extraction, OpenCV for frame capture & preprocessing, PyAutoGUI for control cursor.The dataset contain custom seven gestures classes that used for fined-tune a pretrained vision transformer model using two phase transfer learning strategy.

To improve performance in real time environment use smoothing and gesture debouncing techniques and confidence thresholding mechanisms for stable and reliable interaction. Experimental evaluation demonstrates that the approach achieves high classification accuracy on static test data while maintaining effective real-time performance during live interaction.

II. LITERATURE REVIEW

Gesture-based human–computer interaction systems have gained significant attention as an alternative to traditional input devices such as keyboards and mice. Early virtual mouse systems primarily relied on traditional computer vision techniques including color-based tracking, contour extraction, and object detection for recognizing hand movements [1], [6]. Although these approaches enabled basic cursor control functionality, so performance was vary on different conditions, background in real-time environment.

As machine learning and deep learning techniques are get advance, gesture recognition systems began incorporating frameworks such as OpenCV, MediaPipe, and Convolutional Neural Networks (CNNs). MediaPipe-based systems improved real-time hand tracking by detecting 21 hand landmarks with low computational cost, enabling efficient gesture recognition and cursor interaction [2], [3]. Several studies combined MediaPipe with OpenCV and PyAutoGUI to implement cursor movement, clicking, dragging, and scrolling operations using predefined hand gestures [4], [5]. These approaches demonstrated improved usability and reduced dependency on handcrafted features.

CNN-based gesture recognition systems further enhanced classification accuracy by automatically learning spatial features from hand images [5], [8]. These systems achieved strong performance in static gesture classification tasks and demonstrated practical implementations for virtual mouse systems. However, CNNs primarily focus on local receptive fields and often require deeper architectures to capture long-range spatial dependencies between different hand regions. As a result, their performance drops under varying conditions, background complexity, and dynamic real-time interaction scenarios.

Recently, for CNNs based computer vision applications get strong alternative that was Vision Transformers (ViTs). Unlike CNNs, Vision Transformers process images as sequences of patches and utilize self-attention mechanisms to capture global contextual relationships across the entire image [9]. This enables improved understanding of spatial relationships between fingers and hand regions, making ViTs highly suitable for gesture recognition tasks. Several recent studies have demonstrated the effectiveness of Vision Transformers in image classification and object detection applications [7], [8]. Comparative analyses between ViT, Swin Transformer, and Transformer-in-Transformer (TNT) architectures further highlight the capability of transformer-based models to achieve superior feature representation and scalability [11]. Additional research on lightweight transformer architectures and hybrid attention mechanisms has shown promising improvements for small-scale datasets and real-time computer vision systems [12], [13].

Despite these advancements, limited research has focused on integrating lightweight Vision Transformer architectures into real-time gesture-controlled cursor systems with practical deployment considerations. Many existing systems either rely on rule-based gesture recognition or do not evaluate real-time usability under dynamic interaction conditions. Furthermore, few studies investigate smoothing mechanisms, gesture debouncing, and real-time performance variations between static evaluation and live deployment.

Motivated by these limitations, the proposed work presents a real-time gesture-based cursor control system using a Vision Transformer for robust hand gesture recognition. The system integrates MediaPipe-based hand detection, Vision Transformer-based gesture classification, and real-time cursor interaction mechanisms to provide accurate and efficient contactless human–computer interaction.

III. METHODOLOGY

The proposed system follows a modular pipeline for real time cursor control using hand gestures use Vision Transformer (ViT). The overall architecture consists of five major stages: frame acquisition, hand

detect, preprocessing input data, gesture classification, and cursor control.

3.1 System Architecture

The architecture of the proposed system is shown in Fig. 1. The system starts by capturing real-time video frames using a webcam through OpenCV. And by using MediaPipe hand framework each frame is processed, which detects hand landmarks and extracts the region of interest (ROI) containing the hand.

The extracted ROI is normalized using ImageNet preprocessing statistics & resized to 224×224 pixels before being passed to the Vision Transformer model. The system uses the vit_tiny_patch16_224 architecture pretrained on ImageNet for gesture classification.

The Vision Transformer divides the input image into fixed-size image patches and then converts into patch embedding. Then embeddings are processed through transformer encoder blocks which contain multi-head self-attention mechanisms and feed-forward layers. The self-attention mechanism enables the model to capture global spatial relationships between different hand regions and finger positions to improve gesture recognition performance.

The vision transformer is passed through fully connect classification layer to predict predefined gesture classes by generating output classification token. The predicted gesture mapped the cursor operation using PyAutoGUI, which enables real time interaction with OS.

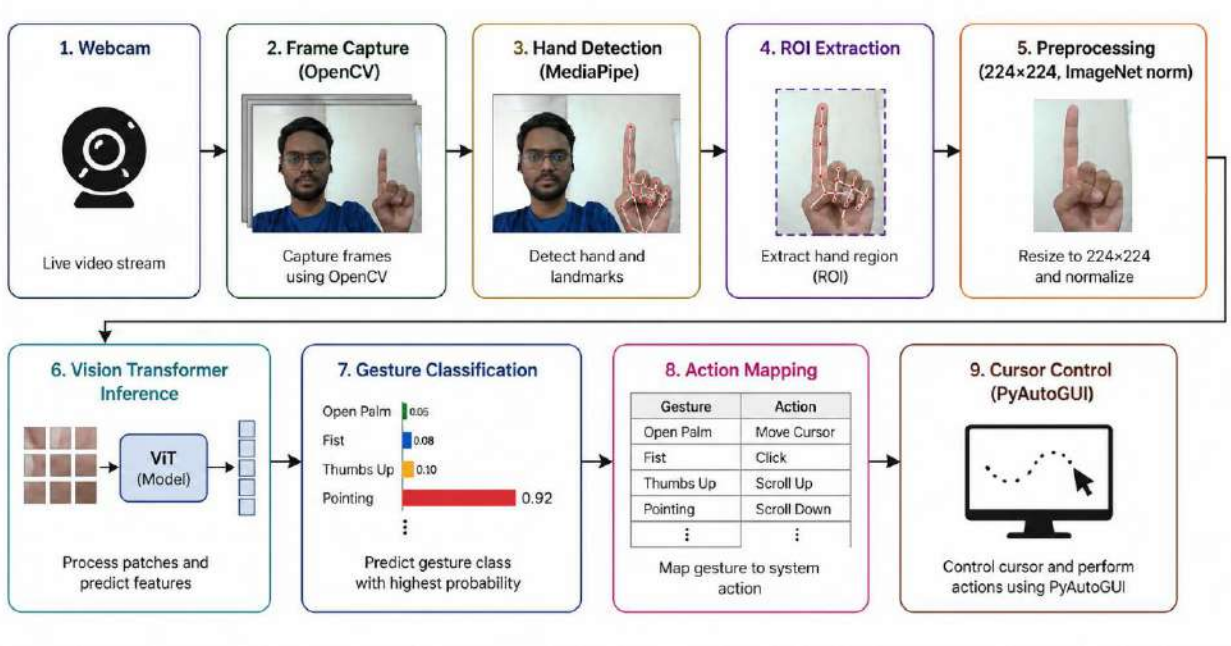

Fig.(1) System Architecture

3.2 Gesture Classes and Actions

The system supports seven gesture classes, each mapped to a specific cursor operation, as shown in Table 1.:

Gesture	Cursor Action	Details
Open Palm	Move Cursor	Cursor follows your index fingertip position
Index Point	Left Click	Triggers after 2 steady frames(debounce)
Two Fingers Up	Right Click	Same debounce as left click
Fist	Neutral / Stop	No action—pause cursor control
Pinch	Drag	Holds left mouse button; move hand to drag
Three Fingers Up	Scroll Up	Scroll up per trigger
Three Fingers Down	Scroll Down	Scroll down per trigger

Table 1: Gesture Classes and Corresponding Cursor Actions

- Collection: Custom images captured via webcam using MediaPipe hand ROI extraction
- Split: 70% train / 15% validation / 15% test
- Augmentation: Random flip, rotation (±15°), color jitter, Gaussian blur

3.3 Vision Transformer Architecture

The architecture of the Vision Transformer used in this system is shown in Fig. (2).

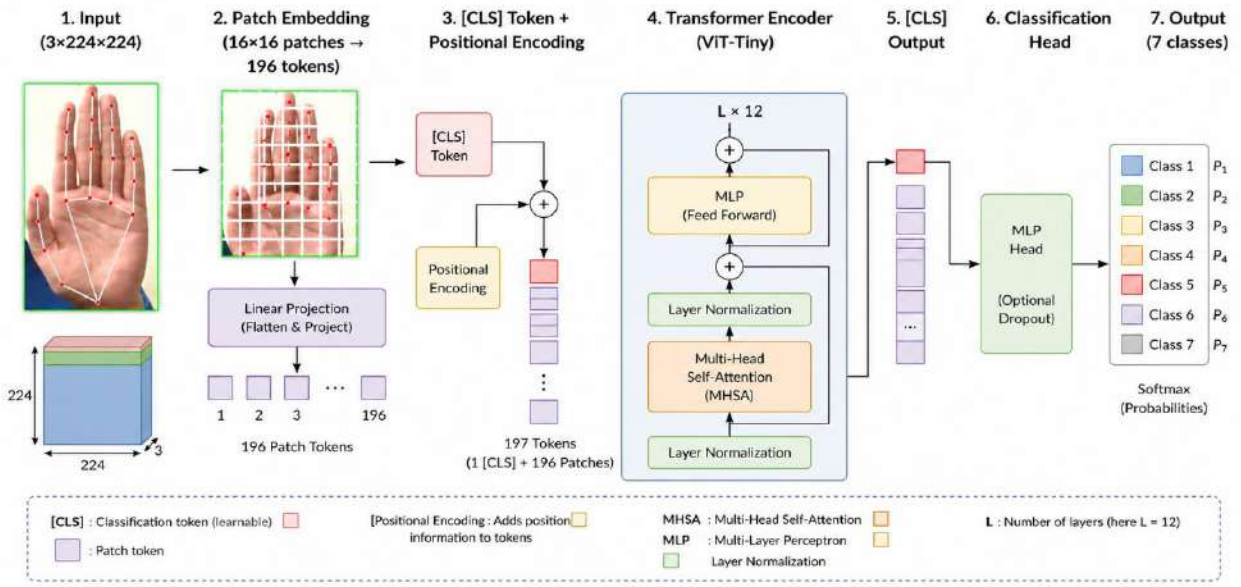

Fig.(2) Vision Transformer Architecture

- Base Model: vit_tiny_patch16_224 (pretrained on ImageNet)
- Parameters: ~5.53 million
- Training Phase 1: Freeze backbone — train classification head only (5 epochs)
- Training Phase 2: Unfreeze backbone — full fine-tuning with 10× lower LR
- Optimizer: AdamW (LR=0.0001, weight_decay=0.0001)
- Scheduler: CosineAnnealingLR
- Loss: CrossEntropyLoss (label_smoothing=0.1)

3.4 Model Training

Initialized ViT model with ImageNet-pretrained weights. And two-phase training strategy was used:

- Phase 1: In first training phase, the pretrained Vision Transformer backbone was freeze while only the classification head was trained. This helped the model learn features that're specific to hand gestures. It did not change the pretrained ImageNet representations much.
- Phase 2: In the second phase, unfroze all the model layers & fine-tuned them using a reduced learning rate. This enabled the transformer attention layers to adapt more effectively to hand gesture recognition while preserving pretrained visual representations.

The model is using the AdamW optimizer with a learning rate of 0.0001 and cosine annealing scheduler to training. Cross-entropy loss with label smoothing is used to improve generalization.

3.5 Real-Time Optimization

To ensure stable cursor movement and avoid unnecessary actions, smoothing and debouncing techniques are used. A gesture debouncing mechanism make sure that actions are triggered only after consistent predictions across multiple frames and moving average filter is used to reduce cursor jitter.

IV. EXPERIMENTAL SETUP AND IMPLEMENTATION:

4.1 Hardware and Software Environment

The system is implemented using Python and tested on a machine with standard computing resources. A webcam is used for real-time video capture. For implementation PyTorch is used which is frame work

in deep learning for training and inference of the Vision Transformer model.

The software stack includes:

- PyTorch and timm library for Vision Transformer implementation
- MediaPipe for real-time hand detection and landmark extraction
- OpenCV for video capture and image preprocessing
- PyAutoGUI for cursor control operations
- NumPy and Matplotlib for data processing and visualization

4.2 Dataset Preparation

A webcam-based data collection is use to create custom dataset. The MediaPipe framework is used to detect the hand and extract the region of interest (ROI) for each frame. The dataset contain seven gesture classes refers to different cursor actions.

The collected dataset is divided into:

- Training set: 70%
- Validation set: 15%
- Test set: 15%

Data augmentation techniques, like rotation, horizontal flipping, color jitter and Gaussian blur are used to improve generalization of the Vision Transformer model.

4.3 Model Configuration

The system uses a Vision Transformer model (ViT-tiny-Patch16-224 model). This model was pre-trained on ImageNet. The lightweight ViT-Tiny model was use to balance high classification accuracy with efficient real-time inference. The size of input image are fixed at 224×224 pixels.

Training is performed using the following configuration:

- Optimizer: AdamW
- Learning Rate: 0.0001
- Weight Decay: 0.0001
- Loss Function: Cross-Entropy Loss with label smoothing
- Scheduler: Cosine Annealing

A two-phase fine-tuning strategy is applied:

- Phase 1: Training only the classification head with frozen backbone
- Phase 2: Full model fine-tuning with a reduced learning rate

4.4 Implementation Pipeline

The real-time system follows a sequential pipeline:

Webcam → Frame Capture → Hand Detection (MediaPipe) → ROI Extraction → Preprocessing → Vision Transformer Inference → Gesture Classification → Cursor Action (PyAutoGUI)

To ensure smooth interaction, additional components are implemented:

- Cursor smoothing using a moving average filter
- Gesture debouncing to prevent repeated unintended actions
- Confidence thresholding for reliable predictions

4.5 Evaluation Procedure

Standard performance indicators, such as accuracy, precision, recall, and F1-score, are used to assess the trained model on the test dataset. Inference latency and frames per second (FPS) are used to measure real-time performance.

In order to confirm that cursor actions like movement, clicking, dragging, and scrolling are accurate, the evaluation also includes qualitative testing of the real-time system.

V. RESULTS

The proposed Vision Transformer-based gesture recognition system was evaluated using both static test dataset evaluation and real-time live interaction testing. The evaluation focused on classification accuracy, real-time responsiveness, latency, and system stability for practical human–computer interaction applications.

5.1 Classification Performance

In the static test dataset all evaluation metrics for trained Vision Transformer model achieved high performance. The final model achieved:

- Accuracy: 99.91%
- Precision (macro): 99.91%
- Recall (macro): 99.92%
- F1-Score (macro): 99.91%

The model was evaluated on a test set which contains 1,093 images across seven gesture classes. The three_fingers_down gesture was incorrectly classified as fist this is only one misclassification was observed during testing. It shows strong capability of the Vision Transformer model in learning different gesture representations.

The confusion matrix analysis shows the most gesture classes like open_palm, pinch, and two_fingers_up, index finger, three_fingers_up, three_fingers_down are achieved near-perfect classification performance.The self-attention mechanism of the Vision Transformer, which effectively captures global spatial relationships between hand features.

5.1.1 Ablation Study

The ablation study shows the vision transformer model can achieve high classification performance across multiple training configuration. Table 2 shows ablation study.

Freeze Epochs	Total Epochs	Fine-Tuning Epochs	Test Accuracy	F1-Score	Training
5	10	5	99.91%	99.91 %	7m 49s
5	15	10	99.91%	99.91 %	9m 39s
5	30	25	99.91%	99.91 %	31m 53s
8	10	2	99.18%	99.19 %	6m 12s
8	15	7	99.91%	99.91 %	18m 7s
8	30	22	99.91%	99.91 %	25m 19s

Table 2: Hyperparameter Ablation Study Results

5.2 Real-Time Performance

In addition to static dataset evaluation, the system was tested under real-time webcam interaction conditions to evaluate practical deployment performance in real-time environment.

The system achieved real-time accuracy in different conditions as follow :

Session	Frame Evaluated	Real-Time Accuracy
Session 1	1636	93.52%
Session 2	1693	81.87%
Session 3	1752	82.59%
Session 4	1491	71.97%
Session 5	1603	85.56%

Table 3: Real-Time accuracy across multiple sessions

The real-time evaluation, a live gesture interaction session was conducted in which continuous cursor control operations such as movement, clicking, dragging, and scrolling.

The Vision Transformer model help for responsive interaction & maintained efficient inference performance with low latency.

5.3 System Behavior

To improves system performance, it used smoothing and debouncing mechanisms. For the smoother pointer movement, average filter reduces cursor jitter. The gesture debouncing mechanism prevents unwanted multiple clicks by requiring consistent gesture predictions over consecutive frames.

5.4 Discussion

The experimental results shows that for real-time human computer interaction (HCI) system & recognized gestures, the vision transformer is highly effective. As compare with CNN based computer vision application & recognized gesture, the proposed system gives benefits improved global feature representation through self-attention mechanisms.

And the model achieved nearly perfect performance on static test dataset, & some reduction was observe during real-time evaluation. This performance gap expected in real-time environment conditions.

The results shows that in real-time environment the proposed system successfully balances classification accuracy, computational efficiency. The lightweight Vision Transformer architecture combined with MediaPipe-based ROI extraction enables efficient gesture recognition while maintaining practical real-time interaction performance.

For future gesture-controlled computing applications, the proposed system provides a robust and contactless alternative to conventional input devices and demonstrates the potential of Vision Transformers.

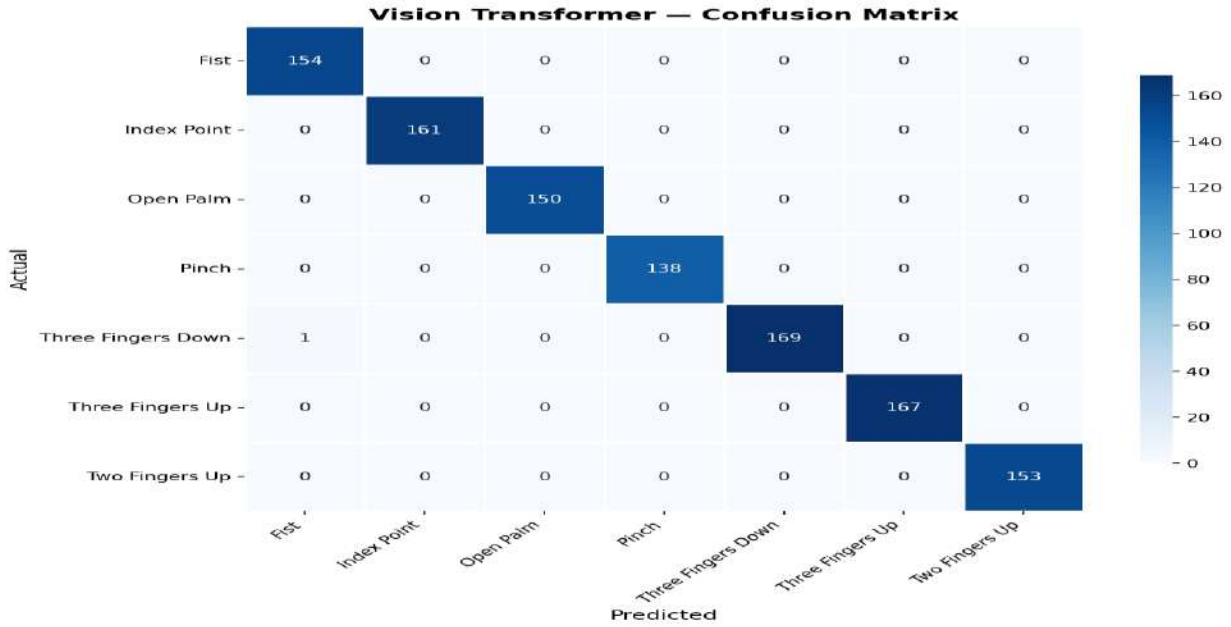

Fig.(3) Confusion Matrix

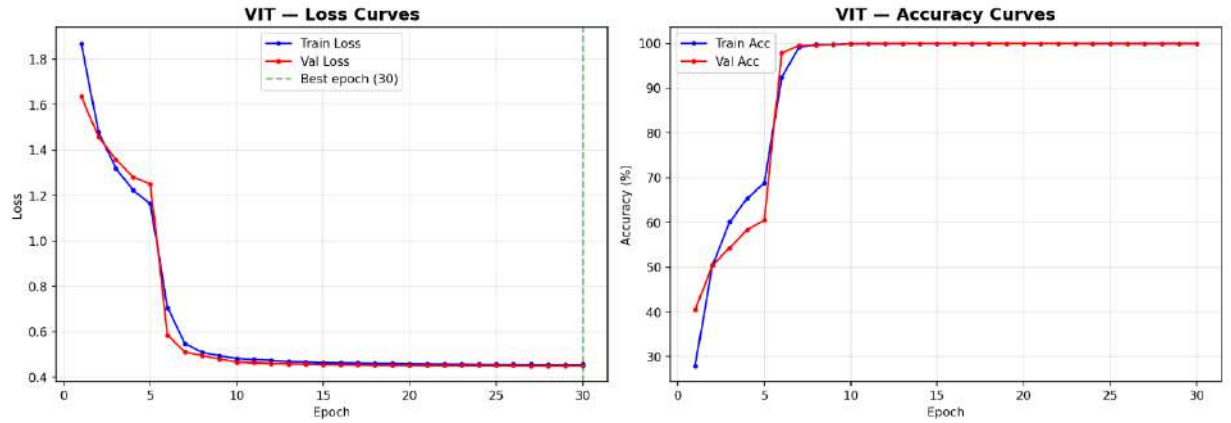

Fig.(4) Training Curves

VI. CONCLUSION AND FUTURE SCOPE

This paper presented a real-time cursor control with gesture-based system using a Vision Transformer for accurate and efficient hand gesture recognition in contactless human–computer interaction applications. The system was integrated with MediaPipe for hand detection and ROI extraction, OpenCV for real-time frame processing, and PyAutoGUI for operating cursor control, forming a complete end-to-end gesture interaction framework.

A lightweight Vision Transformer (vit_tiny_patch16_224) pretrained on ImageNet was fine-tuned using a custom multi-hand gesture dataset containing seven gesture classes corresponding to different cursor operations. The self-attention mechanism of the Vision Transformer enabled effective learning of global spatial relationships between hand features, resulting in robust gesture classification performance.

Experimental evaluation demonstrated that the proposed system achieved 99.91% accuracy on the static test dataset while maintaining effective real-time interaction performance with up to 93.52% live gesture recognition accuracy. The integration of cursor smoothing, gesture debouncing, and confidence thresholding mechanisms significantly improved interaction stability and reduced unintended cursor actions during real-time operation.

The results show that Vision Transformers are a choice instead of traditional systems that use CNN for recognizing gestures in real time when people interact with computers. The system we proposed works well because it balances how accurate it is, how fast it is and how easy it is to use, which shows that transformer-based architectures can be used for applications that are controlled by gestures.

Even though the system works well in time there are still some problems like it can be affected by changes in lighting, blurry motion and when people move their hands really fast. To make the system better we can work on making it stronger, in environments make it use less resources so it can work on simpler devices and make it recognize gestures that are changing and let many people interact with it at the same time. We can also look into using transformer architectures that use less resources and systems that use many ways for people to interact with computers, which can make interacting with computers in real time even better.

REFERENCES

[1] M. Ranawat, R. Shankarmani, M. Rajadhyaksha, and N. Lakhani, "Hand Gesture Recognition Based Virtual Mouse Events," in 2021 2nd International Conference for Emerging Technology (INCET), Belgaum, India, May 21–23, 2021, pp. 1–4. doi: 10.1109/INCET51464.2021.9456388.

[2] G. Jagnade, M. Ikar, N. Chaudhari, and M. Chaware, "Hand Gesture-based Virtual Mouse using Open CV," in 2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), 2023, pp. 820–825. doi: 10.1109/IDCIoT56793.2023.10053488.

[3] P. C. D. Kalaivaani, R. Kumaresan, A. J. Manow Ranjith, and S. Nandha Kumar, "Hand Gestures for Mouse Movements with Hand and Computer Interaction model," in 2023 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, INDIA, Jan. 23–25, 2023. doi: 10.1109/ICCC156745.2023.10128273.

[4] Apoorva, R. Rani, A. Shankar, G. Jaiswal, A. Bondia, and A. Sharma, "Gesture-Controlled Virtual Mouse and Finger Air Writing," in 2024 14th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2024, pp. 370–375. doi: 10.1109/CONFLUENCE60223.2024.10463446.

[5] N. Balakrishnan, T. Krishnamoorthi, K. Aakash, and A. S. Sathishkumar, "Virtual Mouse Using Machine Learning," in 2025 3rd International Conference on Communication, Security, and Artificial Intelligence (ICCSAI), 2025, pp. 1831–1835. doi: 10.1109/ICCSAI64074.2025.11064183.

[6] M. Shetty, C. A. Daniel, M. K. Bhatkar, and O. P. Lopes, "Virtual Mouse Using Object Tracking," in 2020 5th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2020, pp. 548–553. doi: 10.1109/ICCES48766.2020.9137854.

[7] G. E. Rani, M. Sakthimohan, S. S. Surya, S. Kalaiselvi, T. Bernice, and V. Gunasekran, "Cursor Movement Based on Object Detection Using Vision Transformers," in 2023 2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies (VITECON), 2023. doi: 10.1109/VITECON58111.2023.10157042.

[8] S. Sudarshan, A. Kavitha, Mohana, K. Nataraj, G. Ambika, and B. S. Sudhangowda, "Object Detection using Vision Transformer and Deep Learning for Computer Vision Applications," in Proceedings of the 7th International Conference on Intelligent Sustainable Systems (ICISS-2025), 2025, pp. 1524–1529. doi: 10.1109/ICISS63372.2025.11076364.

[9] A. Dosovitskiy et al., "An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale," arXiv:2010.11929v2 [cs.CV], 2021.

[10] A. Berroukham, K. Housni, and M. Lahraichi, "Vision Transformers: A Review of Architecture, Applications, and Future Directions," in 2023 7th IEEE Congress on Information Science and Technology (CiSt), 2023, pp. 205–210. doi: 10.1109/CiSt56084.2023.10410015.

[11] D. M. Sundaram, Y. Kasukurthi, and S. Jayachandran, "Comparative Study of Vision Transformer (ViT), Swin Transformer, and Transformer in Transformer (TNT) on Image Classification," in 2025 5th International Conference on Soft Computing for Security

Applications (ICSCSA), 2025, pp. 1469–1473. doi: 10.1109/ICSCSA66339.2025.11170958.

[12] A. Verma, S. R. Dubey, and S. K. Singh, “Patch Attention Excitation Based Vision Transformer for Small-Sized Datasets,” in ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2025. doi: 10.1109/ICASSP49660.2025.10888545.

[13] Z. Gong and M. Chanmean, “Multi-Scale Hybrid Attention Integrated with Vision Transformers for Enhanced Image Segmentation,” in 2024 2nd International Conference on Algorithm, Image Processing and Machine Vision (AIPMV), 2024. doi: 10.1109/AIPMV62663.2024.10691911.