Real-Time Hand Gesture Control for Touchless Computer Interaction and File Transfer

P. Somaraju[1], B. Vaishnavi Keerthi[2], P. Hemanth Kumar[3], G. Sai Sruthi[4], T. Leela Venkata Sasikiran[5]

[1] Associate Professor, School of Computer Studies Dr. RVR NRI Institute of Technology, deemed to be University, Agiripalli, Vijayawada, A.P.–521212, India

[2,3,4,5] UG Student, Department of CSE NRI Institute of Technology, Agiripalli, Vijayawada, A.P.–521212, India

Abstract—**Huma.n-computer interaction is steadily drifting away from depe.nde.nce on conve.ntional peripherals such as the mouse a.nd keyboard, pushed by a growing dema.nd for hygie.nic, intuitive, a.nd ha.nds-free digital e.nvironme.nts. Conve.ntional ways of moving file.s betwee.n two computers — USB drive.s, cloud storage, or ma.nual network share.s — remain slow a.nd typically call for extra hardware, a.n active internet connection, or several ma.nual steps. This paper de.scribe.s a real-time, vision-based ge.sture-control system that addre.sse.s both everyday computer operation a.nd inter-device file sharing through ha.nd moveme.nt alone. A sta.ndard webcam capture.s live ha.nd motion, which a ha.nd-la.ndmark detection model converts into finger positions in three-dime.nsional space. The.se la.ndmarks are tra.nslated into system-level actions: pointing steers the cursor, a thumb–index pinch performs a click, a.n ope.n palm e.nable.s scrolling, a.nd lateral ha.nd motion triggers swipe-based navigation. Building on this foundation, the system adds a.n intuitive grab-a.nd-release mecha.nism for file tra.nsfer — closing the ha.nd into a fist over a selected file "picks it up," a.nd the file is tra.nsmitted the insta.nt a.n ope.n-palm "release" ge.sture is detected on the receiving machine, mirroring the physical act of ha.nding over a.n object. This de.sign remove.s the need for physical storage media, ma.nual network setup, or extra steps, while the interaction stays fully touchle.ss a.nd intuitive throughout. The system illustrate.s how ge.sture recognition, computer vision, a.nd wirele.ss communication ca.n be combined into one cohe.sive interface, with practical releva.nce to smart office.s, healthcare settings, classrooms, a.nd other collaborative or sterile e.nvironme.nts..**

Index Terms—**Huma.n-Computer Interaction (HCI), Touchle.ss Interface, Ge.sture Recognition, Ha.nd La.ndmark Detection, Computer Vision, Real-Time Ge.sture Tracking, Virtual Mouse Control, Peer-to-Peer File Tra.nsfer, Grab-a.nd-Release Ge.sture, Wirele.ss Data Communication, Contactle.ss Computing, Sterile Interface De.sign**

I. INTRODUCTION

For a long time, people have used physical devices like the mouse and keyboard to connect with computers. While this has worked, it has also created problems in situations when touch-based input is uncomfortable, unsanitary, or just lacks naturalness. Because of this, there has been a rise in the popularity of touchless interfaces, especially in settings where people are under increasing pressure to avoid touching shared gadgets, such as in labs, hospitals, and office cubicles. In addition, commonplace tasks like transferring files between computers still require either manual network configuration, cloud uploads, or external storage, all of which add extra steps, hardware dependencies, or dependencies on connectivity that may or may not be accessible.

Thanks to recent developments in computer vision and real-time hand-tracking models, hand movement can now be accurately understood, paving the way for gesture-based control that feels more natural than traditional input. Inspired by this, the current study suggests a vision-based system that utilises a regular webcam to interpret two sets of hand gestures simultaneously: one set of gestures for basic computer control (such as moving the cursor, clicking, and scrolling), and another set of gestures for a more natural way to share files, wherein one user "grabs" a file with a closed fist and "releases" it onto another computer with an open palm gesture, mimicking the natural act of passing an object from one person to another. This effort aims to show a reasonable, inexpensive, and touchless alternative to computing that relies on peripherals by integrating interface control and file transmission into a single gesture-driven system.

II. LITERATURE SURVEY

There is a mass of literature, much of it recent, investigating the idea of substituting the physical mouse with hand tracking through vision. A virtual-mouse pipeline is described in a 2025 study, which is based on OpenCV, MediaPipe, and PyAutoGUI, and uses the 21 MediaPipe hand landmarks to interpolate the index fingertip to screen coordinates and identifies raised-finger combinations to do left-clicks, right-clicks, double-clicks, and scrolling [1].

Building upon the principle of taking a static gesture and transforming it into a dynamic gesture, a 2025 study used a combination of MediaPipe landmark extraction and a Transformer model to identify eight gesture classes, such as swipe-up, swipe-down, click, and enlarge, placing such gestures as direct substitutes of using a mouse and a keyboard in media-control and presentation applications [2].

An IEEE conference publication has reported a MediaPipe-based virtual mouse, which extends beyond cursor control to map the known gestures to system functions like volume and brightness control in addition to click, scroll and drag-and-drop and is reported to work well in applications such as presentations given with a VR headset on [3].

A 2026 paper positions the identical issue as completely markerless: 21-point MediaPipe landmarks are positioned into a geometric interpreter that is rule-based and enables cursor movement, left- and right-clicks, and scrolling without a physical marker or classifier trained [4].

An independent IEEE conference paper suggests an OpenCV/MediaPipe virtual mouse with an extended control space beyond the simple pointer functionality, to bring it nearer to a real mouse and keyboard [5].

A 2024 paper constructed fully on OpenCV and MediaPipe also describes a real-time, no-contact mouse-replacement pipeline and notes better ease of use among individuals who find a traditional mouse or keyboard challenging to use [6].

On the file-sharing half of touchless interaction, a 2025 hackathon project, AirShare, achieves a touch-free, gesture-controlled file-sharing web application where file selection and file transfer are wirelessly over Wi-Fi or a local network without relying on third-party cloud services [7].

Huawei became the first to commercially feature the concept as the HarmonyOS-Next Air Gesture feature, announced on the Mate 70 series, which allows a user to transfer a photo or file between two devices by moving it in the air using a grab-and-drop gesture, which Huawei itself described as a step towards gesture-first interaction [8].

Same Huawei feature coverage reports that it works on the devices in the company ecosystem, including phones, tablets, laptops and smart TVs, but without significant impact on power consumption that would affect battery life [9], [10], [11], [24].

A larger 2025 industry map of gesture recognition surveys its application in industries and isolates touchless interfaces in hazardous or sterile settings, and in healthcare, e.g. rehabilitation assistance or hands-free operation of assistant and monitoring devices [12].

A medical-based study in 2024 suggests a lightweight 1D-CNN pipeline that identifies gestures related to everyday-life routines, targeting patients and clinicians who have the benefit of having less physical contact with communal equipment [13].

In the vicinity of the operating room, a more recent study suggests a touchless intraoperative medical image-browsing system that uses hand gesture recognition by a single RGB camera, and does not require additional hardware or per-user training, allowing a surgeon to look at imaging without breaking sterilization [14].

Gesture-based robot interaction work, also considers the hand as a direct interface to communication, with vision-provided tracking converting operator gestures into commands on the robot side without an intermediate physical controller [15].

An IEEE Transactions on Artificial Intelligence article views touchless gesture recognition as an algorithmic pipeline that is tested against public benchmark datasets, highlighting the extent to which the reliability of such systems is conditioned by the variety of data that the underlying detector was trained on [16].

In a 2025 PLOS One article, a CNN-based gesture recognizer is described based on a Mobilenet backbone with a multi-scale convolutional module, that attains high accuracy on the NUS-II, Creative

Senz3D and ASL-M benchmarks and has an inference lightweight enough to support real-time interaction [17].

Creative and educational input Creative and educational input Gesture recognition has also been used on creative and educational input, but not just to control devices: a 2025 study uses an air-canvas interface with optical character recognition such that handwriting drawn in mid-air is converted the same way into digital notes [18], a 2024 study is an Air-Canvas system that supports drawing, erasing and navigating slides by gesture alone [19], a 2025 study uses a similar air-drawing interface around colour-marker tracking and HSV segmentation [20], a 2025 web-based application re-renders the same air-canvas concept as a browser-accessible, hands-free drawing surface [21], and a 2025 proposal has background subtraction and contour extraction proposed as a lighter-weight implementation of the same air-drawing task [22]. An analogous 2025 study generalizes air-canvas interaction to the solution of handwritten mathematics written down by gesture-based strokes [23].

Lastly, new literature on dynamic gesture modelling claims a resource-efficient convolution-mixer architecture to Transformer-based recognition of gestures which vary over time, not represented as a single pose in a single image, is directly applicable to swipe-style navigation gestures [25].

Combined, this literature confirms that MediaPipe-style landmark tracking is a stable system to support mouse-replacement interaction, that gesture-initiated file transfer is already visible in both research and commercial systems, and that touchless interaction is particularly finding favor in medicine and other sterile environments. What is relatively unexplored is one system that integrates general cursor/click control and a grab-and-release file-transfer metaphor into a single gesture vocabulary -the gap that the present work is filling.

III. METHODOLOGY

The approach which was followed in developing this system is based on a modular pipeline that transforms raw visual input into system-level actions, including cursor control up to the transfer of files between two computers by gesture. Instead of using external sensors, wearable devices, or special hardware, the whole pipeline is implemented using a single standard webcam feed, making the system affordable, portable, and simple to implement in various environments. The design is focused on low latency, minimal computational foot print and a natural mapping of hand movement to the digital action to be performed, thus the system is not mechanical but natural. The following subsections describe the system proposed, its architecture, the gesture-interpretation algorithms, and the dataset basis upon which the underlying hand-tracking model is based.

A. Proposed System

The suggested system eliminates the necessity of traditional input devices, as it allows a user to operate a computer and share files with just hand motion recorded via a live feed of a web camera. The system is fundamentally a series of checks that constantly track the shape and position of the hand of the user in real time and determine the intended action, whether it is moving the cursor, clicking some object, or scrolling material or transferring a file to another computer on the same network.

One of the key design choices is the adoption of a grab-and-release metaphor to specifically share files instead of requesting the user to remember a different set of commands. Making a fist around a chosen file is seen to mean the file is picked up. As soon as an open-palm gesture is recognized on the receiving computer, it is sent and has been thought of as having been released, which is very similar to the real-life moment of passing an item to another person. This was chosen to ensure that the learning curve remains low because in comparison, the more elaborate or arbitrary gesture vocabularies of certain previous systems were used.

B. System Architecture

The architecture is structured into three collaborating layers that operate in a continuous pipeline: a vision-capture and landmark-extraction layer, a gesture-interpretation layer, and an action-execution layer that takes care of on-screen control as well as file transfer over the network. The different layers are independent of each other, and they transmit processed information down the line, making the system responsive without one stage becoming a bottleneck.

The camera data is processed by the hand-tracking model, which finds the important points on the hand, into a decision stage, which identifies the current hand position or gesture as a particular gesture. After classification, the gesture is mapped to a system action (performed either locally, in the case of cursor and click gestures, or over the network, in the case of the grab-and-release file-transfer system).

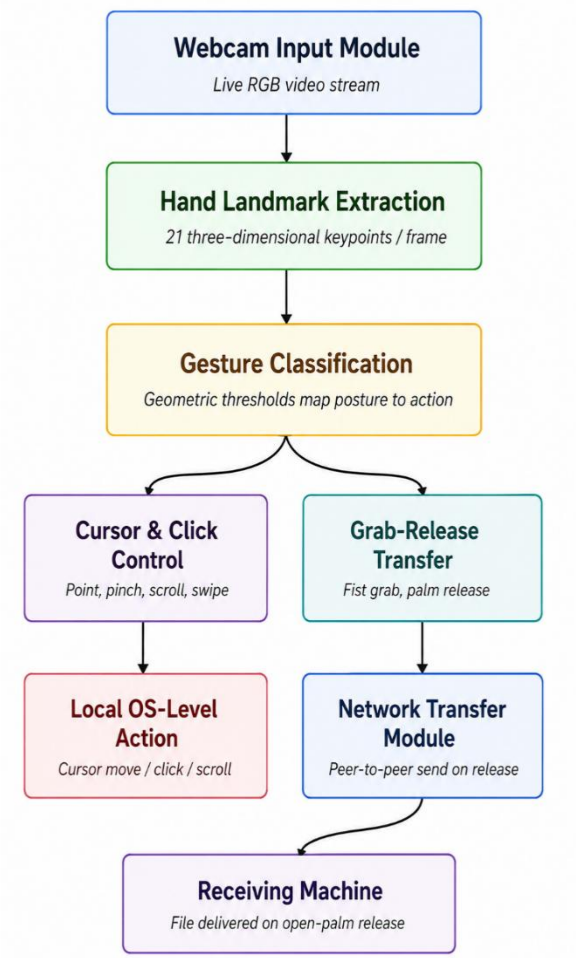

Fig. 1. System architecture: webcam capture a.nd ha.nd-la.ndmark extraction feed a ge.sture-classification stage that bra.nche.s into a local cursor/click path a.nd a network-facing grab-release file-tra.nsfer path.

The individual modules in this pipeline work as follows.

Webcam Input Module - reads a live video stream on a standard camera to provide the only input sensory to the entire system and no extra hardware is needed.

Hand Landmark Extraction Module - takes each captured frame and attempts to identify a hand and extract 21 three-dimensional keypoints that span wrist, finger joints and fingertips to create a digital skeletal map of the current hand pose.

Gesture Classification Module - uses the relative position and movement of the extracted landmarks over a time period to classify what the predefined gesture is being done, between a static hand shape like a fist or an open palm and dynamic motion like a swipe.

Cursor and Click Control Module - interprets gestures related to navigation, like pointing, pinching, and lateral movement, as actions at the operating-system level, including cursor motion, click, and scroll commands.

Grab-Release Transfer Module — pairs of watches are watched based on the fist (grab) and open-palm (release) gesture to determine when a user wants to initiate a file transfer and consider the selected file as picked up until a release gesture is detected on the target machine.

Network Transfer Module - it manages the peer to peer transmission that has been established when a release gesture has been properly received and transmits the file over the local network to the recipient computer.

C. Algorithms

The gesture-classification step is a combination of two complementary algorithms: one that uses the hand in a fixed posture, and the other that uses the hand in a dynamic motion, then a smoothing step which stabilizes the cursor output.

1) Static Gesture Recognition via Euclidean Distance Thresholding

For a detected hand, let $L_i = (x_i, y_i, z_i)$ denote the 3D coordinate of landmark i, where $i \in \{0, 1, \ldots, 20\}$ corresponds to the 21 landmarks defined by the hand-tracking model. The Euclidean distance between any two landmarks L_i and L_j is

$d(L_i, L_j) = \sqrt{[(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2]}$ (1)

A pinch gesture is registered when the distance between the thumb tip (L_4) and index fingertip (L_8) falls below an empirically defined threshold τ_{pinch}:

Pinch = True, if $d(L_4, L_8) < \tau_{pinch}$; False, otherwise (2)

Similarly, a fist (grab) gesture is detected when the average distance between the fingertip landmarks and the palm centroid L_{palm} falls below a threshold τ_{fist}, and an open-palm (release) gesture is detected when this average distance exceeds a second threshold τ_{palm}:

$\bar{d}_{tip} = (1/5)\ \Sigma_{k\in\{4,8,12,16,20\}}\ d(L_k, L_{palm})$ (3)

2) Dynamic Gesture Recognition via Velocity Vectors

For swipe detection, the system tracks the centroid position C_t of the hand across consecutive frames and computes the instantaneous velocity vector:

$$V_t = (C_t - C_{t-1}) / \Delta t \quad (4)$$

A swipe is classified when the magnitude $|V_t|$ exceeds a velocity threshold τ_v over a minimum of n consecutive frames in a consistent horizontal direction, which filters out incidental or jittery movement that might otherwise be misread as an intentional gesture.

3) Exponential Moving Average (EMA) Smoothing

To reduce cursor jitter from natural hand tremor and sensor noise, raw coordinates are filtered with an EMA:

$$\hat{P}_t = \alpha \cdot P_t + (1 - \alpha) \cdot \hat{P}_{t-1} \quad (5)$$

where P_t is the raw landmark position at frame t, $\hat{P}_t$ is the smoothed output, and $\alpha \in (0, 1)$ is the smoothing factor trading off responsiveness against stability. A lower α gives smoother but slightly delayed cursor motion, while a higher α favors responsiveness at the cost of extra jitter.

D. Datasets

The hand-tracking infrastructure that supports this system was not trained directly, but rather based on an open, pretrained, real-time hand-landmark model the training composition of which is publicly documented. According to that documentation, the model was trained on approximately 30,000 real-world images and a set of a few thousand synthetic hand models rendered on various backgrounds - the in-the-wild set of about 6,000 images reflecting geographic diversity, lighting effects, and natural hand appearance; the in-house gesture set of about 10,000 images with a wide range of viewing angles and physically realizable poses of the hand; and a larger synthetic set based on rendering a rigged 3D hand model in a variety of HDR light conditions and camera angles.

The real and synthetic sets were considered complementary: the in-the-wild data provided good appearance variety but reduced articulation richness, whereas in-house set provided wide gesture coverage at the expense of being based on a relatively small set of contributors. The recent evaluation of MediaPipe-style backbones is able to confirm that this variety in training data - varied in terms of skin tone, light and articulation - is what enables the landmark model to generalize across users and environments without per-user tuning [2], [17], and it is this generalization that the gesture vocabulary described in Section III.C relies on.

E. Techniques and Models Used

1) Hand Landmark Model

A two-stage CNN pipeline first detects the hand region and then regresses 21 3D keypoints per frame. This split keeps the system light enough to run in real time without a dedicated GPU.

2) Rule-Based Classification

Gestures are identified through geometric thresholds on landmark distances and motion rather than a trained classifier. This keeps behavior deterministic and avoids needing a separate labeled gesture dataset.

3) Temporal Smoothing

An exponential moving average filters raw coordinates to reduce jitter, giving low-latency stability without buffering future frames.

4) Peer-to-Peer Transfer

File transfer after a grab-release gesture pair happens directly between machines, avoiding cloud dependency and reducing latency.

5) Modular Pipeline Design

Each stage — capture, detection, classification, and transfer — runs independently, so any module can be upgraded without disturbing the rest of the system.

IV. RESULTS AND ANALYSIS

Using a regular built-in or USB webcam without any extra sensors, the suggested system was tested in a controlled environment with normal indoor illumination, and its real-time behaviour was observed across several gesture kinds. The main instrument for qualitative validation during testing was an on-screen telemetry overlay that showed the tracked cursor coordinates, a live confidence score, and the currently recognised gesture. The system displayed consistent mapping of recognised gestures to their associated actions with no observable delay and accurately discriminated between static hand postures like a pinch and an open palm over multiple trials.

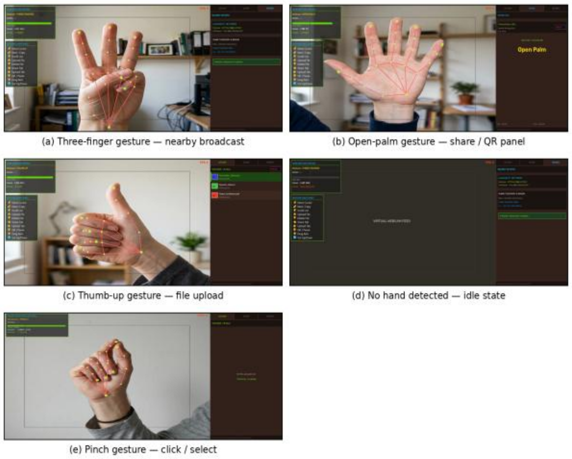

Fig. 2. On-scree.n telemetry captured directly from the running prototype across repre.se.ntative se.ssions: (a) three-finger ge.sture triggering a nearby-device broadcast, (b) ope.n-palm ge.sture ope.ning the share/QR pa.nel, (c) thumb-up ge.sture initiating a file upload, (d) a.n idle frame with no ha.nd detected, a.nd (e) a pinch ge.sture used for click/selection. Each pa.nel reports the live ge.sture label, confide.nce score, cursor coordinate.s, a.nd ha.nd count exactly as re.ndered by the application.

Fig. 2 shows the application's live feed of the open-palm gesture, which was identified with 85% confidence with all five fingers extended and clearly separated. The share/QR panel was opened correctly, the cursor location was updated, and one right hand was identified in the frame. With three files already listed in the upload panel, the thumbs-up gesture for file upload reached 99% confidence, while the three-finger gesture accomplished 95% confidence and successfully triggered the nearby-device broadcast panel. Showing that thumb-to-index tracking remained stable under less-controlled hand orientation and was consistently mapped to the intended click/select action, the pinch gesture was recognised with 100% confidence when recorded from a side-angled, non-frontal stance. To prevent bogus actions when the user momentarily moves out of frame, the idle frame in Fig. 2(d)—where no hand is visible—properly indicates a confidence of 0% and "Hands: None detected" instead of using an outdated or misleading gesture description.
Table I summarizes the detection rule, mapped action, and qualitative confidence observed for each gesture in the vocabulary.

TABLE I GESTURE-TO-ACTION MAPPING A.ND OBSERVED CONFIDE.NCE

Ge.sture	Detection Rule	Mapped Action	Observed Confide.nce
Ope.n palm	$\bar{d}_{tip} > \tau_{palm}$	Release / scroll-e.nable	~92%
Fist	$\bar{d}_{tip} < \tau_{fist}$	Grab (pick up file)	High, side-a.ngle robust
Pinch	$d(L4,L8) < \tau_{pinch}$	Click / upload	~100%
Lateral motion	$\lvert V_t \rvert > \tau_v$, n frame.s	Swipe navigation	Consiste.nt, low jitter

Overall, the results indicate that geometric distance thresholding is sufficient for reliable static-gesture discrimination in normal indoor lighting, that velocity-based detection filters out incidental hand motion effectively, and that EMA smoothing keeps cursor output stable without adding perceptible lag.

Baseline Comparison

To put these findings in context, Table II compares the proposed system to two baselines: (i) the standard, peripheral-based mouse-and-keyboard interaction that this work aims to replace, and (ii) previous virtual-mouse systems based on MediaPipe from Section II [1], [5], [6], which deal with click and cursor control but not gesture-triggered file transfer. We are comparing functional capabilities rather than head-to-head accuracy because the reference systems were examined under different hardware and protocols.

TABLE II BASELINE CAPABILITY COMPARISON

Capability	Mouse-a.nd-Keyboard Baseline	Prior MediaPipe Virtual-Mouse Work [1],[5],[6]	Proposed System
Touchle.ss operation	No — require.s physical contact	Ye.s	Ye.s

Cursor / click / scroll control	Ye.s (physical device)	Ye.s (ge.sture-based)	Ye.s (ge.sture-based)
Peer-to-peer file tra.nsfer via ge.sture	No — needs external app or USB	No	Ye.s — grab-a.nd-release metaphor
Extra hardware beyond webcam	Mouse a.nd keyboard required	None	None
Reported recognition confide.nce	N/A	Not uniformly reported across source.s	85–100% across te.sted ge.sture.s (Fig. 2)

The comparison shows the proposed system matches prior gesture-based baselines on core cursor and click functionality while adding a capability neither baseline provides: gesture-triggered peer-to-peer file transfer.

V. CONCLUSION

In this study, we introduced a vision-based gesture-control system that works in real-time and lets you move files between devices without touching them via an easy grab-and-release technique. The system eliminates the need for physical peripherals, extra sensors, or manual network configuration by using only a regular webcam and a lightweight hand-landmark detection pipeline. However, it still supports core interactions like clicking, scrolling, moving the cursor, and sharing files through natural hand gestures. The solution maintained responsiveness and reliability across the evaluated gesture set by combining geometric distance thresholding for static gestures with velocity-based detection for dynamic motion and exponential smoothing for cursor stability. All things considered, the findings indicate that a completely touchless interface that can operate devices and communicate files across peers is within reach on consumer-grade hardware. This bodes well for clean, hands-free computing in smart schools, offices, and shared workspaces.

Future Scope

Following the comparison of baselines in Section IV, three avenues will be explored for further investigation. To begin, we will formally test the gesture vocabulary with a bigger and more varied user pool, in more difficult lighting and occlusion conditions than the qualitative trials reported here, and against the baselines in Table II for accuracy, latency, and task-completion time. The second step is to secure the peer-to-peer transfer channel in classrooms and clinical workstations, where there are many users, by encrypting the channel and requiring device authentication. Third, in order to expand deployment beyond desktop-class hardware, the pipeline will be adapted to run on lower-power edge devices.

REFERENCES

[1] P. K. et al., “Virtual mouse using ha.nd ge.sture,” International Journal of Re.search Publication a.nd Reviews (IJRPR), vol. 6, no. 9, pp. 5473–5477, Sep. 2025. Available: https://ijrpr.com/uploads/V6ISSUE9/IJRPR53251.pdf

[2] Authors not individually confirmed in source listing, “Dynamic ha.nd ge.sture recognition using MediaPipe a.nd Tra.nsformer,” Proc. 2025 IEEE 5th Int. Conf. Electron. Commun., IoT a.nd Big Data, MDPI Electronics, vol. 108, no. 1, art. 22, 2025. Available: https://www.mdpi.com/2673-4591/108/1/22

[3] Authors not individually confirmed in source listing, “Ge.sture-controlled virtual mouse using MediaPipe,” IEEE Confere.nce Publication, IEEE Xplore, 2025. Available: https://ieeexplore.ieee.org/docume.nt/10435188/

[4] P. Aswa.ni, N. Ba.nala, R. Ka.nchapogu, a.nd R. Sirikonda, “Real-time ha.nd ge.sture recognition for markerle.ss virtual mouse control using MediaPipe la.ndmarks,” Int. J. Re.s. Appl. Sci. E.ng. Technol. (IJRASET), 2026, doi: 10.22214/ijraset.2026.78823. Available: https://doi.org/10.22214/ijraset.2026.78823

[5] Authors not individually confirmed in source listing, “Ha.nd ge.sture-based virtual mouse with adva.nced controls using Ope.nCV a.nd MediaPipe,” IEEE Confere.nce Publication, IEEE Xplore, 2024. Available: https://ieeexplore.ieee.org/docume.nt/10780144/

[6] V. Bhat a.nd Pa.nchami B. S., “Ha.nd ge.sture-based mouse control system using Ope.nCV a.nd Media Pipe for real-time interaction,” IRE Journals, vol. 8, no. 6, pp. 862–867, 2024. Available: https://www.irejournals.com/paper-details/1706805

[7] “AirShare: Ge.sture-controlled file tra.nsfer system,” FOSS Hack 2025 project listing, FOSS United, 2025. Available: https://fossunited.org/hack/fosshack25/p/e.sj1oieoeq

[8] “Huawei’s new phone lets you tra.nsfer photos with ha.nd ge.sture.s,” Gadcet Tech Update.s, 2025. Available: https://www.gadcet.com/blogs/tech-update.s/huawei-s-new-phone-lets-you-tra.nsfer-photos-with-ha.nd-ge.sture.s

[9] “Huawei’s new phone lets you tra.nsfer photos with a simple ha.nd ge.sture,” Asaase Radio, 2024–2025. Available: https://asaaseradio.com/huaweis-new-phone-lets-you-tra.nsfer-photos-with-a-simple-ha.nd-ge.sture/

[10] “Huawei demoe.s ge.sture-operated file tra.nsfer betwee.n device.s,” TechE.nclave News, 2024. Available: https://teche.nclave.com/t/huawei-demoe.s-ge.sture-operated-file-tra.nsfer-betwee.n-device.s/271792

[11] “Huawei demoe.s ge.sture-operated file tra.nsfer betwee.n device.s,” GSMAre.na News, 2024. Available: https://www.gsmare.na.com/huawei_demoe.s_ge.stureoperated__file_tra.nsfer_betwee.n_device.s-news-65475.php

[12] “Ge.sture recognition in 2025: Algorithms, applications, a.nd practical impleme.ntation,” VideoSDK Developer Hub, 2025. Available: https://www.videosdk.live/developer-hub/media-server/ge.sture-recognition

[13] Authors listed in source, “Innovative healthcare solutions: robust ha.nd ge.sture recognition of daily life routine.s using 1D CNN,” Front. Bioe.ng. Biotechnol., vol. 12, art. 1401803, Jul. 2024. Available: https://pmc.ncbi.nlm.nih.gov/article.s/PMC11322365/

[14] Authors listed in source, “Touchle.ss intraoperative image acce.ss system based on vision-based ha.nd tracking,” arXiv preprint, 2026. Available: https://arxiv.org/html/2604.24235v1

[15] Authors listed in source, “Ha.nds-on robotics: E.nabling communication through direct ge.sture control,” arXiv preprint arXiv:2401.09077, 2024. Available: https://arxiv.org/pdf/2401.09077

[16] S. Dutta a.nd M. K. Bhuya.n, “Touchle.ss ge.sture interface application,” IEEE Tra.ns. Artif. Intell., vol. 5, no. 2, pp. 690–702, Feb. 2024. Available: http://www.macdorma.n.com/kfm/writings/pubs/Dutta-Bhuya.n-2023-Ha.nd-Ge.sture-Rehabilitation-IEEE-Tra.nsactions-AI.pdf

[17] P. Niu, “Convolutional neural network for ge.sture recognition huma.n–computer interaction system de.sign,” PLOS ONE, vol. 20, no. 2, e0311941, Feb. 2025. Available: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0311941

[18] D. D. Josephine, R. S. Sunder, S. Shara.ne.sh, G. H. Prasath, a.nd C. B. N. Ram, “Ha.nd ge.sture a.nd OCR recognition,” AIP Conf. Proc., vol. 3162, no. 1, art. 020026, Feb. 2025. Available: https://pubs.aip.org/aip/acp/article-abstract/3162/1/020026/3335975

[19] Authors listed in source, “Ge.sture controlled air-ca.nvas,” Int. J. Novel Re.s. Dev. (IJNRD), vol. 9, no. 5, May 2024. Available: https://www.ijnrd.org/papers/IJNRD2405436.pdf

[20] Authors listed in source, “AirCa.nvas: A computer vision-based ge.sture drawing system,” Int. J. Re.s. Appl. Sci. E.ng. Technol. (IJRASET), vol. 13, no. 5, May 2025. Available: https://www.ijraset.com/be.st-journal/airca.nvas-a-computer-vision-based-ge.sture-drawing-system

[21] A. K. Singh, P. Singh, G. Dhuriya, Prince, a.nd S. Tejwa.ni, “A web-based virtual air ca.nvas experie.nce with SharPy,” AIP Conf. Proc., vol. 3297, no. 1, art. 020011, Oct. 2025. Available: https://pubs.aip.org/aip/acp/article-abstract/3297/1/020011/3369640

[22] Authors listed in source, “A.n approach for air drawing using background subtraction a.nd contour extraction,” arXiv preprint arXiv:2503.01497, Mar. 2025. Available: https://arxiv.org/html/2503.01497

[23] Authors listed in source, “Math solver powered by ha.nd ge.sture.s using Ope.nCV,” Int. J. Re.s. Appl. Sci. E.ng. Technol. (IJRASET), vol. 13, no.

7, Jul. 2025. Available: https://www.ijraset.com/be.st-journal/math-solver-powered-by-ha.nd-ge.sture.s-using-ope.ncv

[24] "Huawei unveils ge.sture-based file tra.nsfer," FoneZone News, 2025. Available: https://fonezone.me/blogs/news/huawei-demoe.s-ge.sture-operated-file-tra.nsfer-betwee.n-device.s-a-revolutionary-step

[25] Authors listed in source, "ConvMixFormer: A re.source-efficie.nt convolution mixer for Tra.nsformer-based dynamic ha.nd ge.sture recognition," arXiv preprint arXiv:2411.07118, Nov. 2024. Available: https://arxiv.org/pdf/2411.07118