DOI: 10.xx/ijdsml.2023.xx

GESTURE RECOGNITION FOR TOUCH-FREE PC CONTROL USING A NEURAL NETWORK APPROACH

Naina Sharma, Vaishali Nirgude, Tanya Shah, Chirag Bhagat, Amithesh Gupta and Yash Gupta
Department of Computer Engineering, Thakur College of Engineering and Technology, India

Abstract

In the pursuit of advancing the field of touch-free human-computer interaction, this paper is focused on developing a gesture enabled PC control system that aims for enhancing user engagement and providing intuitive and flexible control methods, across various applications, particularly those benefiting individuals with mobility impairments. This system has expanding potential use in virtual and augmented reality environments. This study describes a unique method for temporal gesture identification that employs gesture kinematics for feature extraction and classification. Real-time hand tracking and key point identification were performed using MediaPipe. The Euclidean distances between the key points was normalised and input into a Multilayer perceptron model, which classified the gestures and mapped them to specific commands for controlling PC functions. This approach performed well over a large dataset, improving accuracy and usability. The gesture recognition system achieved an average accuracy of 97%, with precision, recall, and F1 score of 0.924, 0.924, and 0.926, respectively, across the five gestures. This system provides the ability of customization to users which allows them to create and map their own gestures to specific commands, in addition to using predefined ones. This level of personalization and flexibility is a significant advancement over existing systems, which typically offer fixed gesture-command mappings.

Keywords:

Human Computer Interaction (HCI), Neural Network, Hand Gesture Recognition, MediaPipe, FaceNet

1. INTRODUCTION

An interface that facilitates communication and information sharing between the user and the computer is known as a Human Computer Interface (HCI). There are two components to this system: software and hardware. We typically integrate a broad range of indications into HMI devices. LEDs, switches, touch screens, and LCDs are some examples of these indicators. Conventional human computer interaction involves using devices such as keyboards, monitors, mice and displays which are required to be connected to the computer. However, in cases like augmented reality and virtual reality conventional HCI might be insufficient. Hence there has been a lot of research lately to create a HCI environment which aligns with human communication behaviour[1], making study of gesture recognition systems a popular and in demand field of HCI. The focus on the Hand Gesture Recognition system has grown tremendously in recent years as a replacement of multi touch technology and because of its efficiency in interacting with machines. From deleting a folder using a simple gesture to baking a cake in the microwave by simply waving our hands like a magician, all of it is made possible by gesture recognition technology [2]. People with disabilities and elderly people who are not able to walk, or talk can communicate with their caregivers using this technology [3]. Hand gesture technology is used in a wide range of situations in the contemporary automation sector, including banking, manufacturing, medical, sign language translation and information technology centres. Its applications are not limited to the gaming industry.

1.1 OVERVIEW

Gesture recognition is a computer technology that is used to map gestures to specific actions or commands. Most used gestures involve use of hands and face. Gesture enabled command technology involves translating physical movements into a format understood by a computer system and to which a computer system can respond to [4]. This can be done by measuring the centre of the palm, location of fingers and shape formed by hand [5]. Two types of gestures- static gestures and dynamic gestures. Static gestures refer to the specific posture or configuration of the hand that is held without any movement whereas dynamic gestures involve a sequence of movements [6]. Convolutional neural networks are one of the most effective techniques to perform image classification tasks for gesture recognition systems due to their ability to learn and extract features from images and video frames. Image classifier involves taking an image or sequences of images as input and categorising them into one of the possible predefined classes that it was trained to classify [7]. A CNN architecture typically includes convolutional layers for detecting features, activation functions like ReLu for introducing non-linearity, pooling layers for down-sampling, and fully connected layers for classifying gestures. Training involves collecting and preprocessing a labelled dataset, using a loss function like categorical cross-entropy, and optimising with algorithms such as Adam or SGD. This process enables the CNN to accurately recognize and map gestures to specific actions or commands.

One of the emerging paradigms of today is NUI - Natural User Interface which is based on the concept that users can interact with computer systems without needing additional hardware for input and output[8].Thus a system of gesture enabled commands are gaining importance due to the growth- by a small percentage , of deaf and hard earring population and also due to growth in touchless and vision based technologies , smart control TV, virtual reality ,video games and augmented reality applications [9].Hand gestures provide for a non-physical connection to the computer, enhancing user comfort and safety. They also enable users to manage complex and virtual environments more easily compared to traditional methods, making them valuable in a wide range of human-computer interaction (HCI) applications. Hand gesture applications require the capability to accurately recognize and interpret a wide range of gestures [10]. Therefore, the focus of this research paper is on building a system which directs computers to perform certain tasks based on hand gestures using convolutional neural networks to accurately identify and classify hand gestures.

Table.1. Gaps and Findings

Title and Journal Name	Techniques used	Performance	Limitations
An Exploration into Human–Computer Interaction: Hand Gesture Recognition Management in a Challenging Environment, 2023 Springer. [11]	• Image enhancement by background noise reduction and colour space conversion • Image segmentation, HSV and RGB colour segmentation • CNN with Max pooling	• Accuracy: • Unsegmented images- 45% • Segmented images- 58%. •	• Sensitive to lighting variations • Non diverse dataset with low quality images • Does not recognize dynamic gestures
Hand Gesture Recognition Using Automatic Feature Extraction and Deep Learning Algorithms with Memory, Big Data Cogn and Cognitive Computing. [12]	• Evaluation of two models: • Manual feature extraction with statistical functions applied through window splitting • Automatic feature extraction with CNN and BiLSTM. Classifiers utilised were ANN, SVM, Softmax for CNN, and Softmax for BiLSTM.	• Testing accuracies: • Manual extraction model: • ANN- 92.936% • SVM- 91.370% • Automatic extraction model: • CNN- 93.971% • BiLSTM- 95.6161%	• Limited scalability and optimization of memory efficiency can be explored.
Finger gesture recognition with smart skin technology and deep learning, Flex. Print. Electron, 2023. [13]	• Hand gesture recognition using a wireless surface electromyography (sEMG) system with electrode arrays. • Signal preprocessing with filtering and Root Mean Square feature extraction for data quality optimization. • CNN, RNN, SVM, and HMM algorithms for classification of gestures based on processed sEMG signals.	• SVM: • Same hand position- 90.4% accuracy • Multiple positions 87.8%. • • CNN: • 78.2% accuracy across sessions and positions.	• Inter-subject variability affects recognition consistency • Real time applicability not supported • Insufficient processing speed • •
Controlling the Computer using Hand Gestures, Multimedia Research, 2022. [7]	• Real time image augmentation -Keras' Image Generator • Modified CNN architecture based on VGG19, featuring convolutional layers, max pooling, fully connected layers, ReLu activation, and softmax. • Adam optimizer	• Testing accuracy- 85.90%	• Limited scope- recognizes only 10 hand gestures for various operations • Future scope can be adding operations like volume control, scrolling and swiping
Research on Gesture Recognition Method Based on Deep Learning, Phys.: Conf. Ser. 1861 012049, 2021. [14]	• An enhanced CNN architecture with the Inception module was employed, featuring image preprocessing, convolutional layers for feature extraction, pooling layers, a fully connected layer, and Softmax regression for gesture recognition.	• 97.8% testing accuracy	• There is need for evaluation under diverse real-world conditions like varied lighting and user demographics • Limited to static gestures.
Hand Gestures for Laptop 2.0., 2021.[15]	• Pi cam for sending images to raspberry Pi • Conversion of BGR image to HSV image • OpenCV library • Task Execution 'PyAutoGUI' is used to map gestures to actions	• **Gesture 5 (Scroll down):** 94% accuracy • **Gesture 4 (Scroll up):** 82% accuracy • **Gesture 3 (Play/Pause):** 100% accuracy • **Gesture 2 (Volume down):** 88% accuracy	• Complicated to set up. • Background colour if skin causes false inputs and unstable system

		• **Gesture 1 (Volume up):** 90% accuracy	
Study on Hand Gesture Recognition by using Machine Learning, 2020. [16]	• Hand gesture recognition is carried out using machine learning algorithms. • The hand sign is captured by HMI. The machine learning algorithm is applied on the image to detect the gesture.	• Accuracy: • Neural Network (CNN) - 94.2 %, • ANN - 95.1% • SVM - 97.5%	• Variations in lighting can have an impact on performance. • Image noise has the potential to diminish accuracy. • Complex backdrops can impair gesture detection.
Gesture Recognition System, 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), 2019. [17]	• Principal Component Analysis (PCA) and Singular value Decomposition (SVD) for input image feature extraction • Feed forward neural network for classifying gestures	• Gesture accuracy- 95.9% • Posture accuracy- 90.3%	• Uniform background and only some types of gestures and postures are recognized. • Recognizes upper body gestures only.
Gesture Recognition Method Based On Deep Learning., 2018. [18]	• RNN is used to process sequence data. • In order to classify features from six-axis attitude sensor data on the wrist for gesture identification, the research used RNN, LSTM, and GRU models.	• The average recognition rate: • RNN - 98% • LSTM - 99.75% • GRU - 99.75%.	• Data Dependency: To prevent overfitting, big, high-quality datasets are necessary. • Sensor Placement: Proper sensor alignment and calibration are necessary for accuracy.

1.2 COMPREHENSIVE HAND GESTURE RECOGNITION METHODS

1.2.1 Hand Gesture Methods:

- **Vision-Based Methods**: Hand gestures are detected and recognised using vision-based technologies, which use cameras and image processing algorithms.[19]
- **Image Processing Techniques**: Skin Colour Detection techniques for identifying hand regions using skin colour segmentation and the use of Edge Detection methods (such as the Canny edge detector) to determine the limits of the hand.
- **Machine Learning Approaches**: Methods for extracting features like edges, contours, and key points from hand pictures and Using machine learning algorithms such as Support Vector Machines (SVM) and k-Nearest Neighbours (k-NN) to classify gestures based on extracted data.
- **Deep Learning Techniques**: Convolutional Neural Networks (CNNs) are trained on massive datasets of hand photographs to identify static gestures and Recurrent Neural Networks (RNNs) or Long Short-Term Memory networks (LSTMs) to recognise sequences of movements; useful in dynamic gesture recognition.
- **Keypoint Detection**: Methods for recognising key points (such as fingertips and knuckles) on the hand utilising models like OpenPose.
- **Skeleton Tracking**: Techniques for determining the location and orientation of the hand skeleton throughout time.

1.2.2 Sensor-Based Methods:

Sensor-based approaches use a variety of sensors to detect the motion and location of the hand and fingers.[19]

- **Inertial Measurement Units (IMUs)** use accelerometers and gyroscopes on the hand or fingers to gather motion data and identify gestures based on movement patterns.
- **Data Gloves**: Sensors detect finger bending and hand movements, allowing for accurate gesture recognition.
- **Leap Motion Controller**: An infrared sensor-based device that monitors hand and finger movements in three dimensions, allowing for exact gesture detection.
- **Electromyography (EMG)**: EMG sensors assess muscle activity in the forearm and can be used to determine hand motions based on signal patterns.

1.2.3 Hybrid Methods:

Hybrid approaches combine vision-based and sensor-based techniques to take use of their respective strengths.

- **Integration of Vision and IMUs**: Using camera data for high-level gesture detection and IMU data for detailed motion tracking.
- **Multi-Modal Systems**: Using many sensors and cameras to increase the accuracy and resilience of gesture recognition systems.

1.3 EXISTING SYSTEMS

The proposed gesture control system for PCs leverages ultrasonic (US) sensors and Arduino to create an intuitive and cost-effective interface. The system involves mounting two US sensors on top of a monitor to measure the distance between the monitor and the user's hand. An Arduino microcontroller reads these distance values and sends the data to a computer via a USB serial connection. Python, utilising the package, PyAutoGUI processes this data to execute various commands based on the measured distance [2]. For instance, a specific distance might trigger a swipe or click gesture, allowing the user to control the computer without physical contact. This approach offers a simple, affordable, and customizable solution for enhancing human-computer interaction.

2. METHODOLOGY

The proposed system architecture of Touch free PC Control system which includes modules for real-time hand gesture detection, interpretation and command execution, facilitating seamless human-computer interaction is presented in Fig.1.

2.1 ARCHITECTURE MODULES

The entire project is divided into three major components with each one being further broken down into multiple modules for low coupling and easier management. The components are as follows:

2.1.1 Authentication module:

Face authentication is a biometric technology that uses facial recognition to identify a person. This module is designed to help users securely access their accounts without having to remember multiple passwords. With this application, users can quickly and easily log in using their face as an authentication factor. Our application is powered by advanced AI technology which enables it to accurately detect faces and recognize them with great accuracy. This makes it a secure and reliable authentication method for any user.

2.1.2 Login module:

DeepFace technology as discussed previously is a powerful tool for authentication and login, allowing users to access their accounts securely with just one frame. It works by capturing a single frame of the user's face and then comparing it to a database of registered users. If the facial features match, the user is granted access to their account. DeepFace provides an extra layer of security and convenience, making it easier than ever for users to sign in quickly and safely. With its ability to detect even small changes in facial features, DeepFace ensures that only authorised individuals can gain access to an account.

2.1.3 Registration Module:

In the registration process, a frame of the user is captured and saved in the user's folder. This frame can be accessed later by the login module, allowing for an efficient and secure way to authenticate users. By capturing a frame of the user during registration, it ensures that only authorised users have access to their accounts. Furthermore, this allows for a more secure authentication process since it ensures that no other person can access someone else's account.

2.1.4 Gesture recognition system:

This module involves real time hand tracking and keypoint identification using MediaPipe, feature extraction using Euclidean distances and multilayer perceptron model for gesture recognition and interpretation.

2.1.5 Customization

- **Add gesture**: Users can add their own gestures and customise it to execute any command. There is no limit to the amount of gestures that can be added.
- **Edit command**: This functionality allows users to change the mapping of any gestures and commands to suit their own preferences at any time.
- **Delete gesture**: Any gesture and its associated command can be deleted according to one's need after which the model is retrained.
- **Reset to default**: There is a set of 10 default gestures and users can change it to default anytime

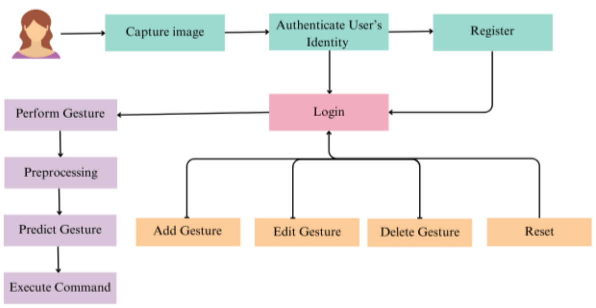

Fig.1. Touch Free PC Control System Architecture

2.2 DATA COLLECTION

Collecting a total of 400 frames of hand key points obtained from MediaPipe. MediaPipe provides a pre-built solution for hand tracking that detects hand keypoints, which are the specific points on the hand that correspond to different joints and landmarks. The hand tracking solution in MediaPipe is based on a machine learning model that is trained to recognize the shape and movement of hands in real-time video streams. The hand tracking solution in MediaPipe can detect 21 key points on each hand, including the tips of each finger, the base of each finger, the centre of the palm, and the wrist. These key points can be used to track the position, orientation, and movement of the hand. FaceNet is used for user face authentication. And as part of data collection mediapipe extracts key points from gestures. Multilayer perceptron appreciates various relationships between the features through feedforward and backpropagation to train the model. This model will be user specific. All information about the user and gesture is stored in a database.

Table.2. List of Default Gestures

Gesture Name	Gesture Command	Command Description
L	Ctrl + o	Used to open a new document
Five	Alt + tab	Switch between tabs
Victory	Ctrl + a	Selects all text
Pinky	Ctrl + c	Copies selected text
One	Ctrl + v	Paste text
Three	Ctrl + s	Saves the file

Four	Ctrl + p	Used to print any document
Spider	Ctrl + f	Launch a finding box
Fist	Ctrl + n	Create a new document
Thumbs	Alt + f	Opens the file menu

2.3 SYSTEM TECHNOLOGY STACK

The system utilises a variety of tools and technologies as mentioned in Table.3. Python was used for implementing the neural network model , Javascript and React for interactive web interfaces, Tkinter for GUI , and SQLite for storing gesture data. UML diagrams facilitated system design.We are using Mediapipe that provides 21 keypoint landmarks for the human Hand. Using these keypoints, we are calculating distance (euclidean distance) from one point to every other point using below formula:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (1)$$

Table.3. Tools and Technologies

	Software
Programming Languages	Python, Javascript
Operating Systems	Window Operating System
Libraries	Tkinter, DeepFace
Databases	SQLite
Others Technologies	UML, React, Visual Studio

Euclidean distance is a measure of the straight-line distance between two points in 3D space. It is calculated by taking the square root of the sum of the squared differences between the corresponding coordinates of the two points. In this modified version, the Euclidean distance is divided by the distance between the palm and a reference point, resulting in a normalised distance measure. The calculated distances are then passed on to our Deep Learning model that identifies and classifies the labels for the gestures performed, based on the training data.

2.3.1 MediaPipe Framework:

MediaPipe framework is used to work upon sensory data consisting of audios and videos and builds production ready perceptron pipelines which is useful for ML practitioners, researchers, students to conduct research work and to develop new Machine Learning technologies. MediaPipe builds pipelines as modular processing graphs which incorporate various components like data transformations, media processing algorithms as well as model inference. Input is audio or video stream which is processed into the graph and output consists of various perceived descriptions like facial landmarks and object localization [20].

MediaPipe implements pipeline in two stages:

- **Stage 1:** Palm Detection and Localization: A palm detector identifies the presence of a hand in the input image and generates an oriented bounding box that circumscribes the hand if a hand is found. BlazePalm palm detector is used to detect. To further enhance efficiency, non maximum suppression is applied after palm detection as it eliminates redundant bounding boxes specifically those of non square aspect ratios. By constraining bounding boxes to squares, the number of potential anchors is reduced to a factor of 3-5.The system utilises an encoder-decoder architecture for feature extraction. This architecture is efficient at capturing broader scene context, thus resulting in accurate detection even for smaller objects [21].

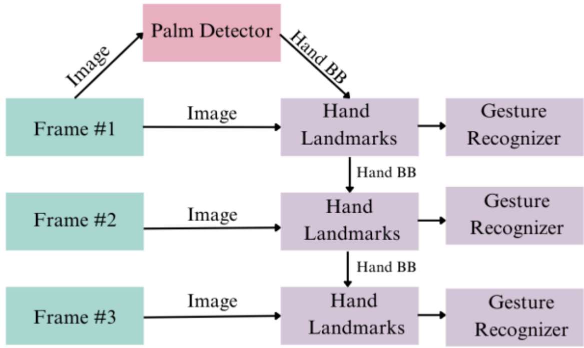

Fig.2. Overview of Hand gesture recognition Perception Pipeline [22]

- **Stage 2**: Hand Landmark Recognition and Gesture Classification: The image cropped in the previous step is now processed by a hand landmark model which predicts three-dimensional (3D) key points that correspond to specific anatomical landmarks on the hand.These key points correspond to 3D hand-knuckle coordinates.This model utilises a regression approach to predict the coordinates of these points which effectively generates a detailed representation of the hand's structure within the detected region. In the end a gesture recognizer utilises these 3D key points to classify the hand gestures into one of the predefined ones [21].

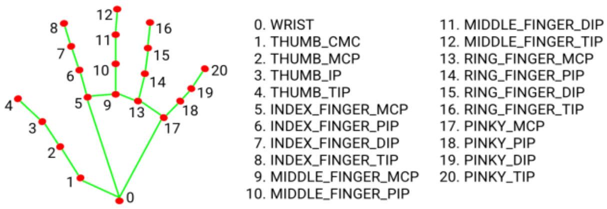

Fig.3. 3D Keypoints detection [22]

2.3.2 Facenet Technolog:

Facenet is one of the facial recognition models that is provided by DeepFace. DeepFace offers a user-friendly interface to a variety of facial recognition models, but Facenet stands out due to its emphasis on accuracy and efficiency. Facenet utilises a deep convolutional neural network (CNN) architecture to learn compact mathematical representation that captures the essence of a person's face known as facial embedding which is a powerful tool for facial recognition tasks. Facenet is adept at face clustering, that is grouping of faces with similar features together. This capability is crucial for organising and searching large facial databases thus enhancing its real-world applicability. It is also robust to variations in pose, partial occlusions and illumination [23].

2.3.3 Multilayer Perceptron Model:

In this paper as well, we are utilising the MLP neural network from the Keras library to develop our Gesture Model. Multilayer Perceptrons (MLPs), is a type of Feedforward Neural Network which is used for learning complex input-output relationships. They consist of an input layer, intermediate hidden layers and an output layer. The input layer receives input features from data, hidden layers perform computations and learn hidden patterns through network of interconnected neurons with non-linear activation functions, and an output layers that provides the final prediction based on type of input and task (e.g., sigmoid for binary classification or softmax for multi-class classification) [24]. In our system the input layer receives data from MediaPipe's hand tracking which involves data about various features of the hand gestures.The output layer generates specific commands or actions based on the input gesture data. MLPs also incorporate Backpropagation which is used to iteratively improve performance by calculating errors, by propagating them backward in the network , and adjusting weights to minimise the difference between predicted and actual outputs , thus refining the accuracy of MLP model in recognizing and interpreting gestures and resulting in a robust and effective system.

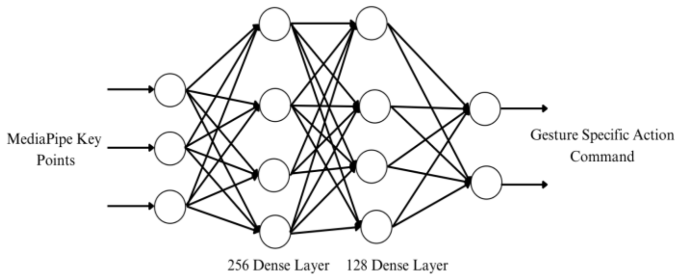

Fig.4. Feed-forward Neural Network

The mathematical equation for MLP can be represented as:

$$f(x) = (w_i * x_i) + b \tag{2}$$

where: m: number of neurons in the previous layer, w: random weight, x: input value, b: random bias

2.3.4 Touch Free PC Control Mathematical Model:

The equation for the Gesture model can be written as follows:

$$o = softmax(W_2 * \text{relu}(W_1 * x + b_1) + b_2) \tag{3}$$

where:

x is the input gesture data of shape [batch_size, 210],

W_1 is the weight matrix of the first dense layer of shape [210, 256],

b_1 is the bias vector of the first dense layer of shape [256],

relu() is the rectified linear unit activation function,

W_2 is the weight matrix of the second dense layer of shape [256, 128],

b_2 is the bias vector of the second dense layer of shape [128],

softmax() is the softmax activation function,

total_labels is the total number of possible gesture labels

Note that the output of the model is a probability distribution over the possible labels, with the highest probability indicating the predicted label.

3. RESULT AND DISCUSSION

The figures below show the application's primary interface components, with a focus on the Authentication Page, Home Page, and Main Application Settings Page. These panels depict the typical user journey, from safe login to accessing core functions and adjusting personal preferences.

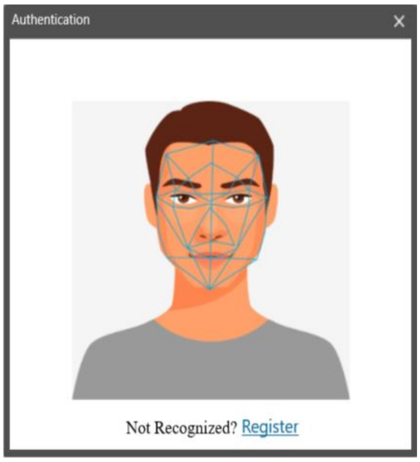

Fig.5. Authentication page (prototype)

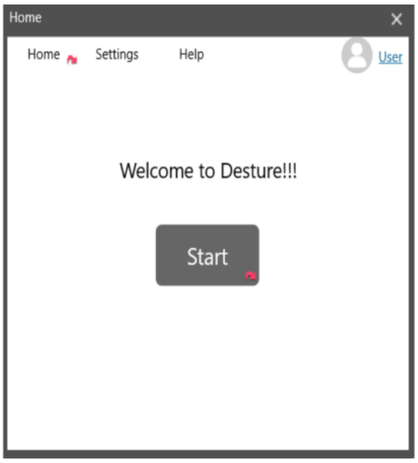

Fig.6. Home page (prototype)

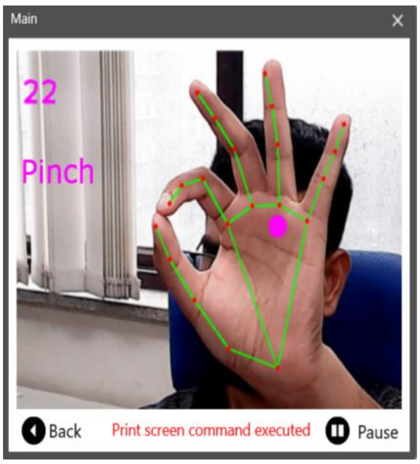

Fig.7. Main Application (prototype)

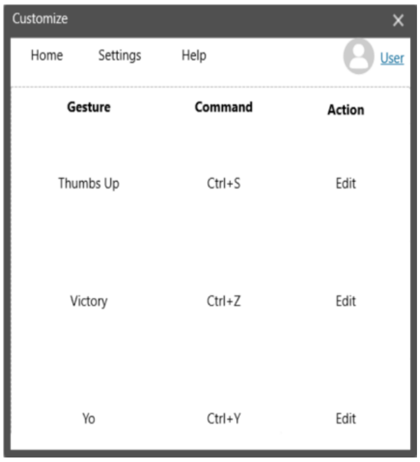

Fig.8.Settings page (prototype)

For the test set, the gesture model had an overall accuracy of 97%, and the face authentication model had an accuracy of 99.63%. Authentication model is able to detect faces in different light conditions. Model is able to distinguish different skin tones. We are collecting 400 frames of hand key points out of which we are using 80% for training data and remaining for testing data. High accuracy in any part of the window. Hand gestures are calculated using relative distance between key points hence there is no constraint on the position of hand on the application window.

Table.4. Accuracy Measure

Gesture	Accuracy
Gesture 1	95%
Gesture 2	97%
Gesture 3	99%
Gesture 4	96%
Gesture 5	97%

Table.5. Performance Measure

Gesture	Precision	Recall	F1-Score
Gesture 1	0.92	0.93	0.93
Gesture 2	0.88	0.89	0.88
Gesture 3	0.96	0.94	0.95
Gesture 4	0.95	0.96	0.96
Gesture 5	0.91	0.90	0.91

4. CONCLUSION

Overall, the results suggest that this gesture-enabled command application is an effective and practical solution for controlling devices using hand gestures and has the potential to provide a more natural and convenient mode of interaction for a wide range of users. Our research demonstrates successful implementation of customizable gesture-enabled command applications. This system allows users to map gestures of their own choice to specific commands, setting it apart from previous efforts which lacked such customization. The current system recognizes only static gestures, but its framework can be expanded to incorporate dynamic gestures. This system has practical applications in virtual and augmented reality interfaces, smart home controls, and assistive technologies for people with disabilities. Future scope can involve expansion of gesture commands, enhancing robustness to background and lighting variations, improving dataset diversity and incorporating dynamic gesture recognition. Doing so will evolve this system to meet diverse, interactive needs and improve its scalability and real-world applicability.

REFERENCES

[1] Gianluca Amprimo, Giulia Masi, Giuseppe Pettiti, Gabriella Olmo, Lorenzo Priano and Claudia Ferraris, “Hand Tracking for Clinical Applications: Validation of the Google MediaPipe Hand and the Depth-Enhanced Frameworks”, *Biomedical Signal Processing and Control*, Vol. 96, pp. 1-6, 2024.

[2] P. Purushotham, R. Madhu, T. Vinod and K. Shekar, “PC Automation using Hand Gestures using Machine Learning”, International Journal of Advanced Research in Science, Communication and Technology, International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal, Vol. 3, pp .1-12, 2023.

[3] M. Yasen and S. Jusoh, “A Systematic Review on Hand Gesture Recognition Techniques, Challenges and Applications”, *PeerJ Computer Science*, Vol. 78, pp. 1-12, 2019.

[4] Riya Jain, Muskan Jain, Roopal Jain and Suman Madan, “Human Computer Interaction - Hand Gesture Recognition”, *Proceedings of International Conference on Image Information Processing*, pp. 1-9, 2022.

[5] Munir Oudah, Ali Al-Naji and Javaan Chahl, “Hand Gesture Recognition Based on Computer Vision: A Review on Technique”, *Journal of Imaging*, Vol. 45, pp. 1-13, 2020.

[6] M. Perimal, S. Basah, M. Safar and H. Yazid, “Hand-Gesture Recognition-Algorithm based on Finger Counting”, *Journal of Telecommunication Electronic and Computer Engineering*, Vol. 10, No. 1-13, pp. 19-24, 2018.

[7] Pradnya Kedari, Shubhangi Kadam and Rajesh Prasad, “Controlling Computers using Hand Gestures”, *International Research Journal of Engineering and Technology*, Vol. 9, pp. 2-9, 2022.

[8] N.C. Camgoz, “Sign Language Transformers: Joint End-to-End Sign Language Recognition and Translation”, *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10023-10033, 2020.

[9] L. Tian, “Video Big Data in Smart City: Background Construction and Optimization for Surveillance Video processing”, *Future Generation Computer Systems*, Vol. 86, pp. 1371-1382, 2018.

[10] D. Sarma and M.K. Bhuyan, “Methods Databases and Recent Advancement of Vision-Based Hand Gesture Recognition for HCI Systems: A Review”, *SN Computer Science*, Vol. 2, pp. 1-40, 2021.

[11] V. Chang, R.O. Eniola and L. Golightly, “An Exploration into Human-Computer Interaction: Hand Gesture Recognition Management in a Challenging Environment”, *SN Computer Science*, Vol. 4, pp. 441-467, 2023.

[12] R.E. Nogales and M.E. Benalcazar, “Hand Gesture Recognition using Automatic Feature Extraction and Deep

Learning Algorithms with Memory", *Big Data and Cognitive Computing*, Vol. 7, pp.102-124, 2023.

[13] Liron Ben-Ari, Adi Ben-Ari, Cheni Hermon and Yael Hanein, "Finger Gesture Recognition with Smart Skin Technology and Deep Learning", *Flexible and Printed Electronics*, Vol. 8, pp. 1-12, 2023.

[14] Yizhi Li, "Research on Gesture Recognition Method Based on Deep Learning", *Journal of Physics: Conference Series*, Vol. 7, pp. 1-6, 2021.

[15] Abhishek Prajapati, Mayuresh Rane, Manish Rajesh Gupta, Manish Suresh Gupta and Aarti Bakshi, "Hand Gestures for Laptop 2.0", *International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering*, Vol. 9, pp. 1-11, 2021.

[16] A. Mohanarathinam, K.G. Dharani, R. Sangeetha, G. Aravindh and P. Sasikala, "Study on Hand Gesture Recognition by using Machine Learning", *Proceedings of International Conference on Electronics, Communication and Aerospace Technology*, pp. 1498-1501, 2020.

[17] P. Sharma and N. Sharma, "Gesture Recognition System", *Proceedings of International Conference on Internet of Things: Smart Innovation and Usages*, pp. 1-3, 2019.

[18] T. Du, X. Ren and H. Li, "Gesture Recognition Method based on Deep learning", *Proceedings of Youth Academic Annual Conference of Chinese Association of Automation*, pp. 782-787, 2018.

[19] Noor Fadel, I. Emad and Abdul Kareem, "Computer Vision Techniques for Hand Gesture Recognition: Survey", *Proceedings of International Conference on New Trends in Information and Communications Technology Applications*, pp 50-76, 2023.

[20] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, "MediaPipe: A Framework for Building Perception Pipelines", *Proceedings of International Conference on Distributed, Parallel, and Cluster Computing*, pp. 1-12, 2019.

[21] F. Zhag, Bazarevsky and A. Vakunov, "MediaPipe Hands: On-Device Real-Time Hand Tracking", *Proceedings of International Conference on Computer Vision and Pattern Recognition*, pp. 1-12, 2020.

[22] Harris Indriani, Moh Agoes and Ali, "Applying Hand Gesture Recognition for User Guide Application using MediaPipe", *Proceedings of International Seminar of Science and Applied Technology*, pp. 6-17, 2021.

[23] I. William, D.R. Ignatius Moses Setiadi, E.H. Rachmawanto, H.A. Santoso and C.A. Sari, "Face Recognition using FaceNet (Survey, Performance Test, and Comparison)", *Proceedings of International Conference on Informatics and Computing*, pp. 1-6, 2019.

[24] Z. Zhang, F. Feng and T. Huang, "FNNS: An Effective Feedforward Neural Network Scheme with Random Weights for Processing Large-Scale Datasets", *Applied Sciences*, Vol. 12, pp. 12478-12485, 2022.