HUMAN COMPUTER INTERACTION USING GESTURES

ABSTRACT

In this paper, we concentrate on leveraging pointing behaviour as a natural interface. In recent years, hand motion recognition-based user interfaces have seen strong development. The majority of visual hand gesture recognition systems function on Arduino-sensor based platform.

Device control using hand gestures refers to the ability of a device to interpret and respond to movements of a person's hand as a means of input. This technology typically uses computer vision and machine learning algorithms to recognize specific gestures and translate them into control commands for the device.

For many human-based electronics products, the gesture control technique has recently emerged as a new creative trend. This method made it possible for users to operate these goods more easily, organically, and conveniently.

Hand gesture control can be applied to various types of devices such as smartphones, laptops, gaming consoles, and smart home appliances, making interaction more intuitive and convenient. This technology has the potential to revolutionize the way people will interact with technology, making it more accessible and natural.

INTRODUCTION

As information technology develops in our society, we should expect a greater integration of computer systems into our surrounds. These conditions will call for novel forms of human-computer interaction with easy-to-use interfaces. Human-computer interaction has consequently become a crucial aspect of our daily life. Since direct hand contact is in fact a natural way for people to communicate with one another and, more lately, with cutting-edge technology in intelligent environments, the effects of human interaction with computing equipment cannot be ignored any longer and they have taken the lead in our concerns.

Gesture recognition is the cornerstone for putting such a system in place. Gesture recognition has been a prominent topic for many years. There are essentially two methods for interpreting gestures in the modern world. Gestures are a convenient replacement for these bulky interface devices as a communication interface for human-computer interaction. The goal of gesture recognition research is to create a system that can interpret certain human gestures and use them to transmit data or operate machinery. After the system has identified the picture regions, the image regions can be processed to detect the hand position in challenging environments or in real-life situations where the user wishes to use the programmed on the fly.

They have applications in different fields such as medical domain, self-driving cars, educational domain, fraud detection, defence, etc.

For the goal of classifying images, there are numerous methods and algorithms, as well as difficulties like data overfitting. The goal of this Human Computer Interaction with Gestures project is to create a real-time application using Python, MediaPipe, TensorFlow and OpenCV. A real-time, open-source library for computer vision and image processing is called OpenCV. With the aid of the OpenCV, MediaPipe, TensorFlow and some Python modules, we'll use it.

Hand gesture recognition has been one of the active research areas in the field of human-computer interface (HCI) due to its flexibility and user friendly. Several applications on hand gesture recognition reported in the literature are, sign language recognition, writing support system for blind people, control of a software interface app so on. The use of hand gesture based system in public places such as ATM, railway station for booking of platform ticket etc. where, the gesture recognition is greatly affected by the variation in light intensity and also due to daylight. In such case the segmentation of hand region is largely affected by intensity of light.

In accordance with its adaptability and user-friendliness, hand gesture recognition has been one of the active study areas in the field of human-computer interface (HCI). Applications for hand gesture recognition have been reported in the literature for things like sign language recognition, writing assistance for the blind, control of a software interface app and so on. The usage of hand gesture-based systems in public spaces, such as ATMs and train stations, where the identification of gestures is significantly impacted by changes in light intensity and daylight. In this situation, light intensity has a significant impact on how the hand region is segmented.

Thus, a reliable and user-independent hand gesture detection system that can function in such a range of light intensity is needed. Although a depth camera can be used for gesture identification, our goal is to construct a vision hand gesture detection system using a webcam because it is a low-cost camera. Numerous objects are seen as noise in the depth image in depth-based hand gesture detection systems due to relatively low light.

MOTIVATION

The motivation behind this project is the ability to overcome the drawbacks of conventional interfaces like arduino-sensor based interfaces, improve accessibility and diversity, and enable natural interactions in novel computing situations is what drives research into human-computer interaction via hand gestures. We can revolutionise how we interact with technology by embracing the organic and expressive qualities of hand gestures, making it more intuitive, inclusive and immersive. By focusing on this area of research, we can open up a wide range of opportunities that will influence how people interact with computers in the future and open the door to game-changing user experiences.

RESEARCH GAP

A fairly small percentage of solutions employ webcams directly; the majority of methods involve Arduino and sensors. If the surrounding environment contains components that resemble human skin, motions may therefore be mistakenly interpreted. The range should include the hand as well. The fundamental issue concerns the overfitting of data sets.

METHODOLOGY

Basic methodology of hand gesture detection

In order to identify hand gestures using vision, we came up with and created a basic system that had multiple stages that we could explain using an algorithm. The gesture recognition system's working flowchart is shown in Figure 1.

This experiment was conducted in a well-lit, controlled area without any other skin-like objects in the background. Any opened application window running on the computer can be given a command based on the recognised gesture to control its various functions.

The process flow for detecting hand gestures with OpenCV and background removal using the Gaussian Mixture-based Background/Foreground Segmentation Algorithm is shown below.

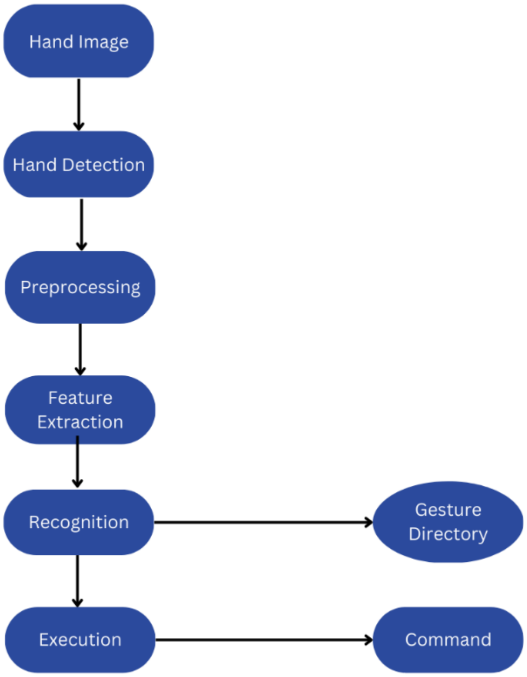

Figure. 1. Flowchart of Gesture Detection (hand gesture)

Proposed methodology

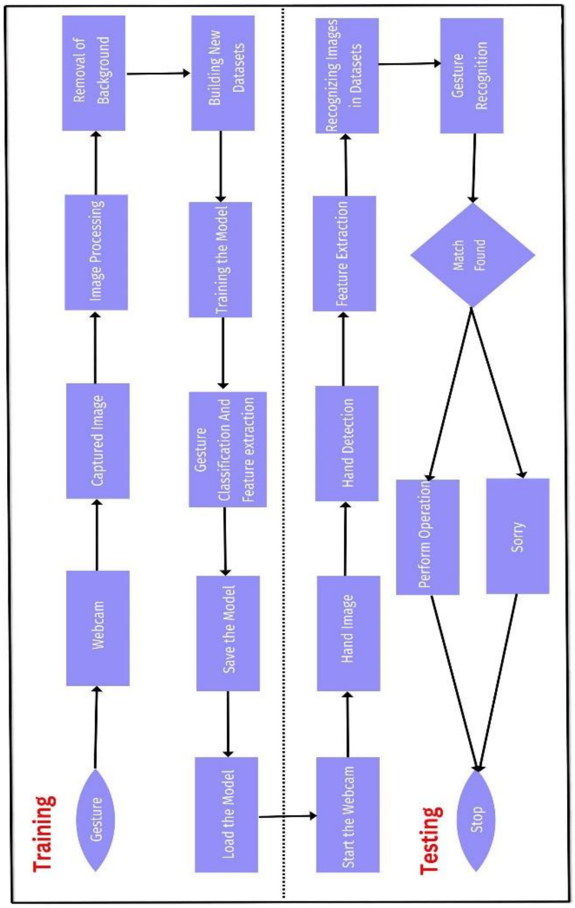

Figure 2. Proposed Methodology of Gesture Detection (hand gesture) using OpenCV

Here is a step-wise methodology for device control using hand gestures explaining the flowchart figure 2.

This flow chart is divided into two phases:

- Training Phase
- Testing Phase

A general explanation of overall methodology is mentioned below as per figure 2.

1. Image Acquisition: Using a camera or depth sensor, capture a live video stream of the hand.

2. Image Pre-processing: If necessary, change the acquired image's color space to grayscale. Use filters to tame the background noise and highlight the hand's outlines. Subtract the background from the hand's image using background subtraction.

3. Hand Detection: To identify the hand region in an image, use computer vision techniques like blob detection, edge detection, or skin Color detection.

4. Hand Segmentation: By separating the hand region from the remainder of the image, isolate the hand region from the background.

5. Extraction of Relevant Hand Features: Extraction of relevant hand features, such as size, shape, orientation, and movement. To extract hand features, use methods like contour detection, convex hull detection, and principal component analysis (PCA).

6. Hand Gesture Recognition: Using methods like machine learning, deep learning, or rule-based algorithms, categories the hand gesture based on the retrieved data. Use tagged data to train a model to recognize hand movements precisely.

7. Device Control: Based on the recognized hand gesture, send orders or carry out operations on the device. Assign various hand gestures to various device commands or activities.

8. Performance Evaluation: Measure the gesture recognition system's accuracy and response time to assess the system's performance. Refine and enhance the system by changing the algorithm and, if necessary, retraining the model.

ALGORITHM

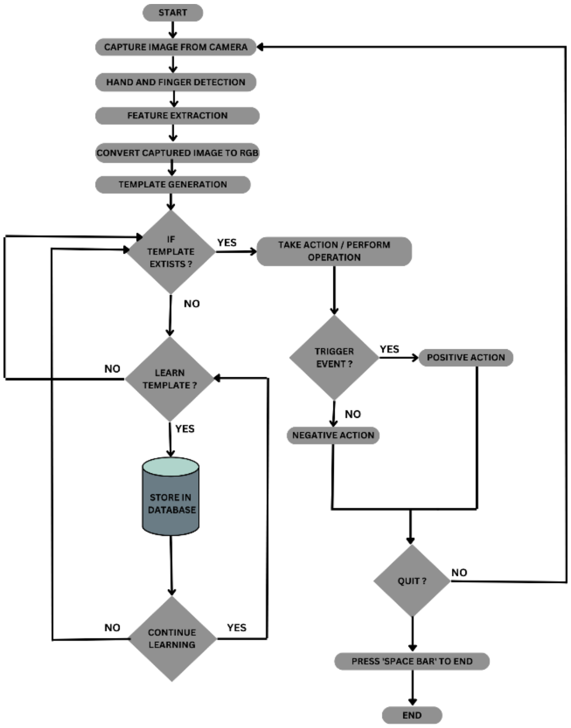

Figure 3. Algorithms of HCI

Algorithms we followed in our implementation:

i. Import necessary libraries like OpenCV, Mediapipe, Numpy, and screen_brightness_control.

ii. Create a VideoCapture object to check the camera.

iii. Initialize the hands module of Mediapipe.

iv. Create a named window and start a thread to manage the window.

v. Start an infinite loop.

vi. Read a frame from the camera.

vii. Convert the frame from BGR to RGB color space.

viii. Process the RGB frame with hands module of Mediapipe and get the landmarks of hands detected in the frame.

ix. If landmarks are detected, draw them on the frame.

x. If fingertips are detected, calculate the distance between them.

xi. Convert the distance value from the hand range (30-350) to respective features range.

xii. Display the frame in the named window.

xiii. If the spacebar is pressed, break the loop.

xiv. Release the camera and close the window

IMAGES

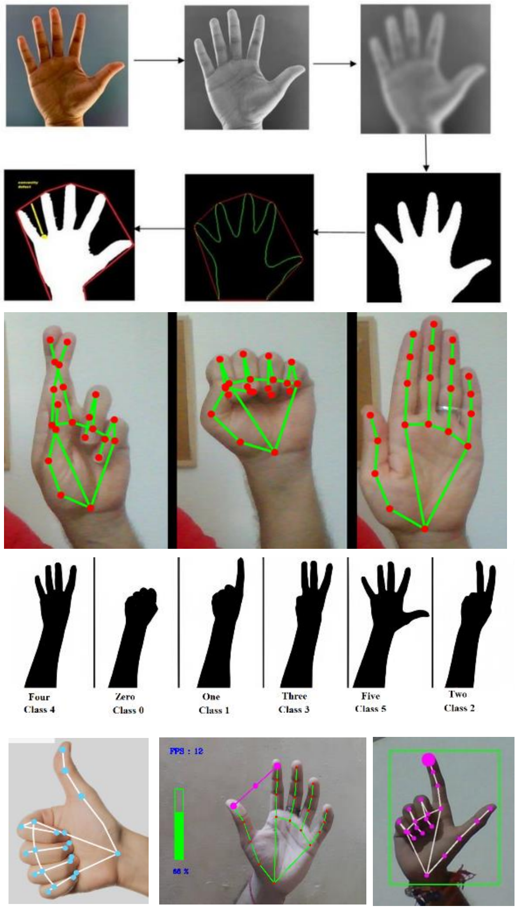

Figure 4. Example of hand gesture detection

RESULT ANALYSIS AND DISCUSSION

- It has been discovered, after conducting the practical testing numerous times, that the system performs exceptionally well in favourable lighting circumstances and in a plain background environment devoid of any skin-like things.
- The algorithm is not particularly reliable because it has a hard time detecting the hand against a complicated background.
- The threshold value that we used to determine the radius using a number of practical estimates also affects how well the system performs.
- Despite this, the system is slightly more responsive than other systems that have been developed in the past because gesture recognition does not require any training.
- For each session, we ran the HGR system and documented the accuracy of the system.
- As a consequence of the studies, we can observe that the system's total accuracy is 95.44%, which is an excellent result.
- The threshold value that we selected is crucial for correctly identifying gestures. After testing the system on a larger number of photos, we can choose a reasonable threshold value to improve the system's performance.
- We've compared our system to a few others and discovered that it has a somewhat higher recognition rate than the others. In several instances, our system had difficulty recognizing particular gestures.

Some limitations are:

- Testing under different lighting conditions It has been noted that our programme had trouble extracting images of hands to analyse further in low illumination situations.
- Another restriction of our programme is that the user's hand must be at least a certain distance from the webcam in order for our distance-based calculation capabilities to function effectively.
- Our code finds it difficult to interpret photos from a distance of less than 2 feet if a user approaches it. This could lead to an inaccurate response or no response at all.
- The webcam's focal angle is also crucial to the software's correct operation.
- The reason for the aforementioned claim is that when the angle is greater than 90 degrees, incorrect input from the user is obtained.

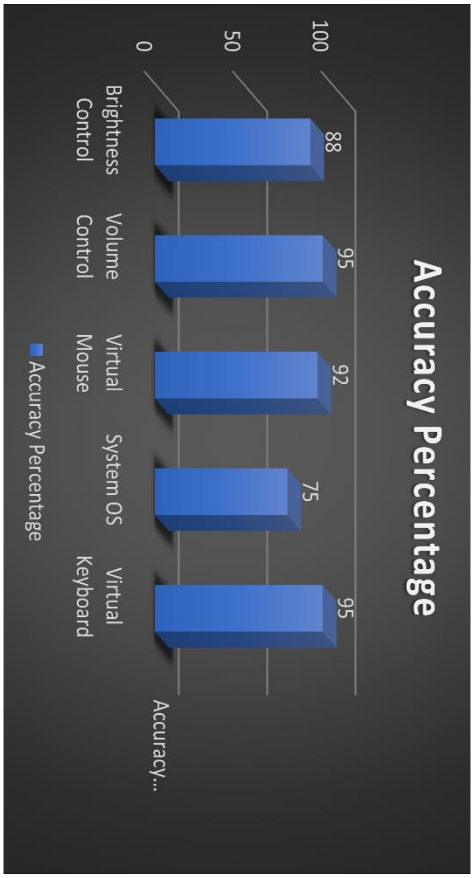

Figure 5. Accuracy percentage of each featured .exe

CONCLUSION

- Gesture-based system control is a technique that allows users to control Windows-based computers with hand gestures rather than using standard input devices like a mouse or keyboard.
- This approach provides a more natural and straightforward user interface, particularly in sectors like gaming, virtual reality, and home automation.
- The need to address problems like precise recognition and hardware limitations still exists, though.
- Despite these challenges, gesture-based system control has enormous promise for revolutionizing how we interact with technology.
- The recognition and detection algorithm needs to be upgraded in order to perform well in crowded and well-lit locations.
- It may be used to control a wide range of programmes, including sports, video players, presentation software, and picture processing tools like Photoshop.
- An average identification rate of 95% has been attained in several experiments on hand gesture recognition utilizing outline analysis.
- The technology has to be enhanced because the present systems still have drawbacks and need to be used in a wider range of situations.
- Future developments could be facilitated by the capacity to handle a variety of computer-related jobs with both hands.
- Additional testing on a larger scale must be conducted to improve the reliability of the results.
- In one application made with this technology, a finger is used to control a mouse and keyboard via a webcam.
- This system can recognize unknown input motions using hand tracking and extraction methods.
- However, this method assumes that the background is stationary to make tracking simpler.

ADVANTAGES AND DISADVANTAGES

Advantages:

- Interaction technique which feels intuitive and natural.
- Enhanced user experience with smooth and immersive interactions.
- Increased accessibility and inclusivity for individuals with physical disabilities.
- Adds a novel and futuristic element to computing experiences.
- Non-contact interaction reduces the risk of germ transmission.

Disadvantages:

- Limited precision compared to traditional input devices.
- There is a learning curve for users to understand and recall particular motions.
- Long-term use might end up in muscle strain and tiredness.
- Sensitivity to environmental elements such as obstacles and lighting.
- Lack of tactile feedback compared to physical buttons or touchscreens.

REFERENCES

- A Real-Time Hand Gesture Recognition System for Daily Information Retrieval from Internet

- Principles of Digital Image Processing Core Algorithms by Wilhelm Burger and Mark J. Burge

- Gesture Recognition Principles, Techniques and Applications by Amit Konar & Sriparna Saha

- S. S. Rautaray and A. Agrawal, Vision Based Hand Gesture Recognition for Human Computer Interaction: A survey, Springer Transaction on Artificial Intelligence Review, pp. 1–54, (2012). C.-C. Chen, H. Chen, and Y.-r. Chen, “A new method to measure leaf age: Leaf measuring-interval index,” American Journal of Botany, vol. 96, no. 7, pp. 1313–1318, Jul 2009.

- Meenakshi Panwar and Pawan Singh Mehra, “Hand Gesture Recognition for Human Computer Interaction”, IEEE International Conference on Image Information Processing (ICIIP 2011), Waknaghat, India, Nov 2011.

- Meenakshi Panwar, “Hand Gesture Recognition based on Shape Parameters”, International Conference on Computing, Communication and Applications (ICCCA), 2012.

- G. Bradski and A. Kaehler, “Learning OpenCV Computer Vision with the OpenCV Library”, O’Reilly Publications, 2008.

- OpenCV Python module https://docs.opencv.org/2.4.13/

- OpenCV installation https://pypi.org/project/opencv-python/

****