Mini Project Report

on

Hand Gesture Recognition Using Machine Learning Techniques

Submitted by

Arjun Sagar N V (20bcs020)

Harshith R N (20bcs056)

Khushi G K (20bcs071)

Rakshitha Y (20bcs107)

Under the guidance of

Dr. Prabhu Prasad

Assistant Professor

DEPARTMENT OF DATA SCIENCE AND INTELLIGENT SYSTEMS

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY DHARWAD

09/05/2023

Contents

List of Figures

1 Introduction

The technique of recognising and deciphering hand gestures produced by people to communicate a certain meaning is known as human hand gesture recognition. Numerous possible uses for this technology exist, including sign language translation, human-computer interface, and virtual reality.

Technology for recognising human hand gestures has advanced significantly thanks to machine learning approaches. These methods analyze massive datasets of hand gesture photos or videos to find patterns using statistical models and algorithms. These models can be taught to recognise a variety of hand gestures with great accuracy by being trained on big datasets.

Pattern recognition and computer vision are the foundations for the machine learning methods used to recognise human hand gestures. The objective of hand gesture recognition is to create algorithms that can accurately decipher and comprehend hand gestures.

Algorithms are trained to recognise hand movements from input data, such as photos or videos, using machine learning techniques. A labeled dataset of hand gestures, with each gesture having a corresponding label, is given to the algorithm during training. The algorithm picks up on the patterns and characteristics in the input data that each gesture label associates with.

Hand gesture recognition often makes use of supervised learning methods like support vector machines (SVMs) and artificial neural networks (ANNs). By identifying the ideal decision boundary that distinguishes the various gesture classes, these algorithms may learn to recognise hand gestures with great accuracy.

Hand gesture recognition also makes use of deep learning methods like convolutional neural networks (CNNs) and recurrent neural networks (RNNs). These algorithms are particularly suited for video-based hand gesture detection because they can be

trained to recognise intricate characteristics and temporal patterns in hand gesture sequences.

Hand gesture recognition also uses computer vision algorithms in addition to machine learning methods. These algorithms can identify and follow hand movements in pictures or movies, as well as extract features for feature-based machine learning algorithms.

2 Related Work

In related research, various methods for solving this issue have been investigated, including supervised and unsupervised learning, deep learning, support vector machines, decision trees, and principal component analysis. Studies have also looked into a range of hand gesture detection applications, including sign language interpretation, human-robot interaction, and virtual reality. These methods have produced promising outcomes, accurately identifying hand motions from pictures or video streams. These research shows how machine learning approaches may enhance human-computer interaction and give users a more comfortable and natural way to connect with technology.

A description of some of the typical methods applied in related work is provided below:

Supervised Learning: In order to recognise hand gestures, supervised learning is frequently utilized. In this method, the system is trained using a labeled dataset of hand gestures. The system learns the connection between the input (hand gesture) and output (matching label) using these labeled samples. In related research, support vector machines (SVMs), decision trees, and artificial neural networks (ANNs) are a few examples of supervised learning techniques.

Unsupervised Learning: Another technique for hand gesture detection involves training the system using an unlabeled dataset of hand motions. Without any prior

knowledge of the labels, the algorithm learns to identify patterns and clusters in the data. Principal component analysis (PCA) and k-means clustering are two examples of unsupervised learning techniques utilized in related work.

Deep Learning: Deep learning is a branch of machine learning that entails building multi-layered artificial neural networks. With excellent success and high accuracy, this method has been used to recognise hand gestures. In deep learning-based hand gesture detection, convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are frequently used.

Applications include sign language translation, human-robot interaction, and virtual reality, among others. Hand gesture recognition has a wide range of uses. For instance, hand gesture recognition is used by sign language translation systems to interpret hand signs into text or speech. Hand gesture recognition is used in human-robot interface systems to promote more intuitive and natural human-robot communication. Hand gesture recognition is a feature of virtual reality systems that enables users to interact with virtual surroundings via hand movements.

The use of machine learning algorithms to recognise human hand gestures has been the subject of numerous prior research investigations. Here are a few noteworthy instances:

The article [3] shows the importance of hand gesture recognition in human-computer interaction, and the difficulties in creating reliable and accurate hand gesture recognition systems are discussed at the outset of the study. The suggested technique comprises employing a Microsoft Kinect sensor to take depth images of hand gestures, preprocessing the photos to get rid of background noise, and then feeding the images into a CNN for classification. The convolutional and pooling layers are followed by fully linked layers in the CNN architecture that was used in the article. The proposed method was tested by the authors using a dataset comprising 10 different hand gestures, and the accuracy was 98.33%. Additionally, they performed tests to assess the method's resilience to various lighting scenarios

and hand orientations, and they discovered that it worked well in both of these situations.

The article [4] is about a system for real-time hand gesture recognition and human-robot interaction is presented in the study. With just one camera, the system can detect ten different hand movements. The three-stage method for the recognition challenge proposed in the research is hand region extraction, feature extraction, and classification. In the first stage, the system uses morphological operations and skin color segmentation to identify and extract the hand region from the input image. In the second stage, features from the hand region are extracted using a pre-trained Convolutional Neural Network (CNN). For this purpose, the CNN is calibrated using a dataset for hand gesture recognition. The hand gesture is classified using an SVM classifier in the third step based on the extracted features. On a dataset comprising 10 hand gestures made by 10 distinct people, the suggested system is assessed. The testing findings demonstrate the suggested system's ability to analyze data quickly (31 frames per second) and with excellent recognition accuracy (97.2%). The potential uses of the suggested system are also covered in the study, including sign language understanding and human-robot interaction. According to the authors, the proposed technology can help deaf people use sign language for communication as well as enabling natural and intuitive interaction between humans and robots.

The article [5] describes the approach for identifying hand gestures using machine learning. The research team gathered a dataset of 1,000 hand gesture photos from various users and expanded it using data augmentation methods. Several machine learning models, such as Support Vector Machines (SVM), Random Forest (RF), K-Nearest Neighbours (KNN), and Convolutional Neural Networks (CNN), were then trained and assessed on the dataset. The CNN model fared better than the other models, obtaining an accuracy of 96.7% on the test dataset, according to the results. In a real-time system for human-computer interaction, the scientists also used the CNN model to enable the system to recognize six different hand movements, including swipe, rotate, zoom-in, zoom-out, play/pause, and stop. The

system had a 93% accuracy rate. The CNN model fared better than the other models, obtaining an accuracy of 96.7% on the test dataset, according to the results. In a real-time system for human-computer interaction, the scientists also used the CNN model to enable the system to recognize six different hand movements, including swipe, rotate, zoom-in, zoom-out, play/pause, and stop. The system's accuracy on the real-time test dataset was 93%.

3 Data and Methods

The Mediapipe library, which we have used, provides a pre-trained machine learning model for hand detection and hand landmark estimation. Specifically, the algorithm used in this code is a deep neural network trained on a set of hand images to accurately detect and initialize hand landmarks. The model uses a combination of convolutional neural networks (CNNs) to achieve high accuracy in hand detection and landmark estimation.

The CNN model consists of several convolutional layers followed by pooling layers to extract features from the input image. These features are then passed through fully connected layers to output the coordinates of the hand landmarks.

A function initializes the pre-trained Mediapipe hand detection model, which internally uses the CNN model for hand landmark estimation. Another function takes an RGB image as input, applies the pre-trained hand detection model to the image, and returns the hand landmarks detected in the image.

3.1 Dataset:

The Hand Gesture Dataset (HGD), a collection of 25,000 photographs of hand movements from 10 different courses, is made available via Google Media Pipe. The hand gesture photos were recorded using a high-resolution camera to build the dataset, which was then manually annotated by a number of human annotators.

To ensure diversity in the dataset, the hand motions are executed by many people with various skin tones, hand sizes, and backgrounds.

The picture dataset and the landmark dataset are the two formats in which the dataset is offered. The landmark dataset comprises the 3D coordinates of the 21 critical spots on the hand for each image, whereas the image dataset just contains the raw images of the hand gestures. The fingertip coordinates, palm center, and wrist center are some of these crucial locations. The training and testing of machine learning models for hand gesture detection uses the landmark dataset.

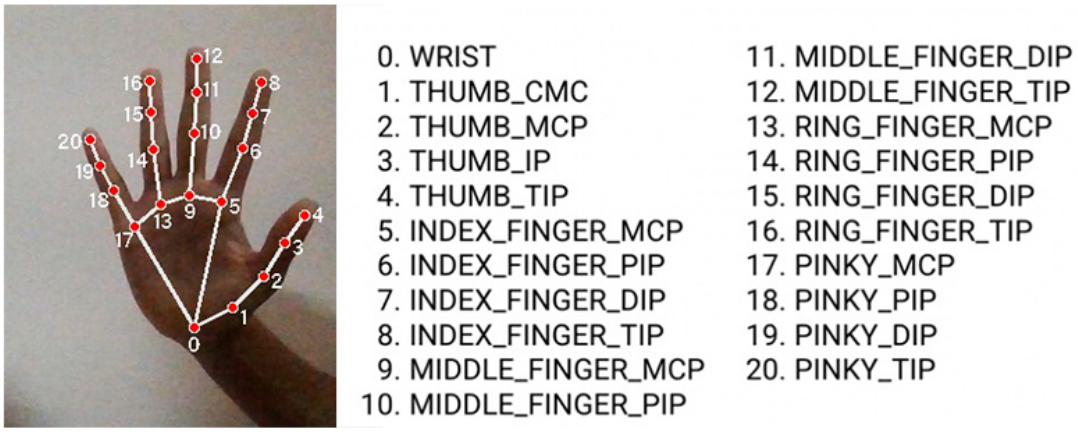

Figure 1. Mediapipe Hand Landmark

3.2 Implementation:

The class handDetector employs the trained model and the mediapipe library to find hands in an image or video stream. The two methods used in this class are find_hands and find_position. Using a pre-trained model and an input image, find_hands searches for any hands in the image. The image is annotated, and the detected hand landmarks are drawn on the source image if any hand is detected. find_position accepts an input image and returns a list of all the hand landmarks that the trained model has identified in that image. If more than one hand is present in the input image, it additionally provides an optional parameter that allows the user to choose which hand to detect. Every recognized landmark on the input image has a circle drawn around it if the hand is detected.

The minimal detection confidence is set at 0.7, and the mediapipe hand detection module is initialized in the constructor. The detected hand landmarks are also initialized in the drawing_utils module to be drawn on the input image.

The pre-trained model is loaded and saved in the class hands object when the class is instantiated. The pre-trained model is used to find hands and their landmarks in the input image when the find_hands or find_position methods are invoked. The landmarks are recorded and returned as a list of positions if any hands are found. The recognized landmarks are additionally drawn on the input image if the hand is detected.

The gesture media control is regulated using hand gesture detection and computer vision. It uses the PyAutoGUI, Pycaw, and OpenCV libraries.

The pycaw.pycaw library is used to import the AudioUtilities, IAudioEndpointVolume, cast, and POINTER objects.

The GetSpeakers() function from AudioUtilities is used to retrieve a list of all the audio output devices on the system. The activate() function is used to enable

volume control for the default device. Cast() is used to build a pointer for the volume control. The volume range is stored in the volRange variable and can be retrieved using the GetVolumeRange() function.

The volume can be controlled using the volume_control() function. The frame, the coordinates of the thumb and index finger tips, and the function are all inputs. The circles around the thumb and index finger tips, the line connecting the two spots, the middle of the line, and the circle surrounding it are all drawn using these coordinates. The Pythagorean theorem is used to determine the line's length, and the np.interp() function is used to translate that length to the volume range. Using the SetMasterVolumeLevel() function, the volume is adjusted. To show the volume change, a rectangle is drawn, and the percentage is calculated by mapping the volume from 0 to 100.

A cv2.VideoCapture() object is created using the cap variable. To build a detector object, the handDetector() function is used.

The picture is read and flipped in the main loop. The hands in the image are located and drawn using the find_hands() method. All 21 points are given index, x, and y coordinates, and the find_position() method is used to determine their locations. The finger list contains the locations of the tips of every finger but the thumb.

The function volume_control() is called if the size of the first list exceeds 9. If the thumb is not folded, the volume can only be altered. The playpause function of the pyautogui library is used to play or pause the media if the thumb is folded and the other fingers are as well. The volume up function of the pyautogui library is called if the thumb is folded and all fingers aside from the little finger are folded. The volume down function of the pyautogui library is called if the thumb is folded and all fingers are folded save for the index finger, which is pointed forward.

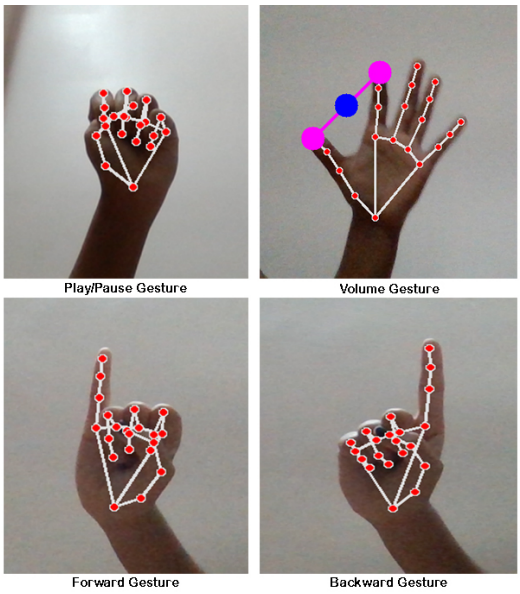

Figure 2. Hand Gestures

With the aid of cv2.imshow(), the video frame is shown. When the q key is pushed, the loop is ended, and the windows are eliminated using cv2.destroyAllWindows().

4 Results and Discussions

The python code 'handDetectorModule' takes the webcam video(series of images) from the system as an input. The code will check if there are any hands in the input images taken by the camera. If yes, then the module will draw landmarks on the hand based on the mediapipe hand module. Then the image is returned with the landmarks drawn. A list is created which will take all the positions of the landmarks of the hand and return that list.

In the python code 'gestureMediaController' we have imported pyautogui for keyboard and mouse controls. We have also imported a pycaw library for audio control. We have created our own gestures using the mediapipe framework image

output given by the 'handDetectorModule'. The fingertips help in the gesture recognition. The results obtained at the end include:

1. **The distance between the index finger and the thumb implements the volume controls(Increase/Decrease).**
2. **If the thumb is folded:**
    - **all the other four fingers are folded, then the media will pause or play.**
    - **If three fingers are folded and the little finger is open, then the media will go five seconds forward.**
    - **If three fingers are folded and the index finger is open, then the media will go five seconds backwards.**

Hand gesture detection for media control is an innovative use of machine learning. Hand motions taken by a camera are detected, analyzed, and then used to control different media playback functions like volume, play/pause, forward, and rewind.

There are many benefits to using gesture recognition for media control. It provides a touchless and hands-free solution, for example, making it a convenient and user-friendly choice. It can also be useful when both hands are busy if there is no direct access to media playback controls.

The application of machine learning techniques, especially image processing and computer vision techniques, is required for the implementation of this technology. These algorithms are useful for real-time tracking the user's hand and position,and then mapping the hand's movement to a specific media control.

The MediaPipe framework, which provides a pre-trained model for hand detection and tracking, is a popular tool for hand gesture recognition. With a set of hand gesture images, this model can be fine-tuned to recognize particular gestures and map them to media control functions.

5 Conclusion

Due to its potential use in a variety of domains, including human-computer interaction, robotics, and virtual reality, human hand gesture detection using machine learning techniques has grown in importance as a subject of study. Convolutional neural networks and other deep learning techniques have demonstrated outstanding performance in the recognition of hand motions from photos and movies using machine learning algorithms. These techniques can be used to accurately classify hand motions in real-time.

A number of researchers have investigated various machine learning strategies and methodologies for hand gesture recognition, and they have produced encouraging results in terms of precision and speed. Nevertheless, there are still several difficulties and restrictions in this area, including the requirement for sizable and varied datasets, a range of lighting conditions, and the occlusion of hand gestures.

The ability to recognise hand gestures using machine learning techniques has significant promise to improve human-machine interaction and open up new possibilities in numerous industries. The accuracy and effectiveness of hand gesture recognition systems are likely to continue to improve as this field of study is pursued, creating new possibilities for interaction and communication between humans and machines.

In order to advance the field of machine learning-based hand gesture identification, it is important for future research to prioritize efforts on multiple fronts. This could involve enhancing the robustness of the recognition systems to increase their resilience to external factors, as well as generating larger and more diverse datasets to improve their generalizability. Additionally, exploring novel applications and multimodal methods, such as combining hand gesture recognition with speech or facial emotions, could lead to more natural and intuitive human-machine interfaces. By pursuing these avenues of research, it is possible to not only improve the accuracy and performance of hand gesture recognition systems, but also pave the way for new applications and enhanced communication between humans and machines.

References

[1] . URL https://techvidvan.com/tutorials/hand-gesture-recognition-tensorflow-opencv/.

[2] . URL https://www.researchgate.net/publication/345142103_Hand_gesture_recognition_using

[3] Norah Alnaim, Maysam Abbod, and Abdulrahman Albar. Hand gesture recognition using convolutional neural network for people who have experienced a stroke. In *2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, pages 1–6, 2019. doi: 10.1109/ISMSIT.2019.8932739.

[4] Abhishek B., Kanya Krishi, Meghana M., Mohammed Daaniyaal, and Anupama H. S. Hand gesture recognition using machine learning algorithms. *Computer Science and Information Technologies*, 1(3):116–120, 2020. ISSN 2722-3221. doi: 10.11591/csit.v1i3.p116-120. URL http://iaesprime.com/index.php/csit/article/view/96.

[5] Edison A. Chung and Marco E. Benalcázar. Real-time hand gesture recognition model using deep learning techniques and emg signals. In *2019 27th European Signal Processing Conference (EUSIPCO)*, pages 1–5, 2019. doi: 10.23919/EUSIPCO.2019.8903136.

[6] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. Real-time continuous pose recovery of human hands using convolutional networks. volume 33, 08 2014. doi: 10.1145/2629500.