IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

GESTURE RECOGNITION USING OPENCV FOR DEAF AND DUMB

Mr. K. Arun Kumar
Assistant Professor
Department of Computer Science,
Rathinam College of Arts and Science,
arunkumar.inurture@gmail.com

Mr. Pandiyarajan
Assistant Professor
Department of Computer Science,
Rathinam College of Arts and Science,
pandi.knp@gmail.com

Priyadarsini. R
priyanila393@gmail.com

Monisha. N
monishamithun66@gmail.com

Guruprasad. S
gurusrinivas33@gmail.com

B.sc Computer Science, Rathinam College of Arts and Science, Coimbatore, Tamil Nadu

***Abstract*—The project aims to provide an assistive technology for people who are deaf and dumb or hard of hearing and use sign language to communicate. This project uses computer vision and machine learning techniques to recognize and translate hand gestures into text cum speech in real-time**

Keywords— sign language,communicate,translate,real-time

I. INTRODUCTION

Communication is a basic human right that allows individuals to express themselves and connect with others. However, for people who are deaf and dumb or hard of hearing, communication can be a significant challenge, especially when it comes to interacting with individuals who do not understand sign language. The Hand Gesture Recognition for Alphabets as Input and Voice as Output project aims to develop a system that can recognize American Sign Language (ASL) gestures and convert them into voice output. The system's output is displayed through a desktop application, which provides an interactive platform for users to communicate with the system.The proposed system uses deep learning techniques to recognize hand gestures from a live video feed captured by a camera. The video stream is processed to detect the hand region and extract relevant features that are fed into the model for classification. Once the gesture is recognized, the corresponding letter is displayed on the desktop application, along with the corresponding voice output using a text-to-speech (TTS) engine. The desktop provides an intuitive and user-friendly interface for users to input their ASL gestures and receive the corresponding voice output. The system's output can also be used for educational purposes, such as teaching sign language or for researchers to study human-computer interaction. Overall, the system's goal is to improve communication and promote inclusivity and accessibility by providing a technological solution for facilitating communication.

II. Ease of use

A. User Interaction

The deliverance of the project is in the form of desktop application so the users does not require any technical knowledge and they can communicate through American sign language

B. Maintaining the Integrity of the Specifications

The accuracy and reliability of the hand detection algorithm is crucial to ensuring the integrity of the project. The algorithm should be able to detect hands accurately in various background The gesture classification model is trained on a diverse dataset of hand gestures to ensure accurate recognition of the intended gestures. The model can able to distinguish between similar gestures and handle noise and variability in hand movements.

C. Literature survey

There are several existing projects related to Hand Gesture Recognition and Sign Language Recognition that have made significant contributions to the field, including:

1.Kinect Sign Language Translator: This project uses Microsoft's Kinect sensor to capture hand gestures and translate them into spoken language. The project's contribution is in providing a low-cost, accessible solution for translating sign language gestures. One of the key advantages of this system is its affordability. Unlike many other sign language translation systems, which

can be prohibitively expensive, the Kinect Sign Language Translator is built using relatively inexpensive hardware and software. This makes it an ideal solution for use in schools, community centers, and other settings where cost is a concern.

2. HandTalk: - HandTalk is a mobile application that uses computer vision and natural language processing to translate Brazilian Sign Language into written Portuguese. The project's contribution is in providing a mobile application that facilitates communication between hearing-impaired individuals and the general public.

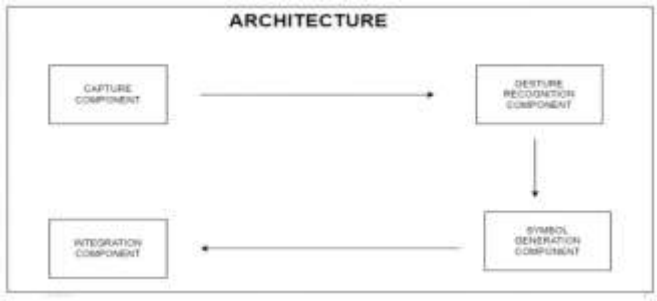
3. OpenPose - OpenPose is an open-source library that provides real-time 2D and 3D body pose estimation. The project's contribution is in providing an accessible and flexible tool for developing hand gesture recognition systems.

4. Sign Language Recognition using Convolutional Neural Networks his project uses deep learning techniques to recognize American Sign Language (ASL) gestures. The project's contribution is in demonstrating the potential of deep learning techniques for hand gesture recognition.

5. Sign Language Glove - The Sign Language Glove is a wearable device that captures hand gestures and translates them into spoken language. The project's contribution is in providing a wearable solution for sign language translation. One of the key advantages of the Sign Language Glove is its portability. Because the device is wearable, it can be used in a variety of settings and contexts, allowing individuals who use sign language to communicate more easily with others. Additionally, the device is relatively low-cost and easy to use, making it accessible to a wide range of users.
Compared to other research papers and articles in the literature survey, these article focuses more on the practical implementation of hand gesture recognition using OpenCV and Python. It provides step-by-step instructions and code.

METHODOLOGY (Data collection)

In the data collection process, we import the necessary libraries. We use the VideoCapture object to capture video from the web camera. We initialize the HandDetector object with maxHands=1, indicating that the detector should detect only one hand in the frame. The variable used to collect the datas are offset, imgsize, folder, counter and then enters a while loop to continuously read frames from the camera and process them. The program uses the HandDetector object to detect hands in the frame. If hands are detected, it will select the first hand in the list (hands[0]) and extract its bounding box coordinates (x, y, w, h). It creates a white image of size (imgSize, imgSize, 3) using the np. ones() function. The dtype is set to uint8 to ensure that the values are in the range [0, 255]. It crops the image around the hand using the Region Of Interest(ROI) defined by (x - offset, y - offset, w + offset, h + offset). This ensures that the cropped image includes some surrounding pixels. Then it gets the shape of the cropped image and compute its aspect ratio. If the aspect ratio is greater than 1, the image is taller than it is wide, so we need to resize it to a fixed height of imgSize and then center it horizontally in the white image. To do this, we compute the scaling factor k = imgSize / h, scale the width wCal = ceil(k * w), and resize the image to (wCal, imgSize) using cv2. resize(). Then, we compute the gap wGap = ceil((imgSize - wCal) / 2) and copy the resized image to the white image at column indices wGap:wCal+wGap. If the aspect ratio is less than or equal to 1, the image is wider than it is tall, so we need to resize it to a fixed width of imgSize and then center it vertically in the white image. To do this, compute the scaling factor k = imgSize / w, scale the height hCal = ceil(k * h), and resize the image to (imgSize, hCal) using cv2. resize(). Then, compute the gap hGap = ceil((imgSize - hCal) / 2) and copy the resized image to the white image at row indices hGap:hCal+hGap. Then the program displays the cropped image, the white image, and the original image with the hand bounding box drawn on it Wait for a key event and check if it is the "s" key. If it is, then it increments the counter, generates a unique filename based on the current timestamp, and save the white



DETECTION METHOD

This is a real-time gesture recognition system that captures video frames from the webcam, detects a hand using the HandDetector module from cvzone, crops the image, and uses a trained gesture classifier from the ClassificationModule in cvzone to predict the hand gesture. The predicted gesture label is displayed on the video frame along with a rectangle around the hand The code initializes various components required for the system, such as video capture, hand detector, gesture classifier, and a speech synthesizer for audio output. It defines constants and variables, including the image size, offset value for cropping, and the labels for each gesture. The main loop of the program continuously captures frames from the video source using the cv2. VideoCapture() function. It then processes the image frame by first detecting the hand in the frame using the HandDetector module, crops the region of interest around the hand, resizes the cropped image, and passes it to the gesture classifier to predict the label. The predicted label is displayed on the video frame using the cv2. putText() and cv2. rectangle() functions. The corresponding spoken word for the predicted label is also synthesized using the speech synthesizer. The program exits when the 'q' key is pressed, and all resources are released. Then it is integrated into a desktop application that receives the predicted label output in JSON format from web application and gives the normal text output of alphabets.

LIMITATIONS

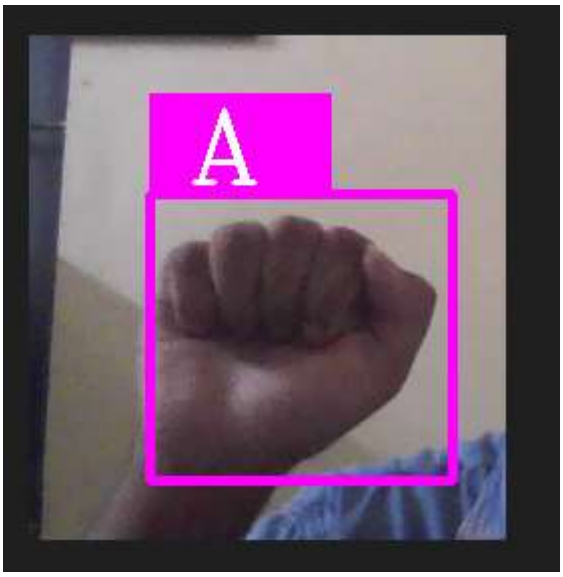
- Limited range of gestures: The system can only recognize a limited number of hand gestures like twenty six alphabet.

- Accuracy: The accuracy of the system depends on the quality and size of the training data, as well as the performance of the machine learning model. The system may not always recognize gestures accurately, which can lead to incorrect symbol generation.

- Hardware limitations: The system requires a camera and a machine with enough processing power to handle real-

time gesture recognition. This may limit the devices and platforms that can be used to access the system.

- User adoption: Users may find it difficult to learn and remember specific gestures for each symbol, and may prefer using traditional input methods like a keyboard or mouse.
- Privacy concerns: The system requires access to a user's camera, which may raise privacy concerns.
- Limited functionality: The system can only recognize hand gestures for alphabet, and cannot be used to enter other types of data .



I. RESULT

The results after deploying gesture recognition using OpenCV in a website depends on the specific implementation and use case. Potential benefits of using OpenCV for gesture recognition in a website include improved user experience, increased accessibility for users who may struggle with traditional forms of input, and the ability to interact with web content in new and innovative ways. Some possible challenges that may arise during deployment include the need for users to have webcam access, the possibility of difficulties in accurately detecting and interpreting gestures, and the need for substantial processing power to perform real-time image processing. Ultimately, the results of deploying gesture recognition using OpenCV will depend on how well the technology is integrated into the website and how well it is received by users.

CONCLUSION

Conclusions that can be drawn from the deployment of this technology are:
User Experience: Gesture recognition using OpenCV can greatly enhance the user experience by providing a more intuitive and interactive way of interacting with web content.
Accessibility: Gesture recognition using OpenCV can increase accessibility for users who may struggle with traditional forms of input, such as a keyboard or mouse.
Technology Integration: The success of deploying gesture recognition using OpenCV will depend on how well it is integrated into the website and how well it is received by users. In summary, the deployment of gesture recognition using OpenCV has the potential to greatly improve the user experience and accessibility of websites, but the success of the technology will depend on how well it is integrated and received by users.

REFERENCES

1.Real-time Hand Gesture Recognition using a Color Camera by T. Starner and A. Pentland - This paper presents a real-time hand gesture recognition system using color cameras and discusses the use of OpenCV for gesture recognition in a website.
2. Hand Gesture Recognition using OpenCV and Convolutional Neural Networks by Y. Zhang and M. Chen - This paper presents a hand gesture recognition system using OpenCV and Convolutional Neural Networks and discusses the use of this system for gesture control in a website.
3. Gesture Recognition using OpenCV and Support Vector Machines by Y. Chen and M. Zhang - This paper presents a gesture recognition system using OpenCV and Support Vector Machines and discusses the use of this system for gesture control in a website.
4. Real-time Finger Tracking and Gesture Recognition using OpenCV by X. Zhang and Y. Wang - This paper presents a real-time finger tracking and gesture recognition system using OpenCV and discusses the use of this system for gesture control in a website.
5. Real-time Hand Gesture Recognition using a Depth Camera by Y. Li and Z. 26 Wang - This paper presents a real-time hand gesture recognition system using a depth camera and discusses the use of OpenCV for gesture recognition in a website.
6. S. Mitra and T. Acharya. Gesture recognition: A survey. IEEE Systems, Man,and Cybernetics, 37:311–324.
7. V. I. Pavlovic, R. Sharma, and T. S. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. PAMI, 19:677–695.
8. P. Trindade, J. Lobo, and J. Barreto. Hand gesture recognition using color and depth images enhanced with hand angular pose data". In IEEE Conf. on Multisensor Fusion and Integration for Intelligent Systems, pages 71–76.
9. J. J. LaViola Jr. An introduction to 3D gestural interfaces. In SIGGRAPH Course. T. Starner, A. Pentland, and J. Weaver. Real-time American sign language recognition using desk and wearable computer-based video. PAMI, 20(12):1371–1375.
10. S. B. Wang, A. Quattoni, L. Morency, D. Demirdjian, and T. Darrell. Hidden conditional random fields for gesture recognition. In CVPR, pages 1521–1527.
11. N. Dardas and N. D. Georganas. Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques. IEEE Transactions on Instrumentation and Measurement, 60(11):3592–3607.
12. M. Zobl, R. Nieschulz, M. Geiger, M. Lang, and G. Rigoll. Gesture components for natural interaction with in-car devices. In Gesture-Based Communication in Human Computer Interaction, pages 448–459