Sign Language Recognition using Machine Learning

[1]Sameer Metkar, [2]Shubham More, [3]Durgesh Kolhe, [4]Omkar Mandavkar,[5]Prof. Aarti Abhyankar

[1]Student, K.C. College of Engineering and Management Studies and Research, University of Mumbai, Thane
[2]Student, K.C. College of Engineering and Management Studies and Research, University of Mumbai, Thane
[3]Student, K.C. College of Engineering and Management Studies and Research, University of Mumbai, Thane
[4]Student, K.C. College of Engineering and Management Studies and Research, University of Mumbai, Thane
[5]Guide, K.C. College of Engineering and Management Studies and Research, University of Mumbai, Thane

Abstract:. Communication is a very important aspect of society. It is said that Man is a Social animal, and for this to be literal an efficient communication is very necessary. Similarly in the stone age, communication happened using drawings etc and as we progressed , society developed script, languages which eventually led to the growth of society. With time globalization started and the spread of languages began, The languages then were classified into verbal and non verbal. People who travelled and couldn't understand a language used non verbal forms to communicate in foreign lands. But Non Verbal languages are not only meant for that but also the only method for People With Hearing and Speaking Disabilities (here onwards mentioned as PWHSD) to communicate without external tools. And since they are small percent of the consensus they are often disregarded or not thought of generally. Though they are a small percent but definitely not a negligible share. To have conversations with them keeping in mind that not everyone can learn another new non verbal language for communicating with them, to facilitate virtual communication with them we have developed a model that can help PWHSD individuals to communicate with others and also help others understand what they are saying. For this we have used a Mediapipe model that can in real time recognize sign language. With the amount of dataset we have provided the accuracy of the model is pretty good.

.

Index terms: Sign Language Recognition (SLR), Computer Vision, Machine Learning, American Sign Language, TensorFlow, Mediapipe, LearnSl.

I. INTRODUCTION

The act of transmitting information from one location, person, or group to another is referred to as communication. The speaker, the message being conveyed, and the listener make up its three elements. Only after the audience receives and comprehends the speaker's intended message can it be said that it was successful. It can be broken down into the following categories: formal and informal communication, oral (face-to-face and across distance), written, non-verbal, grapevine, feedback, visual, and active listening.

Deaf and dumb persons can communicate with one other and with others by using nonverbal cues. A person who is deaf has a hearing impairment that prevents them from hearing, whereas a person who is stupid has a speech impairment that prevents them from speaking. It is challenging to create communication when one cannot talk or listen. Here sign languages play a crucial role in allowing people to communicate without using words. Yet, there is still a difficulty because not many people are familiar with sign language. Although though deaf and dumb individuals can communicate with one another through sign languages, it is still challenging for them to converse with those who have normal hearing, and vice versa, since they lack the ability to hear. Using a technology-driven solution can solve this problem. A solution like this makes it simple to translate sign language gestures into English, a widely spoken language.

A great deal of examination has been finished in this field and there is as yet a requirement for additional re-search. Data gloves, motion capturing systems, or sensors have been utilized for gesture translation [2]. In the past, vision-based SLR systems have also been developed [3]. MATLAB machine learning algorithms were used to create the current Indian Sign Language Recognition system [4]. Both single-handed and double-handed gestures have been studied by authors. The K Nearest Neighbors Algorithm and the Back Propagation Algorithm were used to train their system. Their system was accurate between 93% and 96%. It is an extremely accurate system, but it is not a real-time SLR. Using the Mediapipe object and pose detection API, this paper aims to create a real-time SLR system and train it on a webcam-generated dataset.

The following is the structure of the remainder of this paper after the introduction. The associated work on the SLR system is outlined in Section 2. The process of acquiring and creating data is described in Section 3. The developed system's methodology is the focus of Section 4. The system's experimental evaluation is presented in Section 5, and the paper's future work is discussed in Section 6

II. RELATED WORK

Gesture-based communications are characterized as a coordinated assortment of hand motions having specific implications which are utilized from the meeting weakened individuals to convey in daily existence [3]. They communicate through hand, face, and body movements because they are visual languages. There are more than 300 distinct sign languages available worldwide [5]. Even though there are many different sign languages, only a small percentage of people know any of them, making it difficult for people with special needs to freely communicate with everyone. SLR lets people communicate without knowing sign language. It converts a gesture into a language that is commonly spoken, like English.

Different devices must be utilized for data acquisition via direct measurement or vision-based approaches. In an SLR system, the camera is the primary input device [13]. Other input devices are available, such as Microsoft Kinect, which provides both depth and color video streams simultaneously. The profundity of information helps in foundation division. Sensory gloves and accelerometers are two additional data collection methods in addition to the devices. The Leap Motion Controller (LMC), a touchless controller developed by the San Francisco-based technology company "Leap Motion," is another system used for data acquisition [14, 15]. It can detect and track fingers, and objects that look like fingers, and can operate at around 200 frames per second. Since finding a sign language dataset is difficult, the majority of researchers collect their training dataset by recording it from their signer [2].

An SLR system has been created using a variety of processing techniques [16], [17], and [18]. SLR makes extensive use of the Hidden Markov Model (HMM) [12]. Multi-Stream HMM (MSHMM), which is based on the two standard single-stream HMMs, Light-HMM and Tied-Mixture Density-HMM [2], are the various HMMs that have been utilized.

Different accuracy results have been achieved by utilizing various application systems or processing techniques. The Light-HMM had an accuracy of 83.6 percent, the MSHMM had an accuracy of 86.7 percent, the SVM had an accuracy of 97.5 percent, the Eigen Value had an accuracy of 97%, and the Wavelet Family had an accuracy of 100 percent [2][31][22][32]. Even though different processing models have produced results with high accuracy, the accuracy depends on a variety of other factors as well, including the dataset's size, the clarity of the images in the dataset, and the data acquisition methods and devices used.

Isolated SLR systems and continuous SLR systems are the two types of SLR systems. The system is trained to recognize a single gesture in an isolated SLR. The labels on each image represent a letter, a number, or a particular gesture. Continuous SLR and isolated gesture classification are distinct. The system can recognize and translate entire sentences rather than just a single gesture in continuous SLR [33][34].
Even with all of the research that has been done on SLR, there are still many flaws that need to be fixed by more research. The following are some of the issues and difficulties that need to be addressed: [33][2][4][6].

- Each word must be labeled meticulously using isolated SLR techniques.
- Continuous SLR techniques use isolated SLR systems as building blocks, with sentence synthesis serving as post-processing and temporal segmentation serving as non-trivial pre-processing, which invariably introduces errors into subsequent steps.
- For SLR systems to become commercially available, a less expensive method of data acquisition is required.
- Sensor data acquisition also has some problems, like noise, bad human manipulation, a bad ground connection, and so on.
- The overlapping of the hand and finger introduces inaccuracies in vision-based methodologies.
- There are no large datasets available.
- There are misconceptions about sign language, such as the idea that sign language is the same everywhere and is based on spoken language.

III. DATA ACQUISITION

A real-time sign language detection system is being developed for American Sign Language. For data acquisition, images are captured by webcam using Python and OpenCV. OpenCV provides functions which are primarily aimed at the real-time computer vision. It accelerates the use of machine perception in commercial products and provides a common infrastructure for the computer vision-based applications. The OpenCV library has more than 2500 efficient computer vision and machine learning algorithms which can be used for face detection and recognition, object identification, classification of human actions, tracking camera and object movements, extracting 3D object models, and many more [35].

The created dataset is made up of signs representing alphabets in American Sign Language [36] as shown in Fig. 1. For every alphabet, 25 images are captured to make the dataset. The images are captured in every 2 seconds providing time to record gesture with a bit of difference every time and a break of five seconds are given between two individual signs, i.e., to change the sign of one alphabet to the sign of a different alpha- bet, five seconds interval is provided. The captured images are stored in their respective folder.

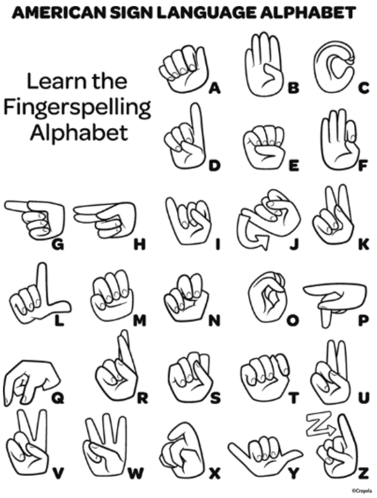

AMERICAN SIGN LANGUAGE ALPHABET
Learn the Fingerspelling Alphabet
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z
©Crepola

Fig. 1. American Sign Language Alphabets

For data acquisition, dependencies like cv2, i.e., OpenCV, os, time, and uuid have been imported. The dependency os is used to help work with file paths. It comes under standard utility modules of Python and provides functions for interacting with the operating systems. With the help of the time module in Python, time can be represented in multi- ple ways in code like objects, numbers, and strings. Apart from representing time, it can be used to measure code efficiency or wait during code execution. Here, it is used to add breaks between the image capturing in order to provide time for hand movements. The uuid library is used in naming the image files. It helps in the generation of random objects of 128 bits as ids providing uniqueness as the ids are generated on the basis of time and computer hardware.

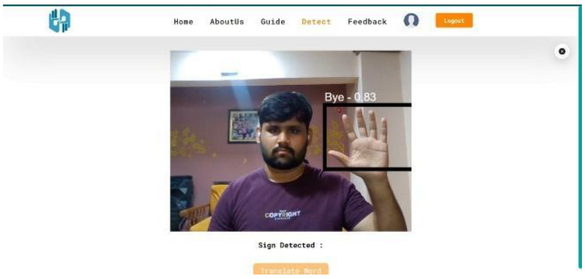


Fig. 2. Showing a Finger and testing detection

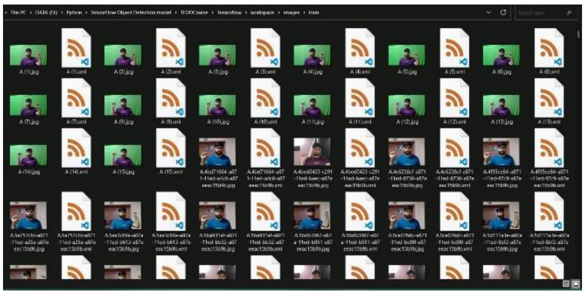

This PC › DATA (D:) › Python › TensorFlow Object Detection model › TFODCourse › Tensorflow › workspace › images › train
A (1).jpg
A (1).xml
A (2).jpg
A (2).xml
A (3).jpg
A (3).xml
A (4).jpg
A (4).xml
A (5).jpg
A (5).xml
A (6).jpg
A (6).xml

Fig. 3. Dataset 1

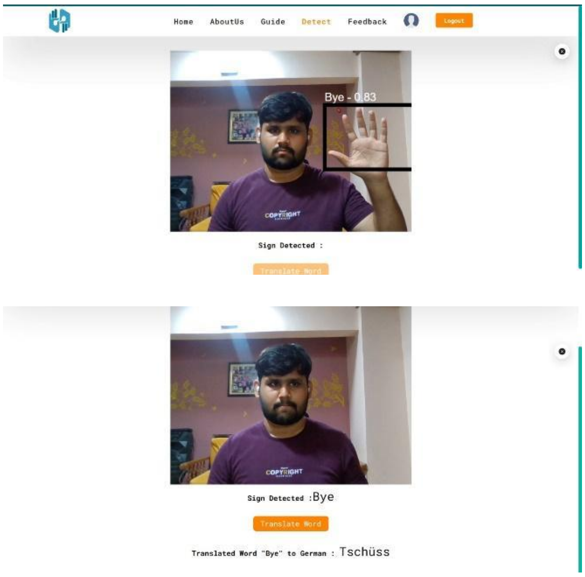
When every one of the pictures has been caught, they are then individually named utilizing the LabelImg bundle. An open-source, free tool for graphically labeling ages is LabelImg. As depicted in Figure, the gesture or sign in the box is used to identify the hand gesture portion of the image. 2, and Figure 3. We made advantage of Google's Firebase, a cloud-based platform, to host the trained model and store user data. Our React application accessed the trained model file that we had placed in Firebase Storage. In order to enable users to log in and access their personalised data, we also implemented Firebase Authentication. Finally, we deployed our React application using Firebase Hosting and made it available to the public. In total, our project's data gathering, preprocessing, model training, integration with React, and hosting using Firebase required a number of processes. We were able to create a real-time sign language recognition programme that can recognise a range of sign language motions by combining these technologies and methodologies.

IV. METHODOLOGY

We started with the MediaPipe Gesture Recognition pre-trained model for our project on sign language recognition. The pre-trained model, however, may not perform well on certain sign language movements that we are interested in because it was trained on a general set of photos. As a result, we used our own proprietary dataset to train the machine on pictures of sign language gestures that we hope to identify. The labeled pictures of sign language motions that we gathered from various sources make up the custom dataset. To make sure the photographs were the same size, aspect ratio, and orientation, we also had to preprocess them.After the model had been trained on our unique dataset, we merged it with React, a well-liked JavaScript front-end toolkit. We were able to load and run the trained model in real-time on a user's web browser using the MediaPipe JavaScript package, which allowed us to recognise sign language motions in real-time without the need for further backend processing. N is the number of matched default boxes, l is the predicted bounding box, g is the ground truth bounding box, g is the encoded ground truth bounding box, and xk is the matching indicator between default box i and ground truth box j of category k. The classification loss is defined as the softmax loss over multiple classes..

The different losses incurred during the experimentation are mentioned in the subsequent section. After training, the model is loaded from the latest checkpoint which makes it ready for real-time detection. After setting up and updating the configuration, the model will be ready for training. The trained model is loaded from the latest check-point which is created during the training of the model. This completes the model mak- ing it ready for real-time sign language detection.

The real-time detection is done using OpenCV and webcam again. For, real-time detection, cv2, and NumPy dependencies are used. The system detects signs in real- time and translates what each gesture means into English as shown in Fig. 5. The system is tested in real-time by creating and showing it different signs. The confidence rate of each sign (alphabet), i.e., how confident the system is in recognizing a sign (alphabet) is checked, noted, and tabulated for the result. We made advantage of Google's Firebase, a cloud-based platform, to host the trained model and store user data. Our React application accessed the trained model file that we had placed in Firebase Storage. In order to enable users to log in and access their personalised data, we also implemented Firebase Authentication. Finally, we deployed our React application using Firebase Hosting and made it available to the public. In total, our project's data gathering, preprocessing, model training, integration with React, and hosting using Firebase required a number of processes. We were able to create a real-time sign language recognition programme that can recognise a range of sign language motions by combining these technologies and methodologies.



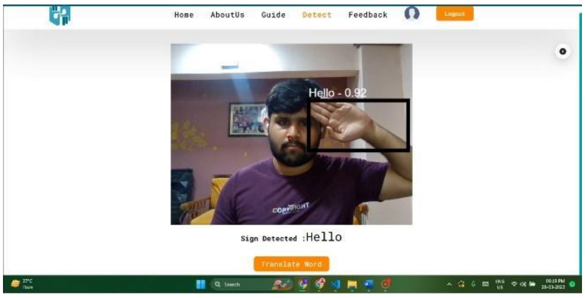
Sign Detected :

Sign Detected :Bye

Translated Word "Bye" to German : Tschüss

V. EXPERIMENTAL EVALUATION

The accuracy of this model is pretty good and around 85% for the given dataset. .The model was trained on a large and varied dataset of sign language videos and uses a deep learning approach to accurately classify hand gestures and movements. It is able to recognize signs in real-time and can be used to improve accessibility and communication for the deaf and hard-of-hearing community. The translation was also on point and hence will help people who know only 1 specific language. During testing the model was tested on all 26 alphabets and some popular common words and it managed to recognize almost all of them with a few exceptions. Also the accuracy of the identified letters and words was pretty good and always nearing or around 85%.



Sign Detected :Hello

. Figure ; Detecting a sign with 92% accuracy

VI. CONCLUSION AND FUTURE SCOPE

Dumb and Deaf people are often deprived of opportunities because of the communication barrier. And if they are included they always have to carry a notepad or some or the other means of communication with them. With the help of our project we can easily help them communicate with the rest and also help others who want to learn Sign Language do it easily and also it is helpful for the Education Institutions and platforms hosting this feature as they increase their brand image .

Hence to make this better we will be working on a all around turnover with totally new efficient algorithms working on better systems. This will help for a faster and better recognition without any constraints about background and color etc. Also we hope to work on expanding dataset and increase knowledge of the model for more effective recognition and communication. Hence our Platform and the Model can be helped and integrated in Day to Day learning of sign language and also have many further applications.

REFERENCES

1. M Aarthi and P Vijayalakshmi, "Sign Language to Speech Conversion", *Fifth International Conference On Recent Trends In Information Technology*, 2016.
2. V. N. T. Truong, C. K. Yang and Q. V. Tran, "A translator for American sign language to text and speech", *2016 IEEE 5th Global Conference on Consumer Electronics*, pp. 1-2, 2016.
3. v. Anuja and Bindu V Nair, "A Review on Indian Sign Language Recognition", *International Journal of computer Applications*, pp. 33-38, July 2013.