JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

SIGN LANGUAGE INTERPRETER USING MACHINE LEARNING

[1]Bushra Mujawar, [2]Shruti Wani, [3]Sakshi Kariya [4]Anush Sharma [5]Kishor Sawarkar

[1]Student, [2]Student, [3]Student [4]Student [5]Associate Professor
[1]Department of Electronics and Telecommunication Engineering,
[1]MCT's Rajiv Gandhi Institute of Technology, Mumbai, India

Abstract : Although sign language is an essential tool for people who have hearing loss, there is still a communication gap between hearing and deaf people. This study suggests a machine learning and image processing-based sign language interpretation system. For people who are deaf or hard of hearing and those who are not skilled in using sign language, the system intends to deliver real-time translation of signs into text and speech. The suggested method classifies signs based on hand gestures taken by a camera using a convolutional neural network (CNN). The results show that the proposed system can translate sign language motions into meaningful words or sentences with a high degree of accuracy and usability. The system achieves an overall accuracy of 89% in classifying the signs of American Sign Language (ASL)

***Keywords* - convolutional neural network (CNN), image processing, machine learning, sign language.**

I. INTRODUCTION

Effective communication is one of the essential components of social integration. Deaf and mute people frequently use sign language to communicate, but it can be difficult for people who are not familiar with it to understand. A child with hearing loss has enormous obstacles in the development of their language and speech skills. Hearing loss limits the child's tutoring, additional schooling, and future professional opportunities. The purpose of this project is to use Sign Language as a first step in bridging the communication gap between deaf and mute people and the general public. The future expansion of this project to incorporate words and common phrases could significantly improve communication between deaf and mute people and the general public. It could also aid in the creation of autonomous systems that can better comprehend and support them. Due to the complexity and variety of sign language, the accuracy of these applications still faces significant challenges.

In order to recognise sign language alphabets by comparable motions, this paper aims to design a system that recognizes motions recorded by a camera instead of high-end devices like gloves or the Kinect, and then we utilise computer vision and machine learning algorithms to extract specific aspects and classify them. Convolution neural networks (CNNs) are demonstrating that they are the best tool for processing these recognition systems. The effectiveness of deep learning techniques in real-world recognition may be their lone drawback. Processing gestures requires powerful computing. The goal of this work is to investigate how machine learning might be used to create a real-time sign language interpreter system that can reliably translate signs. The report also analyses the similar topics' potential drawbacks and potential future developments.

II. LITERATURE SURVEY

The suggested framework's literature review reveals that multiple initiatives have been undertaken to tackle sign identification in videos and photos utilizing a variety of techniques and algorithms.

A real-time interpretation of hand motions in sign language using deep convolutional neural networks is presented by Devashish Gupta et al. [6] with the goal of creating a hardware prototype that is both efficient and affordable for use with deaf and dumb people. One way for identifying the hand position and pattern and translating it to the appropriate message or purpose is hand gesture recognition. In order to bridge the gap between hearing-impaired and hearing-normal people, Yuvraj Groover et al. [15] utilized a variety of translation approaches that have been created.

A system focused on multiple methods and technologies was offered by Smit Patel et al [10] to assist close this gap using deep learning. Here, they suggest a system that uses a web camera to recognize sign language and forecast the appropriate sign. The system makes use of max pooling, ReLU activation function, convolution neural networks, and deep learning algorithms. Their goal is to develop software that is user-friendly, more economical, and effective without sacrificing the required outcomes.

A trustworthy communication interpretation programme was developed by Shubhendu Apoorv et al[1] to translate Indian sign language into understandable output. Machine learning and image processing are used to complete the assignment. This project proposal can be used for everyday communication as well as for teaching different gestures to an autonomous gesture-based system.

An adaptation was created by Rachana Patil et al. [9] to examine the challenges associated with character classification in Indian Sign Language (ISL). In this project, a Convolutional Neural Network-based system for Sign Language recognition has been created.

A method for deaf and dumb persons to recognize signs was proposed by Geethu Get el [7] and built on an ARM CORTEX A8 processor board utilizing the convex hull algorithm and the template matching algorithm.

III. Proposed Methodology

3.1 Creation of Sign Language Recognition Dataset:

There are 400 photos in our dataset for each of the 26 possible hand signs. All of the alphabets' fingerspellings are represented by 26 hand signals, with the exception of J and Z, which need temporal information to be classified. The dataset is acquired while individuals move their hands about on the image plane and along the z-axis in order to collect data from diverse angles. 10,000 photos total are included which are utilised for testing and training the dataset.

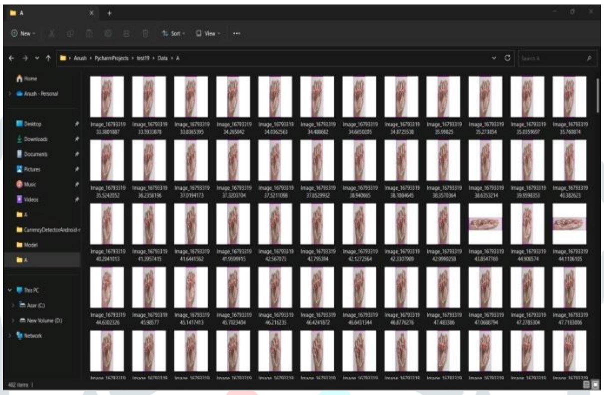

Fig 1. Images of data collected of alphabet A

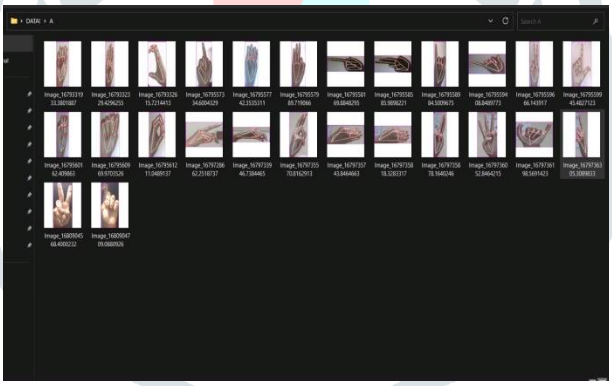

Fig 2. Images of 26 Alphabets [Letters (A-Z)]

3.2 Preprocessing:

a. Bounding Calculation: This is the first step in the preprocessing of our model by setting the bounding rectangle by debug_image and hand_landMarks.

b. The landmarks of the hand signs in the rectangle formed by the bounding calculation are calculated using mediapipe.

c. After completing the bounding calculations and calculating the landmarks of the hand signs they are saved in a file in the .csv format.

d. After the csv data file is saved the conversion of relative coordinates to the normalized coordinates is done and then is saved again.

e. Sign Classification: After the normalized coordinates are saved Hand Sign Classification function is performed for two hand. Further the finger gesture classification is performed by creating a finger ID and then calculation of gesture IDs in the latest direction.

f. Text Output: The system aims to understand the calculated gestures by identifying the hand signs, and then translating them into text.

g. Text to Speech: The text output released is then converted to speech using Playsound and PYTTSX3.

3.2.1 2D CNN:

The static images in the dataset are categorised using a Convolutional Neural Network (CNN) model. The local receptive field kernel window is used to extract a layer from a feature map and create a 2D CNN. It decreases the number of free variables and improves the network's capacity for generalisation. The CNN operates by taking the processes listed below:

a.The hidden layers of the model process the input layer, which is defined by turning each image to a series of integers, with each pixel of a 28x28 image being represented by a grayscale value ranging from 0 (black) to 1 (white).[9]

b. The first hidden layer is made up of nodes that take in the weighted sum of the 784 input values and pass it through the Rectified Linear Unit (ReLU) activation function. When the input is negative, the ReLU outputs 0; otherwise, it leaves the input unmodified. The ReLU's outputs serve as the network's next hidden layer's inputs.[9]

c. ReLU Activation: After passing through a non-linear activation function, often ReLU (Rectified Linear Unit), the output from the convolutional layer is then used. This adds non-linearity to the model and aids in the capturing of more intricate characteristics.

d. Pooling: After the ReLU activation layer, the image is down-sampled by a pooling layer, which takes the maximum or average value of a small region of pixels. This makes the model more effective by reducing the spatial size of the representation.

e. Fully Connected Layers: After flattening the pooled output into a 1D vector, fully connected layers are applied to the vector. One output value is generated for each class in the dataset by these layers after a linear transformation of the input data.

f. Softmax Activation: By using a softmax activation function on the output of the fully linked layer, the model's final output is produced. As a result, a probability distribution is generated over the dataset's classes, and the class with the highest probability is chosen as the model's anticipated output.

3.3 Important Libraries Used:

1. Numpy: The Numpy library in python is an open sourced library that has the main function of performing great variety of mathematical operations on arrays. This same function is performed here in our project by this library of converting the data in the form of array pixels from the image form and reshape it to the required input size of our model ie.1, 224, 224, 3 keeping our data type as float. After converting the image data set to the numerical form of the data set the numerical data set is then normalized by numpy.

2. Mediapipe is an open-source library developed by Google that provides a collection of machine learning models for various computer vision tasks, including hand tracking and pose estimation. Here, mediapipe uses the hands function from the mediapipe libariers to mark landmarks.

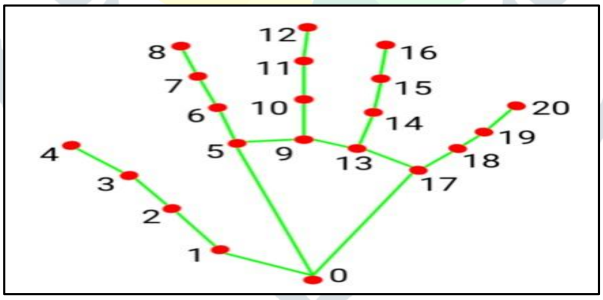

Fig 3. Hand Landmarks calculated using Mediapipe

0. WRIST	11. MIDDLE_FINGER_DIP
1. THUMB_CMC	12. MIDDLE_FINGER_TIP
2. THUMB_MCP	13. RING_FINGER_MCP
3. THUMB_IP	14. RING_FINGER_PIP
4. THUMB_TIP	15. RING_FINGER_DIP
5. INDEX_FINGER_MCP	16. RING_FINGER_TIP
6. INDEX_FINGER_PIP	17. PINKY_MCP
7. INDEX_FINGER_DIP	18. PINKY_PIP
8. INDEX_FINGER_TIP	19. PINKY_DIP
9. MIDDLE_FINGER_MCP	20. PINKY_TIP
10. MIDDLE_FINGER_PIP	

Fig 4. Landmarks and their corresponding names

3. Pyttsx3: This is a Python module that provides a simple and convenient way to convert text to speech. It uses the Text-to-Speech (TTS) technology to generate synthesized speech from the text input. Pyttsx3 is imported and initialized using pyttsx3.init() to create a TTS engine. Then, the desired text is passed to engine.say() method to convert it to speech. Finally, engine.runAndWait() is called to wait for the speech to finish before the program exits.

4. Playsound: The "playsound" module is a Python library that provides a simple way to play audio files. It can be used in conjunction with a text-to-speech (TTS) system to convert text into speech and play the resulting audio output. We first import the necessary modules, "playsound" and "gTTS" (Google Text-to-Speech). We then create a gTTS object with the text that we want to convert into speech, specifying the language as 'en' for English. Next, we save the TTS output as an audio file named 'output.mp3'. Finally, we use the "playsound" module to play the 'output.mp3' file, which contains the speech generated from the input text.

5. Matplotlib & Seaborn: These libraries were used to make several interactive data visualization reports.

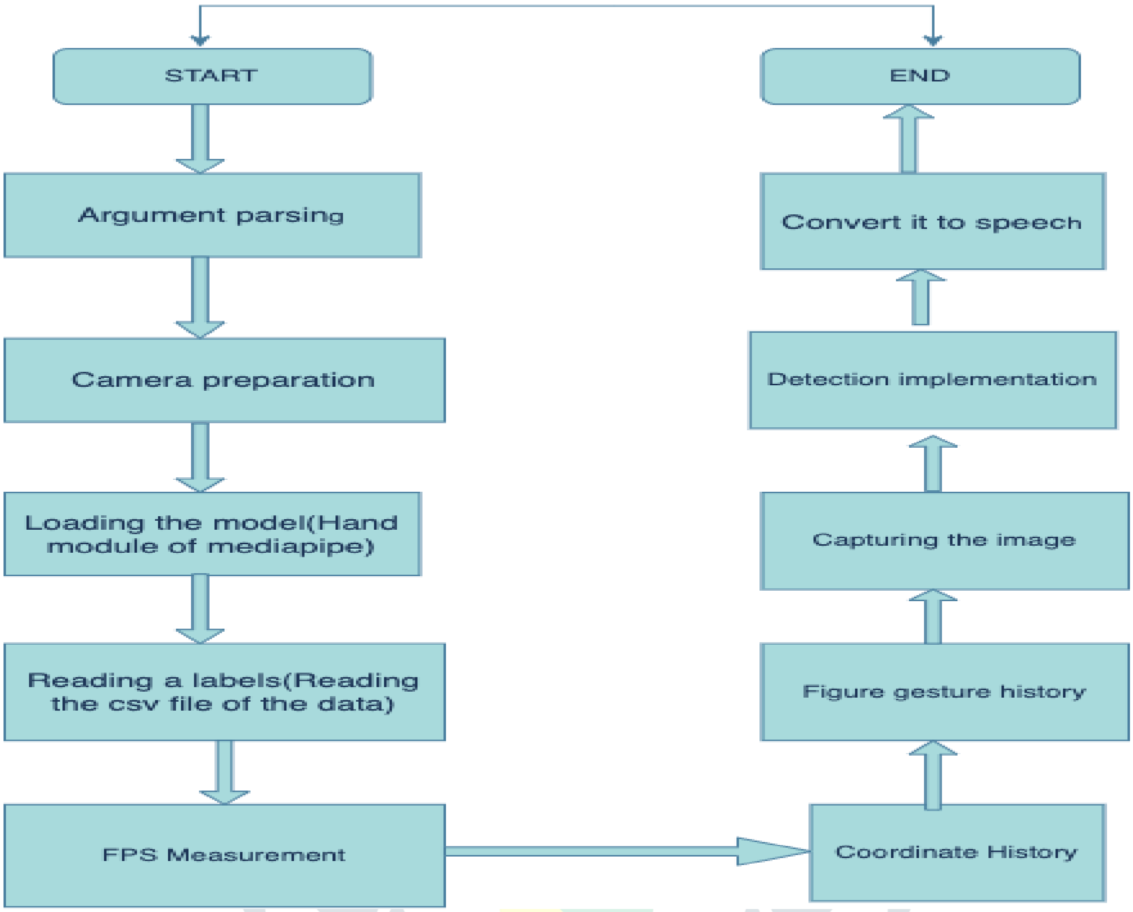

Fig 5. Architectural diagram of the proposed system

IV. Results

640*480 capture photos are analysed at a frame rate of 30 frames per second using a 12th Gen Intel® Core i5-12500H CPU Windows 11 8 GB/512 GB SSD. Figures 6 and 7 below exhibit hand gesture identification using the suggested algorithm and movements that have been identified for the recognition of the letters A and U.

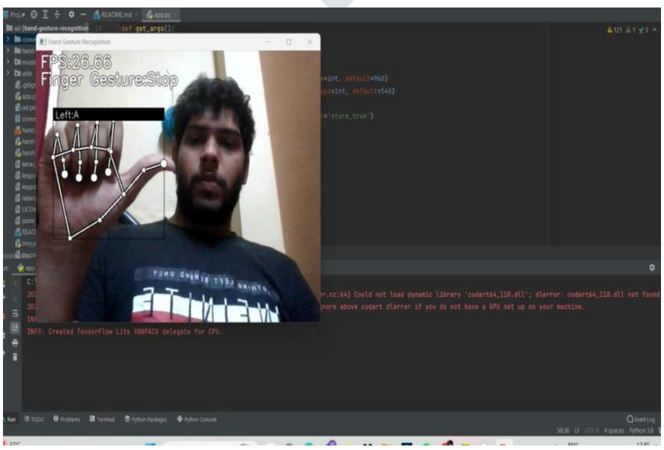

Fig 6. Real Time Sign Language Detection (Alphabet A)

Fig 7. Real Time Sign Language Detection (Alphabet U)

The confusion matrix that was used to determine the system's overall accuracy is shown in the image below. Identification of the anticipated and actual classes, as well as the number of true positives, true negatives, false positives, and false negatives, were used in the calculation. The outcome of the same calculation was 0.89, or 89% when expressed as a percentage.

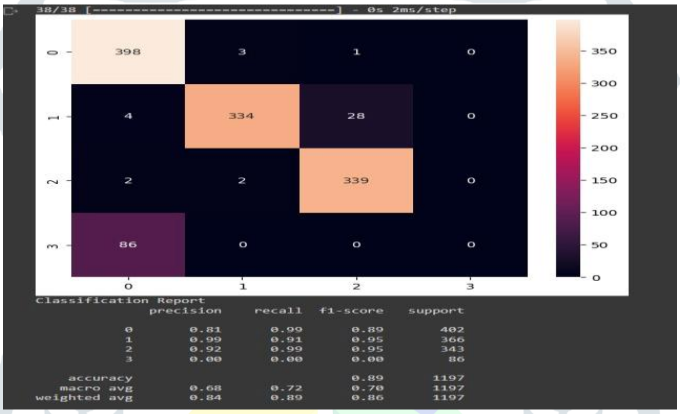

Fig 8. Shows the confusion matrix used for accuracy

V. CONCLUSION AND FUTURE SCOPE

A Sign Language Recognition (SLR) system is a method for correctly recognising and translating a collection of predetermined signs into text or speech with the proper context. The development of effective human machine interactions clearly demonstrates the significance of gesture recognition. In this research, a Convolutional Neural Network (CNN) model was successfully built and its validation accuracy was 89%.

It is necessary to make more advancements to the image processing area to enable bidirectional interaction, allowing the system to translate between spoken language and sign language as well as the other way around. Developments in this area will encourage research on both, which have not yet undergone considerable study.

Indian Sign Language [ISL] recognition should be added to the already existing work, along with a multilingual recognition option. The system's capabilities would be improved, and its potential uses in India would also be expanded.

Adaptation to additional technologies: creating immersive learning experiences by combining sign language recognition with other technologies like virtual reality and augmented reality.

Multimodal recognition: Combining sign language recognition with other modalities such as facial expressions and body language to improve the accuracy and understanding of signed messages.

VI. ACKNOWLEDGMENT

We would like to extend our sincere gratitude towards Dr. Kishor Sawarkar for his relentless support and valuable guidance throughout the project. Additionally, we would like to thank everyone who reviewed the work for their patience and insightful criticism, which allowed us to significantly improve it.

REFERENCES

[1] Apoorv, S., Kumar Bhowmick, S. and Sakthi Prabha, R. (2020) "Indian sign language interpreter using image processing and machine learning," IOP Conference Series: Materials Science and Engineering, 872, p. 012026. Available at: https://doi.org/10.1088/1757-899x/872/1/012026.

[2] A. S. Ghotkar, R. Khatal, S. Khupase, S. Asati and M. Hadap, "Hand gesture recognition for Indian Sign Language," 2012 International Conference on Computer Communication and Informatics, Coimbatore, India, 2012, pp. 1-4, doi: 10.1109/ICCCI.2012.6158807.

[3] A. S. Nikam and A. G. Ambekar, "Sign language recognition using image based hand gesture recognition techniques," 2016 Online International Conference on Green Engineering and Technologies (IC-GET), Coimbatore, India, 2016, pp. 1-5, doi: 10.1109/GET.2016.7916786.

[4] B. Kang, S. Tripathi and T. Q. Nguyen, "Real-time sign language fingerspelling recognition using convolutional neural networks from depth map," 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), Kuala Lumpur, Malaysia, 2015, pp. 136-140, doi: 10.1109/ACPR.2015.7486481.

[5] C. M. Jin, Z. Omar and M. H. Jaward, "A mobile application of American sign language translation via image processing algorithms," 2016 IEEE Region 10 Symposium (TENSYMP), Bali, Indonesia, 2016, pp. 104-109, doi: 10.1109/TENCONSpring.2016.7519386.

[6] D. Gupta, J. P. Mohanty, A. K. Swain and K. Mahapatra, "AutoGstr: Relatively Accurate Sign Language Interpreter," 2019 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS), 2019, pp. 322-323, doi: 10.1109/iSES47678.2019.00080.

[7] G. G. Nath and C. S. Arun, "Real time sign language interpreter," 2017 IEEE International Conference on Electrical, Instrumentation and Communication Engineering (ICEICE), 2017, pp. 1-5, doi: 10.1109/ICEICE.2017.8191869.

[8] N. Adaloglou, T. Chatzis, I. Papastratis, A. Stergioulas, G. T. Papadopoulos, V. Zacharopoulou, G. J. Xydopoulos, K. Atzakas, D. Papazachariou, and P. Daras, “A comprehensive study on Deep Learning-based methods for sign language recognition,” arXiv.org, 19-Mar-2021. [Online]. Available: https://doi.org/10.48550/arXiv.2007.12530

[9] R. Patil, V. Patil, A. Bahuguna, and G. Datkhile, “Indian Sign Language Recognition using Convolutional Neural Network,” ITM Web of Conferences, vol. 40, p. 03004, 2021, doi: 10.1051/itmconf/20214003004
[10] S. Patel, T. Pardhi, and Z. Shah, “ProQuest | Databases, EBooks and Technology for ResearchS.” [Online]. Available: https://about.proquest.com/.

[11] Srivastava, S. et al. (2022) “Sign language recognition system using tensorflow object detection api,” Communications in Computer and Information Science, pp. 634–646. Available at: https://doi.org/10.1007/978-3-030-96040-7_48.

[12] Suharjito a, R. Anderson, G. P. Kusuma, and AbstractSign Language Recognition is a breakthrough for helping deaf-mute people and has been researched for many years. Unfortunately, “Sign language recognition application systems for deaf-mute people: A review based on input-process-output,” Procedia Computer Science, 13-Oct-2017. [Online].

[13] S. Yang and Q. Zhu, "Video-based Chinese sign language recognition using convolutional neural network," 2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN), Guangzhou, China, 2017, pp. 929-934, doi: 10.1109/ICCSN.2017.8230247.

[14]Verma, P. and Badli, K. (2022) “Real-time sign language detection using tensorflow, opencv and python,” International Journal for Research in Applied Science and Engineering Technology, 10(5), pp. 4483–4488. Available at: https://doi.org/10.22214/ijraset.2022.43439.

[15] Y. Grover, R. Aggarwal, D. Sharma and P. K. Gupta, "Sign Language Translation Systems for Hearing/Speech Impaired People: A Review," 2021 International Conference on Innovative Practices in Technology and Management (ICIPTM), 2021, pp. 10-14, doi: 10.1109/ICIPTM52218.2021.9388330.