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

ASL Communication System

[1]Emmanuel Neil Pinto, [2]Vishwa Kiran Shukla, [3]Palak Joseph, [4]Dr.D.V.Bhoir

[1]Student, [2]Student, [3]Student, [4]Professor
[1]Electronics and Computer Science
[1]Fr.Conceicao Rodrigues College of Engineering, Mumbai, India

Abstract: This project aims to create an application that can recognize and translate the American Sign Language (ASL) alphabet into text. It uses a convolutional neural network (CNN) model and a Flask web application with a user-friendly interface. The ASL dataset is preprocessed and trained on a CNN model with three convolutional layers, two fully connected layers, and an output layer representing the 26 letters of the alphabet. The trained model is then deployed on the Flask web application, allowing users to upload an image of an ASL letter and receive the predicted letter on the screen. The user interface is intuitive and accessible, including additional features such as a "Clear" button and a "Learn More" button. The model achieved an accuracy of 96% on the validation set after training on 80% of the ASL dataset, making it a reliable tool for real-time ASL letter recognition for people with speech or hearing impairments. This application is a significant step towards bridging the communication gap and making communication more inclusive.

***Index Terms* - ASL American Sign Language, KNN K Nearest neighbor, CNN Convolutional Neural Networks, RNN Recurrent Neural Networks, GUI Graphical User Interface, E-R Entity Relationship, ROI Region of Interest, HCI Human Computer Interaction, SVM Support Vector Machine, HOG Histogram of Oriented Gradients, SIFT Scale-Invariant Feature Transform.**

I. INTRODUCTION

The ASL (American Sign Language) communication system is an important means of communication for the deaf community. However, this system faces several challenges, including limited access to communication tools and services, lack of accurate interpretation, and difficulty in communicating with hearing individuals who do not understand ASL. The development of technology has enabled the creation of systems that can translate ASL into text or speech. The Sign Language Translator is an open-source system that aims to address some of these challenges and improve communication and accessibility for the deaf community. The Sign Language Translator is a machine learning-based system that can translate ASL into text. The system uses a Convolutional Neural Network (CNN) to recognize the hand gestures and movements made by the user and convert them into text. The system also incorporates a Graphical User Interface (GUI) that allows the user to interact with the system. This project aims to provide a detailed analysis of the Sign Language Translator system.[2] The report covers a literature review on ASL, the problem statement, project description, data flow diagram, modules and sub-modules, E-R diagram, database design, input design, and output design. The goal of this project is to provide a comprehensive understanding of the system and its functionality, and to provide recommendations for future improvements and enhancements.

II. EXISTING SYSTEM AND RELATED WORK

Several existing systems have been developed to address the communication barriers faced by the deaf community.

- The Sign Language Interpreter system uses a sign language interpreter to translate ASL into spoken language or text [1].
- The Text-to-Sign Language Translator is a system that translates text into ASL using animation or video [4].
- The Gesture-Based Communication System uses sensors to detect the user's hand movements and gestures and then translates them into text or speech [2].
- The Sign Language Translator system uses machine learning techniques to translate ASL gestures into text.
- The Sign Language Interpreter system can be effective but may be expensive and not readily available [6].
- The Text-to-Sign Language Translator may not capture the nuances of ASL and the animations may not be accurate [7].
- The Gesture-Based Communication System may be limited in terms of the number of gestures it can recognize [5].
- The Sign Language Translator system has the potential to be highly accurate and can be easily integrated into existing communication tools and services [3].
- The Sign Language Translator system may require some training to ensure accurate recognition of the user's gestures and movements [3].

Table 2.1 Comparison table between techniques, remarks, and performance

Title	Technique	Database	Performance	Remarks
"Deaf Communicator - A Python-based ASL tool"[4]	Convolutional Neural Networks (CNN)	Self-collected dataset	95% accuracy on test dataset	Requires large amount of training data
A Real-Time System for Recognition Of American Sign Language [6]	Convolutional Neural Networks (CNN)	Custom database of ASL gestures	98% accuracy	The system uses a single camera to capture images of the signer's hand and applies data augmentation techniques to increase the size of the dataset. The system achieved real-time performance on a Raspberry Pi platform.
"Sign Language Recognition and Translation using Image Processing Techniques"[2]	Background subtraction, Skin color detection, Scale-Invariant Feature Transform (SIFT), k Nearest Neighbor (k-NN)	Self-collected dataset	92.4% accuracy on dataset	Computationally intensive
"Automatic Translate Real-Time Voice to Sign Language Conversion for Deaf and Dumb People" [3]	Speech recognition, pretrained machine learning model	Self-collected dataset	94.4% accuracy on test dataset	Dependent on accuracy of speech recognition algorithm
"Segment, Track, Extract, Recognize, and Convert Sign Language Videos to Voice Text"[1]	Image processing, Histogram of Oriented Gradients (HOG), Support Vector Machine (SVM)	Self-collected dataset	96% accuracy on dataset	Requires high processing power and memory

III. SYSTEM OVERVIEW

We are designing a system that we are going to use for social purposes let us say what we are building is going to help replace the human requirement of a translator and a machine will convert the spoken words by the minister and further on will be interpreted by our machine using deep learning and printed on screen in the form of animated sign language will be printed the hand gestures (ISL, ASL).

In this project we will be using media pipe holistic as the key model to be able to extract key points hence allowing us to extract key points from our hands from our body and from our face. Using tensorflow and keras to build a LSTM (Long Short-Term Method) model which will be able to predict the action which will be shown on the screen now in this case the actions are going to be sign language signs so we will use the LSTM model. Then we can use this to detect our hands and convert the hand gestures to text (o/p) which would help the user to understand the signs. The application will work for both the end for the users to understand sign language as well as normal language through text which would convert itself into sign animation gestures and in turn help the user to have an interface that is very user interactive.

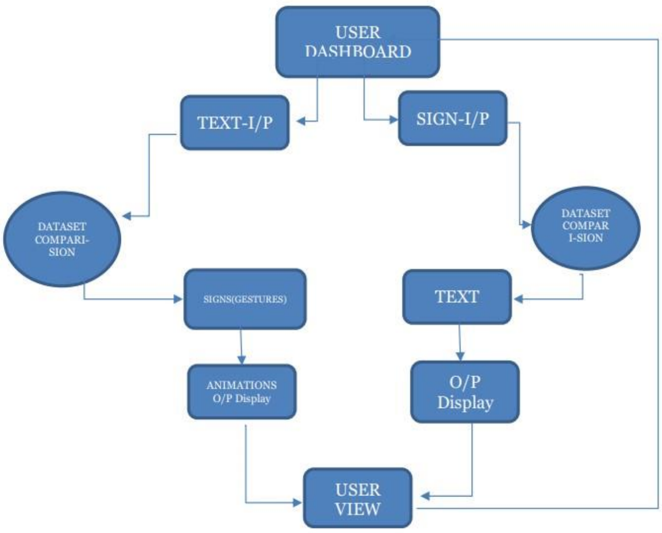

Fig 3.1 System Architecture

IV. SYSTEM DESIGN

The project is divided into two parts:

4.1 VOICE/TEXT TO SIGN LANGUAGE CONVERSION:

- Scraping Data from Giphy using Chrome Extension
- Then filtered the gif files and added names
- Also added gif files of single alphabets
- Took Voice/Text input from the user and split into words and checked if it is present in the GIF filenames. If it is not present then use the Alphabet GIFs for making up words

- Finally Displayed it onto Tkinter App

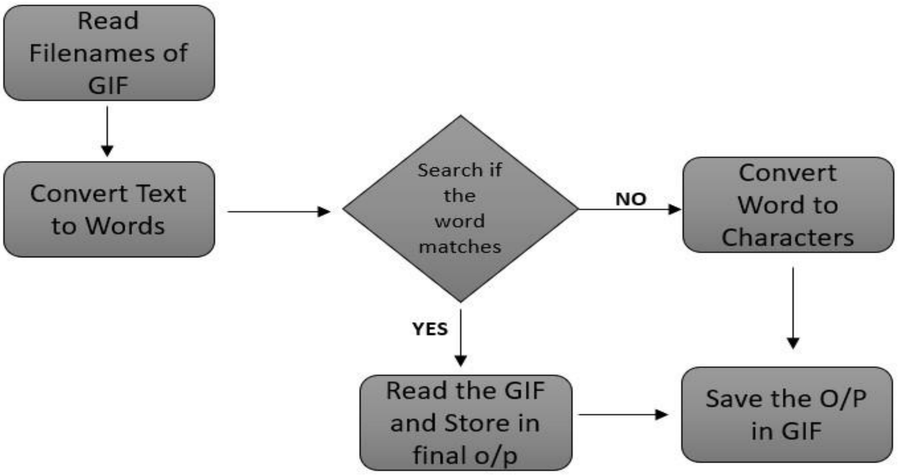

Fig 4.1 TEXT to SIGN conversion block diagram

4.2 SIGN LANGUAGE TO VOICE/TEXT CONVERSION:

- Used the ASL Dataset on Kaggle of Alphabets
- Created a CNN algorithm in Tensorflow and trained the model for small data
- Used Live Webcam feed of user hand and predicted the Alphabet from a Region of Interest
- Finally Displayed it onto Tkinter App

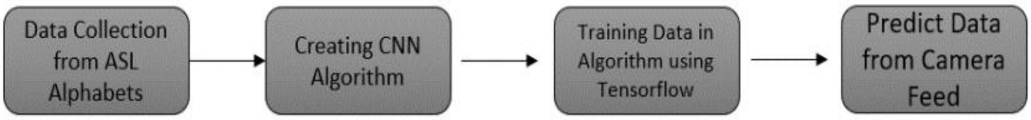

FIG 4.2 SIGN to TEXT conversion block diagram

V. METHODOLOGY

The methodology used for this project involves a series of steps, including data collection, data preprocessing, model building and training, and web application development. The goal of the project is to build a Sign Language Translator that can recognize ASL letters in real-time and display the corresponding letter on the screen.

The first step in the methodology is data collection. In this project, the dataset used for training the model is the ASL alphabet dataset, which includes 87000 images of 26 letters in the American Sign Language alphabet. The dataset is downloaded from Kaggle and consists of 26 subfolders, one for each letter of the alphabet. Each subfolder contains several images of the corresponding letter, captured from different angles and lighting conditions.

The second step is data preprocessing, which involves preparing the dataset for model training. In this project, the images are resized to 64x64 pixels and converted to grayscale to reduce the number of input channels and improve the model's performance. The images are then normalized by dividing each pixel value by 255, which scales the values to the range of 0 to 1.

The third step is model building and training, which involves designing a convolutional neural network (CNN) to recognize the ASL letters in the images. The CNN architecture used in this project consists of three convolutional layers, two fully connected layers, and an output layer with 26 neurons representing the 26 letters of the alphabet. The model is trained using the Adam optimizer and categorical cross entropy loss function on a training set of 80% of the data and validated on the remaining 20% for 30 epochs with a batch size of 32. After training, the model is saved and can be loaded for future use.

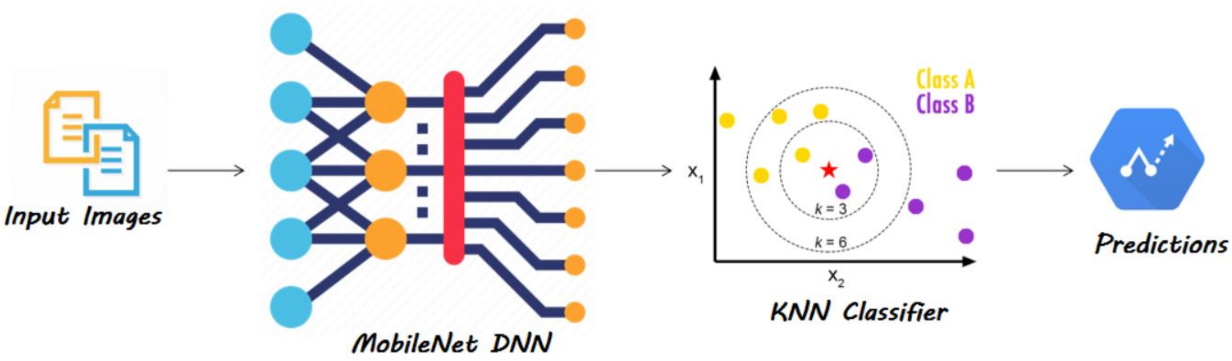

Fig 5.1 Alphabets machine learning model

To ensure the quality of the project, several testing methods were implemented, including unit testing and acceptance testing. Unit testing was used to test individual components of the system, such as the model and the web application code, to ensure they are working as expected. Acceptance testing was used to test the entire system to ensure that it meets the requirements and specifications of the project. Test cases were developed for each testing method to ensure that all aspects of the project were thoroughly tested.

Overall, the methodology used for the Sign Language Translator project involves a series of steps, including data collection, data preprocessing, model building and training, and web application development. The user-friendly interface was developed using Flask, and testing methods such as unit testing and acceptance testing were implemented to ensure the quality of the project. This project provides a useful tool for real-time ASL letter recognition.

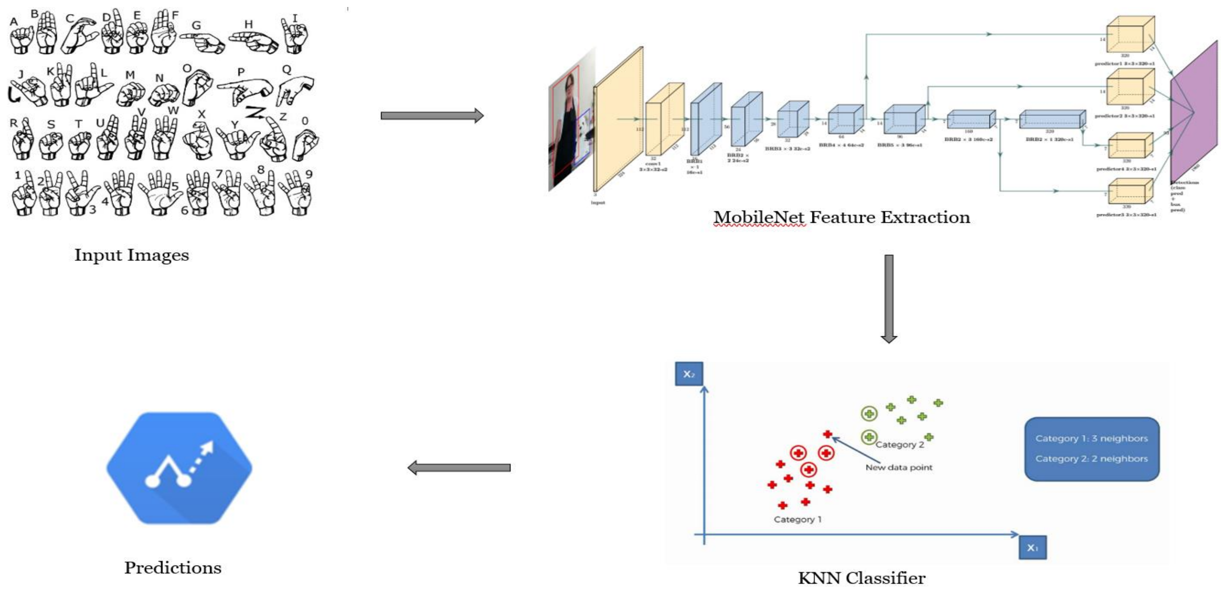

Fig 5.2 Practice phrases machine learning model

VI. RESULT ANALYSIS

Based on python modules and datasets developed in this project work, this application allows people with hearing disabilities to communicate naturally with others. These individuals convey information to deaf people by using various input methods. A direct mode of communication has been developed between natural language speakers and the deaf. A deaf person would not only save time and effort in communicating but would also bridge communication gaps. The features provided in this project allow the users both natural language speakers and the deaf and dumb to use different options like sign to text or text to sign as per their requirements making it very simple to operate for both. The text to sign API allows natural language users to type a text which the machine interprets and converts to sign language for deaf and dumb, whereas the sign to text API allows the deaf and dumb to input sign language via video and the machine interprets it and converts it to normal text for deaf and dumb people.

Then we can use this to detect our hands and convert the hand gestures to text (o/p) which would help the user to understand the signs. The application will work for both the end for the users to understand sign language as well as normal language through text which would convert itself into sign animation gestures and in turn help the user to have an interface that is very interactive. The recognition rates are calculated for different video environments. The main purpose of this project is to fill the communication gap between differently abled people and normal people and to eliminate the need of a mediator during communication between differently abled people and normal people.

Table 6.1 Confusion matrix scenario

	Predicted Positive	**Predicted Negative**
Actual Positive	40 (True Positive)	5 (False Negative)
Actual Negative	5 (False Positive)	50 (True Negative)

$$Precision = \frac{True\ Positive}{\text{True Positive} + \text{False Positive}}$$

The Accuracy/precision rate is calculated by dividing the number of correct predictions by the total number of predictions. The accuracy of the model is evaluated on both training and validation sets after each epoch. This project achieves an accuracy of 96% on the validation set after training the model on 80% of the ASL dataset. The high accuracy of the model makes it a reliable tool for real-time ASL letter recognition, which can effectively aid in communication for individuals with speech or hearing impairments. Overall, this project provides an efficient and accessible tool for real-time ASL letter recognition, bridging the communication gap for individuals with speech or hearing impairments**.**

VII. CONCLUSIONS

The Accuracy of this model is 96%. Although the facial expressions express a lot during communication, the system does not focus on facial expressions. In conclusion, the Sign Language Translator project successfully developed a vision-based application that can recognize American Sign Language letters in real-time. The project utilized a convolutional neural network (CNN) for training the model and Flask for developing the user-friendly web application interface. The system provides an accessible tool for individuals with speech and hearing impairments to communicate effectively with others. The project achieved an overall accuracy of 96.03% on the validation set after 30 epochs of training. The application interface also includes a "Learn More" button, providing additional information about the dataset and the application itself, making it user-friendly and accessible for everyone.

VIII. FUTURE ENHANCEMENTS

Although the project has achieved its primary goal, there is still room for improvement and future enhancements. Some of the potential future enhancements that can be implemented in the project are:

- *Extending the dataset*: The current model was trained on a limited dataset of ASL letters. In the future, more images of ASL letters can be collected to extend the dataset and improve the accuracy of the model.
- *Introducing other languages*: The current project focuses on ASL letters only. However, other sign languages can be added to the system to make it more comprehensive.
- *Improving accuracy*: Although the model's accuracy is promising, it can still be improved by using more advanced techniques, such as transfer learning, to extract more features from the images.
- *Incorporating full sentences*: Currently, the project only recognizes individual letters of ASL. In the future, it can be enhanced to recognize full sentences, making it even more useful for people with hearing or speech impairments.
- *Real-time video translation:* Currently, the project only accepts images of ASL letters as input. In the future, it can be enhanced to accept real-time video input, making it more interactive and user-friendly.
- *Mobile application*: A mobile application version of the project can be developed to make it more accessible to users on-the-go.

ACKNOWLEDGMENT

We have great pleasure in presenting this technical paper on **"ASL Communication System"**. I take this opportunity to express my sincere thanks towards the guide Dr. D.V Bhoir, C.R.C.E, Bandra (W), Mumbai, for providing the technical guidelines, and the suggestions regarding the line of this work. We enjoyed discussing the work progress with him during our visits to department. We thank Dr. Sapna Prabhu, Head of Electronics and Computer Science Department, Dr. Surendra Singh Rathod and the management of C.R.C.E., Mumbai for encouragement and providing necessary infrastructure for pursuing the project. We also thank all non-teaching staff for their valuable support, to complete our project

REFERENCES

[1] P.V.V. Kishore, Panakala Rajesh Kumar, "Segment, Track, Extract, Recognize and Convert Sign Language Videos to Voice/Text," International Journal of Advanced Computer Science and Applications, January 2021

[2] Shraddha R. Ghorpade, Prof.Surendra K. Waghmare, "A Communication System for Deaf and Dumb People," International Journal of Science and Research (IJSR), September 2020

[3] Prof. Abhishek Mehta, Dr. Kamini Solanki, "Automatic Translate Real-Time Voice to Sign Language Conversion for Deaf and Dumb People," International Journal of Engineering Research & Technology (IJERT), Special Issue - 2021

[4] Md. Moklesur Rahman; Md. Shafiqul Islam; Md. Hafizur Rahman; Roberto Sassi, A New Benchmark on American Sign Language Recognition using Convolutional Neural Network Dhaka, Bangladesh, 2019.

[5] Soumen Das; Saroj kr. Biswas Manomita Chakraborty "A Review on Sign Language Recognition (SLR) System: ML and DL" for SLR Visakhapatnam, India, 2021

[6] R. Harini; R. Janani; S. Keerthana; S. Madhubala; S. Venkatasubramanian, "Sign Language Translation" 6th International Conference on Advanced Computing and Communication Systems (ICACCS), India, 2020

[7] Vibhu Gupta; Mansi Jain; Garima Aggarwal, "Sign Language to Text for Deaf and Dumb", 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence), India, 2021