Advanced Real Time System

SIGN LANGUAGE TO SPEECH CONVERSION

Guided by:

Prof. Dr. Matthias Deegener

A Project Report

Submitted by:

Kavita Vaghasiya	**(1442706)**
Vinaykumar Kachhdiya	**(1384165)**
Jatinkumar Nakrani	**(1386383)**

Frankfurt University of Applied Sciences

March 2023

Contents

Acknowledgement

We would like to express our sincere gratitude and appreciation to everyone who has contributed to the successful completion of this project on sign language to speech conversion.

Firstly, We express our gratitude to our project supervisor Prof. Dr. Matthias Deegener for providing us with valuable guidance, support, and feedback throughout the project. Their insightful suggestions and constructive criticism helped us to refine our approach and achieve the desired outcomes.

Moreover, We are also thankful to The Frankfurt University of Applied Sciences for providing us with the necessary resources, equipment, and software tools to carry out this project. Their support and assistance were invaluable in enabling us to overcome various technical challenges and complete the project on time.

We would like to extend our appreciation to team members, our dedicated and hardworking team members, who worked with lot of effort to ensure the success of this project. Each member of the team contributed their unique skills, knowledge, and expertise to the project, and we are proud of the excellent collaboration that we achieved.

Finally, We want to convey our thanks to the participants who kindly volunteered their time to participate in the testing and evaluation of the sign language to speech conversion system. Their valuable feedback and insights helped us to improve the system's performance and usability.

Once again, we thank everyone who contributed to this project and made it a success.

Abstract

Since most people do not know sign language and it is difficult to find an interpreter, we developed a real-time approach to American Sign Language that uses neural networks and is based on fingerspelling. Sign language is one of the oldest and most natural forms of language for communication. With our approach, the hand is first put through a filter, and then, once the filter is applied, the hand is put through a classifier, which determines the type of hand movement. For an alphabet of 26 letters, our technique gives 90.7% accuracy.

Keyword: American Sign Language, Sign Language, Speech Conversion, Neural Networks, Hand Movement, Accuracy.

0.1 Introduction

Sign Language is visual language used by people who is deaf or hard-of-hearing to communicate. However, not everyone is fluent in sign language, which can lead to communication barriers and social isolation for individuals who rely on it as their primary means of communication. In recent years, there has been a growing interest in developing sign language recognition systems that can convert sign language gestures into spoken language in real time.

Communication is the process of exchanging messages and ideas and this process can be done in various ways such as speech, gestures and behaviour. Deaf and dumb people can share ideas and messages with each other through different gestures. These gestures can be movement of hand, body, facial expression and lips movement. This non-verbal communication is called sign language.

For Example, In case a individual who is incapable to talk can stand to perform the system and the system change the human signals as speech and play it so that individuals can really communicate to others.

Real-time sign language to speech conversion project is an innovative solution that uses advanced technologies including computer vision and machine learning to recognize and translate sign language gestures into spoken language. This project has the potential to significantly improve the communication and social integration of people who use sign language by enabling them to communicate more effectively with non-signing individuals.

The project involves the use of a camera that captures the signer's gestures and sends the data to a computer that runs complex algorithms to recognize the gestures and translate them into spoken language. The system can be trained on different sign languages, making it accessible to a diverse group of people from various part of the world.

The implementation of real-time sign language to speech conversion project has several benefits. It can enhance the inclusivity and accessibility of various services, such as education, healthcare, and public services, for deaf and hard of hearing individuals. The project can also promote better social interactions and improve employment opportunities for people who use sign language.

In this report, we will explore the design, development, and implementation of a real-time sign language to speech conversion system. We will figure the technical aspects of our project, including the algorithms used for gesture recognition, the hardware components used, and the software implementation.

The gestures we are using is shown below.

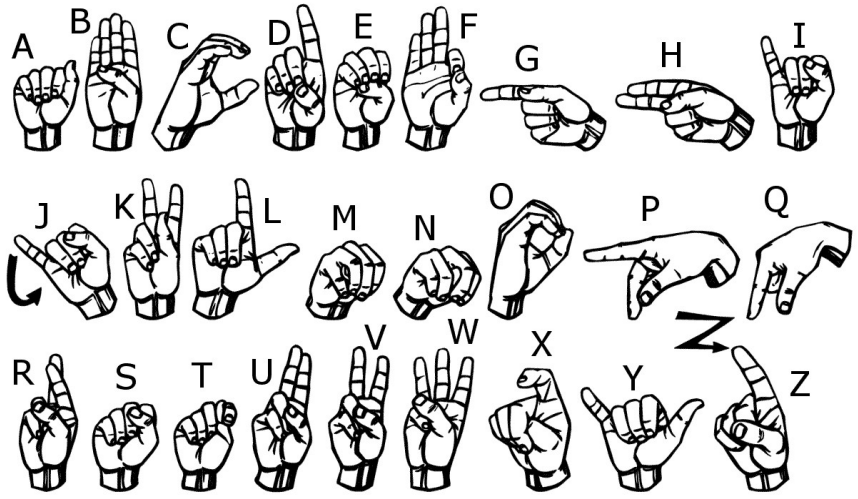


Figure 1: American Signs.

0.2 Motivation

Developing a real-time project for sign language to speech conversion is an exciting and challenging task that has the potential to make a significant effect on individuals who are deaf or have difficulty hearing.

The objective of this project is to establish a system that can accurately translate sign language into spoken language in real-time, allowing for more efficient and effective communication between people who use different modes of communication.

This project will require the integration of multiple technologies, including computer vision, machine learning, and natural language processing. By fortunately completing this project, we can provide an innovative solution to an existing problem, enabling people with hearing impairments to communicate more easily with the hearing world.

This project also has the potential to inspire further developments in the field of assistive technology, opening doors for future advancements and applications.

0.3 Problems & Objectives

0.3.1 Problems

Understanding the typical expressions of deaf and dumb individuals could be a challenging work in genuine life unless it is legitimately specified.

Accuracy is still a concern, as some gesture may be misinterpreted or translated incorrectly.

0.3.2 Objectives

The objective of this project is to recognize the typical expression through pictures so that the communication gap between a typical and hearing-impaired individual can be effortlessly bridged.

0.4 Project Steps & Technology

0.4.1 Project Steps

1. Analyzing and Research Datasets for sign language.
2. Preparing datasets and integrating with our projects
3. Start developing using datasets
4. Testing projects
5. Check performance and accuracy
6. Implementing large amount of data (future)

0.4.2 Technology & Modules

- High configuration Laptop / Mobile with Camera
- Python IDE
- Machine learning Models
- Tensorflow
- keras
- gTTS
- tkinter
- playsound
- PIL
- enchart
- hunspell

0.5 Methodology

The process of recognizing hand gestures and convert it into sign language and converting them to speech involves several steps:

1. Datasets Generation
2. Image Processing
3. Gesture Classification
4. Sentence Formation

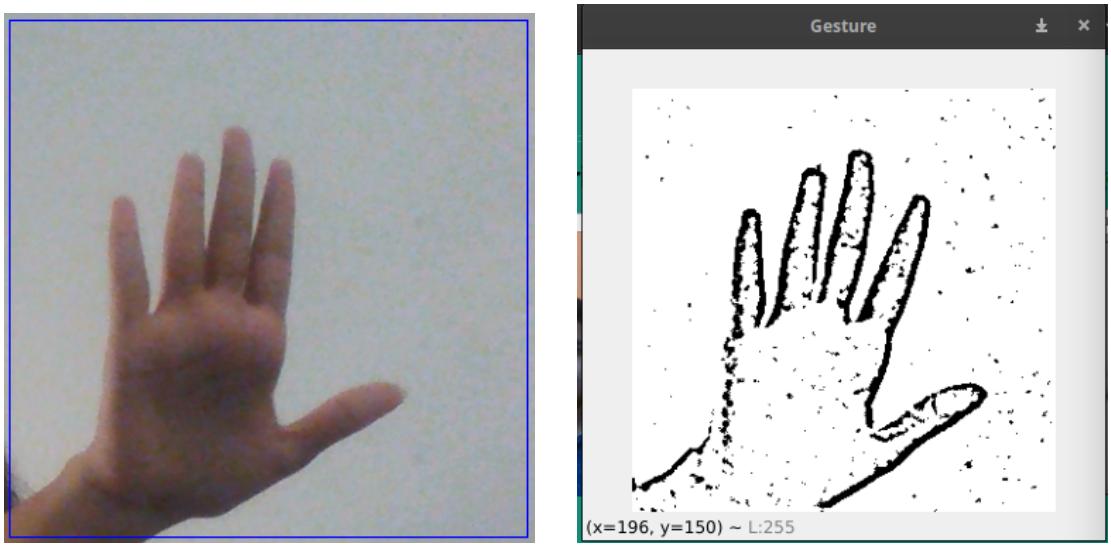
0.5.1 Dataset Generation & Image Processing

We decided to discover ready made datasets for our project but couldn't discover information in set of crude images.

All datasets were in RGB organize additionally we couldn't get sufficient precision for the sign. Hence, we chosen to form our own datasets which can coordinate our requirement.

We have used Open Computer Vision (OpenCV) to generate our dataset.

We have converted crude images(RGB) into grayscale and connected Gaussian obscure to expel superfluous clamor. All the pictures are re-sized to 128 x 128 pixels.



(a) Captured raw image (b) Image after Gaussian Blur effect

Figure 2: Image filtering

We have used Open Computer Vision (OpenCV) to generate our dataset.

To begin with, we have took around 600 images of each sign for training purpose and around 170 images for testing purpose

We have captured each sign displayed in blue frame by camera of our machine shown in above figure.

0.5.2 Gesture Classification

Our system uses two layers of algorithm to select the final symbol.

0.5.2.1 Algorithmn Layer 1

First, we applied Gaussian blur filter and threshold to raw images taken with Open CV

Then processed images passed to CNN model for prediction, if correct letter is detected for more than 50 frames. It is printed on window.

Blank symbol is considered as space between two words.

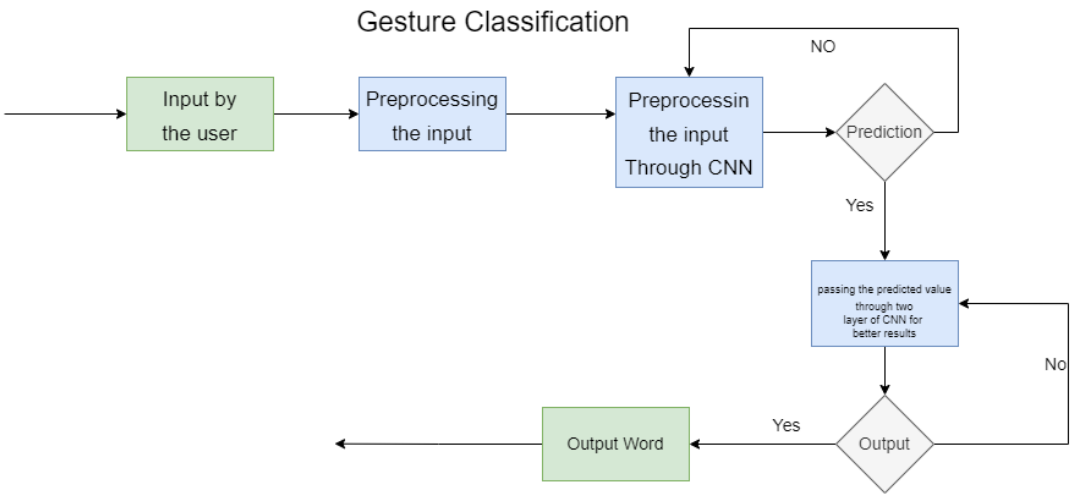


Figure 3: Gesture Classification

0.5.2.2 Algorithmn Layer 2

We detect that some of symbols not giving proper result. Moreover, they showing a similar results matching with other sign.

1. For E : B
2. For F : K and M
3. For M : N and I

To handle this problem, We applied three different classifier to classify these symbols.

1. E, B
2. F, K, M
3. M, N, I

0.5.3 Sentence Formation Implementation & Audio generation

1. We print the character to the string when detected character count exceed the specific value and no other character is close to it by the threshold (in our code we set the value to 50 and the difference threshold to 20).
2. In other case, we empty the current dictionary that contains the current symbol search count in order to avoid the possibility of predicting a wrong character.
3. If the number of blank spaces detected in a plain background exceeds a certain amount and there are no spaces detected in the current buffer, then no spaces will be registered.
4. If a different situation occurs, the program anticipates the conclusion of a word by displaying a blank space and adding the current word to the subsequent sentence.
5. If the sentence is formatted correctly then next step is to generate audio file of sentence. We have use playsound python library to play the audio.

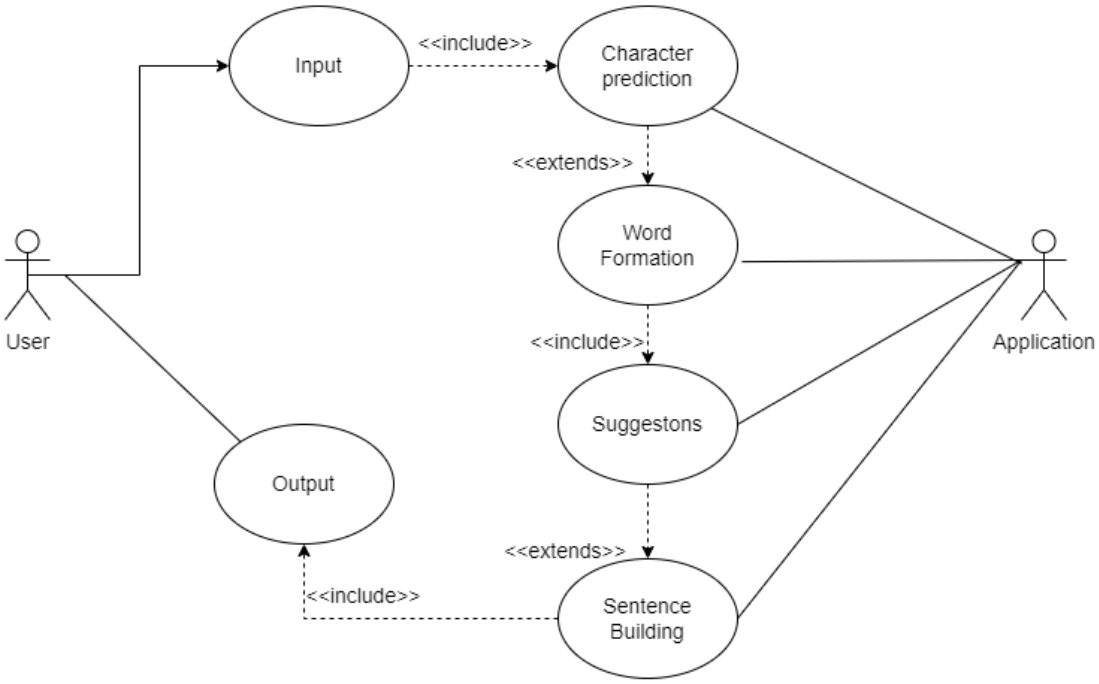


Figure 4: Sentence Formation flow

0.6 CNN(Convolutional Neural Networks) Model for Image Processing

Image preprocessing in a CNN involves resizing, normalization, augmentation, filtering, non-linear activation, pooling, flattening, fully connected layer, and softmax layer. These steps help in extracting meaningful features from the input image and producing accurate predictions.

Below is a concise summary of the stages included in preprocessing an image in a convolutional neural network (CNN) for a real-time project of sign language to speech conversion.

0.6.1 Steps

1. 1st Convolution Layer:
   - The first input image with 128x128 resolution is processed in 1st convolution layer with filter size of 3x3. This will convert in 126x126 resolution image
2. 1st Pooling Layer:
   - Now the with pooling operation of filter size 2x2, resulted image convert into 63x63 pixel.
3. 2nd Convolution Layer:
   - In 2nd convolution layer input of 63x63 pixel images are convert into 60x60 pixel using filter size 3x3.
4. 2nd Pooling Layer:
   - Output image is reduced to 30x30 pixel image by using max pool of 2x2 filter.
5. 1st Densely Connected Layer:
   - 5th layer is fully connected layer with 128 filters. In this layer, the output of 2nd convolution layer is transformed into a sequence of elements arranged in an array of 30x30x32= 28800 and enter to the 2nd Densely Connected Layer.

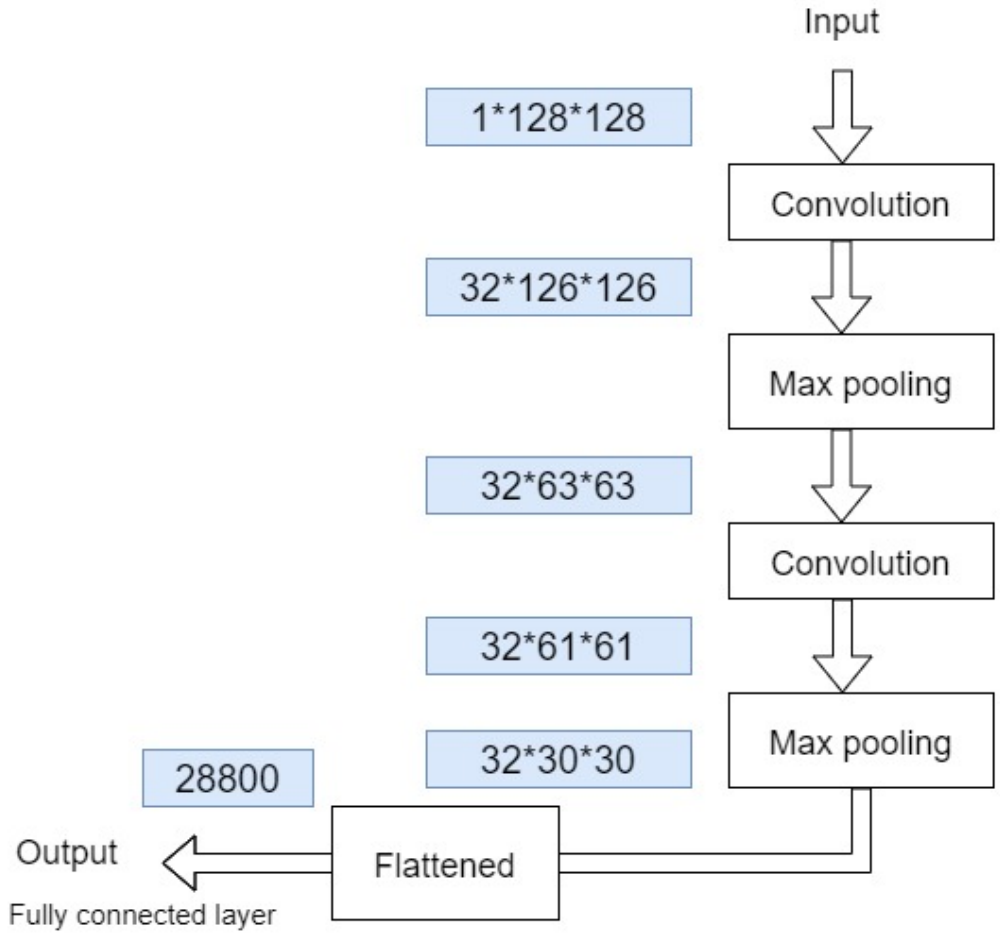


Figure 5: Image preprocessing by CNN layers

6. 2nd Densely Connected Layer:
   - Now the output of 1st Densely Connected Layer is used as an input to fully connected layer with 96 filters.
7. Final Layer:
   - The output of 2nd Densely Connected Layer is serve as an input of final layer which have 'n' number of classes are classifying using datasets.

0.6.2 Layers

0.6.2.1 Activation Function

We have used ReLU (Rectified Linear Unit) to the each of the layer. It finds its most frequent application for activation function in deep learning model.

It allow model to account non-linearity and interaction very well [1]. It helps to removing the vanishing gradient problem by simpler computation method.

0.6.2.2 Pooling Layer

We used the max pooling operation to the each input image with size of 2x2 to reduce the variance and commutations and extract the sharpest feature of images [2].

0.6.2.3 Dropout Layers

To overcome the over-fitting problem, the dropout layer is used wherein few weight are drop from the neural network during training process is resulting in reduce the size of model [3]. This layer dropout random set of node from neural network.

0.6.2.4 Optimizer

We have used Adam Optimizer that implements the Adam algorithm as it gives much higher performance result than the other optimizer [4].

0.7 Auto Correction

We are using Hunspell_suggest python library to suggest correct alternative word for each input. We display 3 set of words and selected word will be append to the current sentence in order to reduce mistakes and assist user.

0.8 Screenshot

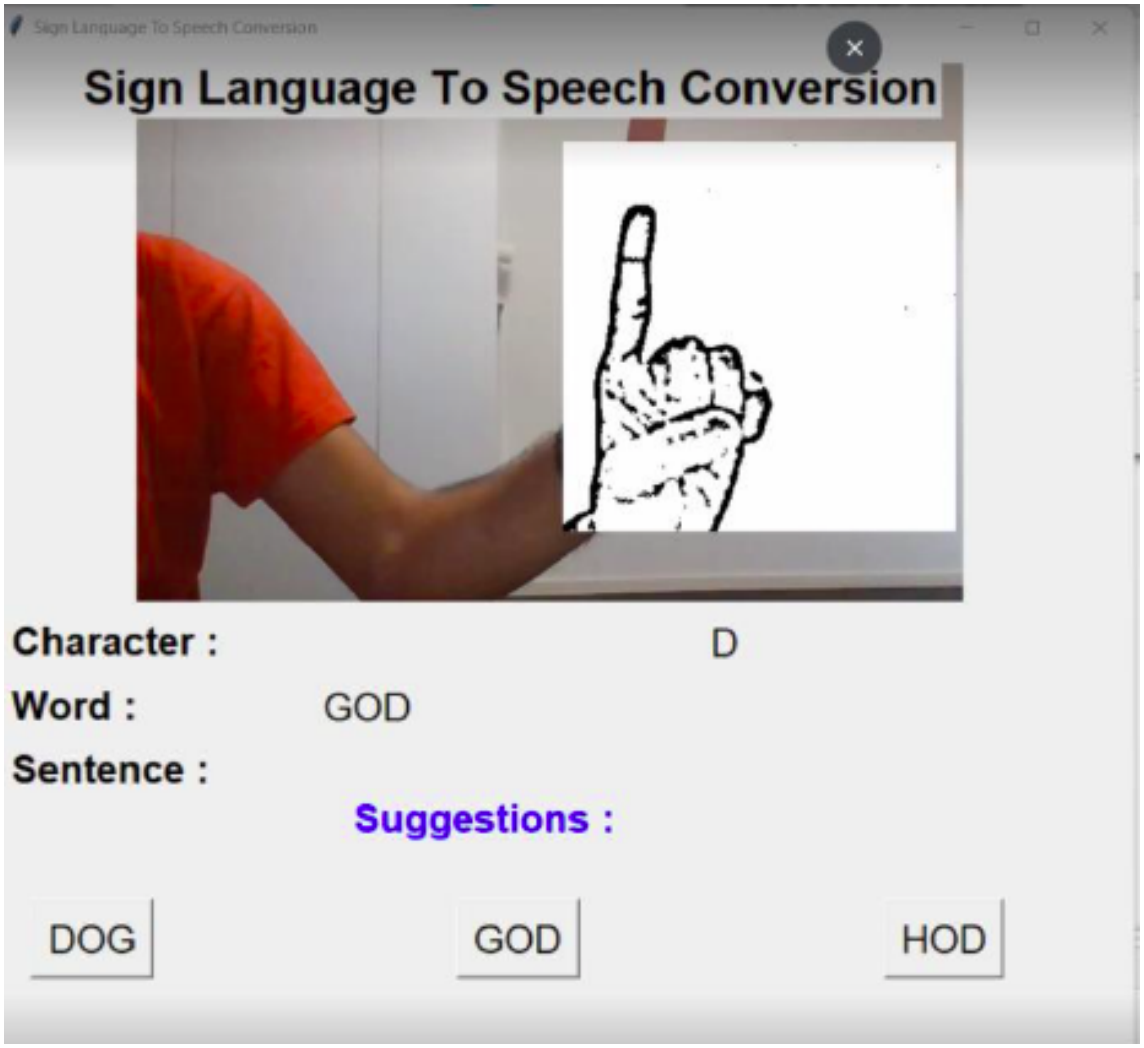


Figure 6: Screenshot of our project

0.9 Training and Testing

To process our input images (RGB), we first convert them into grayscale and then apply a Gaussian blur to eliminate noise. Next, we use an adaptive thresholding technique to extract the hand forward from the background and adjust the size of the images to 128 x 128.

These preprocessed images are fed into our model for both training and testing. The prediction layer uses the SoftMax function to produce a normalized output between 0 and 1, which represents the probability or likelihood of the input image belonging to a specific class.

However, the initial output of the prediction layer may be inaccurate, so we train our model using labeled data and optimize its performance using cross-entropy, a function that measures the difference between the predicted and labeled values.

To reduce the cross-entropy function, we use the Adam Optimizer, which is a Gradient Descent optimization algorithm, to modify the weights of our neural network. TensorFlow provides an inbuilt function to calculate the cross-entropy.

0.10 Results

The model achieved a 87.3% accuracy using only layer 1, and combining layer 1 and layer 2 increased the accuracy to 90.7%, surpassing the accuracy of many current research papers on American sign language that typically use devices like Kinect for hand detection.

The project involved creating a system to recognize Flemish sign language using convolutional neural networks and a Kinect device. The project was successful and achieved an error rate of only 2.5% [5].

A hidden Markov model classifier is used to construct a recognition model in [6] that has a vocabulary consisting of 30 words, and the resulting error rate is 10.90%.

The accuracy of background subtraction in our project may differ, and while other projects use expensive and not easily accessible Kinect devices, our aim was to create an application that could be used with commonly available resources. Therefore, using a regular webcam from a laptop is a significant advantage, as it is both readily available and affordable for most people.

0.11 Ethical Issues

This innovation is one that permits an impaired community to encourage coordinated into an abled community, and may be seen as an digestion and twisting to the rules of favored community.

This may decrease endeavors of hearing individuals to oblige for hard of hearing individuals.

The datasets must be assorted sufficient in arrange to oblige individuals of all skin tones and in all situations.

A predisposition in information may conceivably drawback hard of hearing individuals of a certain ethnic gather.

Require a great light and plain foundation to capture a motion.

0.12 Conclusion

To this extend, we proposed an thought for attainable communication between hearing-disabled and ordinary individuals with the assistance of profound learning and machine learning approaches.

Our data set has been accurately predicted with a final accuracy of 90.7%. By using two layers of algorithms, we were able to improve our predictions by identifying and predicting symbols that are more similar to each other.

As long as the symbols are displayed correctly without any background noise and with sufficient lighting, we can detect nearly all of them.

There's ever the sounding challenge to create a sign dialect framework in information the collection remains invariant of the unconstrained environment.

We are too considering of making strides the prepossessing to anticipate motions in moon light conditions with a better exactness. This venture can be amplified to real-time information.

0.13 Future Work

The team aims to improve the accuracy of their gesture recognition project by exploring various background subtraction algorithms and enhancing pre-processing techniques for low light conditions.

Additionally, they plan to develop a web/mobile application to make the project easily accessible to users.

Additionally, they plan to develop a web/mobile application to make the project easily accessible to users.

Currently, the project only works for American Sign Language (ASL), but with appropriate data sets and training, it could be extended to other sign languages. The current project focuses on finger spelling translation, but to identify contextual signing (where each gesture represents an object or verb), more advanced processing techniques and natural language processing (NLP) would be required.

References

[1] Hao, Wang and Yizhou, Wang and Yaqin, Lou and Zhili, Song, "The role of activation function in cnn," pp. 429–432, 2020. DOI: 10.1109/ITCA52113.2020.00096.

[2] Hang, Siang Thye, Aono, Masaki, "Bi-linearly weighted fractional max pooling," *Multimedia Tools and Applications*, vol. 76, pp. 22 095–22 117, 2017. DOI: 10.1007/s11042-017-4840-5.

[3] Dileep, P and Das, Dibyaiyoti and Bora, Prabin Kuma, "Dense layer dropout based cnn architecture for automatic modulation classification," pp. 1–5, 2020. DOI: 10.1109/NCC48643.2020.9055989.

[4] Mehta, Smit and Paunwala, Chirag and Vaidya, Bhaumik, "Cnn based traffic sign classification using adam optimizer," pp. 1293–1298, 2019. DOI: 10.1109/ICCS45141.2019.9065537.

[5] Dieleman S., Pigou L., Kindermans PJ., Schrauwen B, "Sign language recognition using convolutional neural networks," vol. 8925, 2015.

[6] Zaki, M.M., Shaheen, S.I., "Sign language recognition using a combination of new vision-based features," 2011.