****

**TRIBHUVAN UNIVERSITY**

**INSTITUTE OF ENGINEERING**

JANAKPUR ENGINEERING COLLEGE

KUPONDOLE, KATHMANDU

**A MAJOR PROJECT MID TERM REPORT**

**ON**

**SIGN LANGUAGE TO SPEECH CONVERTER**

**Submitted By:**

ANJAN SIWAKOTI (073/BCT/02)

GORASH KARKI (073/BCT/06)

NIRANJAN SHRESTHA (073/BCT/07)

PRADIP BISTA (073/BCT/08)

A MAJOR PROJECT MID TERM REPORT SUBMITTED TO THE DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING FOR THE PARTIAL FULLFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR’S IN COMPUTER ENGINEERING

**Submitted To:**

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

March, 2020

**ACKNOWLEDGEMENT**

First of all, we express our special gratitude to the Department of Electronics and Computer Engineering, Janakpur Engineering College for providing us the opportunity to explore our interest and ideas in the field of computer engineering through this project. We would also like to acknowledge to all those who provided us the encouragement and possibility to complete this proposal as a part of duty for the project.

We would especially like to express our sincerest appreciation to our project supervisor, Mr. Sudarshan Subedi, who guided us in every step as well as provided us with all the information required during the preparation of this major project mid-term report. Furthermore, we would also like to acknowledge, with much appreciation, the crucial role of our teachers at Janakpur Engineering College, who assisted us during the research and feasibility study of the project.

Lastly, we would like to thank every character that showed the contribution as assistance directly and indirectly in the duration of survey and preparation of this proposal. Their effort and sincerity on the field are always memorable. The positive effort shown by all other related people who inspired us as well as involved in this project in near future will be appreciable.

Thank You.

Anjan Siwakoti(073/BCT/02)

Gorash Karki(073/BCT/06)

Niranjan Shrestha (073/BCT/07)

Pradip Bista(073/BCT/08)

7h March 2020

**ABSTRACT**

We decided on developing an application named “Sign Language To Speech Converter” which is a project intended for helping every person to learn and understand sign language to make conversation easier and understandable for people. Mostly mute people are having hard time conversing with others, as sign language is known by only selected few. Thus, this project is intended to assist the user to be able to converse with any person, so that the people could understand the mute people. Although there are lots of application that can convert text to speech but only few places to learn sign language.

We have decided to use Python 3.0 as our core language for programming. We will be using tensor flow as a machine learning library to store the sign language images as well as their meaning and for the detection of the sign language as input and automated speech as output. We hope that this application will assist the potential users to be able to converse with each other regardless of the disability. To sum up, this application is helpful for each and every person willing to have conversation with the disabled person, this project is solely for mute person to converse with normal person using the modern technology to convert their sign language to automated speech.

Keywords: Sign language, tensor flow, disability, Python 3.0

**TABLE OF CONTENTS**

TITLE PAGE i

ACKNOWLEDGEMENT ii

ABSTRACT iii

TABLE OF CONTENTS iv-v

LIST OF FIGURES vi

LIST OF TABLES vii

LIST OF ABBREVIATIONS viii

1. INTRODUCTION 1

1.1 Objectives 1

1.2 Motivation and Significance 2

1.3 Problem Definition 2

1.4 Scope and Application 2

2. LITERATURE REVIEW 3-7

3. REQUIREMENTS ANALYSIS 8

3.1 Project Requirements 8

3.1.1 Software 8-11 3.1.2 Hardware 11

3.2 Datasets Descriptions 11-12

3.3 Feasibility Analysis 12-13

3.3.1 Operational Feasibility 12

3.3.2 Technical Feasibility 13

3.3.3 Economic Feasibility 13

4. METHODOLOGY 14-16

4.1 Algorithms 14

4.1.1 CNN Training Algorithm 14-15

4.1.2 Algorithm of the System 15-16

4.2 System Design and Architecture 16-18

4.2.1 Traditional Block Diagram 16

4.2.2 Data Flow Diagram 17

4.2.3 Sequence Diagram 17

4.2.4 System Flow Diagram 18

5. WORK DONE AND WORK TO BE DONE 19

6.PARTIAL RESULT AND ANALYSIS 20

7.FUTURE ENHANCEMENTS AND CONCLUSION 21

8. REFRENCES 22-23

**LIST OF FIGURES**

Figure 2.1: Block diagram of sign detection………………………………………… 3

Figure 2.2: Multi-Layer Perceptron Architecture…………………………………… 5

Figure 3.3 Time Allocation For Completion of Project……………………………… 7

Figure 3.1: Convolutional Neural Network architecture……………………………..10

Figure 3.2: Sign language data set……………………………………………………12

Figure 4.4: System Flow Diagram of Sign language to Speech Converter………… .16

Figure 5: Expected Output…………………………………………………………,,.17

**LIST OF TABLES**

Table 3.2 Work Division of all group members 6

Table 4.1: Hardware Specifications 11

**LIST OF ABBREVIATIONS**

UI: User Interface

ML: Machine Language

SL: Sign Language

CNN: Convolutional Neural Network

**1. INTRODUCTION**

A sign language (also signed language) is a language which, instead of acoustically conveyed sound patterns, uses visually transmitted sign patterns (manual communication, body language and lip patterns) to convey meaning—simultaneously combining hand shapes, orientation and movement of the hands, arms or body, and facial expressions to fluidly express a speaker's thoughts. Sign languages commonly develop in deaf communities, which can include interpreters, friends and families of deaf people as well as people who are deaf or hard of hearing themselves.

Wherever communities of deaf people exist, sign languages develop. In fact, their complex spatial grammars are markedly different from the grammars of spoken languages. Hundreds of sign languages are in use around the world and are at the cores of local deaf cultures. Some sign languages have obtained some form of legal recognition, while others have no status at all. In addition to sign languages, various signed codes of spoken languages have been developed, such as Signed English and Warlpiri Sign Language. These are not to be confused with languages, oral or signed; a signed code of an oral language is simply a signed mode of the language it carries, just as a writing system is a written mode. Signed codes of oral languages can be useful for learning oral languages or for expressing and discussing literal quotations from those languages, but they are generally too awkward and unwieldy for normal discourse. For example, a teacher and deaf student of English in the United States might use Signed English to cite examples of English usage, but the discussion of those examples would be in American Sign Language.

Several culturally well-developed sign languages are a medium for stage performances such as sign-language poetry. Many of the poetic mechanisms available to signing poets are not available to a speaking poet.[1]

The IT system has revolutionized the field of machine learning and artificial intelligence. With the help of machine learning we can identify the possibility of machine recognizing the signs made by human to interpret the signs into speech. This project Sign Language to Speech is a program that helps to identify & read the context that the sign language is trying to convey. This project takes different hand signs (gestures) made by the mute person as input data and analyzes this data to interpret the meaning and provide output in the form of automated speech. With this project we are trying to narrow the conversation barrier with the mute person.

* 1. **Objectives**
* To provide the platform to interpret the series of signs (gestures) to speech.
* To make conversation easier and understandable for people with or without disability and to save the time of both parties.

**1.2 Motivation and Significance**

In context of Nepal there are numerous people with speech disability and their one and only option is to learn sign language. Although there are numerous places where people can learn sign language but the people who actually go there to learn SL are either mute or some close members of mute. In our country, only very few normal people know about SL which makes mute people hard to converse in the society. Even if they have to buy their supplies then they have to point out the item until other person recognizes the items otherwise, they have to take someone who can converse for them. We want our project to convey the message that mute people want to express during their conversation with normal people. Our main aim is to reduce this inconvenience for those mute people in order to converse with normal people. In our project we would like to focus on those people who are mute to reduce their inconvenience towards the language barrier. Although mute people are minority in number but the problem, they face in daily life is harsh. So, we would like to focus on those people to reduce their problem regarding speech.

**1.3 Problem Definition**

Many people around the globe communicate with the help of sign language. In order to communicate with each other, people who cannot speak or hear generally use sign language.

All the disabled people may not get opportunity to learn sign language or even if they get to learn they may not be able to memorize it. Sometimes it creates difficulties in communication and learning it may also consume time.

Our project plays a vital role in solving the problem of communication. People who can hear but cannot speak can take great advantage from it. It makes the communication much easier since we can actually hear what a person who cannot speak is trying to say. People who are having a hard time understanding the signs can take a big sigh of relief. In the context of Nepal, where people cannot get education properly, learning sign language is even difficult. Our project can also be used in teaching since we can actually understand the real meaning of a particular sign.

**1.4 Scope and Application**

* Employ this system in the modern devices that can read the sign language to interpret it into automated speech.
* Improve the conversation experience between people (with mute and normal person).
* Familiarize all people with sign language.

**2. LITERATURE REVIEW**

There are various approaches that have been used for converting sign language images into text or speech.

The threshold model with Conditional Random Field (CRF) was an excellent mechanism for distinguishing between vocabulary signs and non-sign patterns (which included out of vocabulary signs and other movements that do not correspond to signs). A short-sign detector, a hand appearance-based sign verification method, and a sub-sign reasoning method were included to improve sign language spotting accuracy.[2]

Another method was automatic sign recognition. Its unique features were an adaptive skin model, DTW on a reference sign for synchronization, robust recognition method which was real-time and person-independent statistics, automatic feature selection for finding the best sign representation and a tolerance parameter TF that changed the behavior of the base classifiers instead of the threshold on the total likelihood. DTW was used only for finding the best path, to synchronize the signal. This method was able to generalize well over different persons, which was troublesome for many other systems. [3]

In another technique, computer vision method was used for recognizing sequences of human-hand gestures within a gloved environment. Vectors were utilized for representing the direction and displacement of the fingertips for the gesture. Modelling gestures as a set of vectors with a motion key allowed the reduction of complexity in modern form and matching, which may otherwise contain multiple and lengthy datasets. [4]

The hand shape was used in recognizing people with high accuracy. It was believed that the scorecard of the hand geometry modality could be promoted to ―high‖ in the distinctiveness and performance attributes of person recognition in that the interface was user-friendly and it was not subject to variability to the extent faces were under confounding factors of accessories, illumination effects and expression. [10] Preliminary tests indicated that hand biometric accuracy was maintained over a span of time. For any hand-based recognition scheme, it was imperative, however, that the hand image be pre-processed for normalization so that hand attitude in general, and fingers in particular be aligned to standard positions [5]

Another comprehensive approach to robust visual sign language recognition system aimed to signer-independent operation and utilized a single video camera for data acquisition to ensure user friendliness. In order to cover all aspects of sign languages, sophisticated algorithms were developed that robustly extract manual and facial features, also in uncontrolled environments. The classification stage was designed for recognition of isolated signs as well as of continuous sign language. For statistical modelling of reference models, a single sign could be represented either as a whole or as a composition of smaller subunits—similar to phonemes in spoken languages. In order to overcome the problem of high interpersonal variance, dedicated adaptation methods known from speech recognition were implemented and modified to consider the specifics of sign languages. [6]

A novel algorithm to extract signemes, i.e. the common pattern representing a sign, from multiple long video sequences of American Sign Language was implemented. A signeme is a part of the sign that is robust to the variations of the adjacent signs and the associated movement epenthesis. Iterative Conditional Modes (ICM) to sample the parameters, i.e. the starting location and width of the signeme in each sentence in a sequential manner were used. In order to overcome the local convergence problem of ICM, it was run repetitively with uniformly and independently sampled initialization vectors. The results on ASL video sequences that do not involve any magnetic trackers or gloves, and also on a corresponding audio dataset were shown.[7]

Yet in another approach, an application’s speech and audio output is translated into text using existing speech-to-text conversion programs. The system translates key text words or phrases into the appropriate sign language. For this translation, pre captured gesture database and Java 3D were used to construct the simple 3D hand model, achieving a rich, interactive, animated environment focusing more on the hand’s degrees of freedom (DOF) rather than texture. However, this approach focused on communicating by hand gestures that can be captured by only fingers and palms. For gestures that required other hand motions, such as wrist rotations and hand translations or facial, expressions, more data needed to be incorporated into the system [8].

A demonstrator for generating VRML animation sequences from Sign Language notation, based on MPEG- 4 Body Animation was developed. The system was able to convert almost all hand symbols as well as the associated movement, contact and movement dynamics symbols contained in any ASL sign-box. [9]

The pattern of the sign in binary form is carefully observed and they are then classified on the basis of Multi-Layer Perceptron architecture which is a feed forward artificial neural network model that maps sets of input data onto a set of appropriate output.

Current research has founded that:

* Leap Motion controller is a sensor which detects the hand movement and converts that signal into computer commands. It consists of two IR cameras and three infrared LED’s. LED generates IR light signal and camera generates 300 frames per second of reflected data. These signals are sending to the computer through USB cable for further processing.

P. Karthick et al. used model that transform Indian sign language into text using leap model. The Leap device detects the data like point, wave, reach, grab which is generated by a leap motion controller. Combination of DTW and IS algorithm is used for conversion of hand gesture into text. Neuron network was used for training the data.

Leigh Ellen Potter et al. used leap motion controller for recognition of Australian sign language. Leap motion controller used to sense the hand movement and convert that hand movement into computer commands. Artificial neuron network is used for training symbols. The disadvantage of that system was low accuracy and fidelity.

* Kinect is Microsoft motion sensor with Xbox 360 gaming console. It consist of RGB camera, depth sensor and multi-array microphone. It recognizes facial movement and speech.

Cao dong et al. [3] used Microsoft kinect to recognize American sign language. Depth camera is kinect sensor used to detect ASL alphabet. Distance adaptive scheme was used for feature extraction. Support vector machine and RF classifier algorithm used for classification purpose. Training of data was done using ANN network. The accuracy of the system was 90%. uan yao et al. used Kinect sensor for recognition of hand gesture. Firstly, it detects hand movement and then matched with counter model. Second task was to locate multi-color glove and detect different color regions. Gaussian color model used for training data and per pixel classifier used for classification. This system has one drawback that is limited accuracy.

* Data glove is a device which uses different sensor to detect hand gesture signal. Hand gesture signal is in the form of analog. ADC is used to convert analog signal into digital form. It consists of flex sensor and accelerometer. Flex sensor is used to detect bend signal [5]. In this method web camera used to capture images. After that, image segmentation has done. Feature like palm, finger extracted from input image. Different hand motion that is half closed, fully closed, semi closed was detected. Data is saved in vector and that vector is used for recognition of alphabets.

Anarbasi Rajamohan et al. used data glove-based method for recognition of American Sign Language. The system consists of flex sensor, accelerometer and tactile sensor. This sensor used to detect hand gesture and converted into code. Accuracy of that system was 90%.

* Vision based is the method web camera used to capture images. After that, image segmentation has done. Feature like palm, finger extracted from input image. Different hand motion that is half closed, fully closed, semi closed was detected. Data is saved in vector and that vector is used for recognition of alphabets.

Paulo Trigueiros et al. used vision-based technique for recognition of Portuguese language. For their implementation, hand gesture was captured in real time. SVM algorithm is used for classification purpose. In this system vowels recognized with accuracy 99.4% and consonants recognized with 99.6% accuracy. Generally, while capturing the image for experiments, head movement is also mixed with hand images. To solve this overlap between hand and head movement, camera is mounted above of signers. But due to this face and body gesture lost. Nilsen et al used less hand gesture for fast recognition process.[14]

For the past decades, research on SLR has been explored. Many studies used sensor-based devices such as Sign Speak. This device used different sensors such as flex and contact sensors for finger and palm movements and accelerometers and gyros for the hand movement; then, by Principal Component Analysis, the gloves were trained to recognize different gestures, and each gesture was then classified into alphabets in real time. The device also used an Android phone to display the text and word received from the gloves via Bluetooth. Sign Speak was found to have 92% accuracy. There are other means of capturing signs by using motion sensors, such as electromyography (EMG) sensors, RGB cameras, Kinect sensors, and leap motion controller or their combinations. Although these sensors provide accurate parameters in measurement of data, they also have limitations; first is their cost, as they require large-size datasets with diverse sign motion they going toned a high-end computers with powerful specifications; next is aesthetics, as the sensors are attached to the fingers and palms of a user, the user can encounter difficulties in setting up the device; ambient lighting conditions or backgrounds in real-world settings may also affect the recognition. Therefore, many researchers jumped from sensor-based to visual-based SLR.

Several methods have been developed in visual-based SLR. Because sign language includes static and dynamic movements, image, and video processing was explored by many.

Wang et al. used color spaces to identify hand gestures and acquired segment images by setting a range of the skin color threshold. Hand gesture segmentation is simply done by using the hand skin threshold method. The system would not produce good results because of lighting conditions, skin color interference, and complex backgrounds that increased noise. There are three types of skin color detection: the explicit range method, the nonparametric method and the parametric method. The explicit range method differentiates the class of pixels into skin- and non-skin-based types from the assigned range of colors. This technique is used mostly because of its non-complex approach and acceptable rate of computation. However, this technique is only limited for a generalized skin color scheme. Another approach was taken by Balbin et al., who used colored gloves for the hands to be identified easily by setting an exact range of the hand skin color threshold (color of the gloves). To recognize the hand gesture, input images underwent various image processing methods or steps. First is pre-processing wherein images were converted into grayscale, and median filter is used to denoise the image. Next, is feature extraction wherein the color of the hand gloves was detected and isolated from the background. Then, the image had undergone pattern recognition. The system used Kohen self-organizing maps, which are the type of a neural network that can learn to identify patterns and group datasets in an unsupervised manner. The system was tested by five persons, and it achieved an accuracy of 97.6%.

These studies propose a complex yet manageable process of skin color thresholding; it can be seen that when only the bare hands of the signer are used, it is difficult for the system to recognize the gesture because of different hindrances such as noise. Other studies used colored gloves to solve the problem, whereas the present study proposed a system that can recognize static sign language without the aid of gloves or hand markings but still produce acceptable results.[20]

**3. REQUIREMENTS ANALYSIS**

* 1. **Project requirements**
     1. **Software**

**Visual Studio Code**

Visual Studio Code is a source-code editor developed by Microsoft for Windows, Linux and macOS. It includes support for debugging, embedded Git control and GitHub, syntax highlighting, intelligent code completion, snippets, and code refactoring. It is highly customizable, allowing users to change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality. The source code is free and open source and released under the permissive MIT License. The compiled binaries are freeware and free for private or commercial use.[15]

**Python**

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

Python is a programming language that lets you work quickly and integrate systems more efficiently.[13]

**Machine Learning**

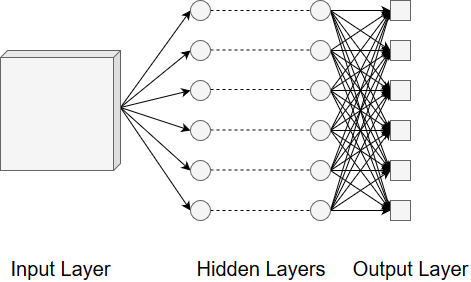
Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop a conventional algorithm for effectively performing the task.

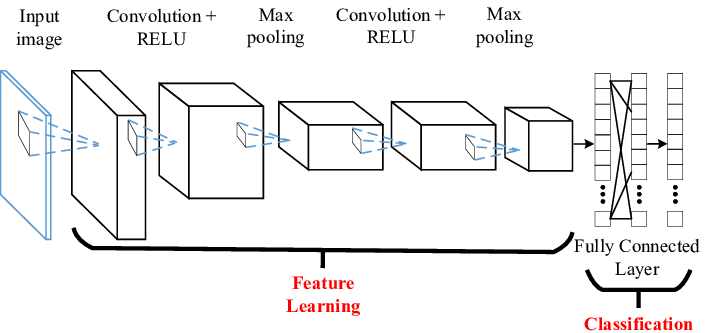
Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a field of study within machine learning, and focuses on exploratory data analysis through unsupervised learning. In its application across business problems, machine learning is also referred to as predictive analytics.[12]

**Convolutional Neural Network**

Convolutional Neural networks are designed to process data through multiple layers of arrays. This type of neural networks is used in applications like image recognition or face recognition. The primary difference between CNN and any other ordinary neural network is that CNN takes input as a two-dimensional array and operates directly on the images rather than focusing on feature extraction which other neural networks focus on.

The dominant approach of CNN includes solutions for problems of recognition. Top companies like Google and Facebook have invested in research and development towards recognition projects to get activities done with greater speed.[12]





*Figure 3.1: Convolutional Neural Network architecture*

**Tensor Flow**

TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications.[12]

**TFlearn**

TFlearn is a modular and transparent deep learning library built on top of TensorFlow. It was designed to provide a higher-level API to TensorFlow in order to facilitate and speed-up experimentations, while remaining fully transparent and compatible with it.

* Easy-to-use and understand high-level API for implementing deep neural networks, with tutorial and examples.
* Fast prototyping through highly modular built-in neural network layers, regularizers, optimizers, metrics...
* Full transparency over TensorFlow. All functions are built over tensors and can be used independently of TFLearn.[12]

**OpenCV**

OpenCV (Open source computer vision) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez (which was later acquired by Intel). The library is cross-platform and free for use under the open-source BSD license.

OpenCV supports some models from deep learning frameworks like TensorFlow, Torch, PyTorch (after converting to an ONNX model) and Caffe according to a defined list of supported layers.[18]

**NumPy**

NumPy is the fundamental package for scientific computing with Python. It contains among other things:

* a powerful N-dimensional array object
* sophisticated (broadcasting) functions
* tools for integrating C/C++ and Fortran code
* useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.[19]

**Anaconda**

Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. Package versions are managed by the package management system conda. The Anaconda distribution includes data-science packages suitable for Windows, Linux, and MacOS.[17]

**3.1.2 Hardware**

The system we are going to use have following specifications:

|  |  |
| --- | --- |
| System | Requirements |
| **Operating System** | Windows 10 |
| **Processor** | Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz (8 CPUs), ~1.8GHz |
| **RAM** | 8192MB RAM |
| **GPU** | NVIDIA GeForce MX130( 4161 MB) |

### 

Table 4.1: Hardware Specifications

**3.2. Data Set Descriptions**

A data set (or dataset) is a collection of data. In the case of tabular data, a data set corresponds to one or more database tables, where every column of a table represents a particular variable, and each row corresponds to a given record of the data set in question. The data set lists values for each of the variables, such as height and weight of an object, for each member of the data set. Each value is known as a datum. Data sets can also consist of a collection of documents or files.

We have researched and found out the datasets can be found from some of the websites like Kaggle, Google Datasets, Buzzfeed, Socrata etc.

For the project we would like to use to data sets of sign languages. Some images of datasets of sign languages are shown below:



*Figure 3.2: Sign language data set*

**3.3 Feasibility Analysis**

A feasibility study is a study that includes the analysis of the software if it is cost effective from the economic view, if it can fulfill the requirement technically, and if it is adaptable in the required environment. It also condiments the groundwork and determine whether the project should be taken or not. Finally, the net result will be rough plane for proceeding with the project.

**3.3.1 Operational Feasibility**

Operational feasibility includes operational analysis of the overall system. Our system is operationally feasible as it will have a good user interface along with better usability and understandability. Also, it aims to fulfill day to day user requirements for communicating in sign language.

**3.3.2 Technical Feasibility**

The proposed project is technically feasible as our system will be developed in normal operating system with the help of Python3.0 along with Tensor flow 2.0 as library for implementation of Convolution neural network. With the help of some of the libraries and machine learning functionality, all the desired functional requirements for this system can be implemented.

**3.3.3 Economic Feasibility**

The system being developed have normal hardware requirement. The Software being used is available for free, resulting in low development cost. Hence, the system is economically feasible.

1. **METHODLOGY**

**4.1. Algorithms**

The algorithms that we will be using for this project are listed and described below:

**4.1.1 CNN Training Algorithm**

CNN0 is trained on all training sets and the parameters of each layer are as follows:

* Layer 1 consists of 20 10 × 10 convolution kernels that perform convolution operations on the input images. The step length is initially set to 4. Then, 3 × 3 max pooling windows with a step length of 2 are used for down sampling.
* Layer 2 consists of 40 5 × 5 convolution kernels that perform convolution operations on the feature maps. The step length is initially set to 2. Then, 3 × 3 max pooling windows with a step length of 2 are used for down sampling.
* Layer 3 consists of 60 3 × 3 convolution kernels that perform convolution operations on the input feature maps. The step length is set to 1. The remaining three layers are fully connected layers. A dropout layer is applied for full connection to avoid overfitting. The keep pro (proportion) parameter is set to 0.5 (i.e., 50% of the neurons at each of the fully connected layers participate in the operation). The number of output nodes is 20. Considering that the ReLU activation function possesses powerful expression ability and is free from the vanishing gradient problem, enabling the convergence rate of the model to be maintained stably, we used the ReLU function in this study for all activations. The learning rate was adjusted to 0.001 after the experiment.

CNN1 is trained on the low-frequency training sets, and the parameters for each layer are as follows:

* Layer 1 consists of 20 12 × 12 convolution kernels that perform convolution operations on the input images. The step length is initially set to 2. Then, 5 × 5 max pooling windows with a step length of 4 are used for down sampling.
* Layer 2 consists of 40 5 × 5 convolution kernels that perform convolution operations on the input feature maps. The step length is initially set to 1. Then, 4 × 4 max pooling windows with a step length of 2 are used for down sampling.
* Layer 3 consists of 60 4 × 4 convolution kernels that perform convolution operations on the input feature maps. The step length is set to 1. Then, 4 × 4 max pooling windows with a step length of 2 are used for down sampling.

The final three layers are all fully connected layers whose parameters are the same as those of CNN0.

The model increases the training weight of the low-frequency samples using special training channels. The training samples are first processed separately to achieve sample equalization. Then, during the labeling process, the final labeling result is jointly determined by the two channels. Because the low-frequency channel is trained only with low-frequency samples, the parameters of this channel are more suitable for identifying low-frequency samples, which reduces the labeling impact of training with insufficient numbers of low-frequency samples.

And thus, it is divided into training and labeling algorithm and implemented as:

The algorithm corresponding to the training phase is as follows.

 Step 1. Launch the algorithm corresponding to the training phase. Sum the number of samples corresponding to each tagged word and determine the low-frequency annotation word set.

 Step 2. Through the program, all the low-frequency samples in the training sample are extracted to form a low-frequency training set.

 Step 3. Construct a CNN model with two channels: CNN0 and CNN1. CNN0 corresponds to the channel with a small convolutional kernel and a large step, and CNN1 corresponds to the channel with a large convolutional kernel and a small step. The first layer of the CNN1 channel is fully connected to the second layer of the fully connected layer for feature fusion.

 Step 4. Input all the training sets into the CNN0 channel and input only the low- frequency training samples into the CNN1 channel. Conduct model training until the model becomes stable.

The labeling phase algorithm is as follows.

 Step 1. Input the test image into both channels (CNN0 and CNN1) of the trained two- channel CNN for feature extraction

 Step 2. Fuse the output vectors of the two channels in a 2: 1 manner (the specific ratio is experimentally determined)

 Step 3. Combine the decision results of the two channels to perform image annotation.

**4.1.2 Algorithm of the System**

**S**tep1: Start

Step 2: Open the application

Step 3: Operation of the application

Step 3.1: Check whether the object is detected or not.

If the object gets detected go to webcam analysis then step 3.2 else

Display error

Step 3.2: Check whether the hand is detected or not.

If the hands get detected go to step 3.3 else

Display inaccurate result/ prediction

Step 3.4 Determine the hand sign.

Step 3.4.1: Check whether the sign is valid or not using Convolution neural network and training Algorithm.

If the sign is valid go to step 3.4.2 else

Display inaccurate result/ prediction

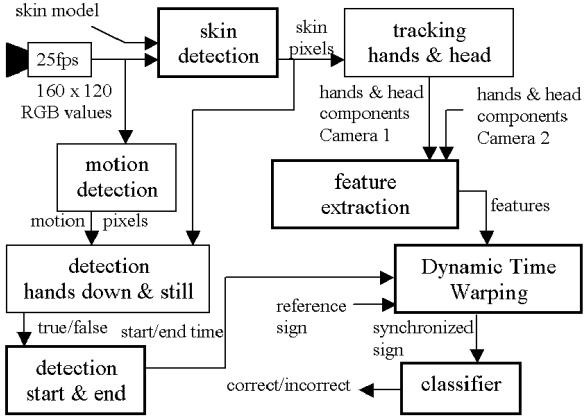
Step 3.4.2: Process the sign using tensor flow as framework.

Step 4: Display the result and interpret in form of speech

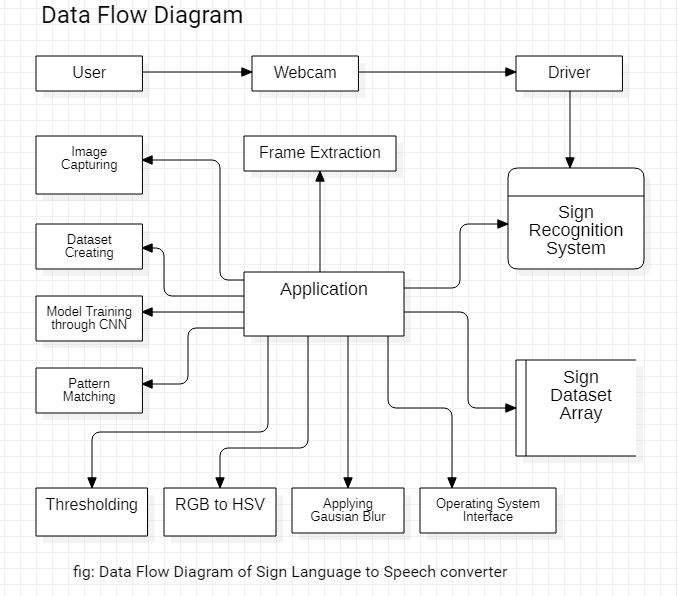
Step 5: End

* 1. **System Design and Architecture**

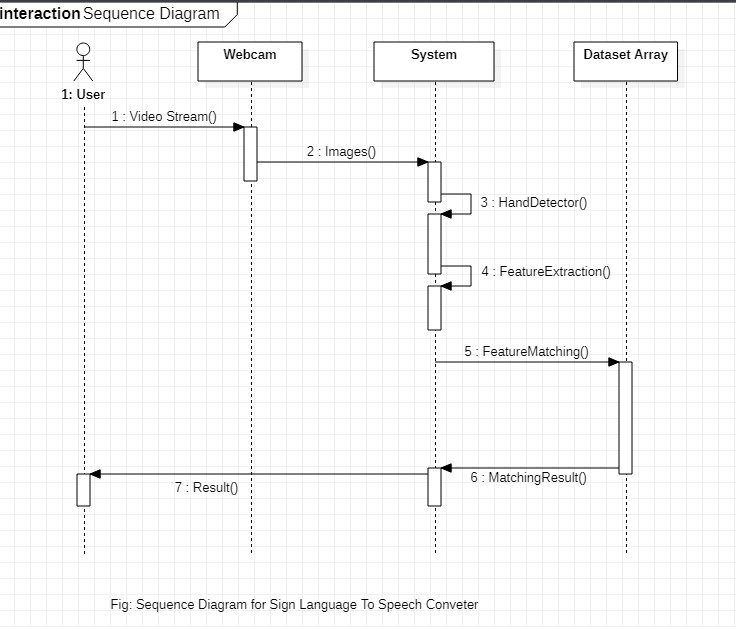
**4.2.1 Traditional Block Diagram**



*Figure 2.1: Block diagram of sign detection [3]*

**4.2.2 Data Flow Diagram**

*Figure 4.2.2: Data Flow Diagram*

**4.2.3 Sequence Diagram**

*Figure 4.2.3: Sequence Diagram*

**4.2.4 System Flow Diagram**

Open the Application

Operation of the system

Display Inaccurate

Result/ Prediction

Is object detected?

**No**

**Yes**

Webcam Analysis

**No**

**Yes**

Is required object a hand?

**No**

Is sign valid?

**Yes**

**No**

**Yes**

Display and interpret the result in speech

Analyze the sign using CNN and training algorithm with application of tensor flow

*Figure 4.4: System Flow Diagram of Sign language to Speech Converter*

1. **WORK DONE AND WORK TO BE DONE**

The work of the project that have been done until now are:

1. Collected the data sets and created the dataset by capturing the image of hand sign using webcam.
2. Created the model for the sign language to text
3. Fed the convolution neural network with the data sets.
4. Trained the model with the datasets by using TensorFlow
5. Detect the hand sign using the OpenCV as a camera input through webcam.
6. Able to predict less number of data sets only with less accuracy.

The work that are left to be to be done are:

1. Successfully predict all the hand signs with high accuracy.
2. Concatenate the letter predicted to make a word.
3. Combine words to form a sentence.
4. Implement the whole system or application to Webapp.
5. **PARTIAL RESULT AND ANLAYSIS**

From the project we have completed up to now has following output as show below:

*Figure 5: Expected Output*

After the development of the product we are expecting the following result as outcome from the working of the product.

* Able to learn more about ML and its functionality.
* No problem conversing regardless the normal person knows about sign language or not.
* Familiarize with the working of each components related to the project.
* Able to interpret the meaning behind series of signs being used.

**7. FUTURE ENHANCEMENTS AND CONCLUSION**

Thus, if the product is successfully developed and has high accuracy and efficiency then it can be enhanced into mobile application as well so that any time any where they can use this application. It can be used in in various government and non-government organization, various machineries and also from this technique various devices can be controlled with the help of hand sign too.

With the product that is proposed, Sign Language to Speech Converter, we aim to provide user a good experience and help users with conversation with sign language easily anywhere. We also plan to make this product as well in android app too and many more features in the near future which will make the users to use our product more efficiently and smoothly.

**8. REFERENCES**

1. T.H. Samiha (TP034305) “Hand gesture recognition system” <https://www.slideshare.net/AfnanRehman/hand-gesture-recognition-systemfyp-report-50225618> Internet:, July. 18, 2017 [Aug. 13, 2019].
2. Sign Language Spotting with a Threshold Model Based on Conditional Random Fields Hee-Deok Yang, Member, IEEE, Stan Sclaroff, Senior Member, IEEE, and Seong-Whan Lee, Senior Member, IEEE
3. SIGN LANGUAGE DETECTION USING 3D VISUAL CUES J.F. Lichtenauer G.A. ten Holt E.A. Hendriks M.J.T. Reinders Information and Communication Theory Group Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS) Delft University of Technology, The Netherlands
4. Visual Gesture Recognition by J. Davis and M. Shah
5. Shape-Based Hand Recognition by Erdem Yörük, Ender Konuko˘glu, Bülent Sankur, Senior Member, IEEE, and Jérôme Darbon
6. Recent developments in visual sign language recognition by Ulrich von Agris Æ Jo¨rg Zieren Æ Ulrich Canzler Æ Britta Bauer Æ Karl-Friedrich Kraiss
7. Automated Extraction of Signs from Continuous Sign Language Sentences using Iterated Conditional Modes by Sunita Nayak, Sudeep Sarkar, Barbara Loeding
8. Hand-Gesture Computing for the Hearing and Speech Impaired Gaurav Pradhan and Balakrishnan Prabhakaran University of Texas at Dallas Chuanjun Li Brown University
9. TEXT-TO-SIGN LANGUAGE SYNTHESIS TOOL
10. John Wiley & Sons, Crystal, D. Dictionary of Linguistics and Phonetics. The Language Library. 2011
11. Huenerfauth, M. Generating American sign language classifier predicates for English-to-asl machine translation. PhD thesis, University of Pennsylvania, 2006.
12. Google Brain Team “TensorFlow” <https://www.tensorflow.org> Internet:, December 2019[November 9, 2015]
13. Guido Van Rossmum “Python programming” <https://www.python.org> Internet:, December 2019[ 1990]
14. Rawate A.M,” A Review Paper on Sign Language Recognition System For Deaf And Dumb People using Image Processing”, ISSN: 2278-0181, March-2016
15. Microsoft,” Visual Studio Code” Internet: https://code.visualstudio.com, Apr.29,2015[July. 29, 2019].
16. A Jeff, S Joel “Stack Overflow” Internet: https://stackoverflow.com Sep 15,2008[July.1,2019]
17. Anaconda, Inc.,”Anaconda” Internet: <https://anaconda.com>, July 2012[December 19 2019]
18. Intel Corporation, Willow Garage, Itseez,” OpenCV” Internet: <https://opencv.org> , June 2000[February 20 2020]
19. Travis Oliphant, ”NumPy” Internet: <https://github.com/numpy/numpy/releases> ,2006[February 20 2020]
20. Lean Karlo S. Tolentino,” Static Sign Language Recognition Using Deep Learning”, Vol. 9, No. 6, December-2019