

**An Efficient Interpretation  
Model for  
People with Hearing and Speaking Disabilities**



Inspiring Excellence

**Supervisor: Dr. JiaUddin**

M. M. Mahmud Sayeed 11201045

Anisha AnjumHossain 13201067

SamrinPriya 13301101

**Department of Computer Science and Engineering,**

**BRAC University**

**Submitted on: 22<sup>nd</sup> August 2017**

## **Declaration**

We hereby declare that this thesis report is our hard work put together, which has not been submitted anywhere for any award. All the contents written here are totally based on our unyielding labor and findings, dedicated to complete our undergraduate thesis. In this paper, we used information from other sources, which have been acknowledged and referred appropriately in the reference section.

Dr. JiaUddin

Signature of Supervisor

M. M. Mahmud Sayeed

Signature of Author

Anisha Anjum Hossain

Signature of Author

Samrin Priya

Signature of Author

## **Acknowledgement**

All thanks to our Almighty, the superior of all, our merciful creator; who has given us the strength, patience and goodwill to complete this research.

Our thesis Advisor, Dr. Jia Uddin has been very kind to help and guide us throughout the journey. He has shared his knowledge and experience, to guide us the whole time to complete this thesis paper. It was an honor for us to work under him.

A very special thanks to our friends, Sauvik Kundu and Toqi Tahmid, students of this University and Department of Computer Science and Engineering, our fourth and fifth horsemen. Both of them are great people with big hearts and have helped us whenever we faced difficulty throughout this journey.

Lastly, thanks to our family and friends who have supported us all throughout the time. Things would not have been possible without their constant support.

# Table of Contents

<b>DECLARATION.....</b>	<b>ii</b>
<b>ACKNOWLEDGEMENT.....</b>	<b>iii</b>
<b>TABLE OF CONTENTS.....</b>	<b>iv</b>
<b>LIST OF FIGURES.....</b>	<b>vi</b>
<b>ACRONYMS AND ABBREVIATIONS.....</b>	<b>vii</b>
<b>Abstract.....</b>	<b>1</b>
<b>Chapter 1: Introduction</b>	
1.1 Motivation.....	2
1.2 Thesis Goal.....	2
1.3 Thesis Outline.....	2
<b>Chapter 2: Background Study</b>	
2.1 System Description	
2.1.1 Introduction to the MKS.....	4
2.1.2 Kinect SDK.....	4
2.1.3 Kinect Hardware.....	5
2.1.4 Kinect Drivers.....	5
2.1.5 Microsoft Visual Studio (IDE).....	6
2.1.6 Camera.....	6
2.1.7 Display/Screen.....	6
2.2 Related Research.....	7
<b>Chapter 3: Design and Implementation</b>	
3.1 Proposed Model.....	9
3.2 Model Description	

3.2.1 Speech to Gesture and Text.....	9
3.2.2 Gesture to Image and Text	
3.2.2.1 CNN (Convolutional Neural Network).....	10
3.2.2.2 Image matching using CNN.....	11
3.2.2.3 Data Augmentation.....	11
3.2.2.4 Fine-Tune PretrainedConvNets.....	11
3.2.2.5 Final Prediction.....	12
<b>Chapter 4: Result and Analysis</b>	
4.1 Dataset	
4.1.1 SGT Dataset.....	13
4.1.2 GIT Dataset.....	15
4.2 Result	
4.2.1 Analysis for SGT.....	16
4.2.2 GIT Analysis.....	19
4.2.2.1 Trained Images.....	19
4.2.2.2 Test Validation.....	20
<b>Chapter 5: Conclusion and Future Work</b>	
5.1 Conclusion.....	21
5.2 Future Work.....	21
<b>References.....</b>	<b>22</b>

## List of Figures

Figure 1	Kinect Components.....	4
Figure 2	Hardware and Software Interaction with an Application.....	5
Figure 3	Proposed Model.....	9
Figure 4	Pseudo code Speech to Gesture and Text (SGT) .....	10
Figure 5	How CNN works.....	11
Figure 6	Family – Sign Gesture.....	13
Figure 7	Question – Sign Gesture.....	14
Figure 8	Adjective – Sign Gesture.....	14
Figure 9	Gesture Dataset.....	15
Figure 10	Gesture and Text (i).....	16
Figure 11	Gesture and Text (ii) .....	16
Figure 12	Stop (taking input) .....	17
Figure 13	Clear.....	17
Figure 14	GIT – Home.....	19
Figure 15	GIT – Food.....	19

## Acronyms and Abbreviations

API – Application Program Interface

ASL – American Sign Language

CLR – Common Language Runtime

CNN- Convolutional Neural Network

D&M – Deaf and Mute.

DTW – Dynamic Time Warping

FPS – Frames per Second

FSM – Finite State Machine

GIT – Gesture to Image and Text.

Google STT – Speech-To-Text

Google TTS – Text-To-Speech

GUI – Graphical User Interface

HMM – Hidden Markov models

HSV – Hue, Saturation, Value

IDE – Integrated Development Environment

IR – Infrared

ISL – Indian Sign Language

MFCC – Mel Frequency Cepstral Coefficients

MVS – Microsoft Visual Studio

MKS – Microsoft Kinect Sensor

MSIL – Microsoft Intermediate Language

RGB – Red, Green, Blue

SGT – Speech to Gesture and Text.

SL – Sign Language.

SVM – Support Vector Machine

VRS – Video Relay Service MVS

WFP – Windows Forms Applications

WFD – World Federation of the Deaf

## **Abstract**

Sign language is a medium of communication between individuals who have hearing and speaking deficiency. Generally they are called Deaf and Mute. To have a better, effective and simpler communication between the vocal and non-vocal society, it is very important to understand their language without difficulty. Previous research works have shown us different types of efficient methods of interaction between these two sets of people. On the down side, however, they tend to focus on one sided conversation only. Our paper focuses on two very important things; one is converting the American Sign Language (ASL) to text word-by-word and the other is audio to gesture & text conversion. We used the microphone of Microsoft Kinect Sensor device; a camera; along with common recognition algorithm to detect, recognize sign language and interpret the interactive hand shape to American Sign Language text. We decided to build efficient system that can be used as an interpreter among the hearing impaired and the normal people. Besides helping as an interpreter, this research may also open doors to numerous other applications like sign language tutorials in the future.



# Introduction

## 1.1 Motivation

Our world has an immense cultural and lingual diversity. According to World Federation of the Deaf (WFD), there are approximately 70 million Deaf and Mute (D&M) people among us. Each language is unique with distinct culture and is expressed by visible hand gestures. “Just as spoken languages have evolved throughout the world, various signed languages have also emerged in different parts of the world. The languages may differ from region to region and country to country”. [23]

This D&M part of society is always isolated from the people because the only method of communication is Sign Language (SL) whereas interpreters are not always available. Thus they are continuously dependent on another individual to convey what they need to say to people who do not understand SL. “Too often, lack of accessibility bars deaf people from sharing and learning information with others” [1].

## 1.2 Thesis goal

To assist them, in overcoming these feelings of helplessness and reliant, we are developing a complete computer based system. Our system would focus on two main methods of interaction; i) Speech to Gesture and Text conversion (SGT) and ii) Gesture to Image and Text conversion (GIT) word-by-word. In order to achieve this we would use a device called the Microsoft Kinect Sensor (MKS).

## 1.3 Thesis Outline

In this thesis, we will discuss about –

- Chapter 02 – In this part, we discuss about the machineries, the software environment or tools we have used to do the thesis is described and also about the papers which we read through to complete this thesis.

- Chapter 03 – This section discusses about how we designed our system with the appropriate models.
- Chapter 04 – Here we talk about how the design, we mentioned in the earlier chapter, helps us.
- Chapter 05 – How the program works, the results and performance is described in this portion, after the system implementation.

And lastly,

- Chapter 06 – Conclusion and Future Work are presented in this phase.

## Background Study

### 2.1 System Description

#### 2.1.1 Introduction to the MKS (Microsoft Kinect Sensor)

Kinect is a motion sensing input device developed by Microsoft in early 2010 [10]. Kinect is mainly used in Xbox 360 console and for Windows PCs [10]. For Windows, a Kinect sensor consists of an RGB camera which can store up to three channel data of resolution 1280×960. It has an IR (infrared) emitter, which emits lights beams, and an IR depth sensor, that reads the reflected beams and processes the information to measure the distance between the object and the sensor. It also has a multi-array microphone that can capture sound and detect the location of the source and direction of the audio wave. It has a practical ranging limit of 40cm – 3.5m for Windows and the frame rate is 30 FPS (Frames per Second) [11]. Figure 1 depicts the picture of a Kinect.

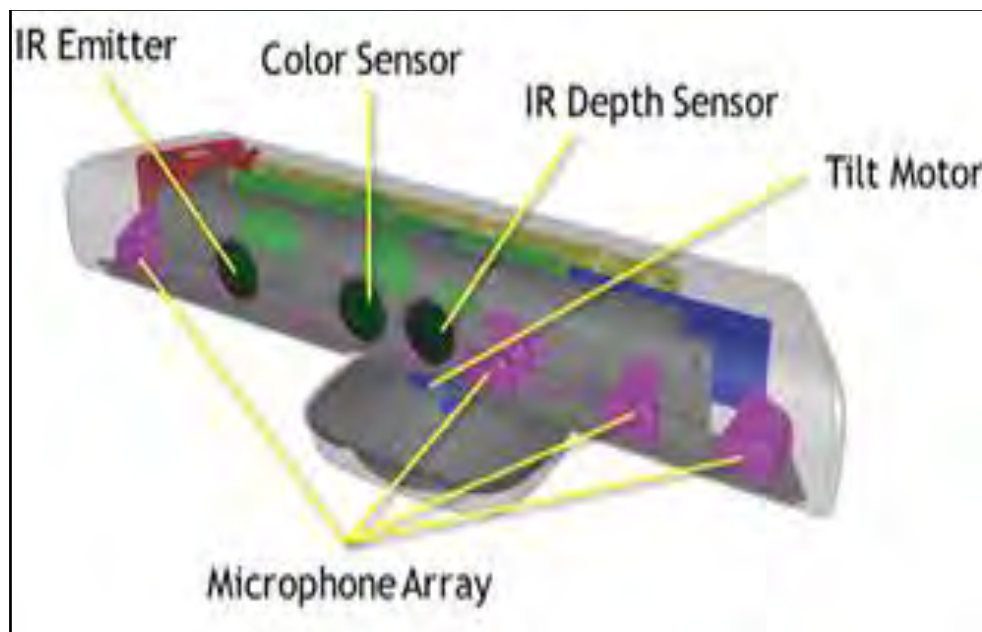


Figure 1: Kinect components [14]

#### 2.1.2 Kinect SDK

Kinect SDK enables developers to create applications in languages like, C++, C# or Visual Basic by using Microsoft Visual Studio (MVS) [12]. It is capable of capturing front body 2D motion, gesture, facial and voice recognition [11], skeletal tracking and advanced audio

capabilities [13]. To track gesture, we need Kinect sensor to record skeletal and depth data and capture the RGB video stream.

Core of the Kinect for the set of Windows API, supports fundamental image and device management features like, access to the Kinect sensors that are connected to the computer; access to image and depth data streams from the Kinect image sensors and delivery of a processed version of image and depth data to support skeletal tracking. Figure 2 shows interaction method of sensor Hardware with software of an application.

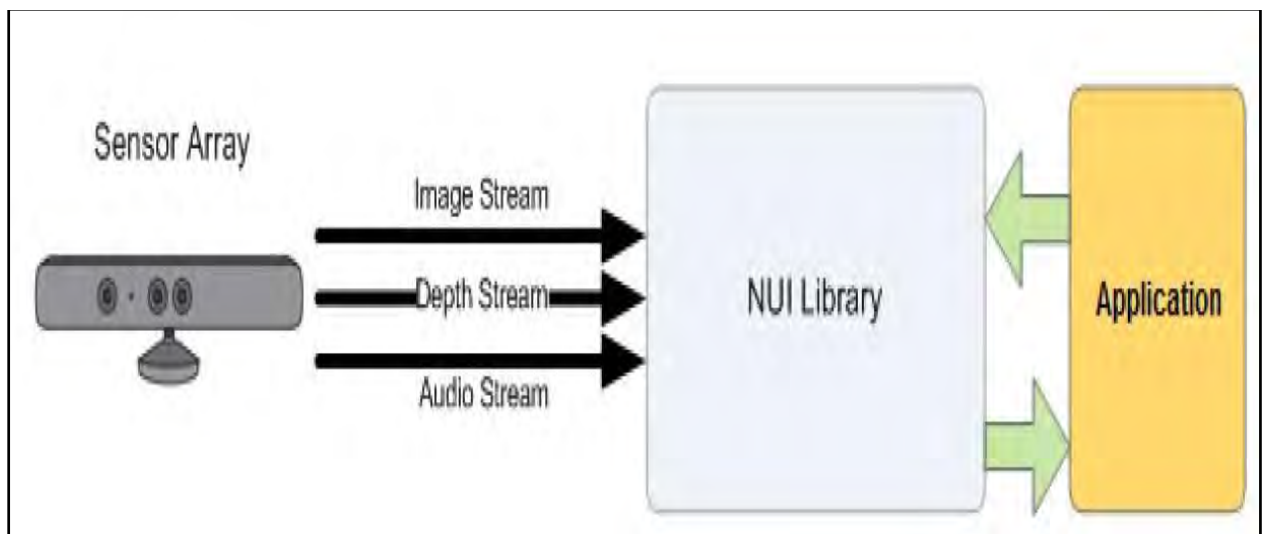


Figure 2: Hardware and software interaction with an application [3]

In addition, the software has the capability to recognize and track human body. The software runtime converts depth data into about 20 skeleton joint points of human body to track up to two persons in front of the camera [14].

### 2.1.3 Kinect Hardware

Including the Kinect sensor and the USB hub, through which the sensor is connected to the computer [15].

### 2.1.4 Kinect Drivers

Microsoft provides Windows (7, 8, 8.1, or 10) drivers for the Kinect sensor [15].

### **2.1.5 Microsoft Visual Studio (IDE)**

Microsoft visual studio (MVS) is an integrated development environment (IDE) from Microsoft. It is an integrated solution which enables the users to develop console and graphical user interface (GUI) applications along with Windows Forms Applications (WFP), web sites, web applications, web services and so on[16].

MVS supports almost all kinds of programming languages including built-in languages such as C, C++ [16] (via Visual C++), VB.NET (via Visual Basic .NET), C# (via Visual C#), and F# (as of Visual Studio 2010) [17]. All of these languages are built on top of the .NET Runtime (known as Common Language Runtime or CLR) and produce the same intermediate output in Microsoft Intermediate Language (MSIL) [19]. Like any other IDE, it includes a code editor that supports syntax highlighting and code completion, using IntelliSense for variables, functions, methods, loops and LINQ queries [20].

IntelliSense is supported for the included languages, as well as for XML and for Cascading Style Sheets and JavaScript when developing websites and web applications [21, 22]. Auto-complete suggestions appear in a modeless list box over the code editor window, in proximity of the editing cursor. In Visual Studio 2008 onwards, it can be made temporarily semi-transparent to see the code obstructed by it [21]. The code editor is used for all supported languages.

### **2.1.6 Camera**

We have to take gestures as individual images for input in the Gesture to Image and Text (GIT) part. Sign gestures can be captured by different types of cameras, like smart phone camera, webcam etc.

### **2.1.7 Display/Screen**

To give the user a mirrored image impression there is a screen or display in front of the user. Data processing has to be accurate and mirroring the real time video essential in order to give the user an impression of mirror.

## 2.2 Related Research

Many research works, related to sign language, have been done. People, who are not deaf, usually do not try to learn the sign language for interacting with those who have hearing difficulties. Finding an experienced and qualified interpreter every time is a very difficult task and not always affordable. This becomes a cause of isolation for the deaf people. However, if a computer can be programmed in such a way that it can translate sign language to text format and vice versa, the gap between the normal people and the deaf community can be minimized.

“A gesture in a sign language is a particular movement of the hands with a specific shape made out of them” [25].

Sign language can provide gesture for letters to perform words that do not have corresponding sign in that language. Though there are papers to recognize such letters, using gesture from the MKS, and forming words combining those letters. A sign language usually provides sign for whole words and papers relating signs to specific words are limited.

Kakde et al. worked on signs recognition and convert to text and speech using web camera for taking inputs, then performing the YCbCr color transformation and Principle Component Analysis (PCA) algorithm for feature extraction [2]. In [3], Ghorpade et al. have a new technique called artificial speaking mouth for dumb people. They have used hand gloves to perform hand gesture, acoustic pre-processing for voice input, Mel Frequency Cepstral Coefficients (MFCC) for feature extraction and Hidden Markov models (HMMs) for speech recognition.

Ear hear is an android application for deaf people to communicate with normal people, which has been worked on by Subhaashini et al. [4]. The Speech-to-Sign and Sign-to-Speech technology are implemented using VRS (Video Relay Service). Similarly, Krishna also designed an android application, which will detect the ISL via mobile camera and convert it into corresponding text or voice output [6]. He has used gesture recognition using video and image processing using histogram method to keep track of the pixels using RGB (red/green/blue) and HSV (hue/saturation/value) for skin color dictation.

Matiwade et al. have developed a hand glove with flex sensor that can interpret English letters, numbers and some Indian sign language (ISL) for the project they have used Gesture, Flex

Sensor, ARM7TDMI, Text-To-Speech Conversion algorithms [5]. While Ahire et al., [7], have explained in their paper how they extracted gesture from real time video and mapped it in human understandable speech. In the second section they have converted natural language into animated sign gesture. They have used frame fragmentation Google Text-To-Speech (TTS) API and Google Speech-To-Text (STT) API in their project.

Archana and Gajanan have proposed two algorithms; (a) Rule based and (b) Dynamic Time Warping (DTW) for ISL word recognition where they have deduced that DTW is better than rule based approach and for sentence creation, they used inverted indexing concept [8]. Shreyashi [9] captured the signs of all 26 alphabets by webcam; pre-processed for feature extraction using HSV color model and compared the obtained features by using the PCA algorithm. For Image segmentation, they used Otsu algorithm and for feature extraction, they used Centroid, skin color and principal component as main feature.

## Design and Implementation

### 3.1 Proposed Model

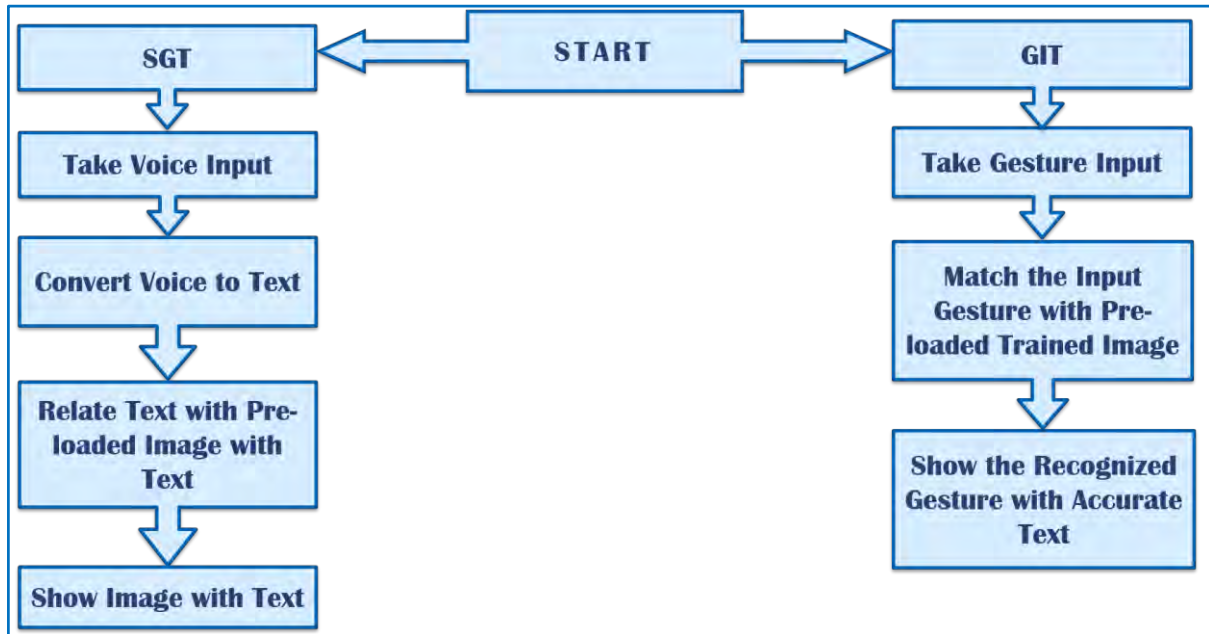


Figure 3: Proposed Model

Left (i. Speech to Gesture and Text)

Right (ii. Gesture to Image and Text)

### 3.2 Model Description

#### 3.2.1 Speech to Gesture and Text

The method, shown in Fig. 3(i), aims to make a normal person able to communicate successfully with a deaf and mute person. A normal human being is capable of giving natural language as input from his side. So, in order to process this natural language, the first step is to generate text data out of audio speech data.

It is easy to process to text data as compared to audio speech signals. There are several platforms those provide us with all the required functionalities and provisions to translate audio speech to corresponding text for further processing.

All the extracted characters from the converted text are then matched with the gesture images that are already stored in data set. Finally, the user will get a sequence of gesture images for the corresponding text, including a subtitle.



We have already tried SGT in C sharp language where we have given the input from a normal microphone. We have used XML for making the UI of the system, speech reorganization engine library for recognizing the speech. Here a sentence can take a maximum of six words and show the gesture along with the subtitle.

However, there are some difficulties we are facing while taking the audio input, because of the background noise sometimes the expected gesture are not showing. To minimize the problem we have decided to use the in-built Kinect microphone. Moreover, one of the challenges that we are willing to take in future work is making a perfect sentence with correct grammar rule. Our current output shows a sentence as “father mother go home” but we will change the structure of the sentence as “Father and mother are going home”. So, basically we will use the to-be verbs, i.e. am, is, was, were etc. in the sentence.

Pseudo code:

```
Initialize KinectSensor, ImageArray, Choices
load kinect
  check connectivity
    initialize images, labels
    recEng_speech_method()
    if click-> speak
      print sequence of images with labels
    clear_AllClick_Method()
    else click-> stop
```

Figure 4: Pseudo code SGT

### 3.2.2 Gesture to Image and Text (Fig. 3(ii))

#### 3.2.2.1 CNN (Convolutional Neural Network)

“A CNN consists of an input and an output layer, as well as multiple hidden layers. The hidden layers are either convolutional, pooling or fully connected” [27].

Neural networks are able to make predictions by learning the relationship between features of the data and some observed response.

In Convolutional (filtering and encoding by transformation) Neural Networks (CNN), every network layer acts as a detection filter for the presence of specific features or patterns present in the original data. The first layers in a CNN detect large features that can be recognized and

interpreted relatively easy. Later layers detect increasingly smaller features that are more abstract and are usually present in many of the larger features detected by earlier layers. The last layer of the CNN is able to make an ultra-specific classification by combining all the specific features detected by the previous layers in the input data.

### 3.2.2.2 Image matching using CNN (Convolutional Neural Network)

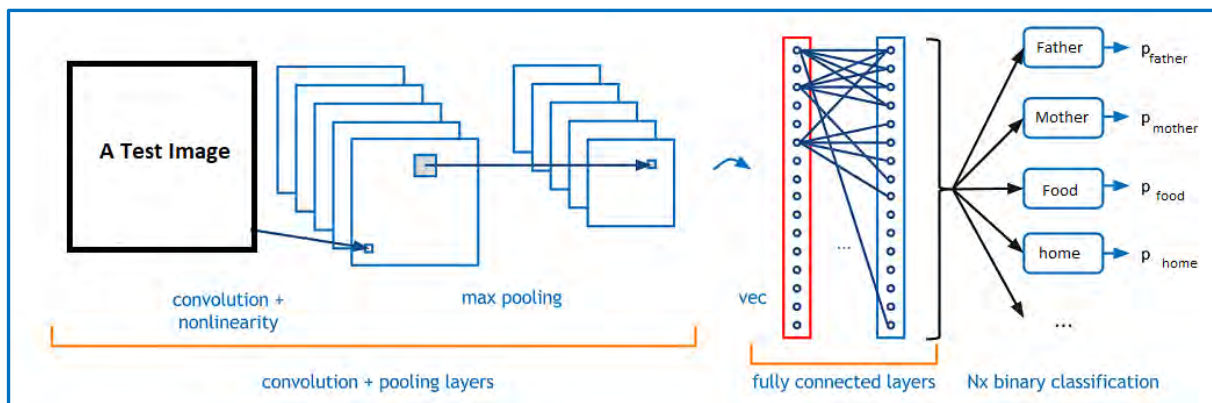


Figure 5: How CNN Works

### 3.2.2.3 Data Augmentation

Due to the small amount of data, data augmentation is used to augment the dataset. Rotation, horizontal flip, width shift range, height shift range, zoom, channel shift have been used for this process.

### 3.2.2.4 Fine-Tune Pretrained ConvNets

In [28], “For image classification pretrained convolutional neural network is used”. TensorFlow and Keras are used to train our model. We have used Inception V3 model pretrained with ImageNet dataset. To train our model, we exclude the last fully connected layer from Inception V3 and include a new fully connected layer which size is the number of our class.

All of the layers of the model were “frozen”, except the last fully connected layer, and then trained for 5 epochs. Then we unfreeze the layers that we froze previously and again train the

model for 24 epochs. The training accuracy of the image matching model is 94.38% and testing accuracy is 95.65%.

### **3.2.2.5 Final Prediction**

“In final result our program gives an output showing which class each test image belongs to and shows the probability of falling into that category” [29].

## Result and Analysis

### 4.1 Dataset

We have collected lots of sign gestures and use those gestures in our database. These sign gestures are stored in the database.

#### 4.1.1 SGT Dataset

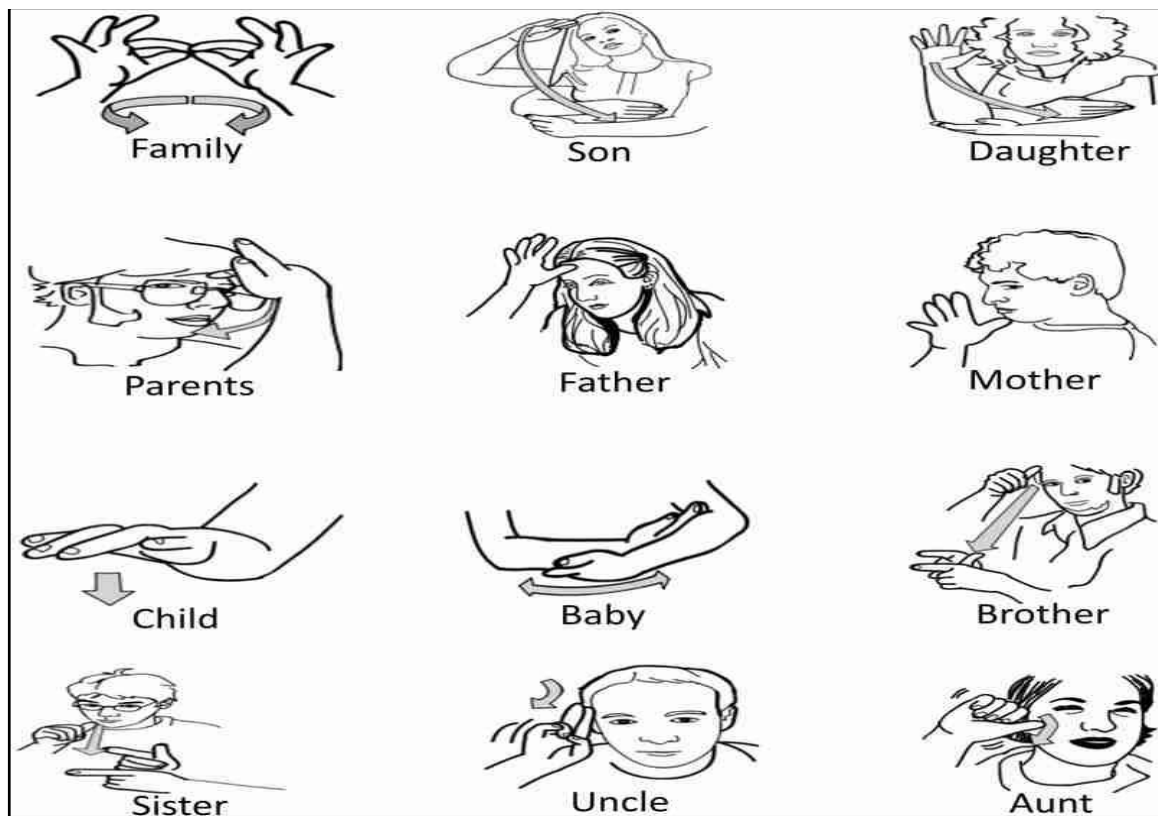


Figure 6: Family – Sign Gesture

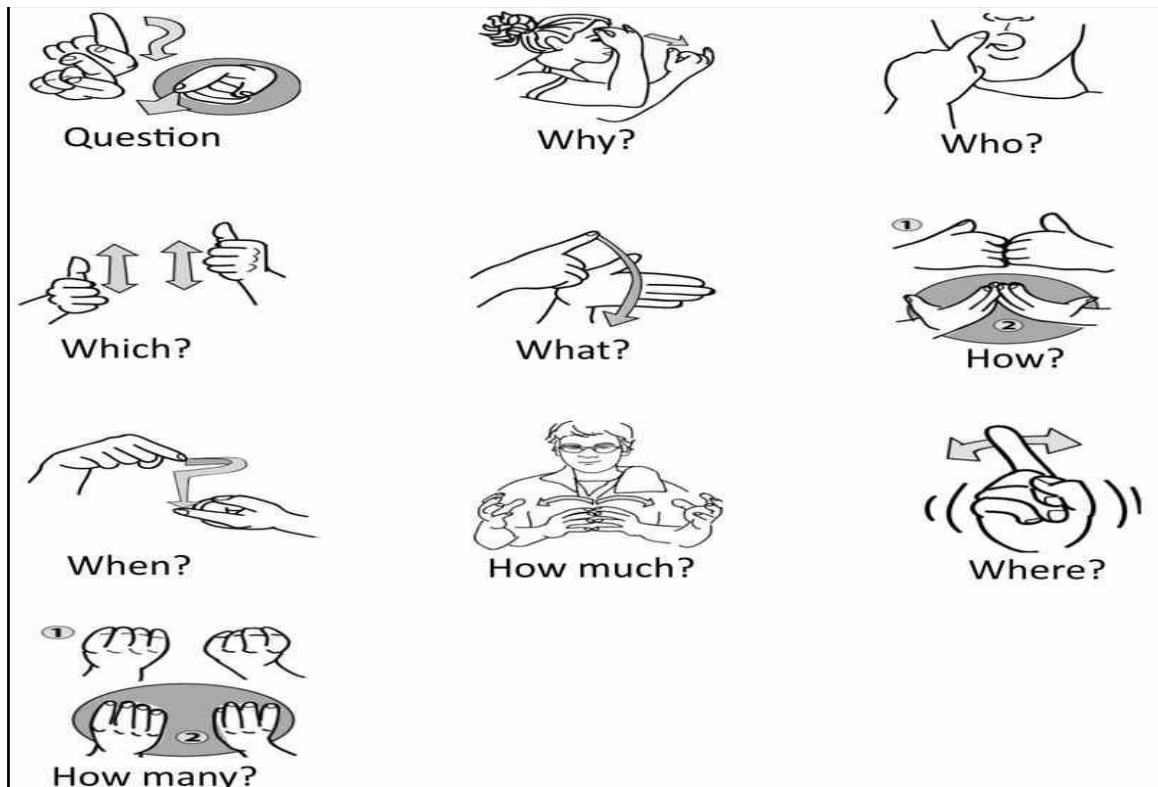


Figure 7: Question – Sign Gesture

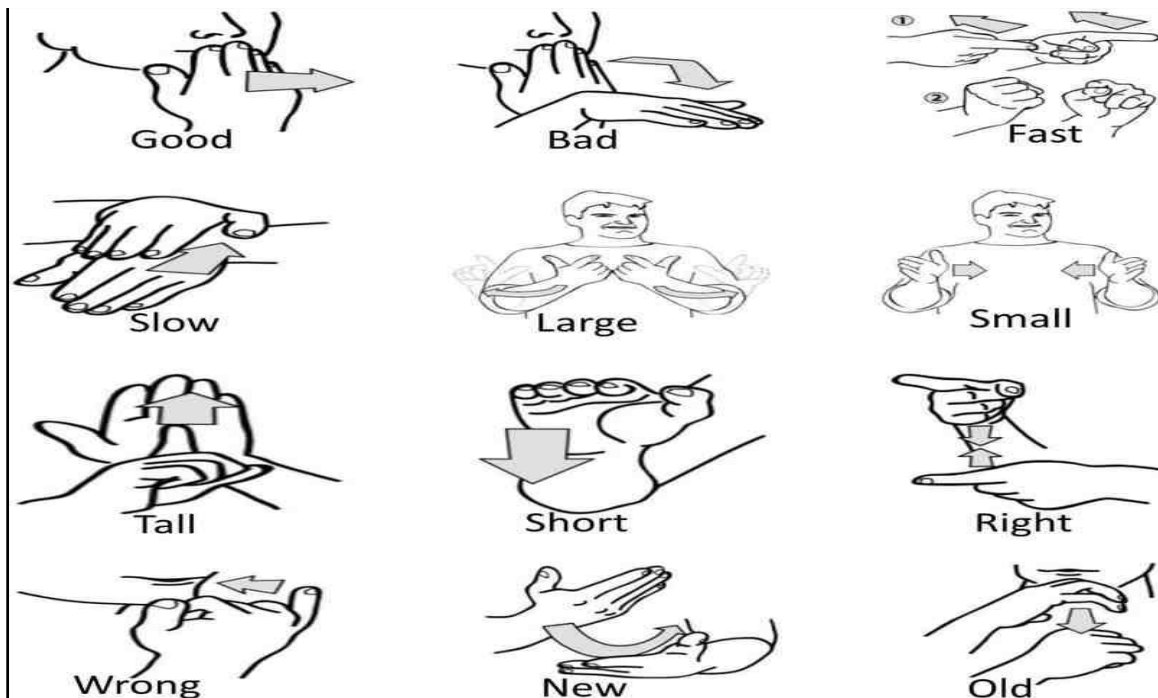


Figure 8: Adjective – Sign Gesture

### 4.1.2 GIT Dataset

On the other half of our project, we have collected some data of various gestures from different people. In our dataset, these people are performing different sign gestures. We have collected the input data with the help of a smart phone camera. Here some people are performing sign gestures in front of the camera, which we take as the dataset for the second part of our project.



Figure 9: Gesture Dataset

Top Left to Right (i. father, ii. home, iii. food, iv. Home)

Bottom Left to Right (v. important, vi. home, vii. mother, viii. Thirsty)

## 4.2 Result

### 4.2.1 Analysis of SGT

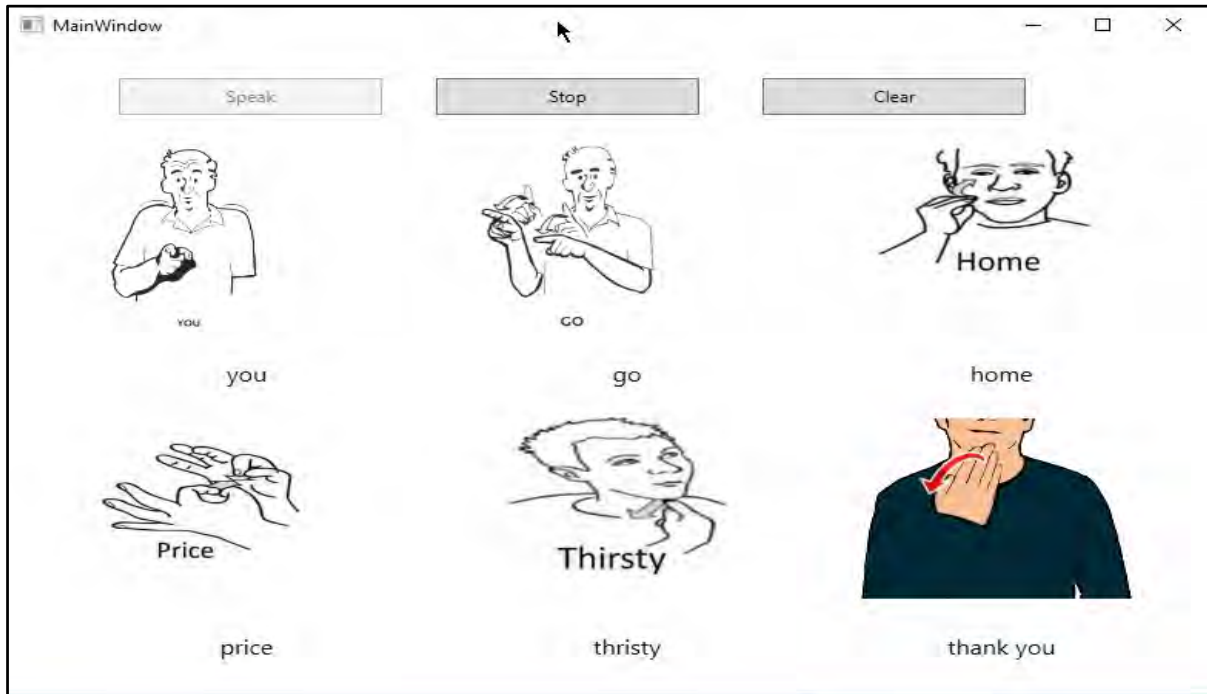


Figure 10: Gesture and Text (i)

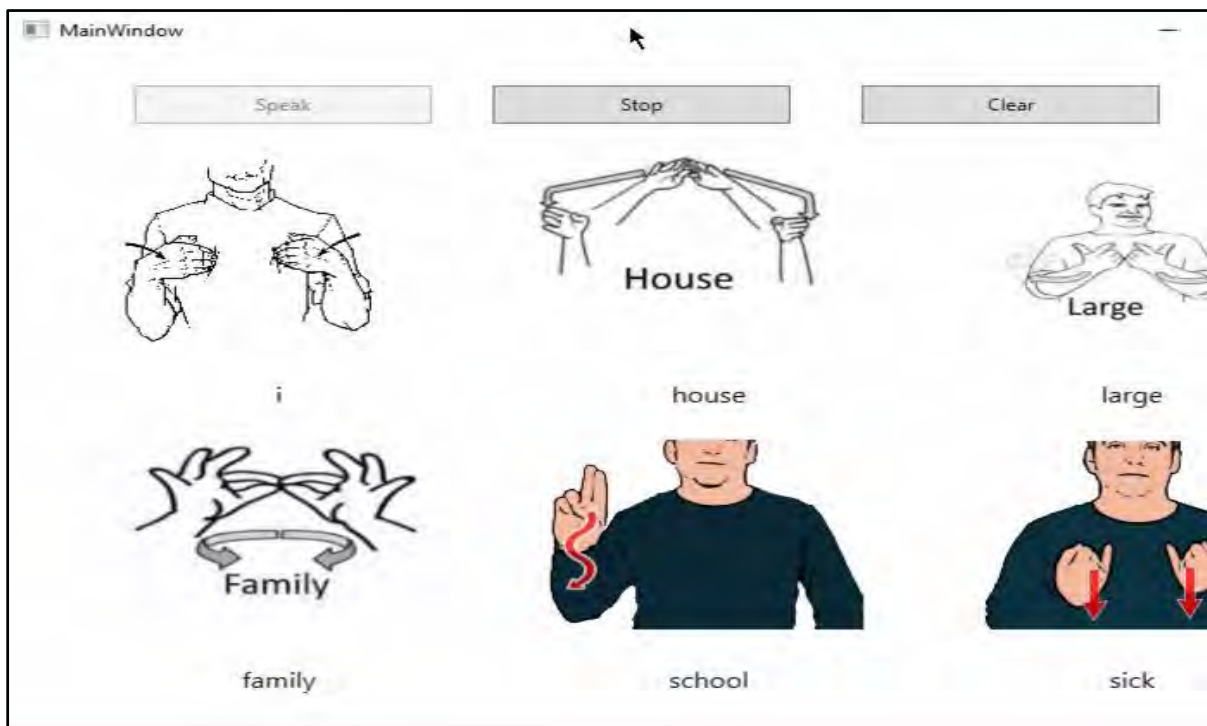


Figure 11: Gesture and Text (ii)

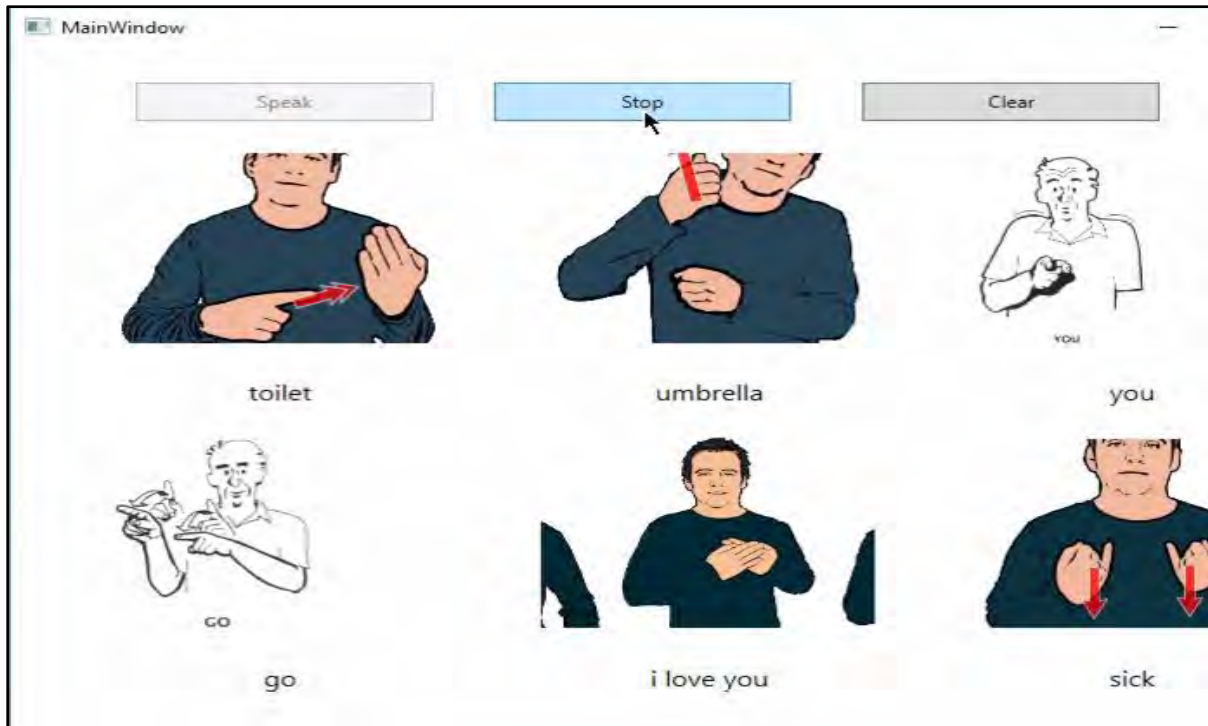


Figure 12: Stop (taking input)

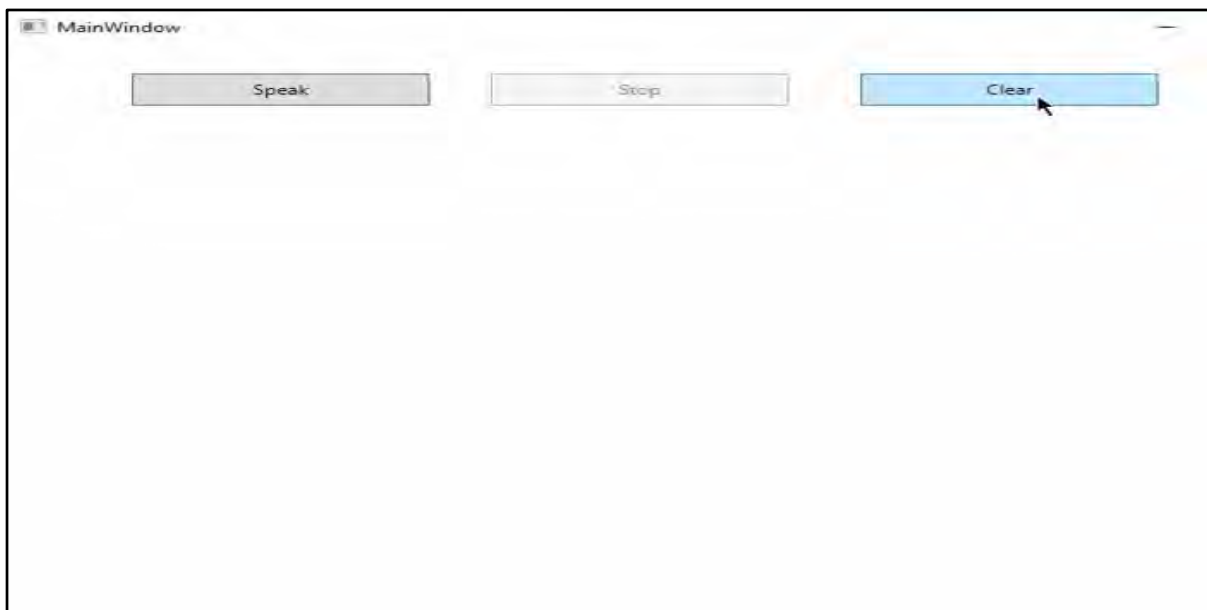


Figure 13: Clear Window

Before starting the procedure of converting Sign to Gesture and Text (SGT), we should go through the following two tests;

- ◆ **Check background noise:** Kinect monitors the ambient noise level in the activity area. If the level is too high, the sensor displays a message asking the user to check for noises



in the room and then to restart the calibration after the user has muted any background noise.

- ◆ **Check speaker volume:** User plays a series of tones for Kinect to hear to calibrate the audio playback level. If the audio tones are too low, a message asks the user to turn up the volume of the speakers. If the sound levels are loud enough, a comment informing that the audio check was successful will be seen.

If the tests are successful, the Kinect sensor's microphone self-calibrates. This is automatic, so the users do not have to do anything during the process. It should take only a few seconds for the self-calibration to complete.

To convert Speech to Gesture and Text (SGT), we access a window as shown in Figure 10.

- **Speak**

When we click the “Speak” button, our voice is taken as input through the Microsoft Kinect Microphone. The words in our voice input are compared with the gesture images preloaded in the database. The matched words are displayed as subtitle along with their respective gesture images above it. Figure 10 and 11 show an example.

- **Stop**

As we press the “Stop” button, the system discontinues taking any further voice input but does not remove the output from the screen as we can see from Figure 12.

- **Clear**

From Figure 13, we can see that as we give the “Clear” command, the screen removes all the previous outputs. We can reuse the technique by clicking the “Start” button again.

## 4.2.2 GIT Analysis



Figure 14: GIT – Home



Figure 15: GIT – Food

### 4.2.2.1 Trained Images

We have trained our algorithm using the model TensorFlow and Keras sensor. In [30], “TensorFlow is an open-source software library for machine learning across a range of tasks, and developed by Google to meet their needs for systems capable of building and training

neural networks to detect and decipher patterns and correlations, analogous to the learning and reasoning which humans use”. TensorFlow is the best way to build deep learning models.

Keras is an open source library neural network written in Python. Designed to enable fast experimentation with deep neural networks, it focuses on being minimal, modular and extensible.

In this segment of our project, we have very carefully trained all the images of same gesture, as shown in Fig. 13, but with different person so that it can remember that how the image is actually defined.

#### **4.2.2.2 Test validation**

Convolutional Neural Network (CNN) tests an input image by breaking it into overlapping image tiles; feed each image tile into a small neural network and save the result from each tile into a new array down sampling. In this research, the computer compares the input with the trained gestures, Fig. 13; identifies it with the help of the trained ones and calculates out how accurate the result is. Here in this portion, we have got a satisfying output after the training.

## **Conclusion and Future Work**

### **5.1 Conclusion**

Our research aims to provide a Two-Way Communicator between People with Hearing-Speaking disabilities and ourselves and also to bridge the interaction gap between these two sections of the society. In Speech to Gesture and Text (SGT), we have used the Kinect microphone and the obtained result is better than that from a normal microphone. On the other hand, in Gesture to Image and Text (GIT), we have used Convolutional Neural Network (CNN) algorithm and the framework TensorFlow and Keras to train images and get the desired output.

### **5.2 Future Work**

We would focus on developing a system on gesture recognition that interprets and converts hand signals to text, i.e. GIT using the Microsoft Kinect System (MKS). Our aim would be to develop such a system, which would take sequence of images that contains SL as input and gives us sentences as output.

Plus, a part of our system is Kinect hardware based and so it might not be affordable or accessible for everybody. Thus, we aim to build a compact system, in the future, so that people can freely access and use it in their smart phone.

## References

- [1] <https://wfdeaf.org/our-work/human-rights-of-the-deaf/>
- [2] Manisha U. Kakde & Amit M. Rawate, "Hand Gesture Recognition System For Deaf And Dumb People Using PCA", *IJESC*, Research Vol. 4, July 2016.
- [3] Shraddha R. Ghorpade & Surendra K. Waghmare, "Full Duplex Communication System For Deaf & Dumb People", *IJETAE*, (ISSN 2250-2459, ISO 9001:2008, Certified Journal Volume, May 2015).
- [4] G.V.S. Subhaashini, S. Divya, S. Divya Suganya & Tharani Vimal, "Ear Hear Android Application for Specially Abled Deaf People", *IJCSEC*, Vol. 3, Issue 3, 2015, Pg.1108-1114 ISSN: 2347-8586.
- [5] Sunita V. Matiwade & DR. M.R. Dixit, "Electronic Support System To Interpret Sign Language Of Communication Used By Deaf And Dumb Community", *JETIR*, Vol. 2, Issue 9 JETIR (ISSN-2349-5162), September 2015.
- [6] Sangeetha, K. and Bharathi Krishna, L., "Gesture Detection For Deaf And Dumb People", *IJDR*, Vol. 4, Issue 3, pp.749- 752, March 2014.
- [7] Prof. Prashant G. Ahire, Kshitija B. Tilekar, Tejaswini A. Jawake & Pramod B. Warale, "Two Way Communicator Between Deaf And Dumb People And Normal People", International Conference on Computing Communication Control and Automation, 2015.
- [8] Archana S. Ghotkar & Gajanan K. Kharate, "Dynamic Hand Gesture Recognition And Novel Sentence Interpretation Algorithm For Indian Sign Language Using Microsoft Kinect Sensor", *Journal Of Pattern Recognition, Research* 1, 24-38, July 2015.
- [9] Shreyashi Narayan Swant, "Sign Language Recognition System To Aid Deaf And Mute People Using PCA", *IJCSET*, ISSN:2229-3345, Vol.5, May 2014.
- [10] <http://www.dmi.unict.it/~battiato/CVision1112/Kinect.pdf>
- [11] <https://msdn.microsoft.com/en-us/library/jj131033.aspx>

- [12] S. Giovanni, Y.C. Choi, J. Huang, E.T. Khoo, K. Yin, "Virtual Try-On Using Kinect And Hd Camera," MIG-2012, vol. 7660, pp. 55-65, 2012.
- [13] <https://msdn.microsoft.com/en-us/library/jj131033.aspx>
- [14] G. Yolcu, S. Kazan, and C. Oz, "Real Time Virtual Mirror Using Kinect," Baikan, Journal of Electrical & Computer Engineering, vol. 2, pp. 75-78, 2014.
- [15] A.B. Habib, Asad, W.B. Omar, "Magic Mirror Using Kinect," BRAC University, 2015.
- [16] <http://home.hit.no/~hansha/documents/microsoft.net/tutorials/introduction%20to%20visual%20studio/Introduction%20to%20Visual%20Studio%20and%20CSharp.pdf>
- [17] Brenner, Pat, "C99 Library Support in Visual Studio 2013", Visual C++ Team Blog., Microsoft, July 2013.
- [18] D. Chai and K. N. Ngan, "Face Segmentation Using Skin-Color Map in Videophone Applications", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 9, no. 4, pp. 551-559, 1999.
- [19] A. Sur, "Visual Studio 2012 and .NET 4.5 Expert Development Cookbook", Vol.1, Chapter 1, "Introduction to Visual Studio IDE Features", 2013.
- [20] Guthrie, Scott. "Nice VS 2008 Code Editing Improvements", July 28, 2007.
- [21] Guthrie, Scott. "VS 2008 JavaScript IntelliSense", June 22, 2007.
- [22] Guthrie, Scott. "VS 2008 Web Designer and CSS Support", July 25, 2007.
- [23] <https://wfdeaf.org/faq/>
- [24] R. Dhanagopal, B. Manivasakam, "Sign Language", GSTF International Journal On Computing, Vol.1, No.2, February 2011.
- [25] Noam Slonim, Ehud Aharoni and, Koby Crammer, "Hartigan's K-means vs. Lloyd's K means – is it time for a change?", IJCAI, 2013.

- [26] Hong, P., Huang, T.S. and Turk, M., “Constructing finite state machines for fast gesture recognition”, Proc. 15th International Conference on Pattern Recognition, Vol. 3, pp.691–694, 2000.
- [27] [https://en.wikipedia.org/wiki/Convolutional\\_neural\\_network](https://en.wikipedia.org/wiki/Convolutional_neural_network)
- [28] <https://github.com/BVLC/caffe/wiki/Model-Zoo>
- [29] [https://en.wikipedia.org/wiki/Probability\\_distribution](https://en.wikipedia.org/wiki/Probability_distribution)
- [30] [https://www.google.com/searchclient=firefox&q=what+is+tenslorflow+sensor&oq=what+is+tenslorflow+sensor&gs\\_l=psy-](https://www.google.com/searchclient=firefox&q=what+is+tenslorflow+sensor&oq=what+is+tenslorflow+sensor&gs_l=psy-)
- [31] [https://en.wikipedia.org/wiki/Probability\\_distribution](https://en.wikipedia.org/wiki/Probability_distribution)