**A**

**Minor Project Report**

on

**Conversion of Sign Language to Speech**

Submitted for partial fulfillment for the degree of

**Bachelor of Technology**

(Information Technology)

in

Department of Information Technology

by

Shubhika Garg

179302148

Sidharth Goenka

179302149

Under the Guidance of

Mrs. Kavita Jhajharia

(June-2020)

**SCHOOL OF COMPUTING AND INFORMATION TECHNOLOGY**

**MANIPAL UNIVERSITY JAIPUR**



**CERTIFICATE**

March 4th,2020

This is to certify that the project titled **Conversion of Sign Language to Speech**is a record of the bonafide work done by **SidharthGoenka**(179302149) and **Shubhika Garg** (179302148) submitted in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology (B. Tech) in **(INFORMATION TECHNOLOGY)** of Manipal University Jaipur, during the academic year 2019-20.

**Mrs. KavitaJhajharia**

*Project Guide, Dept. of INFORMATION TECHNOLOGY*

*Manipal University Jaipur*

**Dr. Pankaj Vyas**

*HOD, Dept. of INFORMATION TECHNOLOGY*

*Manipal University Jaipur*

**ABSTRACT**

One of the major problems that the community of humans is facing is that the people with disabilities find it hard to cope up with the normal people. Many deaf and hard-of-hearing people are unable to communicate with others. Various channels of communication are now available for the benefit of these people with the emerging technologies. Sign language is a non-verbal form available for these people. It is a visual way of expressing by the use of body language, expressions of the face and the movements of the hands.. However, till date not a lot of research has been done in this subject for the betterment of these people. This project will help them to communicate with the world in a better way. The motive of the paper is to convert the human sign language to voice with human gesture understanding and motion capture.

This is effectuated by the use of machine learning which is able to recognize the Sign Language and then convert the predicted letter to text and further to speech. The machine is able to predict the letter based of a database containing different gestures for each letter of the English alphabets and digits. There are a few systems available for sign language to speech conversion but none of them provide natural user interface, one which is easy to use, cheap to implement and also with high percentage of accuracy.

The aim of this project is not only to help the people with but the inability to speak but to provide it in such a way that can be understood and used by anyone, anytime and anywhere. The project with bring with it a user friendly graphical interface(GUI) which will be very easy to operate.

This project will provide them a platform where they can easily express themselves and the people around them will be able understand them very easily irrespective of their knowledge about sign language.

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table No** | **Table Title** | **Page No** |
|  |  |  |
| 1 | Confusion Table Matrix | 10 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No** | **Figure Title** | **Page No** |
|  |  |  |
| 1 | American Sign Language | 2 |
| 2 | Edge detection of an image using Canny | 8 |
| 3 | Hand detection and contour | 8 |
| 4 | Basic layout of the application | 9 |
| 5 | Android application capturing sign | 9 |
|  |  |  |

**CONTENT**

Abstract………………………………………………………………………………. .i

List of Figures……………………………………………………………………….... ii

List of Tables…………………………………………………………………………..iii

**Chapter 01 Introduction**………………………………………………………….......1

* 1. Introduction to work done/Motivation....................................................1
  2. Problem Statement……………………………………………………..1
  3. Aim and Objective of the Research……………………………………2

**Chapter 02 Intelligent Gesture Detection System**………………………………......2

2.1 What is American sign language?……………………………………….....2

2.2Application………………………………………………………………….2

**Chapter 03 Background Overview**………………………………………………......3

3.1 Conceptual Overview…………………………………………………..3

3.2 Technologies Involved…………………………………………………5

**Chapter 04 Methodology**…………………………………………………………......6

4.1 Detailed Methodology……………………………………………….….6

**Chapter 05 Implementation and Results**……………………………………………8

**Chapter 06Future Work and Conclusion**…………………………………………..9

6.1Progress Chart/ Timeline Chart……………………………………….........9

6.2 Future Work and Conclusion…………………………………………..10

**References**…………………………………………………………………………….11

**INTRODUCTION**

* 1. **Introduction to work done/Motivation:**

The key objective of this project is to translate sign language to text and then text to speech for speech-impaired to communicate with other people. Sign language is the most conventional and demonstrative way for the speech impaired people. The project intends to lower the barrier in communication between mute communities and others.

The framework renders assistance for speech-impaired to communicate with the rest of the world using sign language. This leads to the elimination of the mediator who generally acts as a medium of translation. This would contain a user-friendly environment for the user by providing speech output for a sign gesture input.

The projected methodology interprets language into speech. The system overcomes the necessary time difficulties of dumb people and improves their manner.

There exists some latest works on image processing, detecting the hand gesture using the color of the hand and edge detection. Such methods use conversion of image into binary form which can be better understood by a system and then the edges on those pixels are connected to form an outline. Some of the previous works have been done in white background, which is not possible all the time in real life. Maximum works are on hand gloves which are very costly and difficult to carry everywhere. Other techniques like the vision based approach use cameras to capture the image gesture.

The final project can be operated in real-time using relevant image processing technique which is not expensive. Compared to the existing projects this method is simple, compact as well as easy to implement on the go. This system converts the language to text and further to voice that can not only be useful to people with no disabilities but also for the blind people.

**1.2Problem Statement:**

People who lack the sense of hearing and the ability to speak are usually deprived of normal communication with other people in the society. They find it arduous to understand and communicate with them. These people have to rely on an interpreter or on some sort of visual communication. An interpreter is not always available and visualcommunication is mostly difficult to comprehend. These people cannot adapt to the surrounding environment quickly and respond to other normal people.

Sign Language is the primary means of communication in the community of hearing-impaired and mute people. As a normal person is unaware of the grammar or meaning of various gestures that are part of a sign language, it is primarily limited to their families and/or their community. At this emerging era of technology, it is quintessential to make these people feel part of the society by helping them communicate smoothly.

1

**1.3 Aim and Objective of the Research:**

The primary aim of the project is to provide aid for the speech-impaired to communicate with those people who do not understand the sign language. Also this system should be very cost effective as well as easy to implement. The system should not be complicated so that even with a person with least of knowledge can utilize it, which would make is more acceptable to the large audience. The only basic requirement will be the subject to have the knowledge of American Sign Language(ASL).Hence, the objective of this project is to develop a system that is cheap as well as easy to implement and can be used by the masses.

**GESTURE DETECTION SYSTEM**

**2.1 What is American Sign Language?**

American Sign Language (ASL) is a natural language that serves as the leading sign language of the community of the hearing-impaired people in the United States. Apart from North America, localisms of ASL and ASL-based creoles are used in many countries around the world, including much of West Africa and parts of Southeast Asia. ASL is also widely learned as a second language, serving as a lingua franca.

ASL signs have various phonemic components, such as movement of the face, the torso, and the hands. ASL is not a form of pantomime although iconicity plays a larger role in ASL than in spoken languages.  ASL has verbal agreement and aspectual marking and has a productive system of forming agglutinative classifiers.

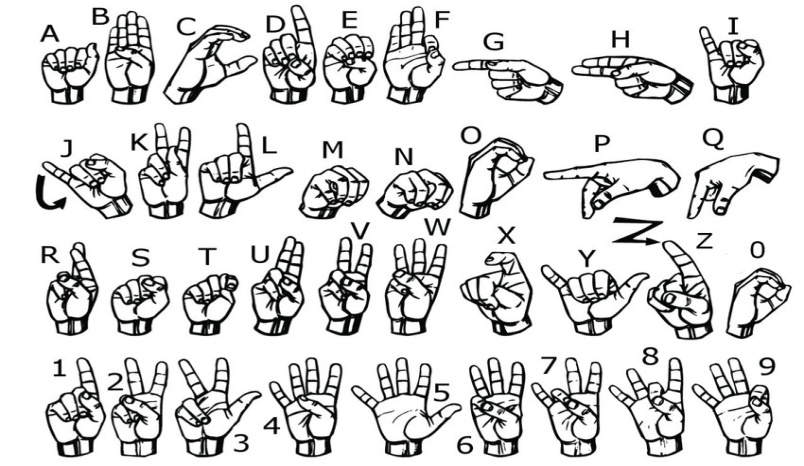
****

Fig 1: American Sign Language

2

**2.2 Application:**

The American Sign Language (ASL) can be used by the speech impaired person to convey what he/she wants to say by the help of gestures. These gestures can be uniquely identified by the person at the other end to understand letter by letter what the person is trying to say. As each of the gesture is different from the other, hence there is no point of confusion.

The person on the other end only needs to have the basic knowledge of the gesture representation of each of the letters of the English alphabets and numbers, and with the help of only this can easily figure out the message being conveyed by the speech impaired person.

**BACKGROUND OVERVIEW**

**3.1 Conceptual Overview**

The project is used to create a trained model with the use of a dataset that is self-created to predict the letters of the English alphabets using gestures as per American Sign Language (ASL). This project uses various machine learning techniques and algorithms to predict with the highest possible accuracy.

Existing database on ASL is used to train the machine so that it can predict the result even when it encounters a data that it has not been trained.

1)**Linear Regression**

Linear regression is amongst the most-simple and easy machine learning algorithm. It is easy to understand and can be applied very easily on a given dataset and get a resultant model which can then be used for similar dataset which the model is not trained for. It basically tries to find a relation between a label and its features. These features of the belonging label are what that helps to predict that particular label. Here we select a label and plot it, then we draw a regression line. The regression line separates the predicted and the actual values. The main aim of the regression line is to maximize the distance between these two values. This distance is known as margin. The more the value of margin, better will be the accuracy of the system.

There are two types of regression: Simple Linear Regression and Multiple Linear Regression.

a*)* ***SIMPLE LINEAR REGRESSION***

It has only one independent variable corresponding to one dependent variable. The relation between two continuous variables can be easily found using simple linear regression. Here we draw a regression line with the equation y= mx + c where m and c are constants. It is 2-D in nature.

The basic aim of the regression line to be drawn in such a way that maximum number of points lie on the line. This can either be fitted or over-fitted. If the dataset is over-fitted, this means that all the data points

3

lie on the regression line and if it is under fitted then less than average number of data points lie on theline. Here we generally try to fit average number data points onto the line for better accuracy. The machine then performs as many number of iteration as the number of points and after each iteration, actual values are compared to the predicted value to find accuracy.

b) ***MULTIPLE LINEAR REGRESSION***

In this type of regression there are multiple independent variable for each dependent variable. Here we try to get an outcome by understanding a linear relationship between both the independent and dependent variables. Since there are multiple features associated to each data point, therefore drawing a linear regression line is not always possible, so we visualize it rather than drawing it. The regression line can be 2-D or 3-D or even 4-D in nature. For this type of regression to work properly there shouldn’t be too much correlation between the independent variables.

The variation or deviation of the predicted value from the actual value is also important for which a term known as coefficient of determination or “R squared” is extensively used in regression analysis. The value of “R squared” increases as the number of variable increases in multiple linear regression.

**2) Canny Edge Detection**

The canny edge detection is to detect out the wide variety of edges from an image. It is carried out by a multi stage algorithm which uses canny filter, which is a multi-stage edge detector. The intensity of the gradient from an image is what is computed on the basis of derivative of Gaussian to filter out the edges. The use of Gaussian filter is to remove the noise from the image and then the potential edges are thinned down to individual pixel curves. In this process the non-maximum pixels are removed. Finally, edge pixels are removed or kept based on hysteresis thresholding.

The general process of edge detection using canny includes:

1. Detecting edges with low error rate.
2. The detected edges should accurately form on the centre of the edge.
3. One edge should only be marked once to avoid overlapping.
4. Image noise should be suppressed so that no false edges are created.

**3) Support Vector Machine (SVM)**

Support Vector Machine (SVM) is a machine learning algorithm. It is used for regression purposes and classifications. The basic working principle of SVM is to draw a hyperplane such that it divides a gives database into two parts. Each of these two parts has a positive and a negative value associated to it.

This hyperplane is created by taking the average distance between two data points that are towards the

4

edges. These data points are called the support vectors. Support vectors are essential parts of SVM algorithm as they are responsible for the orientation of the hyperplane.

Here we try to create a hyperplane in such a way that our data points are in the maximum possible distance from the hyperplane and that each set of the data lie on the correct side of the plane. By doing so

we make sure that when a new data arrives there are high chances of it being on the correct side also. This results in higher accuracy of results

**4) Gradle Build System**

Gradle is an open source system which helps in building an android application. It does all the heavy works such as testing and compiling of the java codes using appropriate tools. As android cannot directly run .java and .xml files, hence they are first needed to be converted to their respective apk file which can then be installed into an android operating system. The gradle system takes all the java files and converts them into .dex files and then compresses all of them into one single apk file. “Build.gradle” is the script that helps with all these tasks.

**3.2 Technologies Involved**

The project uses software tools like Eclipse and Spyder which are program developing platforms for java and python respectively. Eclipse is a simple but powerful java IDE used for java development and it contains various java packages and libraries. Similarly, Spyder helps in python programming as it also contains most libraries such as OpenCV, numpy, pandas. OpenCV stands for open source computer vision and it is used for real time image recognition and machine learning. It has a library of many algorithms which helps the machine predict the object in the image. In this project it is used to recognize hand gestures.

Android Studios is another software tool that is being used in this project. It is a powerful tool that helps with app development using java. It contains all the necessary required libraries such as numpy and OpenCV. Additionally, it provides a GUI based platform for app designing and layout and a very intuitive way of collaborating each element of the app through java code.

Lastly, Google gTTS API has been used to convert resultant text to speech. An API is a piece of code from a software that is open source and are generally available to perform a specific task.

**METHODOLOGY**

**4.1 Detailed Methodology**

As American Sign Language dataset is our basic requirement hence we created one using a simple python code that captures raw images of the gesture of each label such A, B, C and so on. This python code allows the user to decide how many images he wants to capture for each label (more the number of images, higher the accuracy) and the duration after which each label is captured. Once the code starts executing it keep capturing the required number of images and keep storing them into a path provided by

5

the user and keep naming them accordingly. The format of the title of these images are important as they will be used by the machine to understand the similar labels.

After capturing the required number of images, we then divide the images into two very unequal halves, one containing significantly less number of images. For example, if we capture 100 images then we will divide the in a ratio of 85:15. We store these images into two separate folders and name them “train” and “test” respectively. Here, the test images will be used to verify the train images having the same label. This will help the machine predict the alphabet once it recognizes a gesture through OpenCV.

Once the dataset is completely created we will use support vector machine algorithm to create a XML file that will be used by our app for predictions. Each image is converted to its grey image and contour algorithm is applied. This reduces the amount of unwanted features from the image and only the hand gesture part is left out once thresholding mask is applied to it.

At this point only the required part of the image is left out and applying canny edge detection algorithms at this stage results in a very less noise and cleaner edges.

Then the feature points are extracted from the image and plotted on a graph and linear regression is used to separate them by the best possible line which is given by y = mx + c where, m and c are constants.

As the number of images increases and with that the number of feature points also increases. With this increase there comes a point where we can no longer draw a straight line that clearly differentiates two distinct points separately. To solve this problem, we draw another axis vertical to the previous and mark points corresponding to each feature point on the new axis by squaring its value. Now since the plot is in 2D hence we need a plane to separate the two distinct identifiable points. Again when more feature points are added this 2D plane is then converted to a 3D plane, applying the same procedure.

Once the above procedure is completed, the machine has processed all the images associated to each label and extracted all the feature points from them. Using this information, a table can be created for each label that will contain the true positive, true negative, false positive and false negative values.

|  |  |  |
| --- | --- | --- |
|  | *ACTUAL POSITIVES* | *ACTUAL NEGATIVES* |
| ***PREDICTED POSITIVES*** | TRUE POSITIVES(TPs) | FALSE POSITIVES(FPs) |
| ***PREDICTED NEGATIVES*** | FALSE NEGATIVES(FNs) | TRUE NEGATIVES(TNs) |

6

For example,

LABEL: A

No. of images: 20

True Positive: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 16, 17, 20

True Negative: 12, 14, 18

False Positive: 13, 15

False Negative: 19

The numbers associated to the TP, TN, FP and FN are the position of an image of that particular label. This shows us the instances where the machine was correctly able to match one image of test subject to that of the train subject. Lesser the number of TN, FP and FN images, higher will be the accuracy.

This helps us to create a confusion matrix containing each label:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E | F |
| A | 14 | 0 | 0 | 0 | 0 | 0 |
| B | 0 | 13 | 0 | 0 | 0 | 0 |
| C | 0 | 0 | 17 | 0 | 0 | 0 |
| D | 0 | 0 | 0 | 15 | 0 | 0 |
| E | 0 | 0 | 0 | 0 | 14 | 0 |
| F | 0 | 0 | 0 | 0 | 0 | 16 |

This confusion matrix data is stored into and .xml file, that acts as a trained file for the application to use when it encounters new dataset for which it is not trained for.

7

**IMPLEMENTATION AND RESULTS**

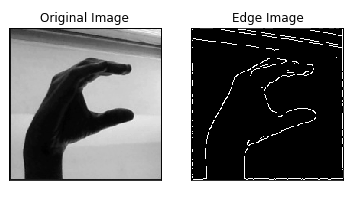
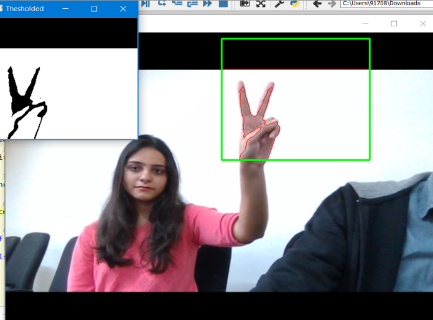
 

Fig 2: Edge detection of an image using Canny Fig 3: Hand detection and contour

8

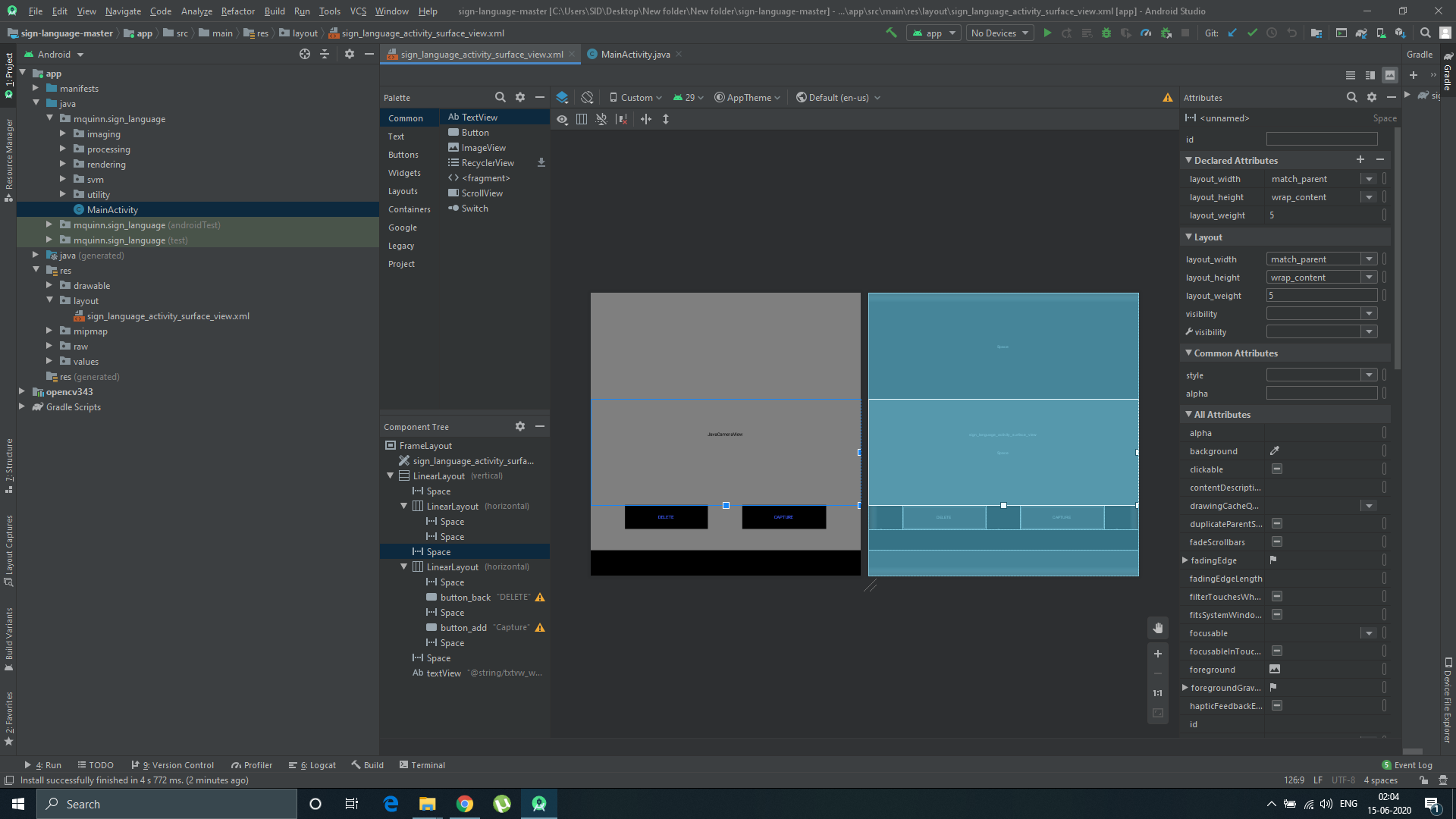
****

Fig 4: Basic layout of the Android application

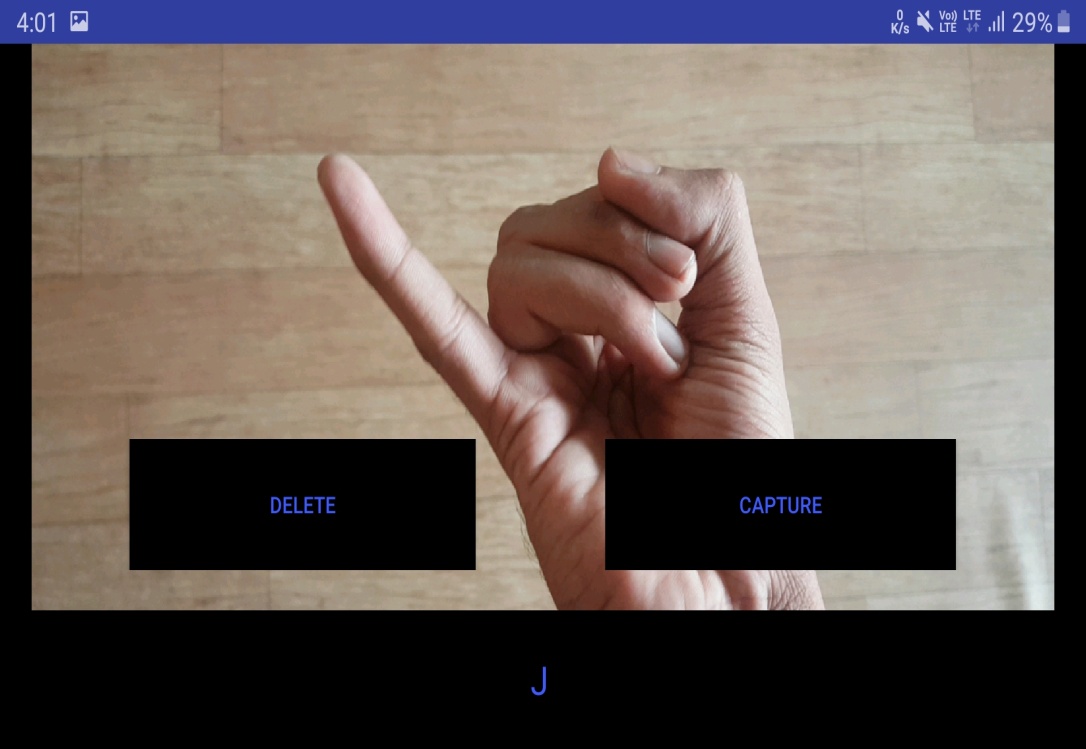


Fig 5: Android application capturing sign

9

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| A | 17 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| B |  | 16 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| C |  |  | 20 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| D |  |  |  | 18 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| E |  |  |  |  | 20 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| F |  |  |  |  |  | 20 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| G |  |  |  |  |  |  | 14 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| H |  |  |  |  |  |  |  | 20 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| I |  |  |  |  |  |  |  |  | 17 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| J |  |  |  |  |  |  |  |  |  | 16 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| K |  |  |  |  |  |  |  |  |  |  | 20 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| L |  |  |  |  |  |  |  |  |  |  |  | 19 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| M |  |  |  |  |  |  |  |  |  |  |  |  | 14 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| N |  |  |  |  |  |  |  |  |  |  |  |  |  | 13 |  |  |  |  |  |  |  |  |  |  |  |  |
| O |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 19 |  |  |  |  |  |  |  |  |  |  |  |
| P |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 17 |  |  |  |  |  |  |  |  |  |  |
| Q |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 18 |  |  |  |  |  |  |  |  |  |
| R |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 19 |  |  |  |  |  |  |  |  |
| S |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 14 |  |  |  |  |  |  |  |
| T |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 14 |  |  |  |  |  |  |
| U |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 19 |  |  |  |  |  |
| V |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 18 |  |  |  |  |
| W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 19 |  |  |  |
| X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 20 |  |  |
| Y |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 20 |  |
| Z |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 20 |

Table 1: Confusion matrix table

**FUTURE WORK AND CONCLUSION**

* 1. **Progress chart / Time line chart**

1. Jan 2020

* Gathering the required knowledge on the project.
* Learning basic concepts of machine learning.
* Collecting the basic knowledge of all the tools required.
* Searching for a suitable ASL dataset.

10

1. Feb 2020

* An appropriate dataset was created and divided into train and test model as required.
* Fetching out the edges of the image in the dataset.
* Text to speech conversion through API was done.
* Learning the basic fundamentals of android studios.
* Started designing the layout of the app.

1. Mar 2020

* Learning how to apply different machine learning algorithms.
* Applying different algorithms on the created dataset to see which one suits the best.
* For each algorithm finding out the best possible accuracy.
* Completed designing the app.

1. April 2020

* Assigning each element of the app to a specific task.
* Studying SVM kernel and its implementation.

1. May 2020

* Training the SVM kernel.
* Completing all the paperwork associated with the project.
* Preparing the final project report and power-point slides.

**5.2 Future Work and Conclusion of our project**

By the use of SVM kernel and various regression techniques, the prediction of various gestures has been possible. The android application further enhances the experience by providing a graphical user interface so that almost anyone with a smartphone can use it.

The application is able to recognize the gesture with good accuracy and be used to understand the message being conveyed by the speech-impaired person.

During the entire process there are various parameters that can be altered which can result in better accuracy from the kernel. Various methods can be adopted as per the user requirements. For example, we used canny edge detection algorithm for the feature extraction of an image, but there are other algorithm what can be implemented very easily which can result in even better accuracy. Even the combination of such algorithms can be implemented by the use if he wishes to do so.

This application is very simple to use and is very cost effective.

11

Minimum hardware requirements are needed to run this application. Even a low spec smartphone can run this application with ease. And as the layout of the application is very simple and there are not many options to tinker around with within the application, hence it is very simple for even a general user who doesn’t have much experience with and smartphone can use this app.

This application can be used in a wide variety of places without too many short-comings. The availability of such an application will bring more confidence to the speech-impaired people as they can now communicate with people around them with even greater ease.

If any further study or research is done on this project, then it is possible the enhance the app experience even further. Additionally, the app can be made more user friendly and more efficient. A separate section can be created in the app that will contain all the gestures from the ASL dataset. Even SVM kernel can be trained to support more than just ASL and the learning curve for the same can be attached into the app itself for anyone to easily understand.

As we are not storing the data collected permanently, this could be another future scope for the app development. The data being recorded by the app can be stored within the app itself which can then be used in future to save time and energy. Even an integrated function can be built that converts the predicted text to voice within the app.

**REFERENCES**

*Journal / Conference Papers*

[1]Kishore, P. V. V., and P. Rajesh Kumar. "Segment, track, extract, recognize and convert sign language videos to voice/text." *(IJACSA) International Journal of Advanced Computer Science and Applications* 3 (2012).

[2]Glenn, Chance M., et al. "An image processing technique for the translation of ASL finger-spelling to digital audio or text." *Instructional Technology and Education of the deaf Symposium, Rochester, NY*. 2005.

[3]Pariwat, Thongpan, and PusadeeSeresangtakul. "Thai finger-spelling sign language recognition using global and local features with SVM." *2017 9th international conference on knowledge and smart technology (KST)*. IEEE, 2017.

*Web*

[1]<https://chrisalbon.com/machine_learning/support_vector_machines/svc_parameters_using_rbf_kernel/> [2]<https://blog.codeonion.com/2016/04/09/show-camera-on-android-app-using-opencv-for-android/>

12

13