# ABSTRACT

Sign language is one of the oldest and most natural form of language for communication, but since most people do not know sign language and interpreters are very difficult to come by we have come up with a real time method using neural networks for fingerspelling based american sign language. In our method, the hand is first passed through a filter and after the filter is applied the hand is passed through a classifier which predicts the class of the hand gestures.Our method provides 82.22 % accuracy for the 24 letters of the alphabet.

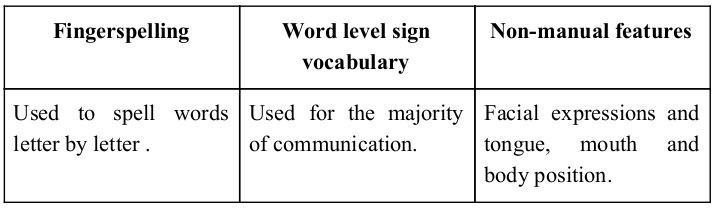
# INTRODUCTION

American sign language is a predominant sign language Since the only disability D&M people have is communication related and they cannot use spoken languages hence the only way for them to communicate is through sign language. Communication is the process of exchange of thoughts and messages in various ways such as speech, signals, behavior and visuals. Deaf and dumb(D&M) people make use of their hands to express different gestures to express their ideas with other people. Gestures are the nonverbally exchanged messages and these gestures are understood with vision. This nonverbal communication of deaf and dumb people is called sign language.

Sign language is a visual language and consists of 3 major components[6]:

In our project we basically focus on producing a model which can recognise Fingerspelling based hand gestures in order to form a complete word by combining each gesture. The gestures we aim to train are as given

in the image below.





In our project we basically focus on producing a model which can recognise Fingerspelling based hand gestures in order to form a complete word by combining each gesture. The gestures we aim to train are as given in the image below.

# PROBLEM STATEMENT:

* Most of the people are not able to understand or interpret the sign language, that lead to lack of communication between normal people and deaf & dumb community.
* People sometimes find it difficult to learn sign language i.e. Standardised Sign Language.
* People need a third person interpreter to communicate with deaf & dumb community. This leads to lack of cohesive bond.

## OBJECTIVES:

* To develop a system which would recognize the signs or gestures made by the deaf and dumb Person.
* To design a web app for which converts meaning of sign whilecommunicating with deaf and dumb person.

# REQUIREMENT ANALYSIS AND SPECIFICATION:

## Information gathering:

* 1. **Vision-Based Sign Language Translation Device** By Yellapu Madhuri1, Anitha.G, Anburajan M, Department of Biomedical Engineering, SRM University, kattankulathur, Tamilnadu, India - Automatic translation of Indian sign language into speech in English to assist the hearing and/or speech impaired people to communicate with hearing people. It could be used as a translator for people that do not understand sign language, avoiding by this way the intervention of an intermediate person and allow communication using their natural way of speaking.
  2. **Hand Gesture Recognition and Voice Conversion for Deaf and Dumb** by Rupesh Prajapati, Vedant Pandey, Nupur Jamindar, Neeraj Yadav, Prof. Neelam Phadnis, Department of Computer Engineering, Shree L.R. Tiwari College of Engineering, Maharashtra, India - Deaf and dumb people cannot communicate with normal person so they have to depend on some sort of visual communication. There are many languages spoken all around the world and interpreted. “Special people”, that is people who have difficulty in speaking and hearing “The dumb” and “The deaf” people respectively find it difficult to understand what exactly the other person is trying to express and so with the deaf people. Sometimes people interpret these messages wrongly either through sign language or through lip reading or lip sync.

# Software Development Model:

1. Incremental Model is a process of software development where requirements are broken down into multiple standalone modules of software development cycle. Incremental development is done in steps from analysis design, implementation, testing/verification, maintenance. Each iteration passes through the requirements, design, coding and testing phases. And each subsequent release of the system adds function to the previous release until all designed functionality has been implemented. The system is put into production when the first increment is delivered. The first increment is often a core product where the basic requirements are addressed, and supplementary features are added in the next increments. Once the core product is analyzed by the client, there is plan development for the next increment.
2. In our project, we are using incremental model. We divided our project into number of steps. First while working in python phase, the iterations of SDLC contains steps for Image data processing, Building Neural Network model, Validating the model’s accuracy, and Saving the model for further use. Each step in each iteration of python phase may or may not require updates in amount of data, optimization for design of Neural Network model; until we meet the desired accuracy. On other side, in android application development phase, the iterations will contain steps like designing app according to requirements, integration of python module into app i.e., implementation, testing of app for skin detection and sign prediction. And as a whole product, incompatibility of code or more requirement leads to go back to python phase in next iterations. And this will continue until the product is complete.

# SOFTWARE REQUIREMENT AND SPECIFICATION:

## PRODUCT PURPOSE:

Most of the people are not able to understand or interpret the sign language, that lead to lack of communication between normal people and deaf & dumb community. People sometimes find it difficult to learn sign language i.e. Standardized Sign Language. People need a third person interpreter to communicate with deaf and dumb community. This leads to lack of cohesive bond and extra man power also. This gives us motivation to develop a system which can interact between deaf & dumb person and normal person.

# PROJECT SCOPE:

“Sign Operated-Alexa” is a web based application which is integrated with the power of technologies such as computer vision and deep neural networks. This system will help normal people to interpret sign language into text instantly with help of web cam. This should reduce communication gap between normal people and dead & dumb community.

## SYSTEM REQUIREMENTS:

### SOFTWARE REQUIREMENTS:

* 1. **CLIENT SIDE:**
     1. Operating System – Windows 10
     2. Language- Python

### HARDWARE REQUIREMENTS:

1. **COMPUTER SYSTEM:**
   1. RAM – minimum 4 GB.
   2. Processor – Intel i3 or above.
   3. Hard disk space – minimum 100 GB.

### DEVELOPMENT TOOLS:

* 1. IDE – Jupyter Notebook.
  2. Language – Python.
  3. Libraries – Keras, Tensorflow , Opencv , Matplotlib, Numpy, Pandas,Seaborn.

## FUNCTIONAL REQUIREMENTS:

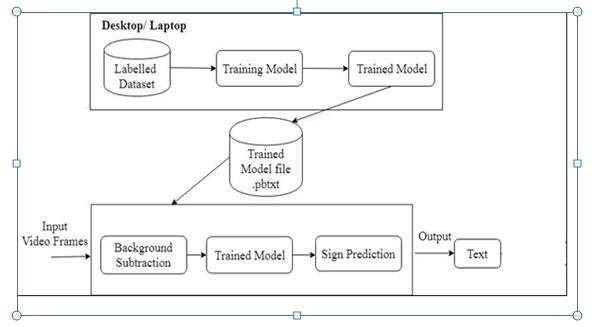
* System should accept hand gestures frames as input.
* System should perform image processing on input such as resizing and converting into array of pixels.
* System should predict the sign by pushing the processed image into neural network deployed into the system.

## NON-FUNCTIONAL REQUIREMENTS:

* System should be reliable. It shouldn’t fail while running and processing task.
* System should be efficient to predict signs with maximum accuracy.
* System should not lag.
* System should work with high performance i.e., throughput. It should be fast enough to predict maximum signs in less time.

# DESIGN:

## SYSTEM ARCHITECTURE:



### Fig. 1 System Architecture

The proposed system architecture has been shown in fig.1. This system will be build for the normal people those which will be communicating with deaf & dumb community . The trained algorithm/trained model at the backend will process the frames of video and recognize the hand gesture made by the disabled person. Further the semantic of signs will be converted to Text.

## Convolution Neural Network :

Unlike regular Neural

Networks, in the layers of CNN, the neurons are arranged in 3 dimensions: width, height, depth. The neurons in a layer will only be connected to a small region of the layer (window size) before it, instead of all of the neurons in a fully-connected manner. Moreover, the final output layer would have dimensions (number of classes), because by the end of the CNN architecture we will reduce the full image into a single vector of class scores.

### TensorFlow :

Tensorflow is an open source software library for numerical computation. First we define the nodes of the computation graph, then inside a session, the actual computation takes place. TensorFlow is widely used in Machine Learning.

### Keras :

Keras is a high-level neural networks library written in python that works as a wrapper to TensorFlow. It is used in cases where we want to quickly build and test the neural network with minimal lines of code. It contains implementations of commonly used neural network elements like layers, objective, activation functions, optimizers, and tools to make working with images and text data easier.

### OpenCV :

OpenCV(Open Source Computer Vision) is an open source library of programming functions used for real-time computer-vision. It is mainly used for image processing, video capture and analysis for features like face and object recognition. It is written in C++ which is its primary interface, however bindings are available for Python, Java, MATLAB/OCTAVE.

### Matplotlib :

Matplotlib is a plotting library available for the Python programming language as a component of NumPy, a big data numerical handling resource. Matplotlib uses an object oriented API to embed plots in Python applications.

### Numpy:

NumPy stands for Numerical Python.NumPy is a Python library used for working with arrays.It also has functions for working in domain of linear algebra, fourier transform, and matrices.NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely.

### Pandas:

Pandas is an open source Python package that is most widely used for data science/data analysis and machine learning tasks. It is built on top of another package named Numpy, which provides support for multi-dimensional arrays. As one of the most popular data wrangling packages, Pandas works well with many other data science modules inside the Python ecosystem, and is typically included in every Python distribution, from those that come with your operating system to commercial vendor distributions like ActiveState’s ActivePython.

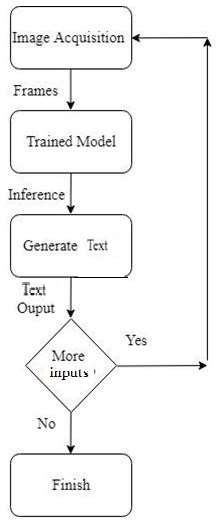
### Seaborn :

**Seaborn** is a data visualization library built on top of matplotlib and closely integrated with pandas data structures in Python. Visualization is the central part of Seaborn which helps in exploration and understanding of data.

## Sign Language Translation:

In this module, Protobuf (.pb) file i.e. binary representation of the trained convolution neural network along with scalar values which represents classes of the images. This protobuf file will be integrated into android app, to which converted/pre-processded frames (into numerical values)when passed will infer the results. At runtime, the video frames captured by android device will be converted into integer array or byte array format. This converted image will be passed to the trained model (.pb) file which in return will return the results in integer value. This integer value corresponds to the class/label of the image which is stored in labels.txt file which will be displayed to the user as a result. These all will lead to outcome of the project.

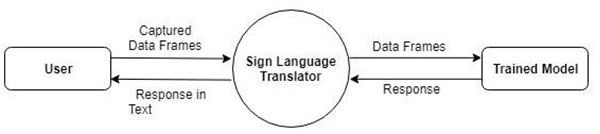
## FLOW CHART:



### Fig. 2 Flow of Sign Language Translation

Fig. 2 depicts the flow chart of system. The initial step is to take images/frames through camera and pass to it to trained model . The trained model will process on gives image and produce an inference. This inference will be in form of alphanumeric which will be converted into voice. If there is more signs to be interpreted continue taking frames and output will generated. System will terminate after exiting the application.

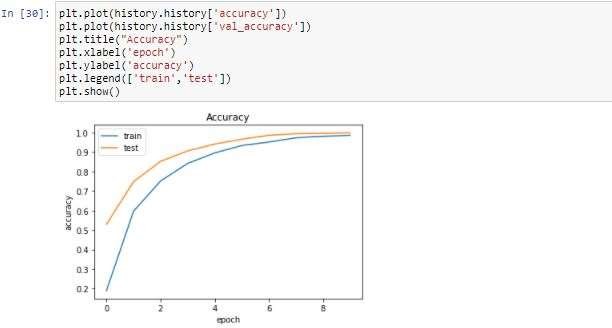
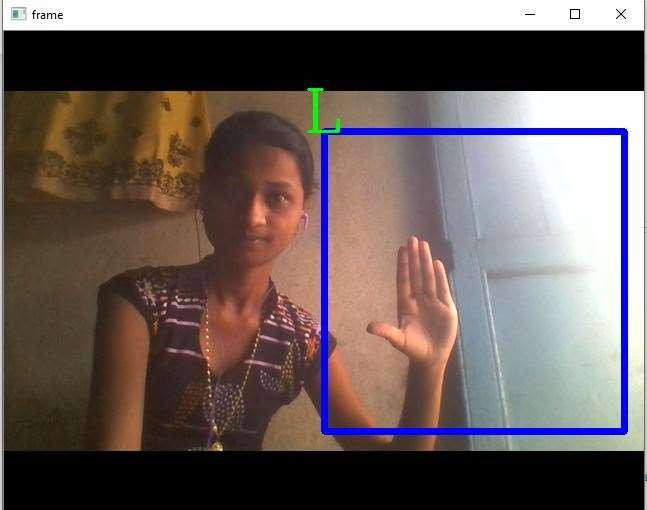
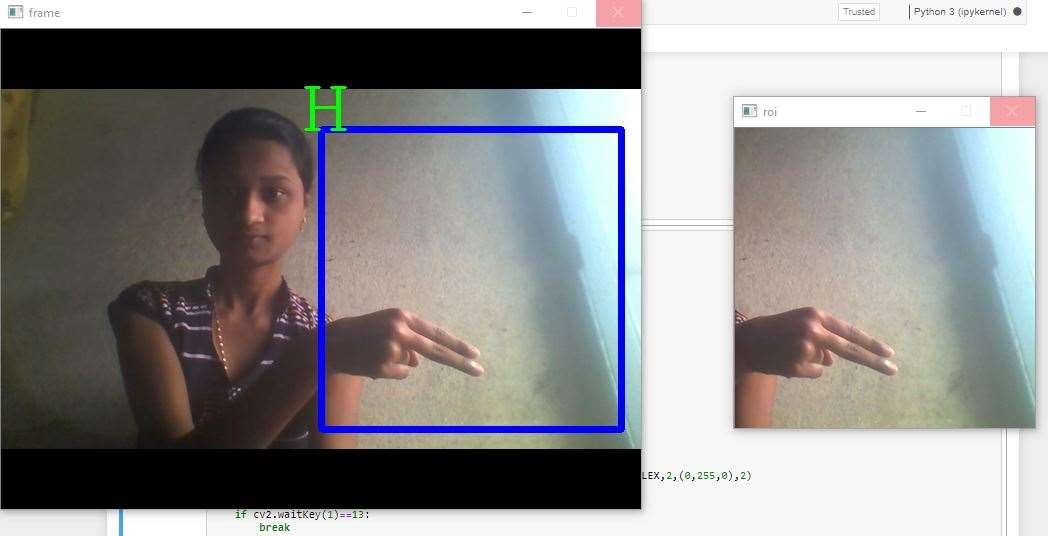
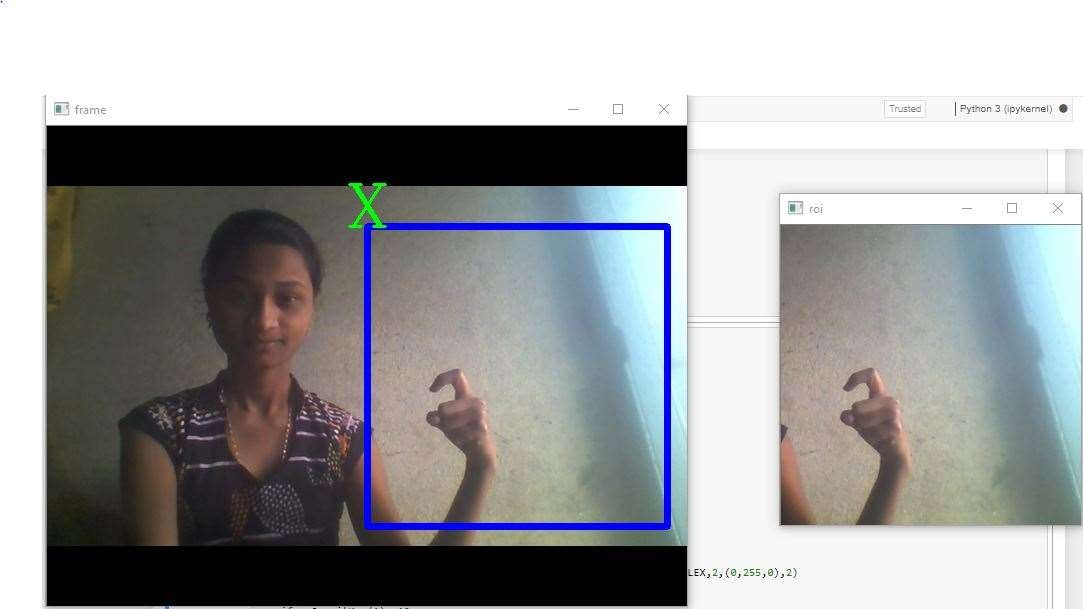
## DATA FLOW DIAGRAM:

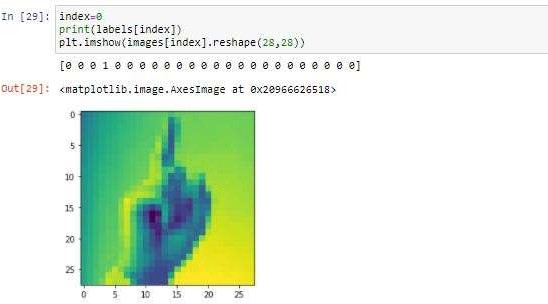
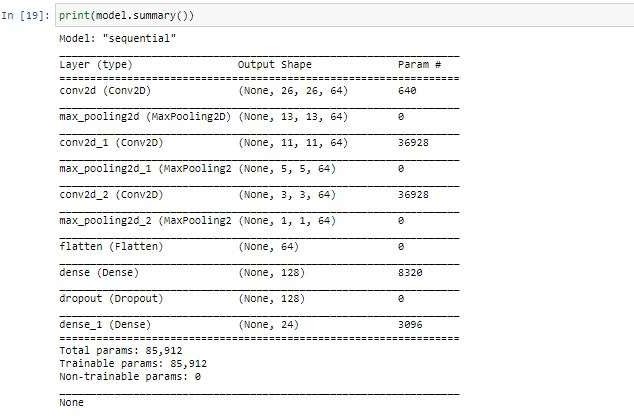


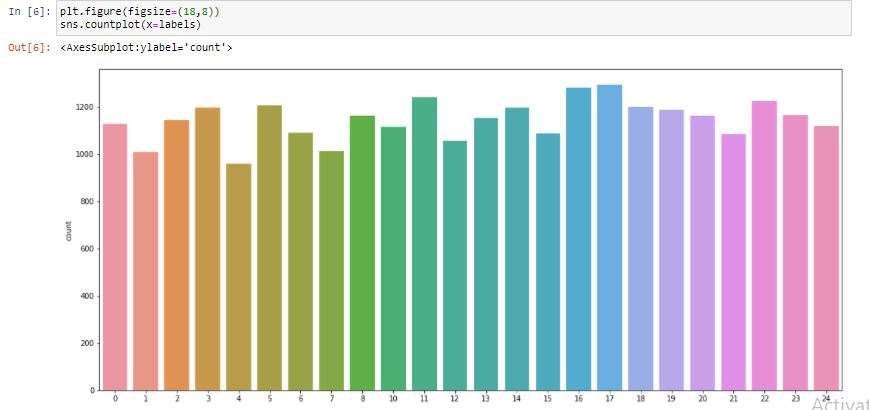
### Fig. 3 DFD of Sign Language Translation

Fig. 3 depicts the data flow for the system. The application takes camera generated frames as an input data to the system. These data frames are passed to the trained model and in reponse to these data frames, the trained model will predict the sign and generate output in string data and given as output to the user.

# IMPLEMENTATION :







## LANGUAGE:

### PYTHON:

We have used python in this project for building Neural Network model, as in past few years’ python has evolved a lot and has its roots into field of machine learning and deep learning due to large repository of libraries available for the work to be done.

## IDE-JUPYTER NOTEBOOK:

Jupyter Notebook is an open-sourced web-based application which allows you to create and share documents containing live code, equations, visualisations, and narrative text. This notebook not only supports Python but also has support for over 40 programming languages. It provides a perfect environment for the data science enthusiast who just started out in their career in this field. This IDE supports markdown and enables you to add HTML components from images to videos. The IDE also includes data cleaning and transformation, numerical simulation, statistical modelling, data visualisation, and many others.

## DATASET- Kaggle Dataset

Kaggle Datasets allows you to publish and share datasets privately or publicly. We provide resources for storing and processing datasets, but there are certain technical specifications:

* 100GB per dataset limit
* 100GB max private datasets (if you exceed this, either make your datasets public or delete unused datasets)
* A max of 50 top-level files (if you have more, use a directory structure and upload an archive)When you upload a dataset we apply certain processing steps to make the dataset more usable.
* A complete archive is created so the dataset can be easily downloaded later
* Any archives (e.g., ZIP files) that you upload are uncompressed so that the files are easily accessible in Notebooks (directory structure is preserved)
* Data types for tabular data files are automatically detected (e.g., geospatial types)
* Column-level metrics are calculated for tabular data which are viewable on the data explorer on the dataset's "Data" tab

When publishing datasets, you might also want to consider the technical specifications of Notebooks if you intend to use (or encourage other Kaggle users to use) Notebooks to analyze the data.

We use data set from GitHub repository Evil Ports and also data set created of our own. The datasetcontains camera snaps of static signs (hand gesture) for example, alphabets A-Z. The overall dataset consists of 900 to 1000 images of each letter, which leads to huge collection and efficiency of results.

### PRE-PROCESSING:

Bottlenecks are created from the images provided. The script can take thirty minutes or more to complete, depending on the speed of your machine. This first phase analyses all the images on disk and calculates and caches the bottleneck values for each of them. 'Bottleneck' is an informal term we often use for the layer just before the final output layer that actually does the classification. (Tensorflow Hub calls this an "image feature vector".) This penultimate layer has been trained to output a set of values that's good enough for the classifier to use to distinguish between all the classes it's been asked to recognize. That means it has to be a meaningful and compact summary of the images, since it has to contain enough information for the classifier to make a good choice in a very small set of values.

## FUTURE SCOPE :

We are planning to achieve higher accuracy even in case of complex backgrounds by trying out various background subtraction algorithms. We are also thinking of improving the preprocessing to predict gestures in low light conditions with a higher accuracy.

## LIMITATIONS:

* The model works well only in good lighting conditions.
* Plain background is needed for the model to detect with accuracy.

# CONCLUSION

Sign Language Translator will act as strong bridge of communication between normal people and deaf & Community. This application will lead people to instantly interpret the sign language without any need third person. This application will help society to communicate the deaf & dumb community and generate a bond between them. We are focusing that this application will help to interact with deaf and dumb person at places such as Shops, Hospitals, Police headquarters etc. Further implementation of our system will enable text and audio support with gesture recognition.

# REFERENCES

## WEBSITES:

* https://ieeexplore.ieee.org/document/7353332/
* https://ieeexplore.ieee.org/document/7449921/
* https://ieeexplore.ieee.org/document/6508395/
* https://[www.handspeak.com/](http://www.handspeak.com/)
* https://docs.opencv.org/2.4/doc/tutorials/tutorials.html
* https://developer.android.com/docs/
* https://[www.tensorflow.org/api\_docs/](http://www.tensorflow.org/api_docs/)
* https://keras.io/