REPORT

ON

"Intelligent Gesture Recognition System for Deaf and Dumb Communication"

Prepared by:
Rupali Taneja - 18csu182
Saumya Achantani- 18csu194
Saumya Gupta - 18csu195

Under the supervision of:

Mrs. Poonam Choudhary
Ms. Priyanka Jain (Industry Mentor)

Department of Computer Science and Engineering
NorthCap University, Gurugram- 122001, India
Session 2020-21

ACKNOWLEDGEMENT

This project has been done as part of our course Introduction to Image Processing and Recognition at The Northcap University. Being extremely interested in everything having a relation to image processing, this group project was a great occasion to give us the time to learn and confirm our interest in this field. In performing our project, we had to take the help and guideline of some respected persons, who deserve our greatest gratitude. The completion of this assignment gives us much pleasure. We would like to show our gratitude to our teacher, **Ms. Poonam Chaudhary** and our **mentor Ms. Priyanka Jain** for giving us a good guideline for the project throughout numerous consultations. Also, she gave us her valuable suggestions and ideas when we required them. Further we want to thank our faculty, **Dr Shaveta Arora** for providing us knowledge about concepts of image processing and recognition.

ABSTRACT

The goal of this project was to build a neural network able to classify which letter of the American Sign Language (ASL) alphabet is being signed, given an image of a signing hand. This goal is further motivated by the isolation that is felt within the deaf community. Loneliness and depression exist in higher rates among the deaf population, especially when they are immersed in a hearing world. Large barriers that profoundly affect life quality stem from the communication disconnect between the deaf and the hearing. Some examples are information deprivation, limitation of social connections, and difficulty integrating in society.

Hand gesture recognition is very significant for human-computer interaction. In this work, we present a novel real-time method for hand gesture recognition. In our framework, the hand region is extracted from the background with the background subtraction method. Then, the palm and fingers are segmented so as to detect and recognize the fingers. Finally, a rule classifier is applied to predict the labels of hand gestures. The experiments on the data set of 1300 images show that our method performs well and is highly efficient.

CONTENTS

1. Objective

Our project aims to bridge the gap between the speech and hearing-impaired people and the normal people. 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.

The project uses an image processing system to identify, especially American alphabet sign language used by the deaf people to communicate and converts them into text so that normal people can understand. Everyone deserves to have their voice heard, no matter the circumstances.

With this being said, sign language is a two-way street. The deaf community needs to know it, along with everyone else around them. They must live in an environment and world that they feel can hear them and what they are trying to say at all times, and to be able to express their feelings, contribute to a conversation, learn, and overall live their lives like other people. Sign languages are the native languages of the Deaf community and provide full access to communication to them. Although sign languages are used primarily by people who are deaf, they are also used by others, such as people who can hear but can't speak.

2. Introduction

Sign language is a visual way of communicating where someone uses hand gestures and movements, body language and facial expressions to communicate. It is a language that provides visual communication and allows individuals with hearing or speech impairments to communicate with each other or with other individuals in the community. . Communication is the process of exchange of thoughts and messages in various ways such as speech, signals, behaviour, 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.

According to the World Health Organization, the number of hearing-impaired individuals has recently reached 400 million. American Sign Language (ASL) is a natural language that serves as the predominant sign language of Deaf communities. The first written document of sign language has come from Plato ,he recorded Socrates saying: If we had neither voice nor tongue, and yet wished to manifest things to one another, should we not, like those which are at present mute, endeavour to signify our meaning by our hands, head, and other parts of the body. ASL Substantially facilitates communication in the deaf community. But there are only ~ 250,000-500,000 speakers which understand it (source: Wikipedia), hence limiting the number of people that they can easily communicate with. A real-time sign language is an important milestone in facilitating communication between the deaf community and the general public. In order to diminish the obstacle and to enable dynamic communication, we present an ASL recognition system that uses Convolutional Neural Networks (CNN) in real time to translate a video of a user's ASL signs into text.

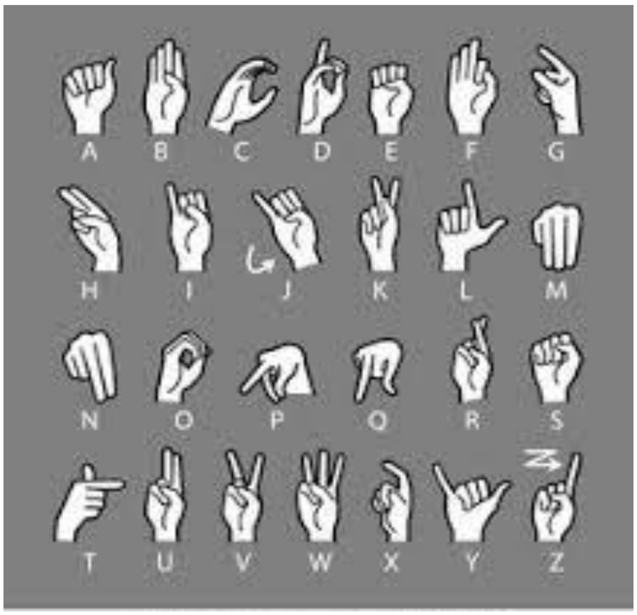


Fig. 1: The American manual alphabet

Fig 2: Quote in sign Language

3. Motivation

For interaction between normal people and D&M people, a language barrier is created as sign language structure is different from normal text. So, they depend on vision-based communication for interaction.

The proposed system is motivated by the isolation that is felt within the deaf community. Large barriers that profoundly affect life quality stem from the communication disconnect between the deaf and the hearing. Some examples are information deprivation, limitation of social connections, and difficulty integrating in society.

If there exists a common interface that converts the sign language to text, the gestures can be easily understood by the other people.

The aim is to develop a user-friendly human computer interface (HCI) where the computer understands the human sign language. Loneliness and depression exists in higher rates among the deaf population, hence there should be some way to keep them motivated. For this, an option of meditation is provided in American sign language.

Further, in Covid-19 times, creating awareness among all the people is of prior importance. Hence another section of covid 19 guidelines is provided.

4. Dataset Description

4.1. Dataset Creation

Steps we followed to create our dataset are as follows:
We used the Open computer vision(OpenCV) library in order to produce our dataset. Firstly, we captured around 250 images of each of the symbols in ASL for training purposes and around 170 images per symbol for testing purposes. First, we capture each frame shown by the webcam of our machine. In each frame we define a region of interest (ROI) which is denoted by a blue bounded square as shown in the image below.

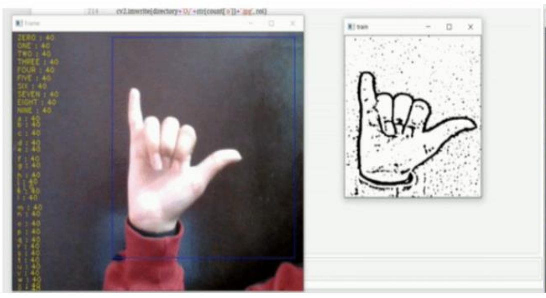


Fig 3: ASL letter "Y"

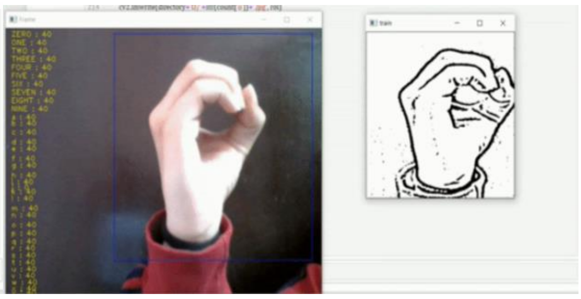


Fig 4: ASL letter "O"

From this whole image we extract our ROI which is RGB and convert it into Gray scale Image as in Fig. 5.

Finally, we apply our gaussian blur filter to our image which helps us extract various features of our image. The image after applying gaussian blur looks like Fig. 6.

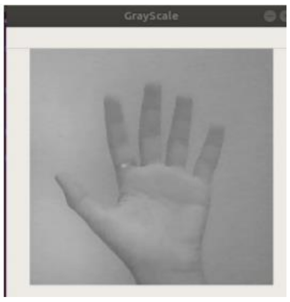


Fig5: Gray scaled Image

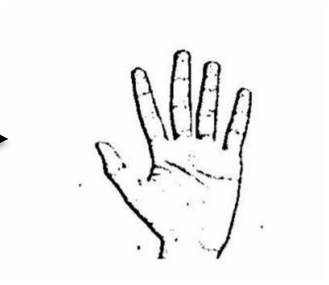


Fig.6: feature extracted.

4.2 Directory Structure

The dataset consists of two folders - train and test. The train folder further includes the sub directories of each alphabet. Each sub directory has 250 images.

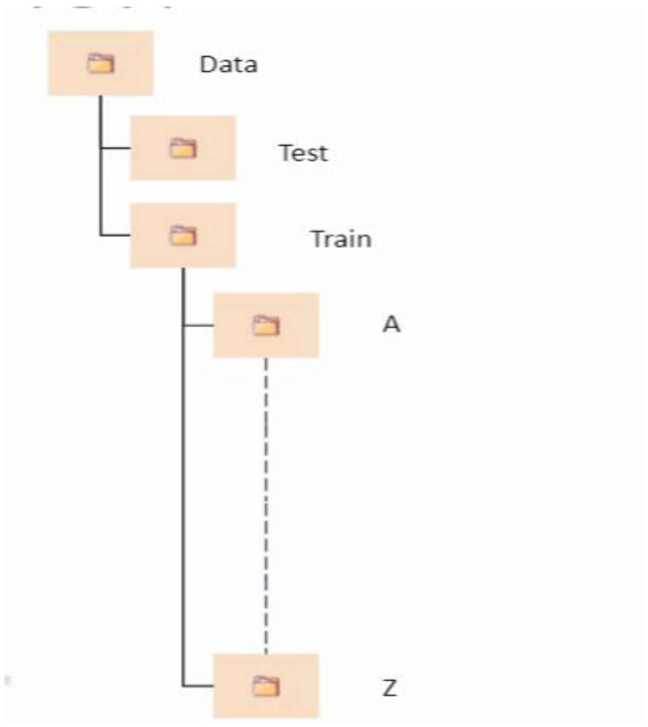


Fig 7: Directory Structure

5. Methodology

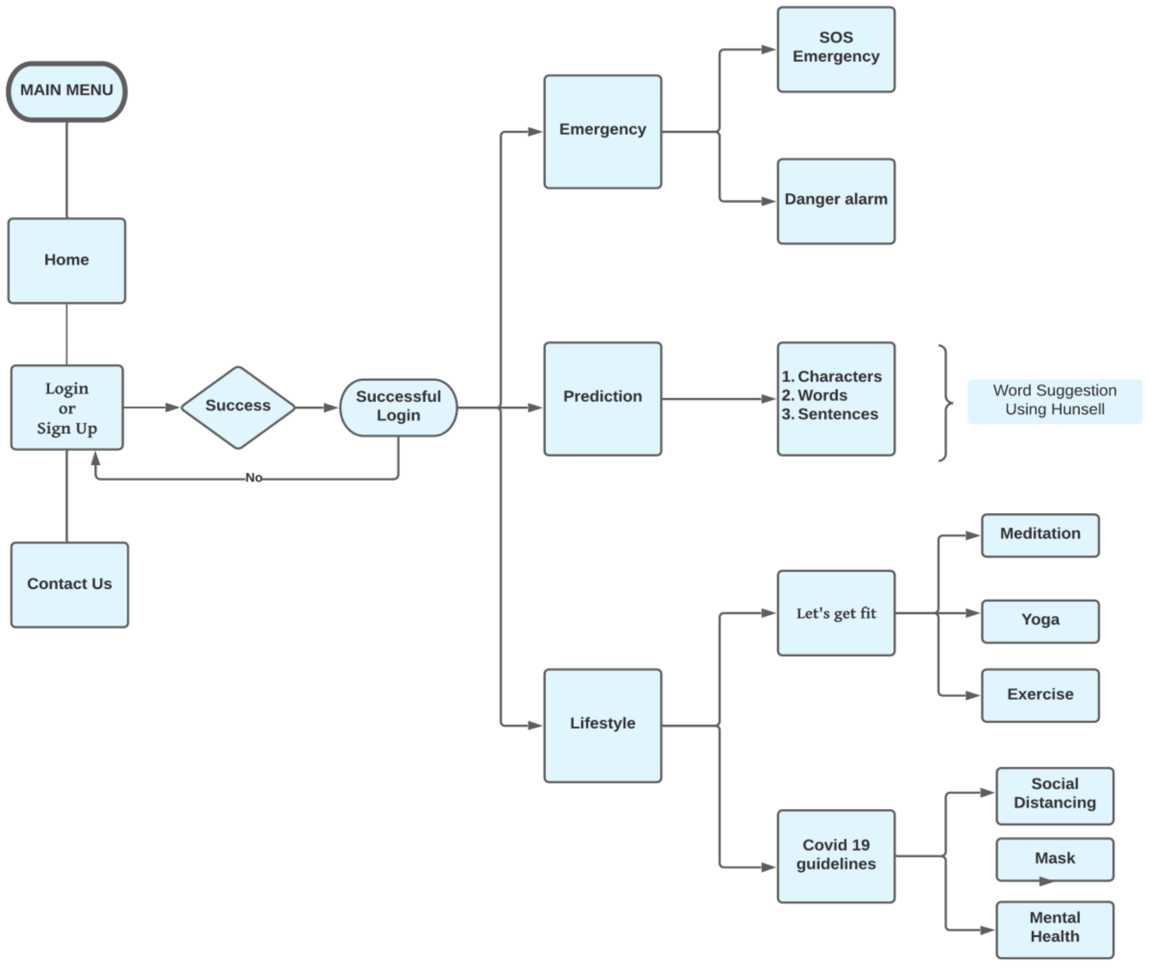


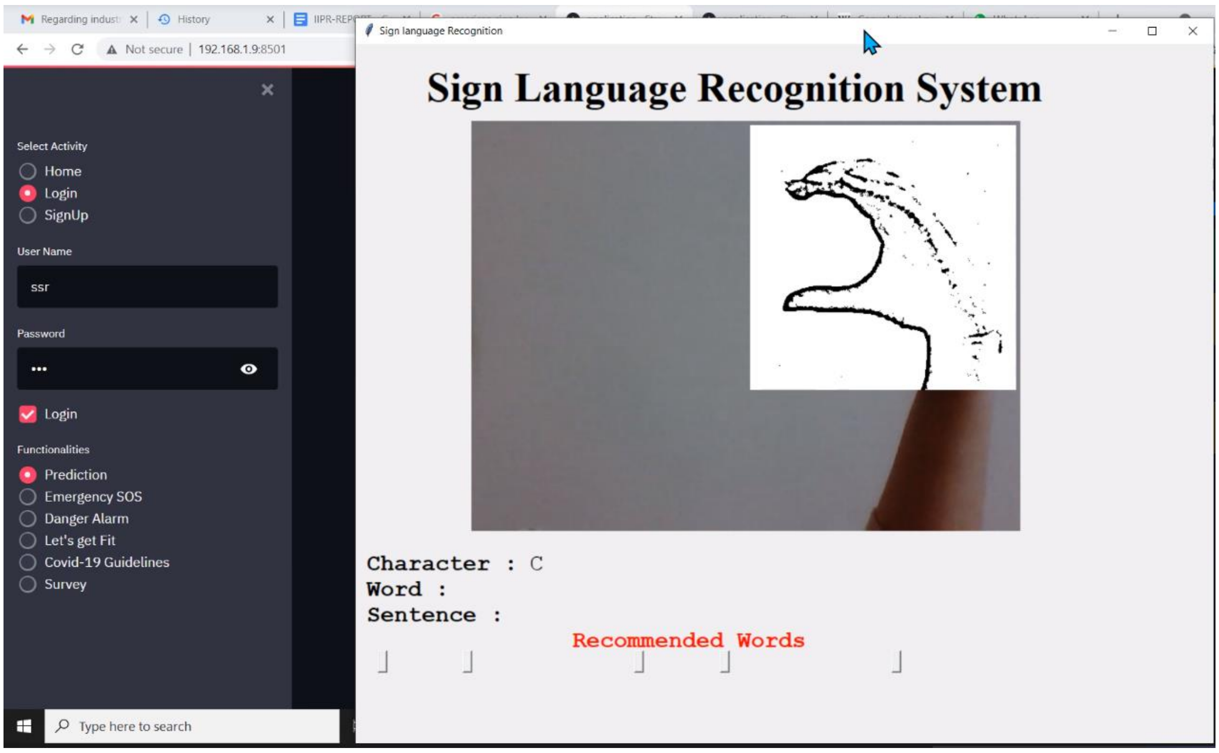
Fig 8: Project Framework

5.1. Functionalities:

5.1.1. **Prediction:** It consists of three tasks to be done in real time:

- Obtaining video of the user sign (input).
- Classifying each frame in the video to a letter.
- Reconstructing and displaying the most likely word from classification scores (output).

If a blank is given after predicting character, it results in word formation followed by sentence formation. The feature of word suggestion is given using Hun spell. Hun spell is a spell checker and morphological analyser designed for languages with rich morphology and complex word compounding and character encoding, originally designed for the Hungarian language. Hun spell is based on MySpell and is backward-compatible with MySpell dictionaries.



Sign Language Recognition System

User Name: ssr

Password: •••

[✓ Login]

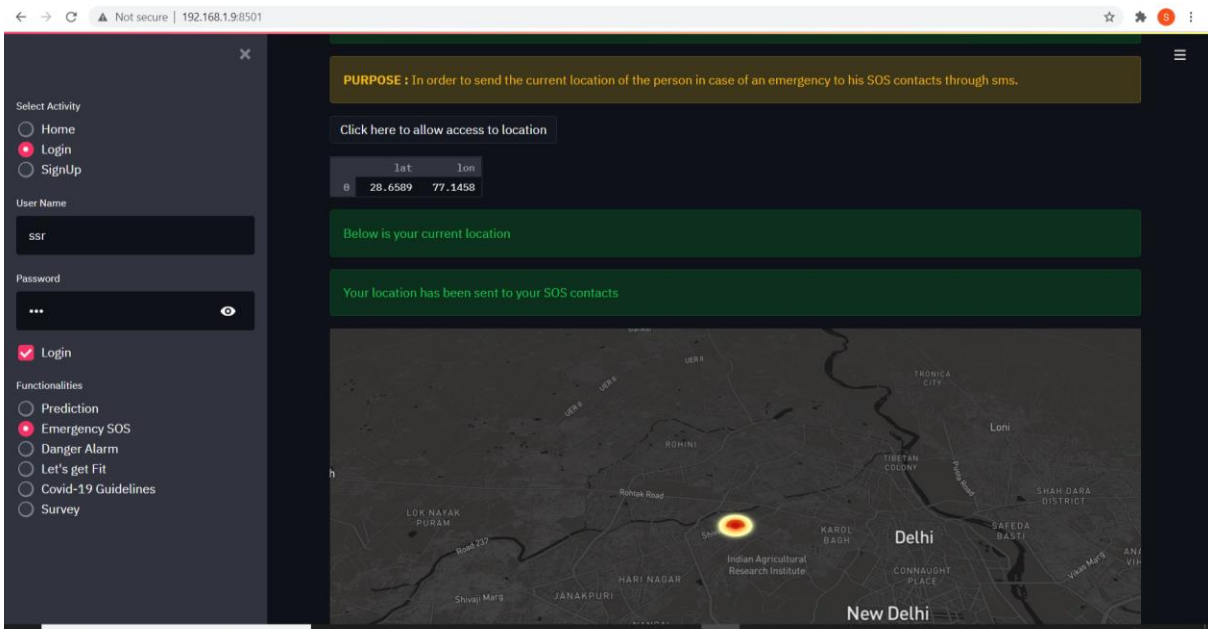
Character : C
Word :
Sentence :

Fig 9: Sign Language to Text Translator

5.1.2. SOS Emergency:

This option facilitates the user to send his/her location as SMS in case of an emergency. www.location.net is the website used to send the current location of the person attached with the SMS to the SOS contact, given by the user at time of sign-up.

Web driver and XPath is a technique in Selenium to navigate through the HTML. structure of a page. Using this, we extracted the latitude, longitude and corresponding current location of the browser and store it to produce a map.



Not secure | 192.168.1.9:8501

Select Activity
Home
[✓ Login]
SignUp

User Name: ssr

Password: •••

[✓ Login]

Functionalities
Prediction
[✓ Emergency SOS]
Danger Alarm
Let's get Fit
Covid-19 Guidelines
Survey

PURPOSE : In order to send the current location of the person in case of an emergency to his SOS contacts through sms.

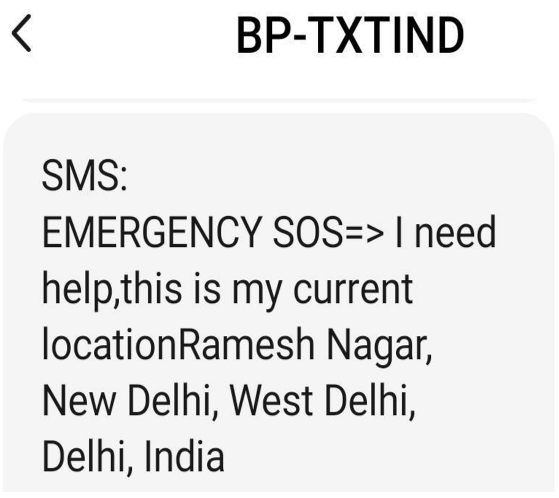
Click here to allow access to location

```
     lat      lon
0  28.6589  77.1458
```

Below is your current location

Your location has been sent to your SOS contacts

Fig 10.a): Emergency SOS current location of user



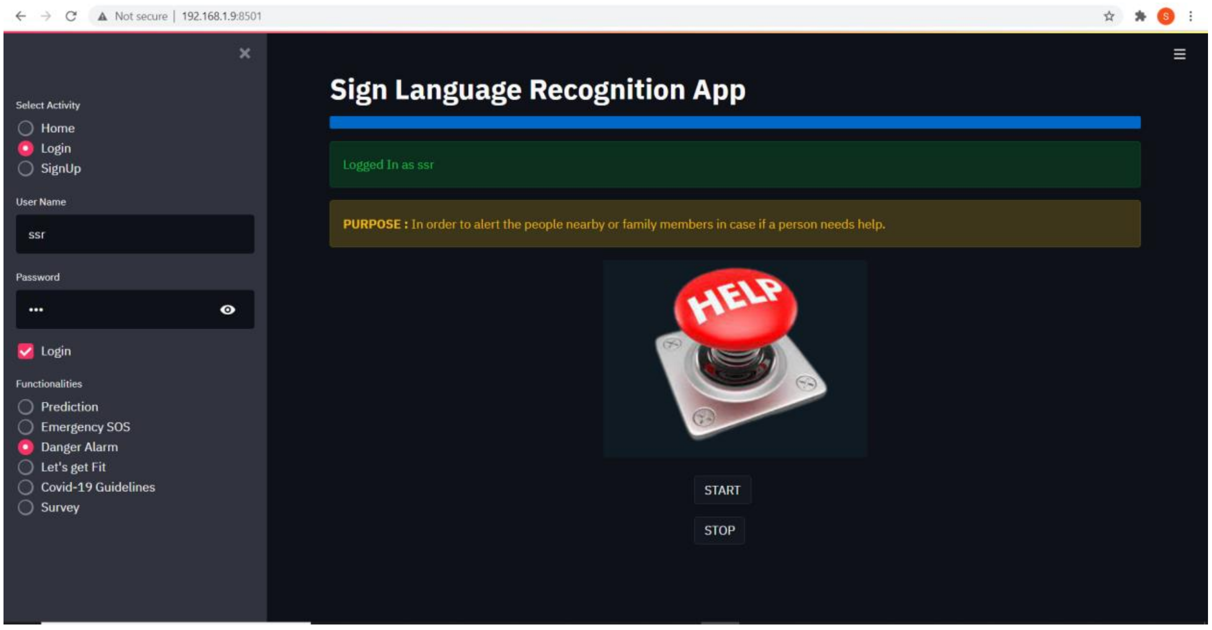
BP-TXTIND

SMS:
EMERGENCY SOS=> I need help,this is my current locationRamesh Nagar, New Delhi, West Delhi, Delhi, India

Fig 10.b) Message send to user via SOS.

5.1.3. Danger Alarm:

Using this, users can alert nearby people in case of an emergency. Buttons are provided to start and stop the alarm. The library used for the same is pygame. First the pygame and a module called mixer from pygame is initialized. pygame. init () initializes the library. mixer is a library, which helps to run audio in the game or any pygame application. As most of the games include audio, you should also initialize the mixer.



Sign Language Recognition App

User Name: ssr

Password: •••

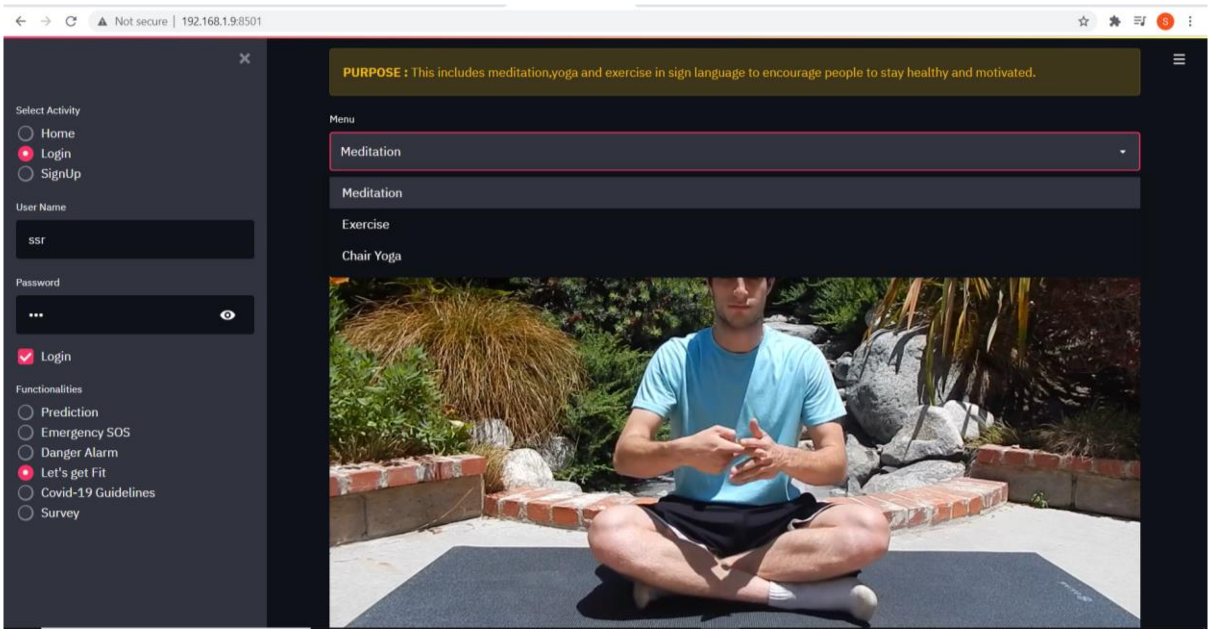
Logged In as ssr

PURPOSE : In order to alert the people nearby or family members in case if a person needs help.

Fig 11. Danger Alarm

5.1.4. Let's Get Fit:

It includes meditation videos to encourage especially abled people to avoid. stress and depression. Further, yoga and exercise videos are also provided to maintain health and stay fit.



Not secure | 192.168.1.9:8501

PURPOSE : This includes meditation,yoga and exercise in sign language to encourage people to stay healthy and motivated.

Select Activity
- Home
- [✓ Login]
- SignUp

User Name: ssr

Password: •••

[✓ Login]

Functionalities
- Prediction
- Emergency SOS
- Danger Alarm
- [✓ Let's get Fit]
- Covid-19 Guidelines
- Survey

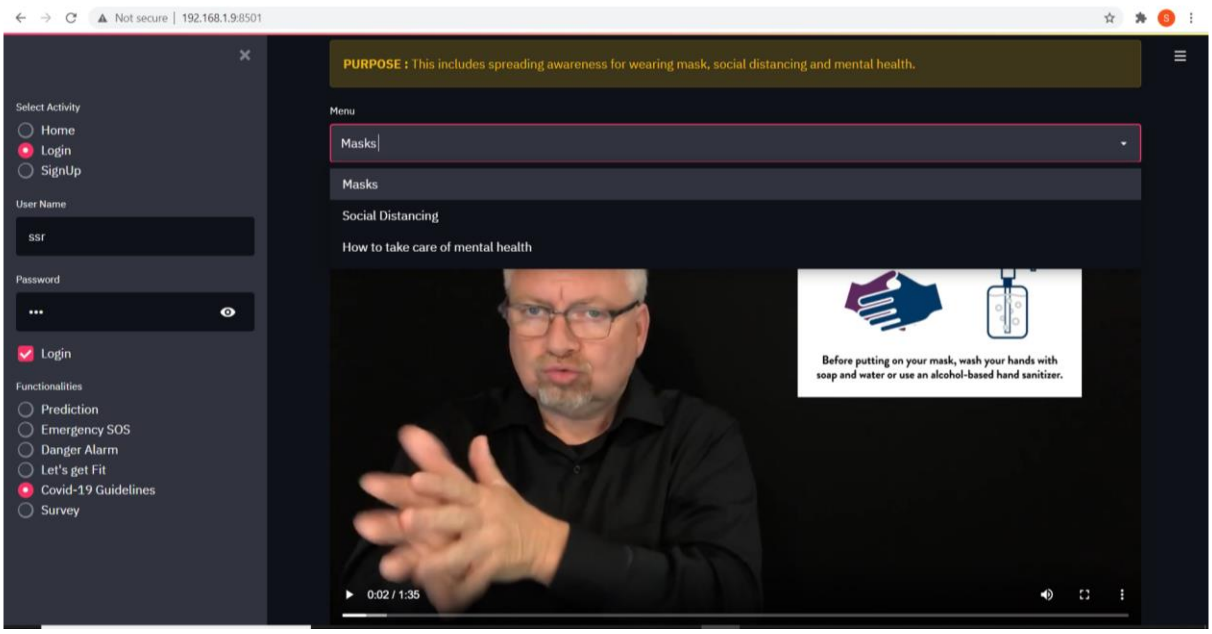
Menu: Meditation

- Meditation
- Exercise
- Chair Yoga

Fig 12. Let us Get Fit.

5.1.5. Covid-19 Guidelines:

In order to create awareness, this section is provided with 3 options: mask, social distancing and mental health. Each option is provided with a separate video to spread knowledge of coivd19 amongst especially abled people.



Not secure | 192.168.1.9:8501

PURPOSE : This includes spreading awareness for wearing mask, social distancing and mental health.

Select Activity
- Home
- [✓ Login]
- SignUp

User Name: ssr

Password: •••

[✓ Login]

Functionalities
- Prediction
- Emergency SOS
- Danger Alarm
- Let's get Fit
- [✓ Covid-19 Guidelines]
- Survey

Menu: Masks

- Masks
- Social Distancing
- How to take care of mental health

Before putting on your mask, wash your hands with soap and water or use an alcohol-based hand sanitizer.

0:02 / 1:35

Fig 13. Covid-19 Guidelines

5.1.6. Survey: To hear thoughts, suggestions, concerns or problems with anything so that we can improve, we have added a contact us page which includes a google form link to get feedback from user.

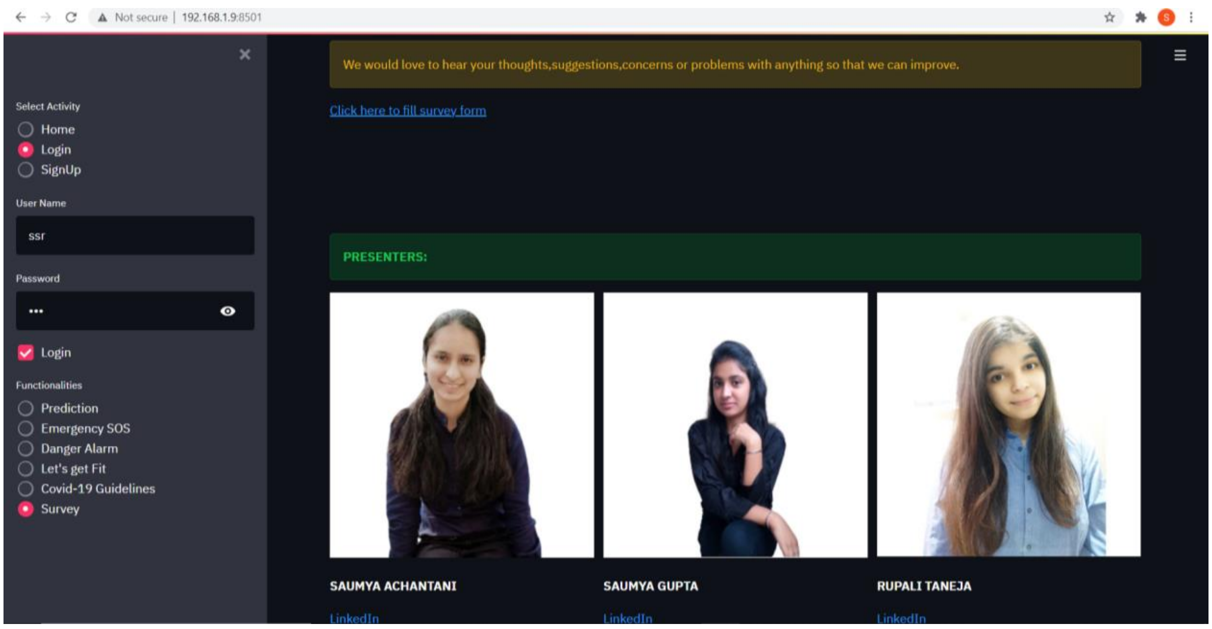
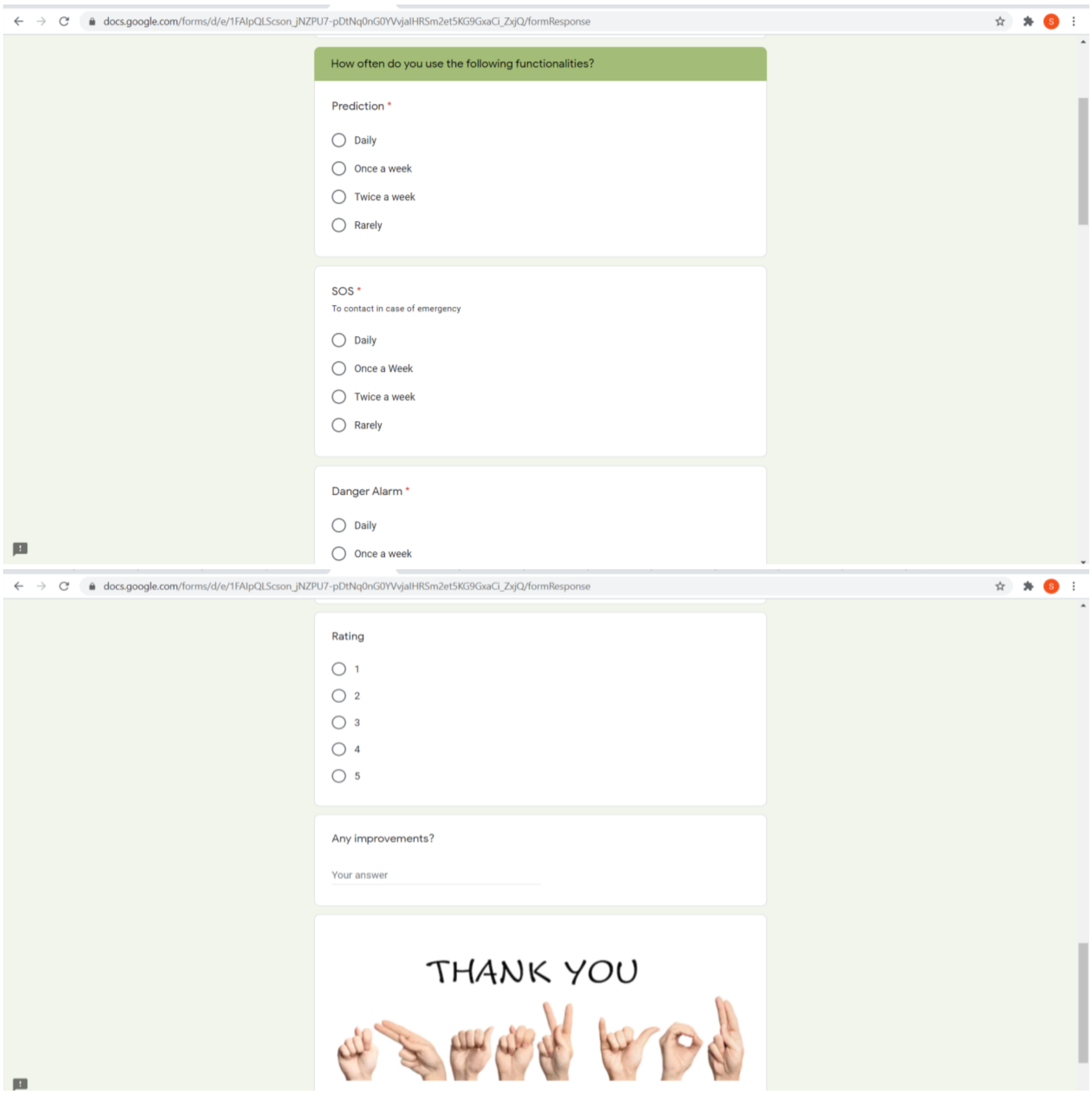


Fig 14. Contact Us


docs.google.com/forms/d/e/1FAIpQLScson_jNZPU7-pDtNq0nG0YVvjalHRSm2et5KG9GxaCi_ZxjQ/formResponse

How often do you use the following functionalities?

Prediction *

- Daily
- Once a week
- Twice a week
- Rarely

SOS *
To contact in case of emergency

- Daily
- Once a Week
- Twice a week
- Rarely

Danger Alarm *

- Daily
- Once a week

Rating

- 1
- 2
- 3
- 4
- 5

Any improvements?

Your answer

Fig 15. Survey form from users

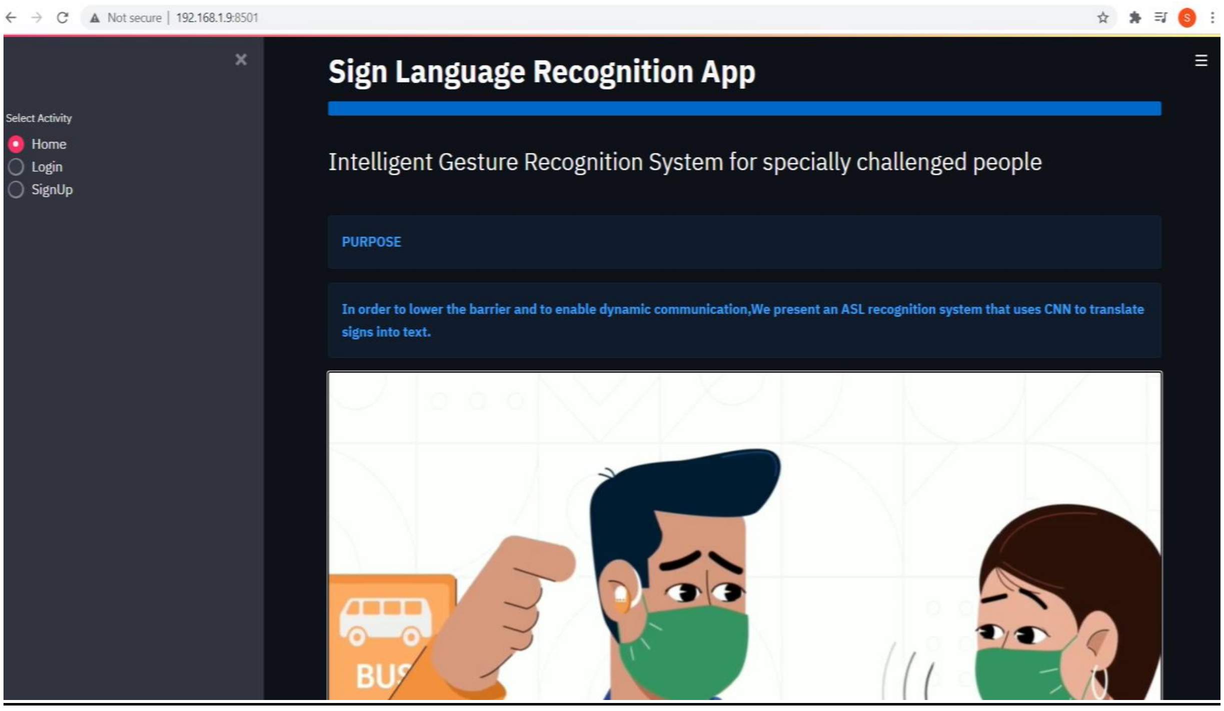
5.2)Deployment of model to Streamlit

What is Streamlit?

- Streamlit is an open-source Python library that makes it easy to create and share beautiful, custom web apps for machine learning and data science. Creating a user interface for a Python project is seamless with Streamlit, a relatively new browser-based Python framework that lets you elegantly showcase a project.
- Streamlit tries to take the application development team out of the picture, by enabling data scientists to develop their own applications.
- A bi-directional Streamlit Component has two parts: A frontend, which is built out of HTML and any other web tech you like (JavaScript, React, Vue, etc.), and gets rendered

in Streamlit apps via an iframe tag. A Python API, which Streamlit apps use to instantiate and talk to that frontend.

- It is an all-in-one tool that encompasses web serving as well as data analysis. Hence, have used streamlit interface to deploy our code into.


Not secure | 192.168.1.9:8501

Select Activity

[✓ Home] Login SignUp

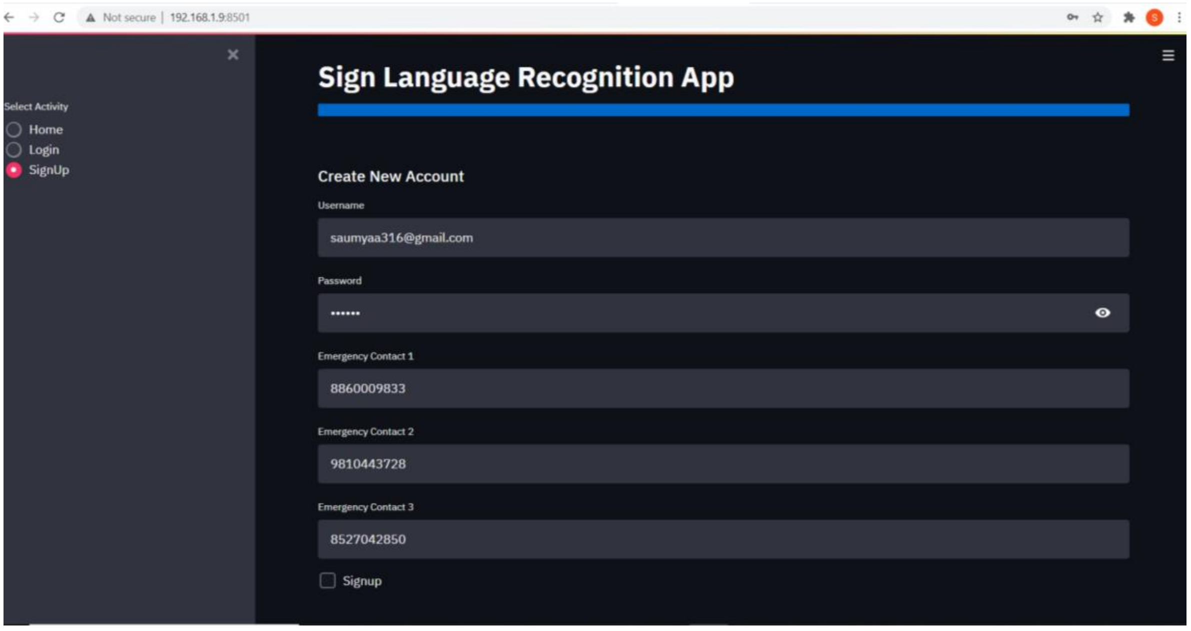
Sign Language Recognition App

Intelligent Gesture Recognition System for specially challenged people

PURPOSE

In order to lower the barrier and to enable dynamic communication,We present an ASL recognition system that uses CNN to translate signs into text.

Fig 16. Home page


Not secure | 192.168.1.9:8501

Select Activity

Home Login [✓ SignUp]

Sign Language Recognition App

Create New Account

Username	saumyaa316@gmail.com
Password	••••••
Emergency Contact 1	8860009833
Emergency Contact 2	9810443728
Emergency Contact 3	8527042850

☐ Signup

Fig 17. Sign Up page; details such as username, password, emergency contact.

6)Model Architecture

6.1)Terminologies

- A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other.
- 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.

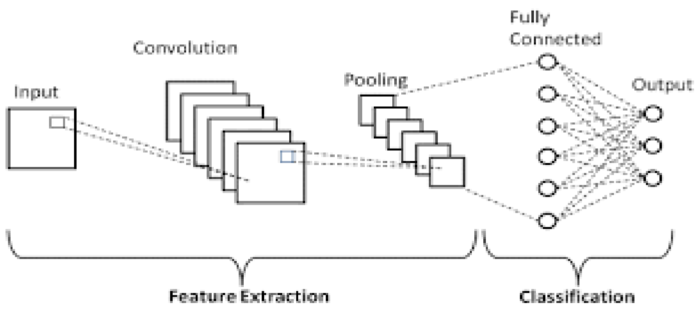


Fig 18. Architecture of CNN

1.Convolution Layer: In convolution layer we take a small window size [typically of length 5*5] that extends to the depth of the input matrix. The layer consists of learnable filters of window size. During every iteration we slid the window by stride size [typically 1], and compute the dot product of filter entries and input values at a given position. As we continue this process well create a 2-Dimensional activation matrix that gives the response of that matrix at every spatial position. That is, the network will learn filters that activate when they see some type of visual feature such as an edge of some orientation or a blotch of some colour.

2. Pooling Layer : We use pooling layer to decrease the size of activation matrix and ultimately reduce the learnable parameters.

There are two type of pooling :

2.1)Max Pooling : Max pooling is a pooling operation that selects the maximum element from the region of the feature map covered by the filter. Thus, the output after max-pooling layer would be a feature map containing the most prominent features of the previous feature map

2.2)Average Pooling : Average pooling involves calculating the average for each patch of the feature map. This means that each 2×2 square of the feature map is down sampled to the average value in the square.

3. Fully Connected Layer : In convolution layer, neurons are connected only to a local region, while in a fully connected region, we connect all the inputs to neurons. The input to the fully connected layer is the output from the final Pooling or Convolutional Layer, which is flattened and then fed into the fully connected layer. We need this layer since every single neuron has a connection to every single one in the next layer, hence now there is a flow of information between each input dimension (pixel location) and each output class, thus the decision is based truly on the whole image.

4. Final Output Layer : After getting values from fully connected layer, we will connect them to final layer of neurons[having count equal to total number of classes], that will predict the probability of each image to be in different classes.

<u>Different Libraries used to train the model</u>

1.TensorFlow : TensorFlow is a free and open-source software library for machine learning. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks. TensorFlow is a symbolic math library based on dataflow and differentiable programming. TensorFlow was originally developed for large numerical computations without keeping deep learning in mind.

2.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.

3.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.

<u>Image Processing Functions:</u>

OpenCV-Python is a library of Python bindings designed to solve computer vision problems.

1. Pre-processing: OpenCV uses BGR image format. So, when we read an image using cv2.imread() it interprets in BGR format by default. We have used cvtColor () method to convert our original image from the BGR colour space to Gray using COLOR_BGR2GRAY.

2.Gaussian Blurring: The Gaussian filter is a low-pass filter that removes
the high-frequency components are reduced.

Gaussian blur soothes uneven pixel values in an image by cutting out the extreme outliers. It is done with the function, cv.GaussianBlur(). We should specify the width and height of the kernel which should be positive and odd. We also should specify the standard deviation in the X and Y directions, sigmaX and sigmaY respectively. If only sigmaX is specified, sigmaY is taken as the same as sigmaX. If both are given as zeros, they are calculated from the kernel size. Gaussian blurring is highly effective in removing Gaussian noise from an image.

$$G(x, y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

3.Adaptive Thresholding

Adaptive thresholding typically takes a grayscale or color image as input and, in the simplest implementation, outputs a binary image representing the segmentation. For each pixel in the image, a threshold has to be calculated.

Adaptive thresholding is the method where the threshold value is calculated for smaller regions. e.g., if an image has different lighting conditions in different areas. In that case, adaptive thresholding can help. Here, the algorithm determines the threshold for a pixel based on a small region around it. So, we get different thresholds for different regions of the same image which gives better results for images with varying illumination.

We used cv2.adaptiveThreshold for this.

The adaptive Method decides how the threshold value is calculated:

- cv.ADAPTIVE_THRESH_MEAN_C: The threshold value is the mean of the neighbourhood area minus the constant C.
- cv.ADAPTIVE_THRESH_GAUSSIAN_C: The threshold value is a gaussian-weighted sum of the neighbourhood values minus the constant C.

The blockSize determines the size of the neighbourhood area and C is a constant that is subtracted from the mean or weighted sum of the neighbourhood pixels.

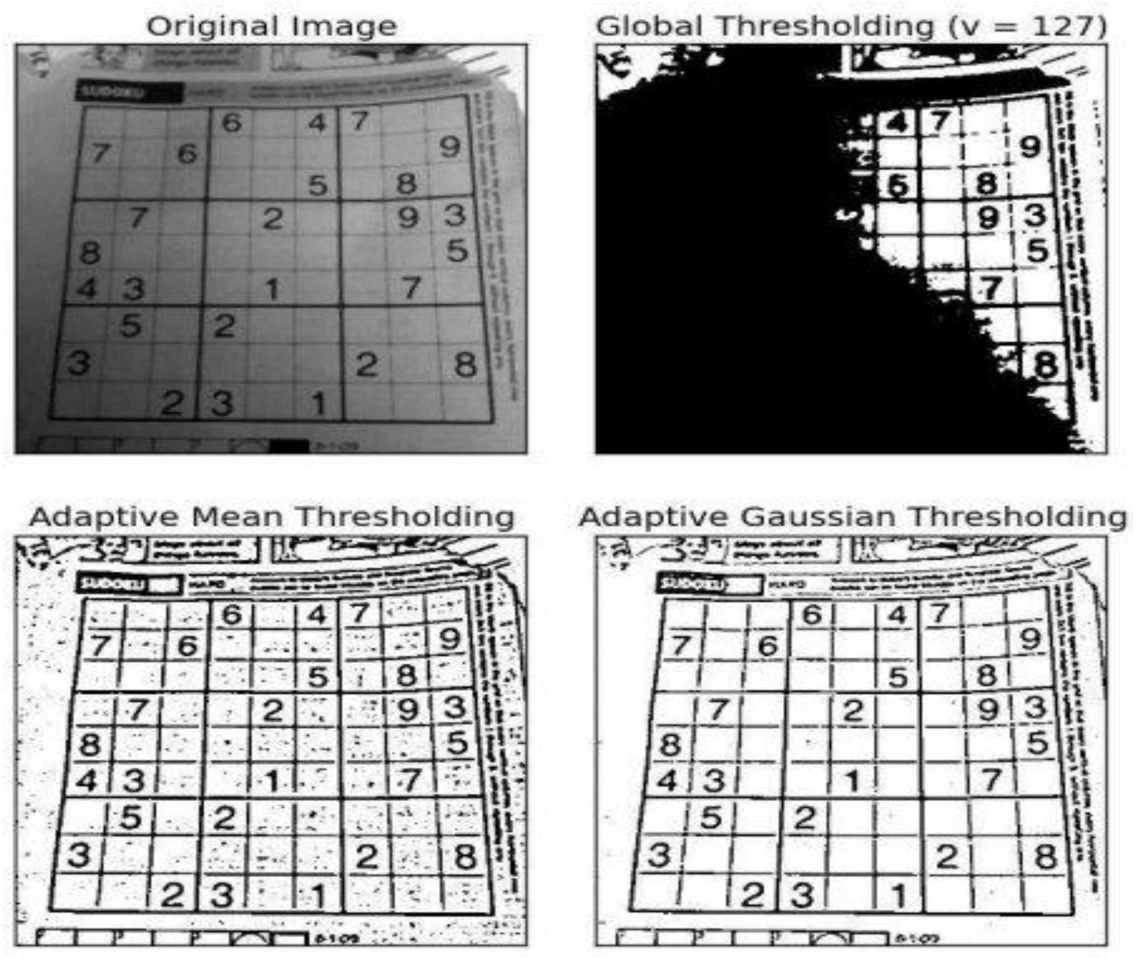

Original Image
Global Thresholding (v = 127)
Adaptive Mean Thresholding
Adaptive Gaussian Thresholding

Fig 19. Thresholding

6.2) Approach

Our approach uses two layers of algorithm to predict the final symbol of the user

Algorithm Layer 1:

- Apply gaussian blur filter and threshold to the frame taken with opencv to get the processed image after feature extraction.
- This processed image is passed to the CNN model for prediction and if a letter is detected for more than 50 frames then the letter is printed and taken into consideration for forming the word.

- Space between the words is considered using the blank symbol.

Algorithm Layer 2:

- We detect various sets of symbols which show similar results on getting detected. 2. We then classify between those sets using classifiers made for those sets only. Layer 1: CNN Model : 16 1. 1st Convolution Layer :The input picture has a resolution of

128x128 pixels. It is first processed in the first convolutional layer
using 32 filter weights (3x3 pixels each). This will result in a 126X126 pixel image, one for each Filter-weights.

- 1st Pooling Layer: The pictures are down sampled using max pooling of 2x2 i.e we keep the highest value in the 2x2 square of array. Therefore, our picture is downsampled to 63x63 pixels.

- 2nd Convolution Layer: Now, these 63 x 63 from the output of the first pooling layer is served as an input to the second convolutional layer. It is processed in the second convolutional layer using 32 filter weights (3x3 pixels each). This will result in a 60 x 60-pixel image.

- 2nd Pooling Layer : The resulting images are downsampled again using max pool of 2x2 and is reduced to 30 x 30 resolution of images.

- 1st Densely Connected Layer : Now these images are used as an input to a fully connected layer with 128 neurons and the output from the second convolutional layer is reshaped to an array of 30x30x32 =28800 values. The input to this layer is an array of 28800 values. The output of these layer is fed to the 2nd Densely Connected Layer. We are using a dropout layer of value 0.5 to avoid overfitting.

- 2nd Densely Connected Layer: Now the output from the 1st Densely Connected Layer is used as an input to a fully connected layer with 96 neurons.

- Final layer: The output of the 2nd Densely Connected Layer serves as an input for the final layer which will have the number of neurons as the number of classes we are classifying (alphabets + blank symbol).

Activation Function:

- Activation function decides, whether a neuron should be activated or not by calculating weighted sum and further adding bias with it. The purpose of the activation function is to introduce non-linearity into the output of a neuron.
- We have used ReLu (Rectified Linear Unit) in each of the layers(convolutional as well as fully connected neurons). ReLu calculates max(x,0) for each input pixel.

- This adds nonlinearity to the formula and helps to learn more complicated features. It helps in removing the vanishing gradient problem and speeding up the training by reducing the computation time.

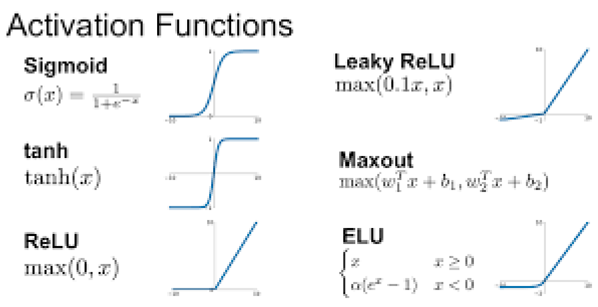


Fig 20. Types of Activation Function

Pooling Layer:

- Pooling layers are used to down sample the volume of convolution neural network by reducing the small translation of the features. pooling layer also provides a parameter reduction.
- We applied max pooling to the input image with a pool size of (2, 2) with relu activation function. This reduces the amount of parameters thus lessening the computation cost and reduces overfitting.

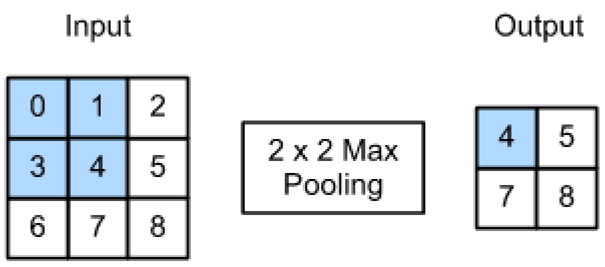


Fig 21. Maximum Pooling Layer

Dropout Layers:

- Dropout is a technique where randomly selected neurons are ignored during training. They are “dropped-out” randomly. This means that their contribution to the activation of downstream neurons is temporally removed on the forward pass and any weight updates are not applied to the neuron on the backward pass.
- The problem of overfitting, where after training, the weights of the network are so tuned to the training examples they are given that the network doesn’t perform well when given new examples.
- This layer “drops out” a random set of activations in that layer by setting them to zero. The network should be able to provide the right classification or output for a

specific example even if some of the activations are dropped out.

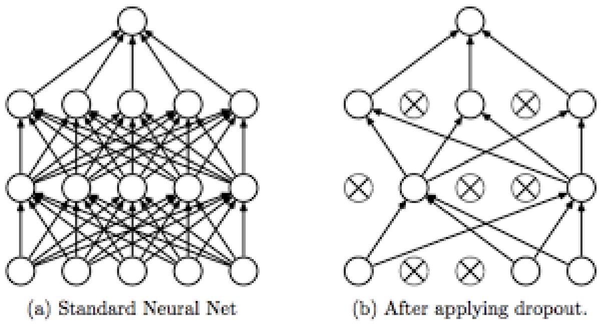


Fig 22. Standard Neural Network

Optimizer:

- Optimizers are Classes or methods used to change the attributes of your machine/deep learning model such as weights and learning rate in order to reduce the losses. Optimizers help to get results faster.
- We have used Adam optimizer for updating the model in response to the output of the loss function. Adam combines the advantages of two extensions of two stochastic gradient descent algorithms namely adaptive gradient algorithm(ADA GRAD) and root mean square propagation(RMSProp)

We are using two layers of algorithms to verify and predict symbols which are more similar to each other so that we can get us close as we can get to detect the symbol shown.

In our testing we found that following symbols were not showing properly and were giving other symbols also :

1. For D : R and U
2. For U : D and R
3. For I : T, D, K and I
4. For S : M and N

So, to handle above cases, we made three different classifiers for classifying these sets:

1. {D,R,U}
2. {T,K,D,I}
3. {S,M,N}

Whenever the count of a letter detected exceeds a specific value and no other letter is close to it by a threshold we print the letter and add it to the current string(In our code we kept the value as 50 and difference threshold as 20).

Whenever the count of a blank(plain background) detected exceeds a specific value and if the current buffer is empty no spaces are detected

In other case it predicts the end of word by printing a space and the current gets appended to the sentence below.

To Save Neural Network Model to JSON, we used the model_from_json () function that will create a new model from the JSON specification. The weights are saved directly from the model using the save weights () function and later loaded using the symmetrical load weights () function.

The model is then converted to JSON format and written to model. Json in the local directory. The network weights are written to model.h5 in the local directory.

Auto Word Suggestion

Hun spell is the spell checker of LibreOffice, OpenOffice.org, Mozilla Firefox 3 & Thunderbird, Google Chrome, and it is also used by proprietary software.

The Hun spell function is a high-level wrapper for finding spelling errors within a text document or getting suggestions, adding new words, etc. It also provides some basic morphological analysis related methods.

In this project we have used Hun spell to auto suggest words which the especially abled person is trying to communicate.

6. Challenges Faced

There were many challenges faced during the project. The very first issue was the dataset. We wanted to deal with raw images and that too square images as CNN in Keras as it was a lot more convenient working with only square images. We couldn't find any existing dataset for that hence we decided to make our own dataset.

Second issue was to select a filter which we could apply on our images so that proper features of the images could be obtained and hence then we could provide that image as input for CNN model. We tried various filters including binary threshold, canny edge detection, gaussian blur etc. but finally we settled with a gaussian blur filter. More issues were faced relating to the accuracy of the model we trained in earlier phases which we eventually improved by increasing the input image size and also by improving the dataset.

7. Conclusion

In this report, a functional real time vision based American sign language recognition for D&M people have been developed for asl alphabets. We achieved final accuracy of 87.0% on our dataset. We are able to improve our prediction after implementing two layers of algorithms in which we verify and predict symbols which are more similar to each other. This way we are able to detect almost all the symbols provided that they are shown properly, there is no noise in the background and lighting is adequate. Gesture recognition algorithm is relatively robust and accurate. Convolution can be slow, so there is trade-off between speed and accuracy.

8. **References**

- https://en.wikipedia.org/wiki/OpenCV
- https://www.sciencedirect.com/science/article/pii/S1877050917320720
- https://docs.opencv.org/master/d7/d4d/tutorial_py_thresholding.html
- https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
- https://en.wikipedia.org/wiki/Sign_language
- https://data-flair.training/blogs/sign-language-recognition-python-ml-opencv/
- https://pypi.org/project/hunspell/
- https://docs.streamlit.io/_/downloads/en/latest/pdf/
- https://www.nidcd.nih.gov/health/american-sign-language
- Pigou L., Dieleman S., Kindermans PJ., Schrauwen B. (2015) Sign Language Recognition Using Convolutional Neural Networks. In: Agapito L., Bronstein M., Rother C. (eds) Computer Vision - ECCV 2014 Workshops. ECCV 2014. Lecture Notes in Computer Science, vol 8925. Springer, Cham
- https://opencv.org/
- https://en.wikipedia.org/wiki/TensorFlow
- https://en.wikipedia.org/wiki/Convolutional_neural_network