AI Assistance for Sign Language

Click here - Github Repo ,

Video Demo

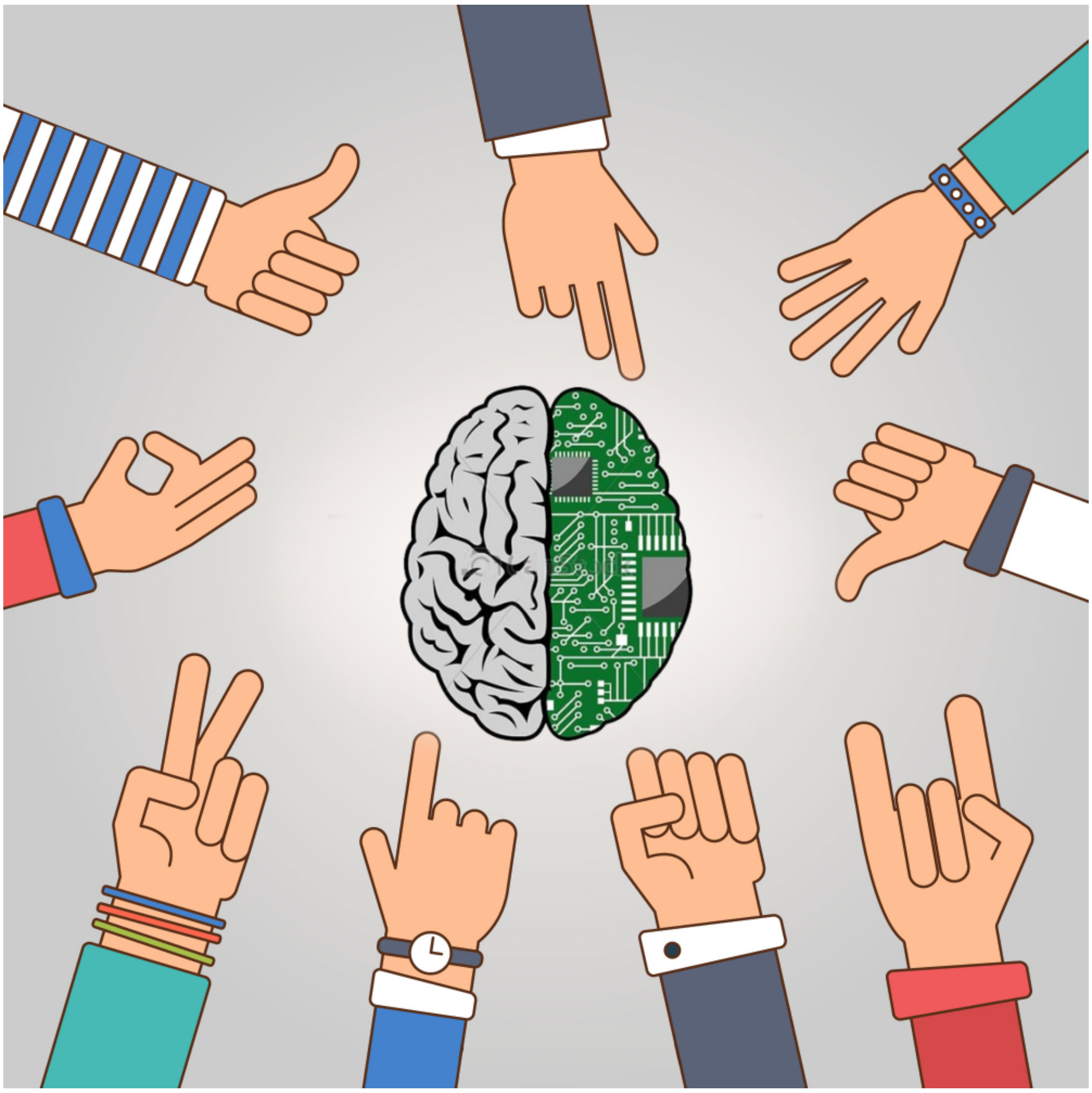
Team: 0 or 1 India

Members:

Harikrishnan

Rohit Arunachalam

Vaka Isha Reddy



PROBLEM STATEMENT

Humans communicate with one another to share their thoughts, feelings and experiences. But this is not the case for the deaf-mute. There are many people who face these disabilities right from birth or because of issues such as Selective mutism, Aphasia syndrome, Alport syndrome, Norrie syndrome etc.
A large number of the population is disconnected from the mainstream hearing-dominated society and lie at the risk of being marginalized because people who are limited to using only speech can't communicate with them. A lack of accessibility to support the conversation between both communities also adds to the problem. Due to this , there is a communication gap between the deaf-mute and those who can speak.To bridge this gap, a non-verbal language called sign language exists.
According to the World Federation of the Deaf, there are more than 70 million deaf people worldwide. More than 80% of them live in developing countries. Collectively, they use more than 300 different sign languages. Sign languages are fully fledged natural languages, structurally distinct from spoken languages.
In this project, we have implemented a solution for INDIAN SIGN LANGUAGE.

SOLUTION

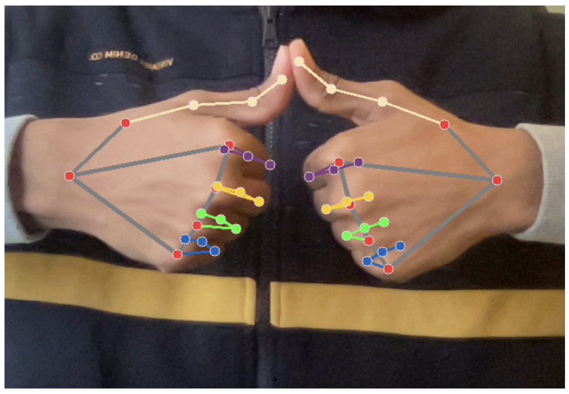
- Our solution to this problem is an AI assistance for sign language tool that works based on computer vision and machine learning using technologies like OpenCV,Mediapipe,BERT,Streamlit.etc
- The tool gets the sign language gesture performed by the person as input from the camera and converts that into speech.Similarly , it converts text to Indian sign language gestures.
- We've trained the ML model using the SVM algorithm.
- We have also added features like word autocomplete,Next word prediction and Backspacing

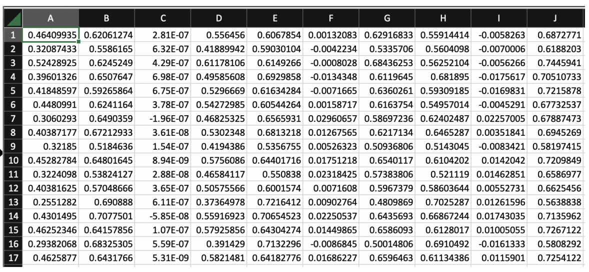
There are unique gestures in each sign language with variation in hand shape, motion profile,position of the hand, face and body parts contributing to each sign. So, visual sign language recognition is a complex research area in computer vision.Our project works only for static hand gestures,however we've started working on dynamic hand gestures.

IMPLEMENTATION OF SIGN LANGUAGE RECOGNITION

Implementation of Sign Language Recognition comprises of primarily 2 steps:

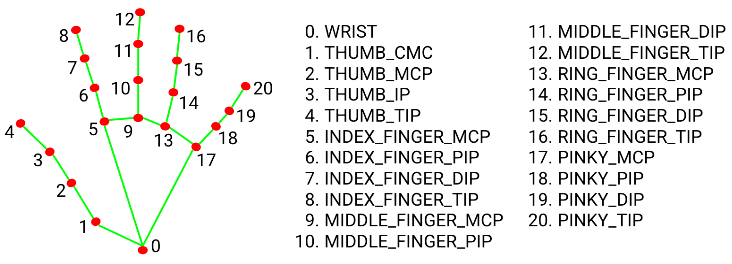
1) Collection of the dataset:
We created 39 datasets (26 Alphabets, 0-9 numbers, and gestures for backspace, space, and autocomplete), each gesture consisting of 900+ images.
We used OpenCV to capture images for the dataset and ran Mediapipe hands ML pipeline on the images to get the coordinates of the hands in these images and stored them in a CSV file





	A	B	C	D	E	F	G	H	I	J
1	0.46409935	0.62061274	2.81E-07	0.556456	0.6067854	0.00132083	0.62916833	0.55914414	-0.0058263	0.6872771
2	0.32087433	0.5586165	6.32E-07	0.41889942	0.59030104	-0.0042234	0.5335706	0.5604098	-0.0070006	0.6188203
3	0.52428925	0.6245249	4.29E-07	0.61178106	0.6149266	-0.0008028	0.68436253	0.56252104	-0.0056266	0.7445941
4	0.39601326	0.6507647	6.98E-07	0.49585608	0.6929858	-0.0134348	0.6119645	0.681895	-0.0175617	0.70510733
5	0.41848597	0.59265864	6.75E-07	0.5296669	0.61634284	-0.0071665	0.6360261	0.59309185	-0.0169831	0.7215878
6	0.4480991	0.6241164	3.78E-07	0.54272985	0.60544264	0.00158717	0.6163754	0.54957014	-0.0045291	0.67732537
7	0.3060293	0.6490359	-1.96E-07	0.46825325	0.6565931	0.02960657	0.58697236	0.62402487	0.02257005	0.67887473
8	0.40387177	0.67212933	3.61E-08	0.5302348	0.6813218	0.01267565	0.6217134	0.6465287	0.00351841	0.6945269
9	0.32185	0.5184636	1.54E-07	0.4194386	0.5356755	0.00526323	0.50936806	0.5143045	-0.0083421	0.58197415
10	0.45282784	0.64801645	8.94E-09	0.5756086	0.64401716	0.01751218	0.6540117	0.6104202	0.0142042	0.7209849
11	0.3224098	0.53824127	2.88E-08	0.46584117	0.550838	0.02318425	0.57383806	0.521119	0.01462851	0.6586977
12	0.40381625	0.57048666	3.65E-07	0.50575566	0.6001574	0.0071608	0.5967379	0.58603644	0.00552731	0.6625456
13	0.2551282	0.690888	6.11E-07	0.37364978	0.7216412	0.00902764	0.4809869	0.7025287	0.01261596	0.5638838
14	0.4301495	0.7077501	-5.85E-08	0.55916923	0.70654523	0.02250537	0.6435693	0.66867244	0.01743035	0.7135962
15	0.46252346	0.64157856	1.07E-07	0.57925856	0.64304274	0.01449865	0.6586093	0.6128017	0.01005055	0.7267122
16	0.29382068	0.68325305	5.59E-07	0.391429	0.7132296	-0.0086845	0.50014806	0.6910492	-0.0161333	0.5808292
17	0.4625877	0.6431766	5.31E-09	0.5821481	0.64182776	0.01686227	0.6596463	0.61134386	0.0115901	0.7254122

MediaPipe Hands utilizes an ML pipeline consisting of multiple models working together: A palm detection model that operates on the full image and returns an oriented hand bounding box. A hand landmark model that operates on the cropped image region defined by the palm detector and returns high-fidelity 3D hand keypoints. Using this ML pipeline we infer 21 3D landmarks of a hand from each frame.



0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

Using Mediapipe,we obtained the coordinates of the hands in each image and stored them in a CSV file.We performed data cleaning in order to obtain an accurate dataset and were left with around 2.1 million coordinates to train the model.

2) Creating a ML Model using a suitable Machine Learning Algorithm and testing the model:

We picked Support Vector Machine(SVM) algorithm to create the ML model for our project.

WHAT IS SVM & HOW DOES IT WORK?

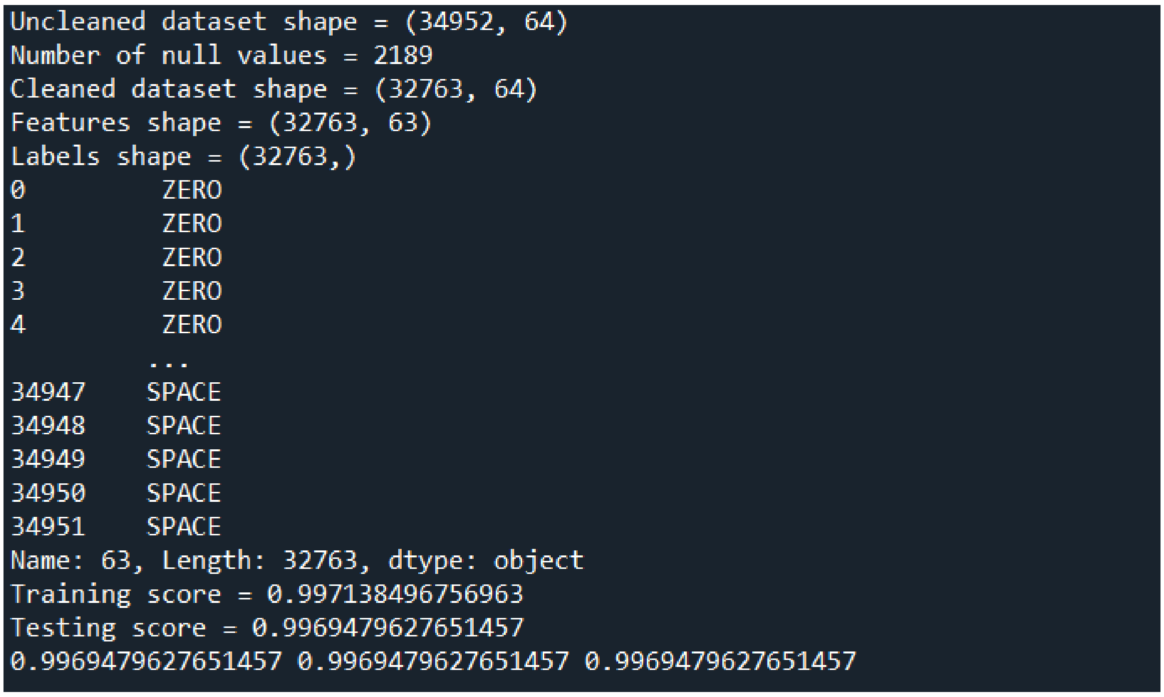
Support Vector Machines (SVMs) are a type of supervised learning algorithm that can be used for classification or regression tasks. They work by finding the hyperplane in a high-dimensional feature space that maximally separates the different classes.

In the context of classification tasks, an SVM model takes a set of labeled training examples as input and uses them to learn a decision boundary that can be used to predict the class of new examples. The decision boundary is found by maximizing the margin, or the distance between the boundary and the nearest training examples from each class.

WHY DID WE CHOOSE SVM OVER OTHER ALGORITHMS?

- SVMs are effective in high dimensional spaces, or when the number of features is much greater than the number of samples.
- SVMs are generally less computationally intensive compared to other algorithms
- SVMs can be more interpretable than others because the decision boundary is found in the original feature space rather than a transformed feature space. This can make it easier to understand how the model is making predictions.

We used 80% of the dataset to train the model and the remaining 20% to test the model. Using the pickle module we created an SVM model.(SAV File)
We tested the model by checking if the hand gestures are recognised with a good level of accuracy in different lighting conditions.

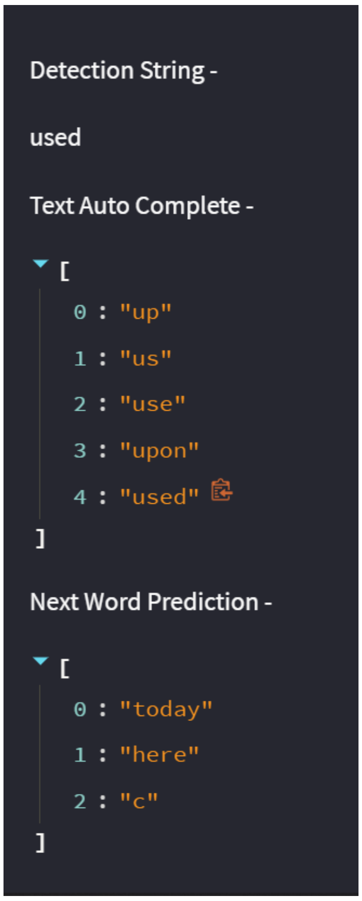


```
Uncleaned dataset shape = (34952, 64)
Number of null values = 2189
Cleaned dataset shape = (32763, 64)
Features shape = (32763, 63)
Labels shape = (32763,)
0          ZERO
1          ZERO
2          ZERO
3          ZERO
4          ZERO
         ...
34947    SPACE
34948    SPACE
34949    SPACE
34950    SPACE
34951    SPACE
Name: 63, Length: 32763, dtype: object
Training score = 0.997138496756963
Testing score = 0.9969479627651457
0.9969479627651457 0.9969479627651457 0.9969479627651457
```

IMPLEMENTATION OF TEXT PREDICTION:

The alphabet that was predicted using SIgn language recognition will be the input for text prediction. If the gesture prediction of the alphabet is wrong then the user can use “BACKSPACE GESTURE” to delete the alphabet from the string , or if the word shown in the auto-complete prediction is correct, then the user can perform the “AUTO COMPLETE GESTURE” to complete the word. There is also another gesture for space which helps in creating space between two words or alphabets.

fast_autocomplete library from python is used to predict the current word and a pre-trained **BERT** model has been used for next word prediction.



Detection String -

used

Text Auto Complete -

```
[
  0 : "up"
  1 : "us"
  2 : "use"
  3 : "upon"
  4 : "used"
]
```

Next Word Prediction -

```
[
  0 : "today"
  1 : "here"
  2 : "c"
]
```

Why BERT?
One reason why BERT may be considered better than other language models is that it is able to take into account the context of the words in a given sentence, rather than just processing the words in isolation. This is because BERT is a bidirectional model, meaning it takes into account the context of the words both to the left and to the right of the target word. This allows BERT to better understand the

meaning of words in context and make more accurate predictions.
Overall, BERT's ability to take into account context and its good generalization performance make it a powerful tool for natural language processing tasks.

IMPLEMENTATION OF TEXT TO SIGN LANGUAGE:

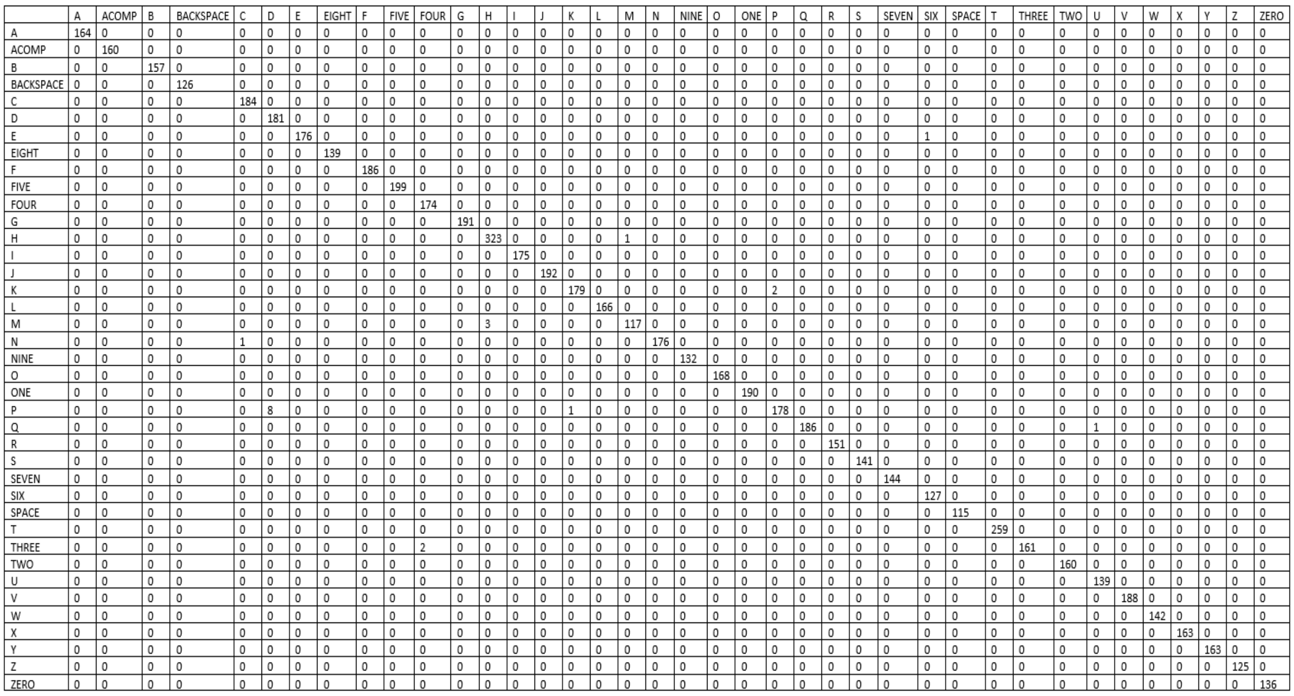
We get the user’s input for the text , iterate over the text and display the hand sign for each alphabet with the help of Pillow module.

RESULT

In this project we have developed a functional real-time computer vision based Sign language assistance tool for the deaf-mute that recognises the Indian Sign Language performed and converts that into speech, and converts text to sign language for interpretation.We have also created our own hand gestures to accommodate features like BACKSPACE, AUTO COMPLETE AND SPACE.

We intended to make this project as a web application so that anyone with access to any device(smartphone,tablets,laptops,computers) and an internet connection can make use of this tool without the need of any prerequisites to download specific software and we have achieved it. (We are in the process of deploying this application in the cloud)

We achieved a prediction accuracy of 99.0% on our dataset in good lighting conditions**.** The confusion matrix of the model is given below.



	A	ACOMP	B	BACKSPACE	C	D	E	EIGHT	F	FIVE	FOUR	G	H	I	J	K	L	M	N	NINE	O	ONE	P	Q	R	S	SEVEN	SIX	SPACE	T	THREE	TWO	U	V	W	X	Y	Z	ZERO
A	164	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ACOMP	0	160	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	157	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
BACKSPACE	0	0	0	126	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	0	0	0	184	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	0	0	0	0	0	181	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E	0	0	0	0	0	0	176	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
EIGHT	0	0	0	0	0	0	0	139	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	0	0	0	186	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FIVE	0	0	0	0	0	0	0	0	0	199	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FOUR	0	0	0	0	0	0	0	0	0	0	174	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	0	0	0	0	0	191	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	0	0	0	0	0	323	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	175	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
J	0	0	0	0	0	0	0	0	0	0	0	0	0	0	192	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
K	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	179	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	166	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	117	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	176	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
NINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	132	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
O	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	168	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ONE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	190	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	178	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	186	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	151	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	141	0	0	0	0	0	0	0	0	0	0	0	0	0
SEVEN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	144	0	0	0	0	0	0	0	0	0	0	0	0
SIX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	127	0	0	0	0	0	0	0	0	0	0	0
SPACE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	115	0	0	0	0	0	0	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	259	0	0	0	0	0	0	0	0	0
THREE	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	161	0	0	0	0	0	0	0	0
TWO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	160	0	0	0	0	0	0	0
U	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	139	0	0	0	0	0	0
V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	188	0	0	0	0	0
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	142	0	0	0	0
X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	163	0	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	163	0	0
Z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	125	0
ZERO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	136

LIMITATIONS

1) Recognition of the sign language gestures may not work in poor lighting conditions.
2) Users are limited to performing only alphabet and number gestures since we have not added gestures for common words and phrases.We will be adding gestures for the latter soon

FUTURE SCOPE

- Sign language recognition for dynamic gestures (gestures that have movement)
- Increasing the Accuracy of the model in poor lighting conditions.
- Addition of a Sign Language Instructor that teaches people to perform Sign language and guide them in doing the gestures correctly

ACKNOWLEDGEMENTS

OpenCV and the applications developed by people using that have greatly fascinated us, and when we got to know about the OpenCV AI competition we were thrilled to know about this opportunity.It has provided motivation and has been the backbone for the development of this project.

We would like to thank our University professors Dr. Santhi Natarajan , Dr.S. Vidhusha , Dr. S. Manisha , Dr. Sundharakumar KB , Dr Pradheep Balaji R and Mr.Madhan Kumar for their constant support and advice by providing valuable inputs and guiding us.Last but not the least , we would like to thank our friends and family for constantly supporting and motivating us throughout.

REFERENCES

- Sign Language Recognition with Support Vector Machines and Hidden Conditional Random Fields Going from Fingerspelling to Natural Articulated Words César Roberto de Souza and Ednaldo Brigante Pizzolato Universidade Federal de São Carlos, São Carlos, Brasil
- Indian Sign Language Recognition using SVM Classifier Deepali G.Malia Nitin S.Limkarb Satish H. Malic a,b.cAssistant Professor in E&T department,JSPM's BIT,Barshi413401,India.
- https://opencv24-python-tutorials.readthedocs.io/en/latest
- https://google.github.io/mediapipe/solutions/hands.html
- https://www.youtube.com/@PragyaGupta
- https://www.kaggle.com/datasets/prathumarikeri/indian-sign-language-isl
- https://docs.streamlit.io
- https://github.com/whitphx/streamlit-webrtc
- https://github.com/pandeynandancse/next_word_prediction_streamlit