SIGN LANGUAGE ACTIONS RECOGNITION

Project report submitted
in partial fulfillment of the requirement for
the degree of

Bachelor of Technology
In
Computer Science Engineering

By

M.Venkata kalyan babu, 18BCS049
T. Umesh anand babu, 18BCS105
M.P. Bharath, 18BCS057
Y.Mokshith ramendra, 18BCS112

Under the guidance of

Dr.B.Jayalakshmi
Assistant Professor
Department of Computer Science and Engineering

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY
Dharwad

CERTIFICATE

It is certified that the work contained in the project report titled “SIGN LANGUAGE ACTIONS RECOGNITION” by “Venkata kalyan babu(18BCS049)”,“Umesh Anand Babu(18BCS105)”,“MP.Bharath(18BCS057)” and “Y.Mokshith ramendra(18BCS112)” has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Dr.B.Jayalakshmi

Assistant Professor

Computer Science Engineering

(May 2022)

DECLARATION

We declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, we have adequately cited and referenced the sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not have been properly cited or from whom proper permission has not been taken when needed.

M.VENKATA KALYAN BABU

18BCS049

T.UMESH ANAND BABU

18BCS105

MP. BHARATH

18BCS057

Y. MOKSHITH RAMENDRA

18BCS112

APPROVAL SHEET

This project report entitled “SIGN LANGUAGE ACTIONS RECOGNITION” by M.venkata kalyan babu (18BCS049), T.Umesh Anand Babu (18BCS105) , M.P.Bharath (18BCS057) and Y.Mokshith ramendra (18BCS112) is approved for the degree of Bachelor of Technology in Computer Science and Engineering.

Supervisor

Dr.B.Jayalakshmi
Assistant Professor
Computer Science Engineering

Head of Department

Dr.Uma Sheshadri
Professor
Computer Science Engineering

Date:11/5/2022
Place: Dharwad

Table of Contents

1. INTRODUCTION

Human communication is essential in our day-to-day lives because it allows us to express ourselves. Speech, along with gestures, body language, reading, writing, and visual aids, is one of the most extensively utilized ways of communication. These modalities of communication, however, create a communication gap for the speaking and hearing handicapped minority. Visual aids or an interpreter are the only known ways to communicate with a disabled individual. However, these approaches have been shown to be difficult and expensive, making them nearly impossible to deploy in an emergency. We might be able to come up with a viable solution through Sign Language Recognition. To convey meaning, Sign Language mostly relies on hand-operated communication. This refers to simultaneous actions involving a variety of hand shapes, hand, arm, and body movement to convey the speaker's thoughts.

Two types of sign language include fingerspelling, which spells out words character by character, and word level association, which uses hand motions to convey word meaning. Fingerspelling is a useful tool in sign language because it enables users to convey names, addresses, and other words that have no value at the word level. Despite this, because fingerspelling is difficult to grasp and use, it is not widely employed. Furthermore, there is no universal sign language, and only a few people know how to use it, making it an ineffective mode of communication.

The problem we're looking at is supervised feature learning for sign language recognition. Recognizing sign language is an exciting computer vision challenge, because closing the communication gap between a disabled person and a normal person who does not comprehend sign language would be tremendously beneficial. To identify the SL actions, we first created a data collection that included all of the different, distinct hand motions. The next step was to use MP holistic to extract key points from the frame, and then record the landmarks of each key point in a numpy array. After that, preprocess the data and construct labels and features, then design and train an LSTM model with the collected data to decide which action was made.

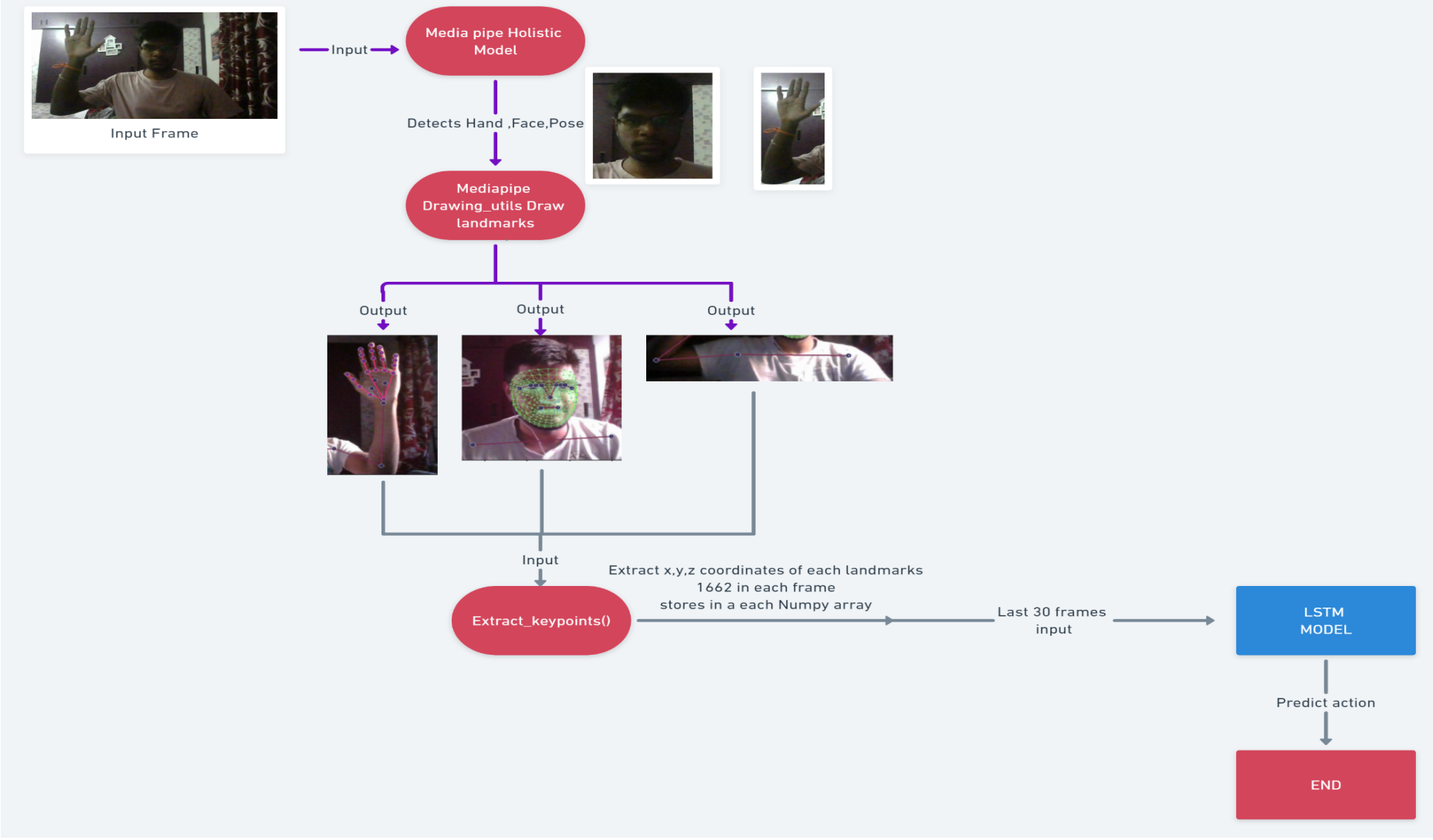
2. MOTIVATION

1.3 million people in India have "hearing impairment," according to the 2011 census. In India, the National Association of the Deaf estimates that 18 million people are deaf, accounting for roughly 1% of the population. These data were the driving force behind our efforts. Because these speech challenged and deaf people lack a proper channel to communicate with regular people, a system is required. The disabled's sign language is not understood by everyone. As a result, our initiative aims to convert sign language motions into text that regular people can understand.

3. PROBLEM DEFINITION

Dumb people use hand signs to communicate, normal people have difficulty understanding their language. Hence there is a need of the systems which recognizes the different signs and conveys the information to the normal people.

Approach of our project



4. ALGORITHM

Step1: Importing and Installing all the required dependencies like Tensorflow, Sklearn OpenCV etc..

Step2: Keypoints Using Mediapipe Holistic.

Step3: Extracting Keypoint Values of hands, pose, face by using OpenCV.

Step4: Setup Folders For Collection.

Step5: Collect Keypoint Values For Training and Testing.

Step6: Preprocess Data and Create Labels and Features by using Sklearn.

Step7: Build and Train LSTM Neural Network.

Step8: Making Predictions.

Step9: Save Weights.

Step10: Evaluating model Using Confusion Matrix and Accuracy.

Step 11: Testing the model in Real time.

5. Methodology:

5.1 DATASET DESCRIPTION

We didn't use any existing datasets instead, we used OpenCV to gather our own images from a webcam, and for each sign, we took 30 films, 30 frames, and 1662 landmarks, so that we could cover all the scenarios that would be useful for accurately detecting signs in real time.

Signs in our Dataset:-

Hello = 0

IloveYou = 1

No = 2

Yes = 3

I’mFine = 4

Howareyou = 5

Help = 6

After = 7

Careful = 8

Thankyou = 9

Iamsorry = 10

Me = 11

Move = 12

Pay = 13

Phone = 14

Protect = 15

Stop = 16

Takecare = 17

Danger = 18

Donot = 19

Water = 20

Bye = 21

5.2 Extraction of keypoints

5.2.1 What is mediapipe

One of the most common and widely used use cases in computer vision is object detection. Several object detection models are used in various applications around the world. Many of these models have been implemented as stand-alone solutions for a specific computer vision task with their own dedicated application. MediaPipe accomplishes this by combining several of these duties into a single real-time end-to-end solution.

MediaPipe is an open-source, cross-platform Machine Learning framework for building complex, multimodal applied machine learning pipelines. It could be used to build advanced Machine Learning models.such as face detection, multi-hand tracking, object detection and tracking, and many more. MediaPipe is a model implementation mediator for systems running on any platform, allowing developers to spend more time experimenting with models and less time worrying about the system.

Possibilities with mediapipe:

1.Tracking and Detection of Human Pose A minimum of 25 2D upper-body landmarks are inferred using RGB video frames for high-fidelity human body pose tracking.

2.Face Mesh in 3D with 468 face landmarks and multi-face support.

3.Based on a high-performance palm detection and hand landmark model, Hand Tracking 21 landmarks in 3D with multi-hand support.

4.Tracking from Every Angle Tracking of 33 poses, 21 per-hand, and 468 facial landmarks in real time and with semantic consistency.

5.Hair Segmentation Super realistic real-time hair recoloring.

6.Detection and tracking of objects Detection and tracking of video objects in a single pipeline.

7.Face Detection Ultra-lightweight face detector with 6 landmarks and support for multiple faces.

8.3D Object Recognition Detection and 3D pose estimation of commonplace objects such as shoes and chairs.

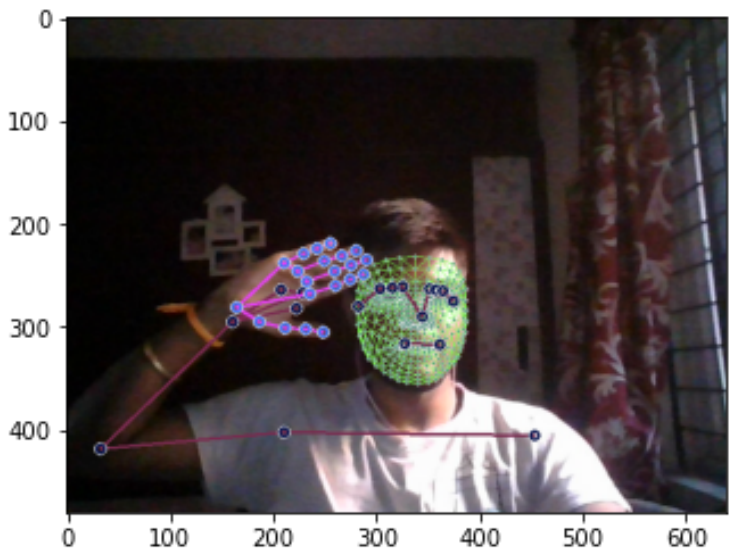
5.2.2 Mediapipe Holistic

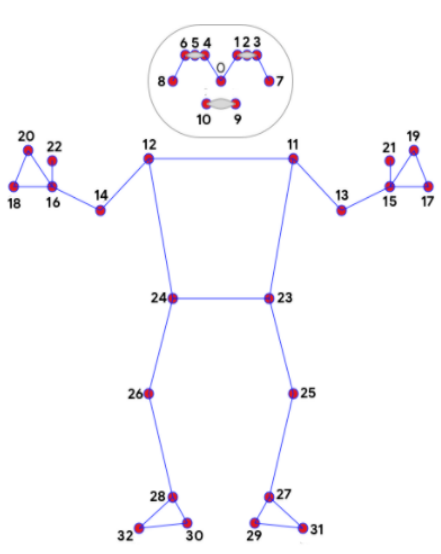
One of the pipelines that includes optimised face, hands, and pose components for holistic tracking is Mediapipe Holistic, which allows the model to identify hand and body poses as well as face landmarks at the same time.One of the most common applications of MediaPipe holistic is detecting faces and hands and extracting crucial features to feed into a computer vision model.

Landmarks that can be Detected using Mediapipe Holistic:

POSE_LANDMARKS:-
A list of pose landmarks. Each landmark consists of the following:

- **x** and **y:** The image width and height are used to normalise landmark coordinates to [0.0, 1.0].
- **z:** Should be ignored because the model is not yet fully trained to predict depth, but this is on the agenda.
- **visibility:** A value between [0.0, 1.0] indicating the likelihood of the landmark being visible (present and not occluded) in the image.

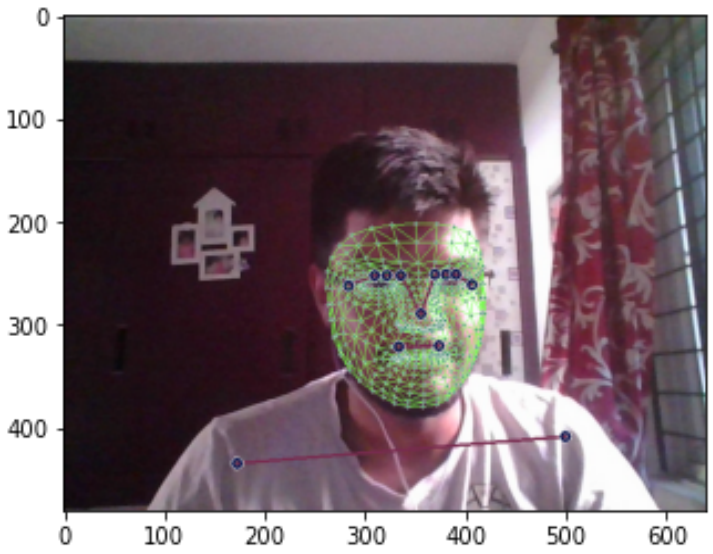




0. nose
1. left_eye_inner
2. left_eye
3. left_eye_outer
4. right_eye_inner
5. right_eye
6. right_eye_outer
7. left_ear
8. right_ear
9. mouth_left
10. mouth_right
11. left_shoulder
12. right_shoulder
13. left_elbow
14. right_elbow
15. left_wrist
16. right_wrist
17. left_pinky
18. right_pinky
19. left_index
20. right_index
21. left_thumb
22. right_thumb
23. left_hip
24. right_hip
25. left_knee
26. right_knee
27. left_ankle
28. right_ankle
29. left_heel
30. right_heel
31. left_foot_index
32. right_foot_index

FACE_LANDMARKS:-

There are 468 facial landmarks on this list. Each landmark is made up of the letters x, y, and z. The image width and height are used to normalise x and y to [0.0, 1.0]. The depth of the landmark is represented by z, with the origin being the depth at the centre of the head. The smaller the value, the closer the landmark is to the camera. The magnitude of z is measured on the same scale as x.

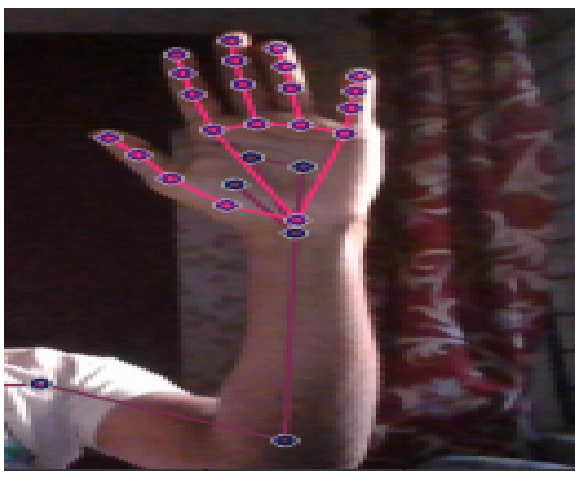


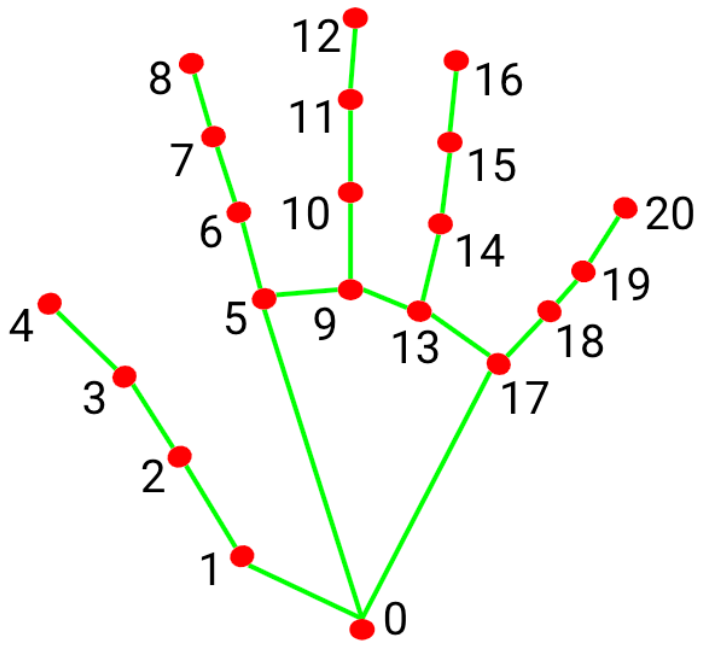
LEFT_HAND_LANDMARKS:-

On the left hand, there are 21 landmarks. x, y, and z make up each landmark. The picture width and height, respectively, normalize x and y to [0.0, 1.0]. The landmark's origin is at the wrist, and the smaller the value, the closer it is to the camera. The scale of z is similar to that of x.

RIGHT_HAND_LANDMARKS:-

A list of 21 hand landmarks on the right hand, represented in the same way as left hand landmarks.





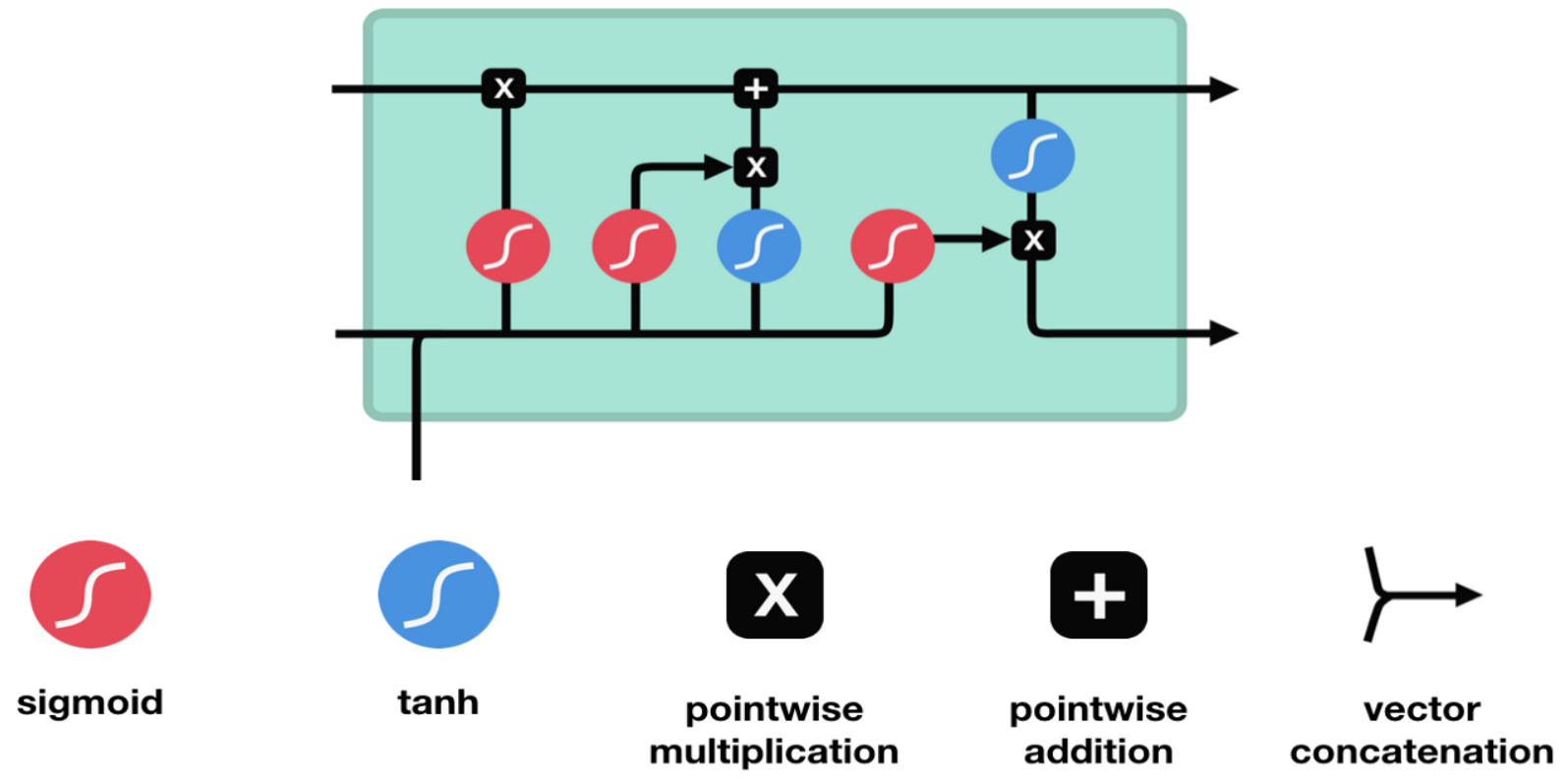
0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP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

5.3. The Model

LSTM:-

LSTM networks are well-suited to classifying, processing and making predictions based on time series data.An LSTM has a control flow that is comparable to that of a recurrent neural network. It processes data and passes information on as it moves along. The processes within the LSTM's cells are the differences.

The LSTM can keep or forget information using the procedures listed below

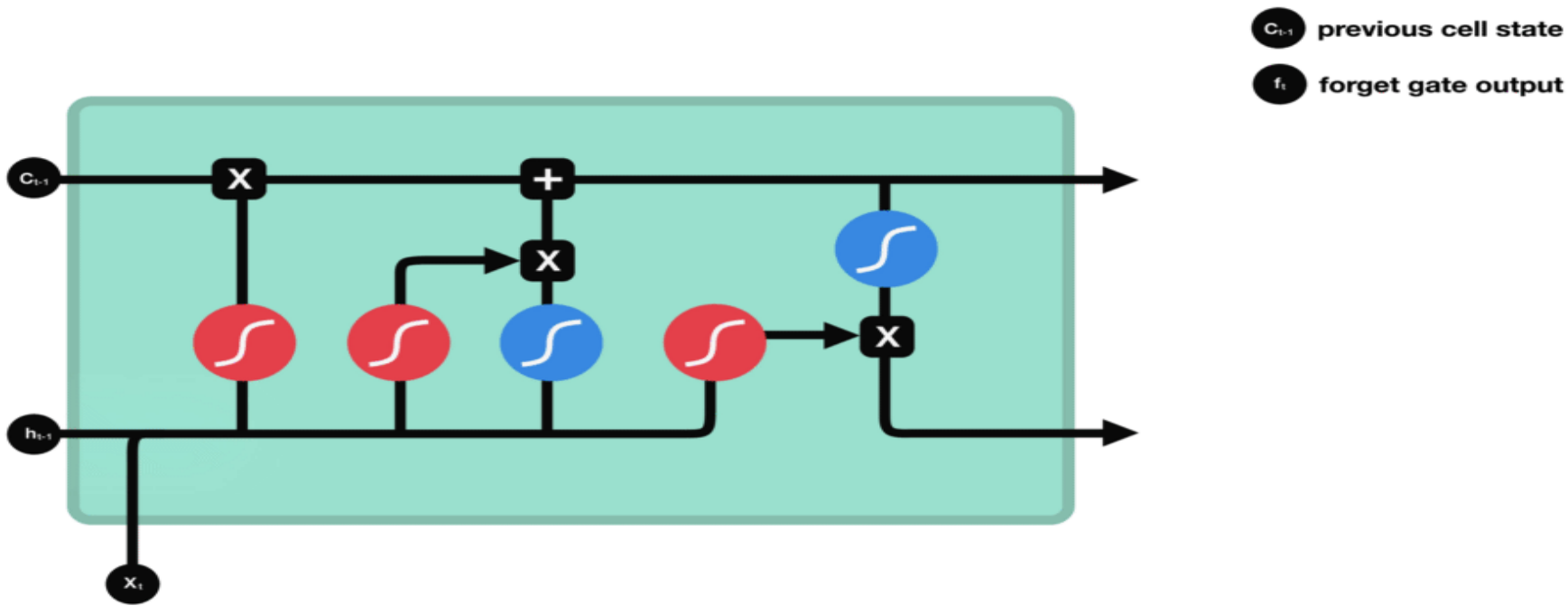


The cell state and its multiple gates are the essential concepts of LSTMs. The cell state functions as a highway that transports relative information down the sequence chain.

You might think of it as the network's "memory." In principle, the cell state can carry meaningful information throughout the sequence's processing. As a result, information from earlier time steps might make its way to later time steps, lessening the short-term memory effects. Information is added or withdrawn from the cell state via gates as the cell state travels.The gates are different from other neural networks that determine whether information about the cell state is ok to keep. During training, the gates might learn what information is important to keep or forget.

Forget gate:

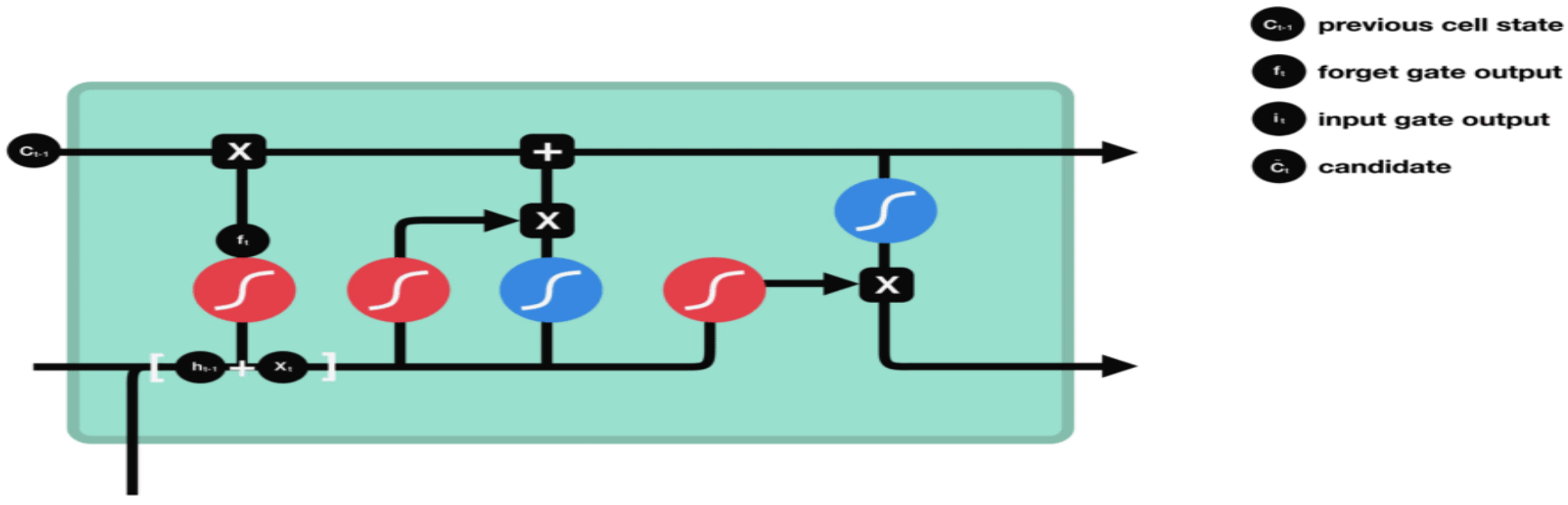
The first one is the forget gate. This gate determines whether information should be deleted or kept same. The sigmoid function passes information from the previous hidden state as well as information from the current input. The results will be between 0 and 1. During pointwise operation between these values and the forget cell, the closer to 0 represents to forget, and the closer to 1 means to retain.



In overview the forget gate carries information like memory decides which information to be kept and forgotten next the input gate decides what information should be added to the forget gate from new information plus previous hidden state information. The output gate determines what information should be carried forward to the next hidden layer.

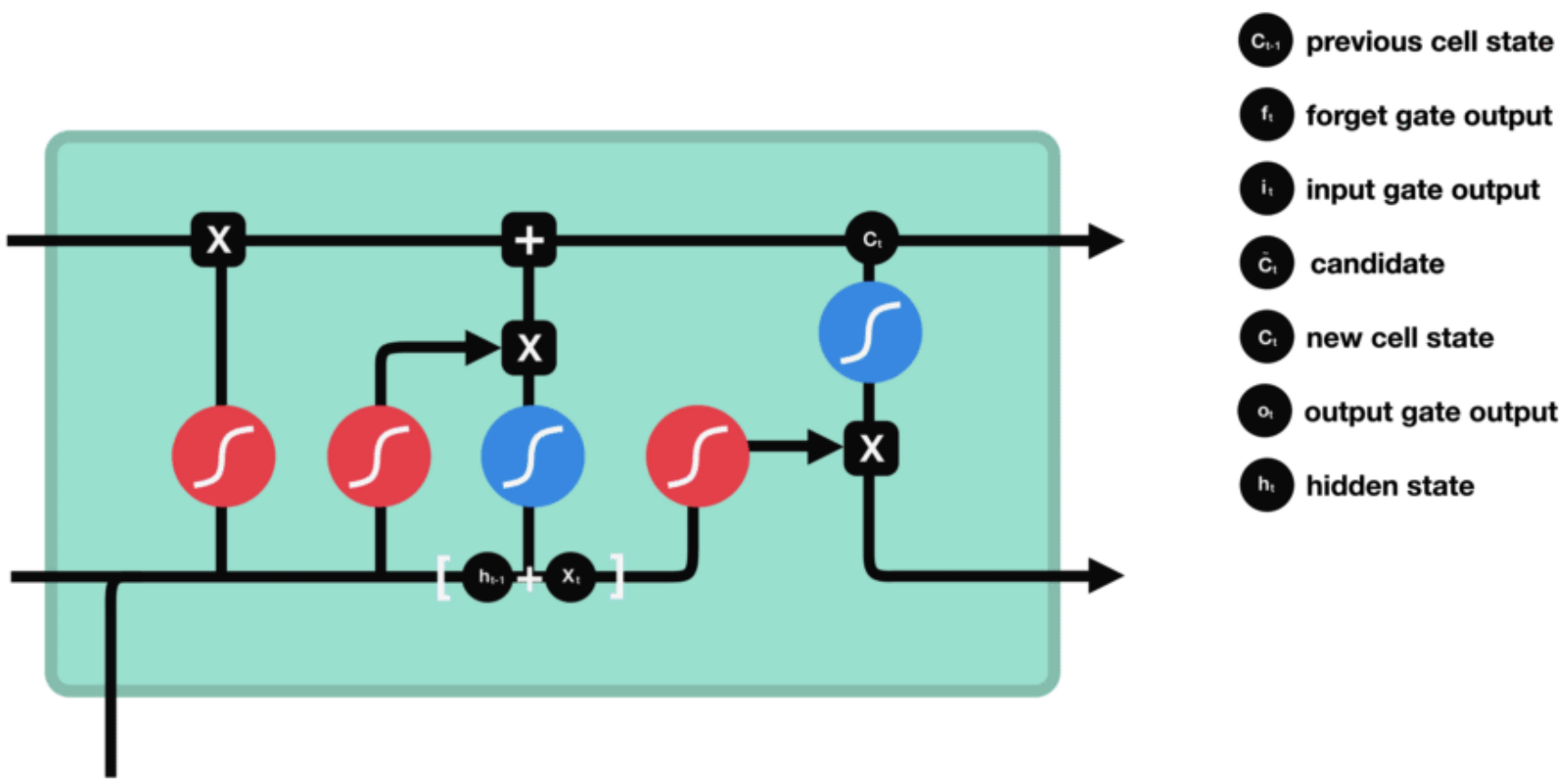
Input gate:

To update the information We have the input gate in the cell state. First, we use a sigmoid function to combine the past hidden state and the current input. This determines which values will be updated by converting them to a value between 0 and 1, with 0 indicating that values should be ignored and 1 indicating that values should be remembered.Then send the hidden state and current input into the tanh function to convert values between -1 and 1 it also helps to regulate the network by converting them into small values like between -1 to 1 but In our case used Relu activation function converts values -ve to 0 +ve to same for to get precise values because these values are the coordinates of our landmarks of our hand ,face, pose.Then we do pointwise operation between Relu output with the sigmoid output will decide which information is important to retain same from the Relu output.



Output Gate:

Last but not least, there's the output gate. The hidden state's next hidden state is determined by the output gate. The hidden state (forget state) keeps track of prior inputs. Predictions are also made using the hidden state. First, we use a sigmoid function to combine the prior hidden state and the current input. Then we call the tanh function with the newly updated forget gate.in our case Relu function.after that operating pointwise operation between Relu output with the sigmoid output to determine what information will the hidden state should transmit to next layer.



Sigmoid:

Gates contains sigmoid activations.It specifies a range of numbers between 0 and 1. This is useful for updating or forgetting data since any integer multiplied by 0 equals 0 and values are forgotten. Because every integer multiplied by one has the same value, it will be preserved. The network will figure out which data is unimportant and should be deleted, and which data should be kept.

6. Deploying the model

We've deployed our model in flask, so it will be running on localhost.

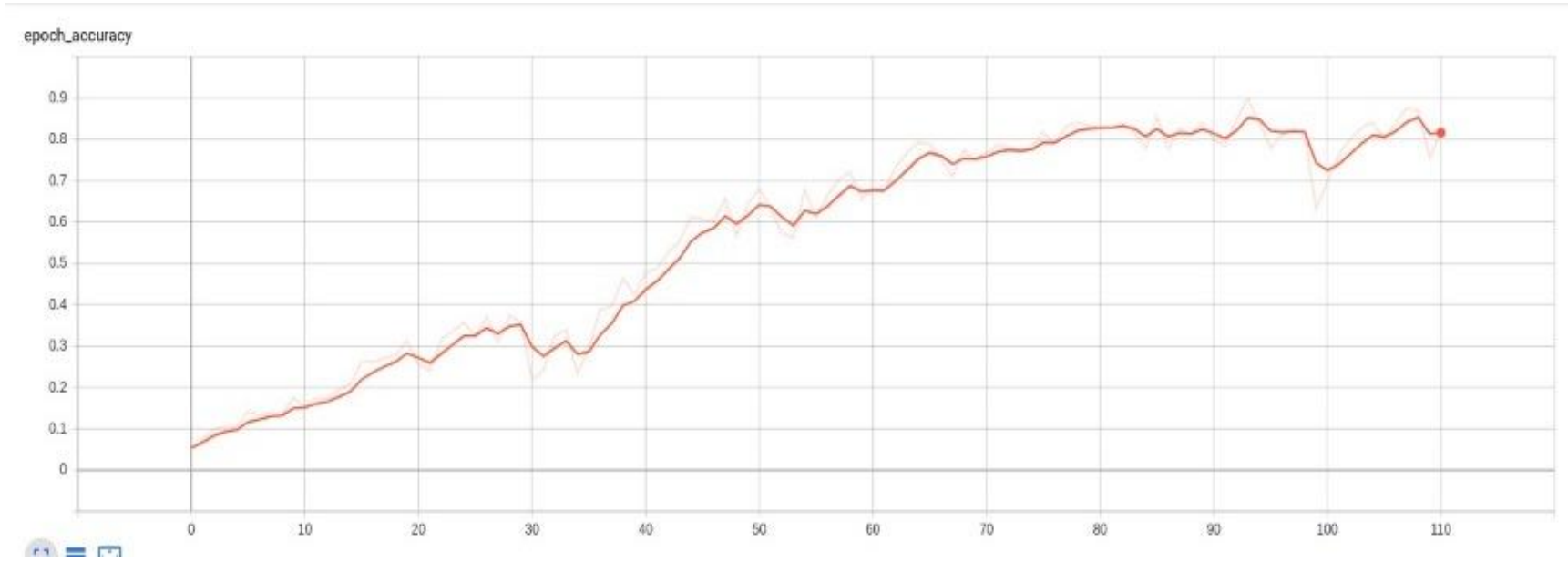
7. Results and Discussion

The final LSTM had a validation loss of 0.1453 and validation accuracy of 93.43% for 200 Epochs.

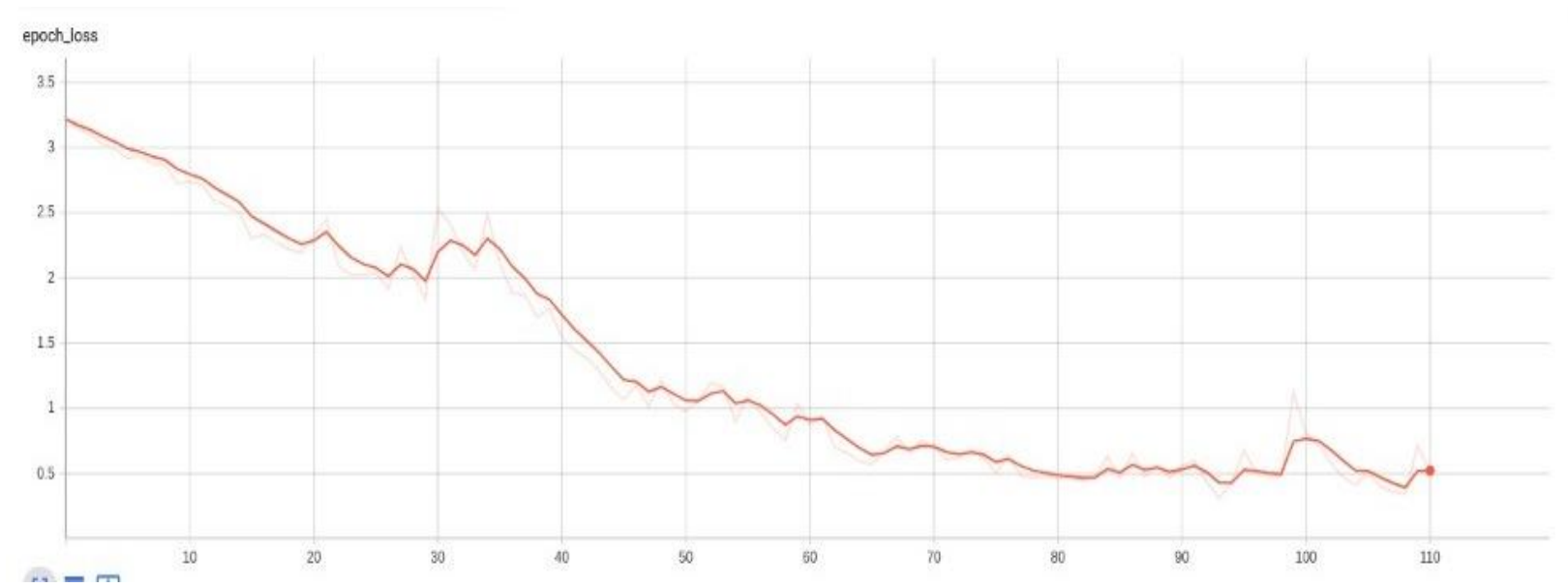
At first, we tested by taking pretrained MobileNet followed by Lstm layers with same no.of sequences we used now i.e 30 different sequences per sign that will be 630 sequences total and we are getting nowhere near the accuracy that was going to be useful. So then we transisted to using Mediapipe holistic combined with LSTM layers, Because it needed less data to produce hyperactive model and it was a much denser neural network so rather than having a 30 to 40 million parameters in a neural network we had around only half of the million neural networks which means it was going to be way faster to train a particular model. And also it was a lot simpler which will be faster when it comes to detecting signs in real time.

7.1.Graphs

1. Accuracy



2. Epoch Loss



7.2.Accuracy per sign

	precision	recall	f1-score	support
0	1.00	1.00	1.00	2
1	1.00	0.80	0.89	5
2	1.00	0.50	0.67	2
3	1.00	1.00	1.00	1
4	1.00	1.00	1.00	2
5	1.00	0.67	0.80	3
6	1.00	1.00	1.00	2
7	1.00	1.00	1.00	3
8	0.00	0.00	0.00	0
10	1.00	1.00	1.00	1
11	0.00	0.00	0.00	0
12	1.00	1.00	1.00	1
13	1.00	1.00	1.00	2
14	1.00	1.00	1.00	3
15	1.00	1.00	1.00	4
16	1.00	1.00	1.00	1
17	1.00	1.00	1.00	5
18	1.00	1.00	1.00	1
19	1.00	1.00	1.00	1
20	1.00	1.00	1.00	1
21	0.00	0.00	0.00	0
accuracy			0.93	40
macro avg	0.86	0.81	0.83	40
weighted avg	1.00	0.93	0.95	40

7.3 Confusion_matrix

```
Out[70]: array([[[38,  0],
                 [ 0,  2]],

                [[35,  0],
                 [ 1,  4]],

                [[38,  0],
                 [ 1,  1]],

                [[39,  0],
                 [ 0,  1]],

                [[38,  0],
                 [ 0,  2]],

                [[37,  0],
                 [ 1,  2]],

                [[38,  0],
                 [ 0,  2]],

                [[37,  0],
                 [ 0,  3]],

                [[39,  1],
                 [ 0,  0]],

                [[39,  0],
                 [ 0,  1]],

                [[39,  1],
                 [ 0,  0]],

                [[39,  0],
                 [ 0,  1]],

                [[38,  0],
                 [ 0,  2]],

                [[37,  0],
                 [ 0,  3]],

                [[36,  0],
                 [ 0,  4]],

                [[39,  0],
                 [ 0,  1]],

                [[35,  0],
                 [ 0,  5]],

                [[39,  0],
                 [ 0,  1]],

                [[39,  0],
                 [ 0,  1]],

                [[39,  0],
                 [ 0,  1]],

                [[39,  1],
                 [ 0,  0]]])
```

8. Future work

In future we wish to improve in specific areas like:-

1.Training the model with more data (Sign) videos with high FPS and achieving better accuracy.

2.We're likewise considering adding NLP to our model so that it looks more realistic like it will read text, hear speech etc…

3.We wish to make our SLR as an Android app so that anyone can download it from the play store and use it whenever required.

9. Summary and Conclusions

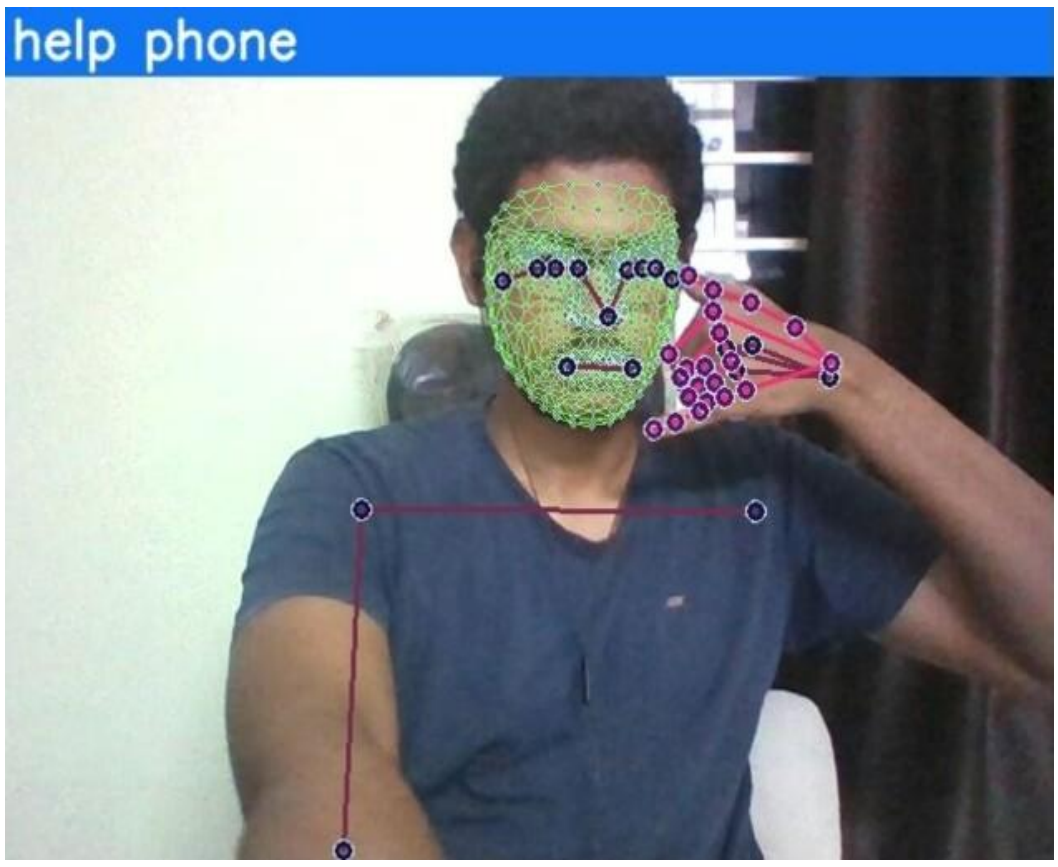
In this project, we've developed a Sign Language detection using Mediapipe Holistic for extracting Key points and using those collected key points we've trained and our Lstm model and that was successful and we got an accuracy of 93%. The question of perfection is another attempt to deal with it in the days to come.

And also the sign language action recognition system (SLR) provides an easy and satisfactory user communication for deaf and dumb people. This SLR provides two way communications which helps in easy interaction between the normal people and disables. The system is a novel approach to ease the difficulty in communicating with those having speech and vocal disabilities. Our aim is to provide an application to the society to establish the ease of communication between the deaf and mute people by making use of image processing algorithms. Since it follows an image based approach it can be launched as an application in any minimal system and hence has near zero-cost.

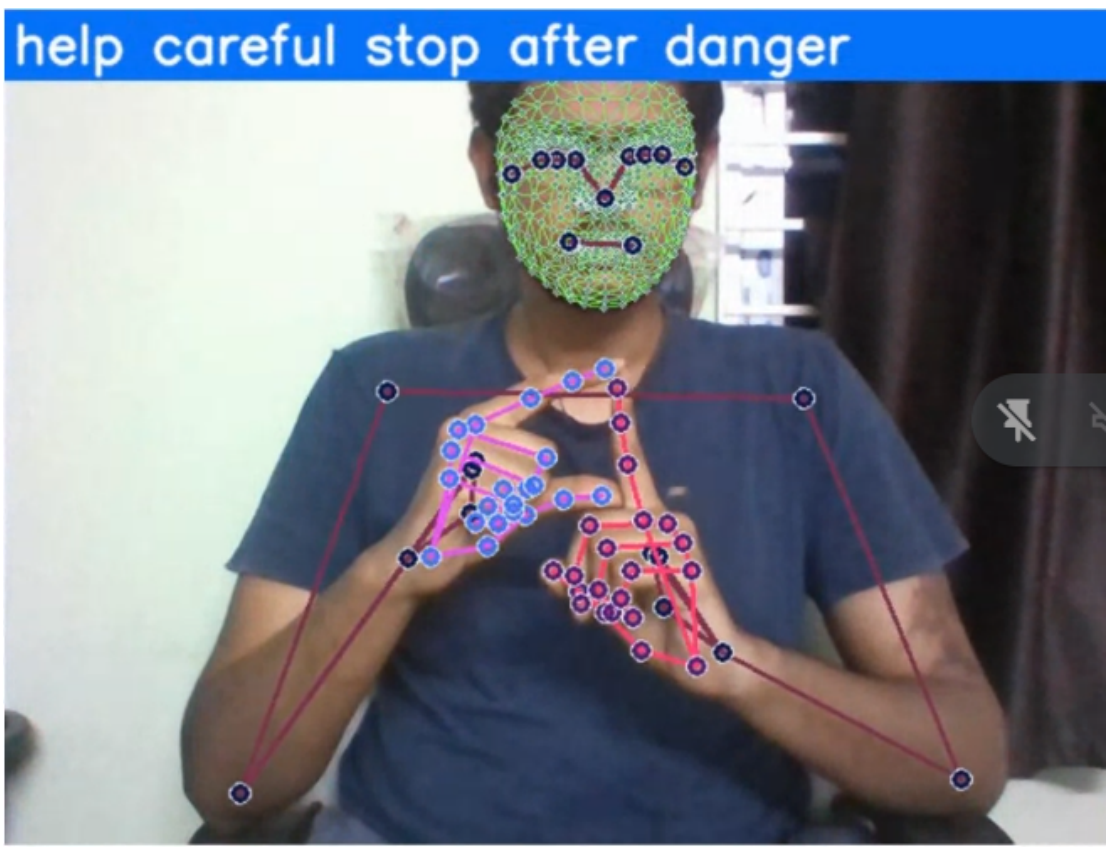
10.Appendix

Output sample 1

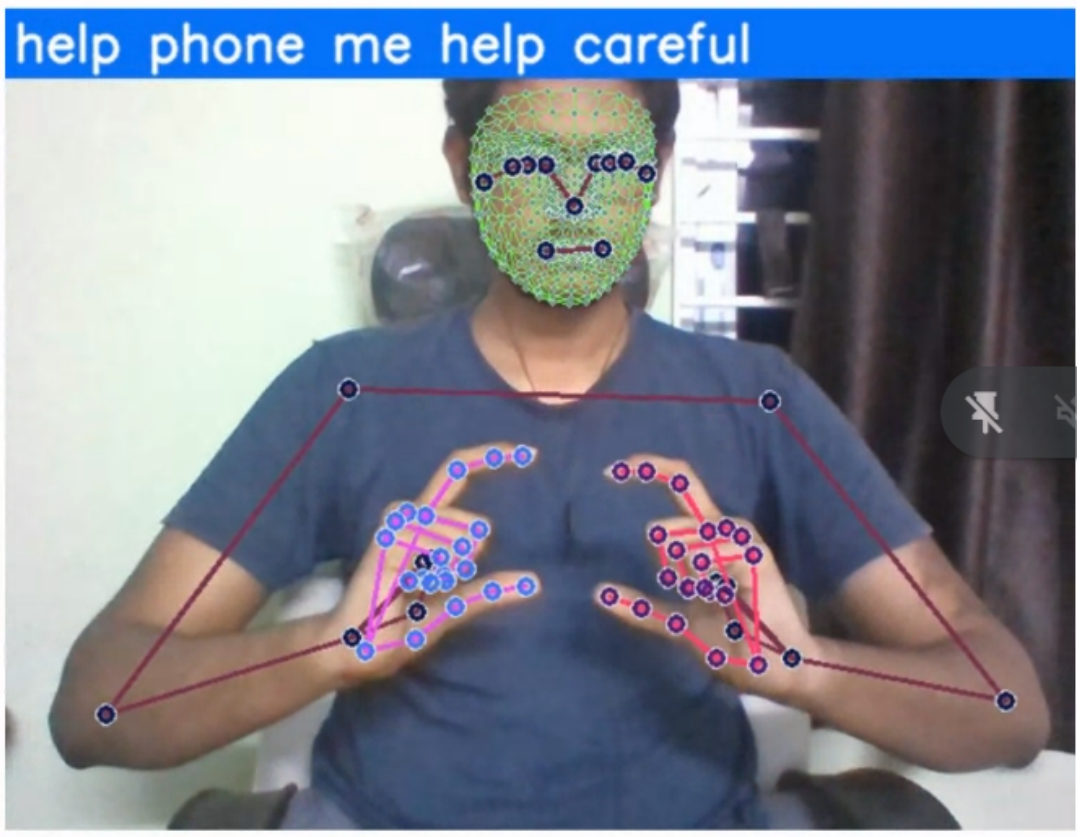
Detecting Phone sign



2. Detecting danger sign



3. Detecting Careful sign



4. Detecting Water sign

11. References

1.https://arxiv.org/ftp/arxiv/papers/2107/2107.13647.pdf

2. https://www.irjet.net/archives/V9/i1/IRJET-V9I1133.pdf

3.https://bansal-pranav.medium.com/indian-sign-language-recognition-using-googles-mediapipe-framework-3425ddce6748

4.https://www.researchgate.net/publication/355402809_Development_of_a_software_module_for_recognizing_the_fingerspelling_of_the_Russian_Sign_Language_based_on_LSTM

5.https://www.sciencedirect.com/science/article/pii/S2667305321000454

6.https://www.researchgate.net/publication/353567881_Egyptian_Sign_Language_Recognition_Using_CNN_and_LSTM

7.https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

8.https://www.ijrte.org/wp-content/uploads/papers/v7i6/F2746037619.pdf

ACKNOWLEDGEMENT

We wish to express our gratitude to Dr.B.Jayalakshmi, Assistant Professor, Department of Computer Science Engineering, IIIT Dharwad, for her constant support and valuable guidance throughout this Project. We are thankful to her for taking out time to guide us, motivate us, and provide valuable feedback despite her busy schedule. The Project benefited a lot from her experienced inputs.

We respect and thank Dr.B.Jayalakshmi for providing us with an opportunity to work on this project titled “SIGN LANGUAGE ACTIONS RECOGNITION” and ensure that we get all the help required to learn and develop something, duly completing the project on time.

M.VENKATA KALYAN BABU
18BCS049
Department of Computer Science Engineering

T.UMESH ANAND BABU
18BCS105
Department of Computer Science Engineering

M.P.BHARATH
18BCS057
Department of Computer Science Engineering

Y.MOKSHITH RAMENDRA
18BCS112
Department of Computer Science Engineering