A

Mini Project Report on

Sign language recognition

Submitted in partial fulfillment of the requirements for the degree

of

Second Year Engineering – Information Technology

by

Harsh Gajera 22104099

Sekhar Gauda 22104044

Aditya Gupta 22104155

Paresh Gupta 22104089

Under The Guidance Of

Ms. Sneha Dalvi

DEPARTMENT OF INFORMATION TECHNOLOGY

A. P. SHAH INSTITUTE OF TECHNOLOGY

G.B. Road, Kasarvadavali, Thane (W), Mumbai-400615

UNIVERSITY OF MUMBAI

Academic Year: 2023-24

CERTIFICATE

This to certify that the Mini Project report on **Sign language recognition** has been submitted by **Harsh Gajera (22104099)**, **Sekar Gauda (22104044)**, **Aditya Gupta (22104155)** and **Paresh Gupta (22104089)** who are a Bonafede students of A. P. Shah Institute of Technology, Thane, Mumbai, as a partial fulfillment of the requirement for the degree in **Information Technology**, during the academic year **2023-2024** in the satisfactory manner as per the curriculum laid down by University of Mumbai.

Ms. Sneha Dalvi
Guide

Dr. Kiran Deshpande — **HOD(Information Technology)**

Dr. Uttam D. Kolekar — **Principal**

External Examiner:
1.

Internal Examiner:
1.

Place: A. P. Shah Institute of Technology, Thane
Date:

ACKNOWLEDGEMENT

This project would not have come to fruition without the invaluable help of our guide **Ms. Sneha Dalvi**. Expressing gratitude towards our HoD, **Dr. Kiran Deshpande**, and the Department of Information Technology for providing us with the opportunity as well as the support required to pursue this project. We would also like to thank our teacher **Mr. Mandar Ganjapurkar** who gave us her valuable suggestions and ideas when we were in need of them. We would also like to thank our peers for their helpful suggestions.

TABLE OF CONTENTS

ABSTRACT

In today's world, hand gestures are used a lot, especially with new technology like Artificial Intelligence and Machine Learning. This helps in many areas like helping deaf people communicate, controlling robots, and even in medical work. This report talks about a modern way to understand American Sign Language (ASL) using computers. It looks at both still and moving hand signs. They used a tool called MediaPipe from Google to find where the hand is, and they made their own set of pictures to test it. They used a type of computer program called LSTM from Tensorflow to understand the signs, and it worked really well. This is a big step in helping people who can't speak to be understood better. Communication is really important for those people who can't hear or speak. They use sign language, but many people don't understand it. Computers can help by learning to understand sign language and turn it into words. Some systems can already do this for Indian Sign Language (ISL), but they are slow. This paper talks about making a system that can understand American Sign Language in real-time. They used a regular camera to watch people signing, and they trained the computer to understand it quickly using a method called transfer learning. Even though they didn't have a lot of examples, the computer still did a good job, which is promising for helping people communicate better.

1.INTRODUCTION:

One of the most crucial pillars of daily life is communication because it allows people to express their ideas and opinions and thus helps them integrate into society. The ability to hear and speak, however, is not shared by all people, and thereby some find it difficult to use. As a result, they are unable to communicate normally and struggle to fit into society.

Sign language recognition is crucial in the field of assistive technology for persons with hearing impairments. This technology enables seamless communication and access to diverse services for this demographic. People with hearing impairments may have substantial difficulties in their daily activities if they do not have proficient sign language recognition technology. They may struggle to communicate their needs, understand information, or even participate in social activities. By developing and improving sign language recognition technology, we can empower these individuals to lead more independent and fulfilling lives, bridging the communication gap and promoting inclusivity within society.

The goal is to study the usage of LSTM (Long Short Term Memory) networks in the recognition of Indian Sign Language (ISL) at the word level. The system's primary job is to quickly and correctly detect, categorise and translate the signs performed in ISL by utilizing neural networks and Computer Vision. The present system in this study can currently handle the recognition of up to 24 ISL vocabulary words in real-time within a matter of a few seconds. Gesture-based sign language recognition systems face numerous hurdles, particularly in the context of American Sign Language (ASL) where the alphabets are widely different from Indian Sign Language (ISL). The primary obstacle stems from the intricate and dynamic hand movements integral to ISL. Another issue pertains to the diversity in signing approaches among different individuals, which complicates the development of a universally applicable recognition model.

Furthermore, the existence of background disturbances and obstructions amplifies the challenge of achieving precise recognition. Nevertheless, the adoption of Long Short-Term Memory (LSTM) networks has demonstrated promising outcomes in enhancing the accuracy and efficiency of ASL recognition systems.

1.1.Purpose

Sign language recognition aims to bridge communication gaps between the deaf and hearing communities by translating sign language into text. This technology detects and interprets the intricate hand movements and gestures to inherent in sign language. Its purpose is to facilitate real-time communication for the deaf, enabling them to interact more seamlessly with others, access educational resources, and participate fully in society. Sign language recognition systems strive for accuracy and efficiency, empowering individuals with hearing impairments to express themselves fluently and be understood by a wider audience.

- **Emergency Situations:** In emergency situations, clear communication is essential for ensuring the safety and well-being of all individuals involved. Sign language recognition allows emergency responders to communicate with deaf individuals quickly and accurately, providing instructions, assistance, and reassurance during crises such as natural disasters or medical emergencies.
- **Education:** In education, sign language recognition facilitates teaching and learning for both deaf and hearing students, providing accessible resources such as online tutorials, interactive lessons and digital textbooks.
- **Personal Development:** Sign language recognition encourages personal development and self-expression among deaf individuals by providing them with tools to communicate confidently and effectively in various contexts. It empowers them to pursue educational, professional, and personal goals with independence and agency.
- **Public Services:** Sign language recognition technology enhances access to public services such as transportation, government offices, and social welfare programs. It ensures that deaf individuals can communicate effectively with service providers, access information, and avail themselves of essential services without facing communication barriers or discrimination.
- **Accessibility:** Sign language recognition enhances accessibility for the deaf and hard of hearing, allowing them to engage with digital content, communicate through technology, and access various services independently.
- **Healthcare:** In healthcare sectors, accurate communication is crucial for providing quality care. Sign language recognition assists healthcare professionals in communicating with deaf patients, ensuring that medical information is conveyed accurately and patient's needs are understood and addressed properly.

1.2.Problem Statement

Problem statement:

People who cannot speak often communicate using symbols and gestures. Understanding these symbols requires knowing their unique language and culture well. Without the right context, these gestures can be misunderstood. This shows how important it is to be culturally sensitive and to communicate effectively when interacting with people who are deaf and unable to speak. Using sign language interpreters can help bridge the communication gap and improve understanding in many situations.

The Solution:

Our project aims to create a virtual talking system that doesn't require sensors, making it accessible to individuals who struggle to communicate verbally. By employing image processing techniques and human hand gestures as input, this concept offers a practical solution for those who are unable to speak with others due to various reasons such as speech disabilities, medical conditions, or language barriers.

By providing an intuitive interface based on hand gestures, this system empowers users to express themselves effectively, promoting equal opportunities and enabling seamless communication in varied environment. Additionally, the implementation of such a system could potentially improve the quality of life for individuals with communication difficulties, enhancing their ability to interact and engage with others in meaningful ways.

1.3. Objectives

- To detect the hand gesture using the camera module on the device.
- Using appropriate datasets for recognizing and interpreting data using LSTM algorithm.
- To deliver accurate text output for the respective hand gesture sign language.
- To attain maximum accuracy as possible.

1.4. Scope

- **Python Fundamentals:** An in-depth analysis of Python programming that covers functions, control statements, data structures, and object-oriented ideas.
- **Model Deployment and Evaluation:** Gaining knowledge about how to deploy models in production environments as well as understanding evaluation metrics for Machine Learning and Deep Learning models.
- **Domain-specific Applications:** Exploring the applications of Python Machine Learning and Deep Learning in domains like computer vision, natural language processing, recommendation systems, and more.
- **Real-Time Recognition:** The system will aim to recognize sign language gestures in real- time, enabling instantaneous communication.
- **Basic Vocabulary:** Initially, the project will focus on recognizing a predefined set of common sign language gestures representing basic words and phrases
- **Single-Handed Gestures:** The project will primarily concentrate on recognizing single- handed sign language gestures.

2.LITERATURE REVIEW:

Sr no.	Title of Research Paper	Key findings	Author	Publication Year
1.	Realtime Sign Language Detection and Recognition	The research paper presents a real-time sign language recognition system for Indian Sign Language (ISL) based on deep learning models such as TensorFlow, Keras, and LSTM for training.	Aakash Deep Aashutosh Litoriya Akshay Ingole Vaibhav Asare Shubham M Bhole Shantanu Pathak	2022
2.	Machine Learning solutions with MediaPipe	This document introduces the basic concepts of the MediaPipe framework developed by Google.	Y Quiñonez, C Lizarraga R Aguayo	2022
3	How to Improve the Accuracy of Your Image Recognition Models	This Research paper provides strategies to improve image recognition models include data augmentation, adding more layers, adjusting image size, increasing epochs, reducing color channels, employing transfer learning, and fine-tuning hyperparameters	Jason D'Souza	2023

3.PROPOSED SYSTEM:

The proposed system for the ASL alphabet recognition is shown in Figure . The proposed system recognizes the American sign language of alphabets. It recognizes gestures with motion and gestures without motion as well. The first step is to collect the dataset for the system which will be discussed in results and discussion section. MediaPipe hands is a high- fidelity hand and finger tracking framework used in the dataset preparation. It makes use of a number of Machine Learning to infer the 21 three dimensional landmarks of a hand in real time from just one frame.

3.1Features and Functionality

1. **Hand Gesture Recognition:**
    - Utilizing the MediaPipe framework, the system employs a machine learning pipeline for hand gesture recognition.
    - It employs a palm detection model to identify hands within the entire image, outputting a hand-bounding box.
    - Subsequently, a hand landmark model operates on the cropped hand region, providing accurate 3D hand keypoints.
    - When necessary, the hand is localized using palm detection if the landmark model fails to do so.
2. **Model Comparison:**
    - In contrast to TensorFlow, an open-source library primarily focused on deep neural network training and inference, MediaPipe offers cross-platform, customizable solutions for live and streaming media.
    - The system leverages MediaPipe for capturing hand keypoints and TensorFlow for training and detecting the machine learning algorithm.
3. **Hardware Compatibility:**
    - MediaPipe is compatible with both CPU and GPU, requiring no additional processing power.
4. **Dataset Creation:**
    - The experimental dataset utilized in the study is generated using the MediaPipe open-source library.
    - The dataset comprises 26 alphabets with corresponding hand gestures, captured over 100 frames.
    - Parameters include the (x, y, z) coordinates of hand joints.
5. **Training LSTM Model:**
    - LSTM is a type of feed-forward neural network suitable for recognizing continuous motion in hand gestures.
    - Accommodates sequences of data, allowing recognition of hand motion gestures.
    - Extracts hand keypoints from the training dataset.
    - Trains on 2000 epochs until accuracy nears 1 and training loss approaches 0.
    - Employs RELU activation for simplicity and enhanced training on CPU.
    - Employs TANH activation for simplicity and enhanced training on GPU.
    - Utilizes Adam Optimizer to update network weights based on training data, adjusting learning rates independently for each parameter.

4. REQUIREMENT ANALYSIS:

1. **Sign Language Recognition:**
   The system aims to accurately recognize and interpret American Sign Language (ASL) gestures in real-time. Utilizing Mediapipe for hand landmark detection and LSTM for gesture recognition, it facilitates both static and dynamic gesture recognition. The dataset consists of 26 ASL alphabets, capturing gestures for 100 frames per alphabet. Additionally, the system ensures efficient dataset management through the integration of Mediapipe for hand landmark detection and data collection.

2. **Real-time Detection:**
   Real-time detection functionality enables immediate interpretation of ASL gestures. The system leverages Mediapipe for efficient hand landmark detection and LSTM for real-time gesture recognition. An intuitive user interface facilitates seamless interaction, while continuous monitoring of webcam input ensures timely detection and interpretation of ASL gestures.

3. **Multi-gesture Recognition:**
   Supporting a wide range of ASL gestures enhances communication comprehensiveness. Through training on diverse ASL gestures, the system proficiently recognizes multiple gestures within a single interaction. This capability augments usability and facilitates smoother communication experiences for users.

4. **Output and Performance Metrics:**
   To ensure accurate gesture detection, the system defines an output format for displaying recognized ASL gestures. Performance evaluation utilizes precision, recall and support metrics based on a confusion matrix analysis. The system aims for high precision and recall across all ASL alphabets, ensuring reliable gesture recognition.

5.PROJECT DESIGN:

5..DFD (Data Flow Diagram):

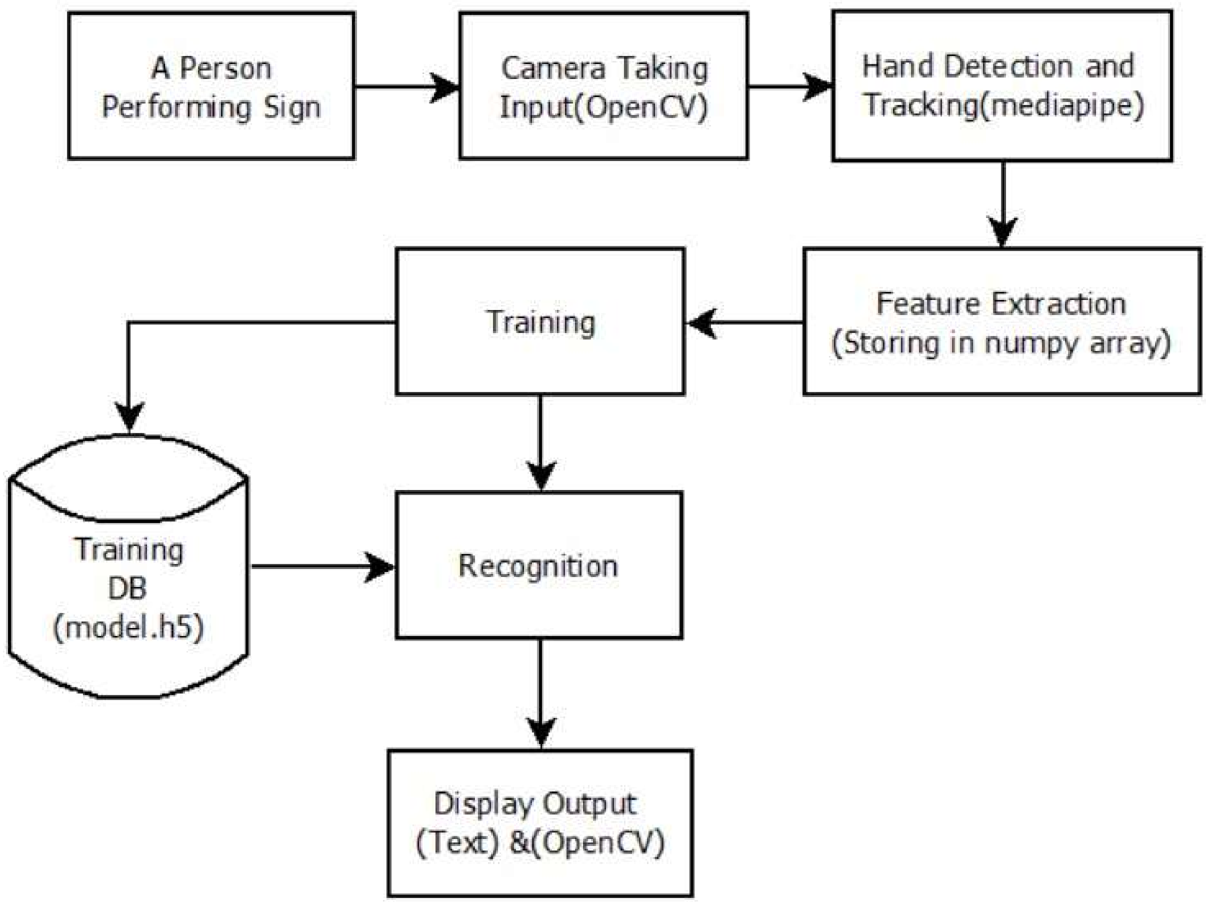


Fig 5.1 Data Flow Diagram.

5.3. System Architecture

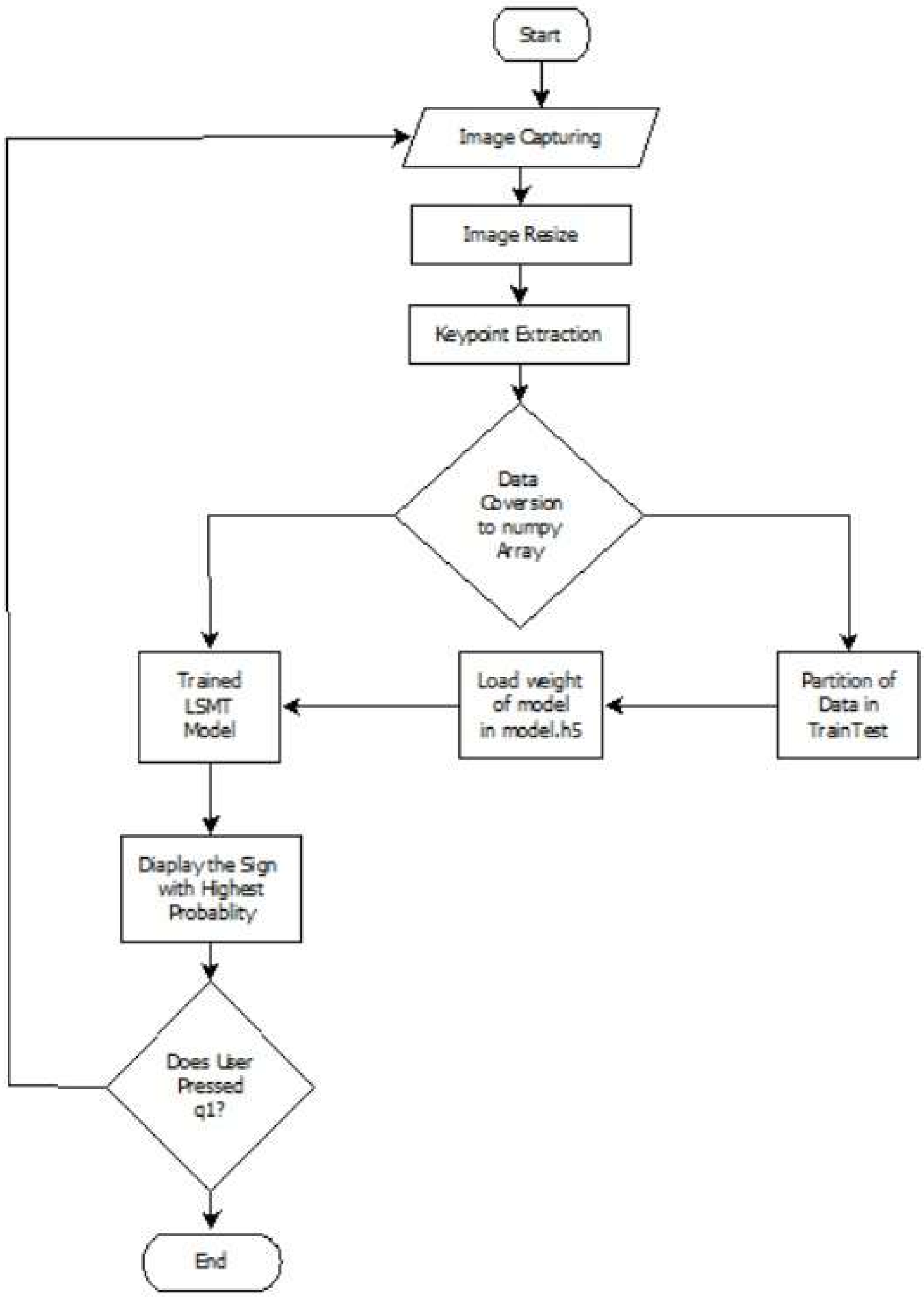


Fig 5.2. System Architecture.

5.4. Implementation:

Mediapipe hand key point detection works with a machine learning pipeline. A palm detection model that works on the entire image is one of the models in the MediaPipe machine learning pipeline. It receives the entire image and outputs a hand-bounding box that is orientated. a hand landmark model that uses the palm detector-cropped image region and outputs highly accurate 3D hand keypoints. Additionally, the hand is cropped by the ML pipeline using the hand landmarks found in the previous frame, and palm detection is only used to localize the hand when the landmark model is unable to do so.

Fig 5.4. American Sign Language.

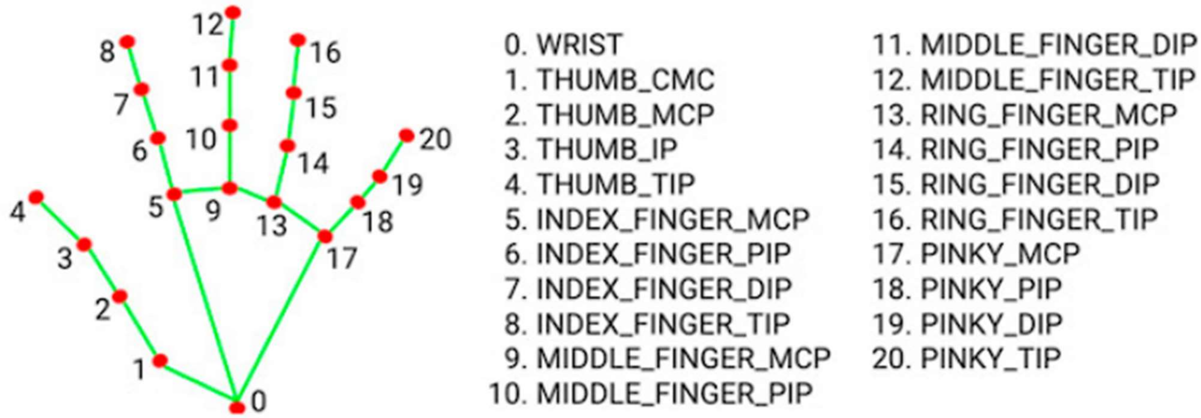


Fig 5.5. Hand Landmarks representation used in Mediapipe.

Fig 5.5. shows the 21 hand landmarks that can be tracked from the Mediapipe hand landmark detector. Arsheldy Alvin(Software Engineer) have used Mediapipe and K nearest neighbours algorithm to determine the hand gestures which helps us in understanding how to use Mediapipe for hand gesture recognition. The difference between tensorflow and Mediapipe is that tensorflow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks.

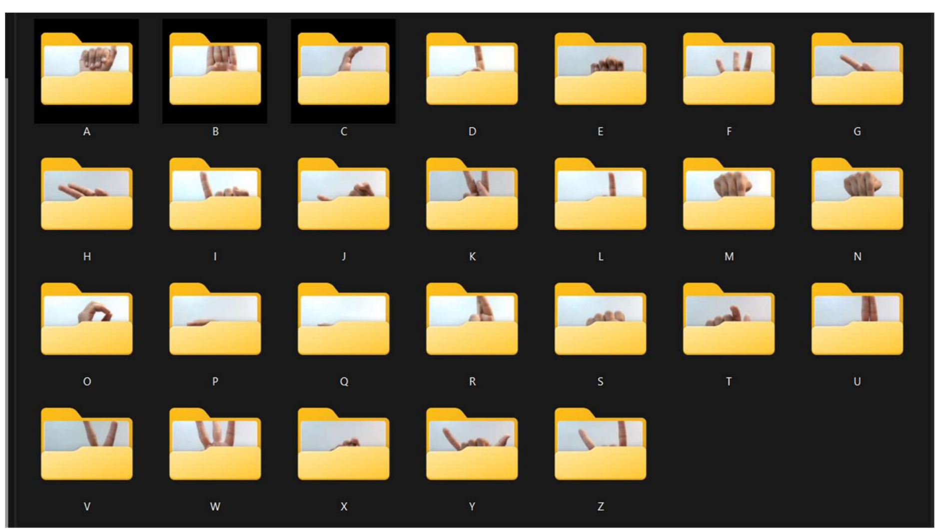


Fig 5.6. Sign Dataset in PNG format.

Fig 5.7. Extraction of landmarks using Mediapipe.

The dataset required for the experimental study of the proposed algorithm is created using MediaPipe which is an open-source library. It captures the hand gestures for 30 frames and stores them as NumPy array for training the model. The dataset has 100 different permutations of a single alphabet and each permutations consist of 101 numpy array. This makes a total of 2,62,600 numpy array, these are indexed and stored. The dataset has different parameters that are x, y and z axes of the hand joints. The dataset is divided into 95% for training and 5% for testing. In Figure 5.8, the numpy array format of Dataset for alphabet 'A'.

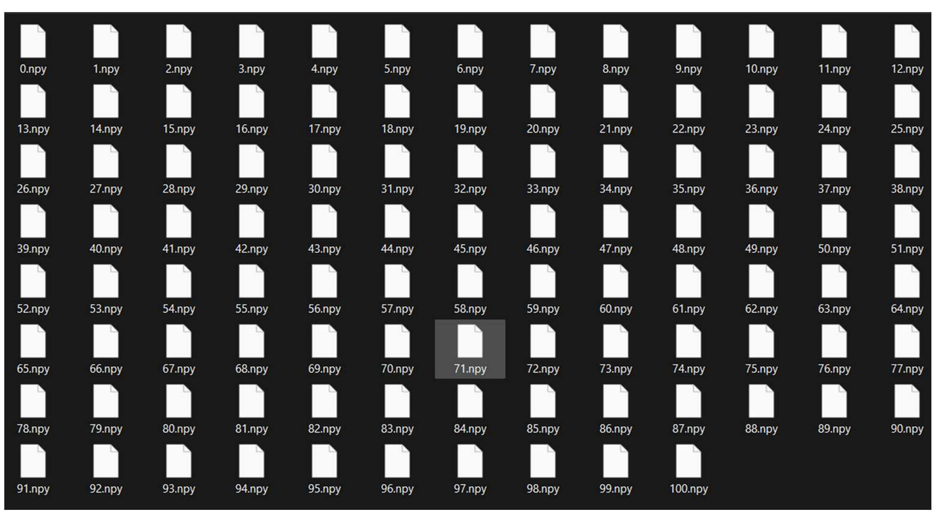


Fig 5.8. Sign Dataset in Numpy array(.npy) format.

LSTM is one of the feed-forward neural networks. Since, in American alphabet hand gestures there are several hand gestures that have hand motion in them instead of a single static pose. So, its better to use LSTM using which we can recognize the continuous motion. The advantage of using LSTM is that it allows not only single data points i.e images, but it also allows the entire sequences of data. An artificial neural network is a network of interconnected neurons, each of which is connected to one another and is modelled after the brain's organic neural networks. We can execute sophisticated operations on the data by combining a number of different algorithms, each of which is capable of completing a different task. A class of neural networks called Recurring Neural Networks, or RNNs, is designed specifically to handle temporal data. The input is processed using the internal state of the RNN neurons, which is accomplished with the aid of loops built into the neural network. RNN neurons have a cell state or memory. From the training dataset, Spatio-temporall features are extracted and the LSTM model classification model is derived. The generated ML model is stored for prediction of gestures in real time.Fig 5.9. shows the Summary of Sequential model made using 3 LSTM Layer and 3 Dense Layers

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 100, 64)           32768

 lstm_1 (LSTM)               (None, 100, 128)          98816

 lstm_2 (LSTM)               (None, 64)                49408

 dense (Dense)               (None, 64)                4160

 dense_1 (Dense)             (None, 32)                2080

 dense_2 (Dense)             (None, 26)                858

=================================================================
Total params: 188,090
Trainable params: 188,090
Non-trainable params: 0
_________________________________________________________________
PS C:\Users\Paresh Gupta\Desktop\SignLanguageDetectionUsingML-main>
```

Fig 5.9. Summary of Model.

6.TECHNICAL_SPECIFICATION:

The dataset is created for American Sign Language where signs are alphabets of the English language. The dataset is created following the data acquisition method described in Section 8 (Result and Discussion). The experimentation was carried out on a system with an AMD Ryzen 5 5500H 3.30GHz processor, 8 GB memory and webcam (HP TrueVision HD camera with 0.31 MP and 640x480 resolution), running Windows 11 operating system. The programming environment includes Python (version 3.9.0), Jupyter Notebook, OpenCV (version 4.2.0), TensorFlow Object Detection API.

6.1 Functional Requirement:

a. **Machine Learning Model:**
    - Long Short-Term Memory (LSTM)

b. **Programming Environment:**
    - Python: Version 3.9.0
    - Jupyter Notebook

c. **Libraries and Frameworks:**
    - OpenCV
    - TensorFlow API
    - Time
    - Mediapipe
    - Mlxtend
    - Sktlearn

6.2 Non-Functional Requirement:

a. **Operating System:**
    - Windows 11

b. **Hardware:**
    - Processor: AMD Ryzen 5 5500H 3.30GHz
    - Memory: 8 GB
    - Webcam: HP TrueVision HD camera with 0.31 MP and 640x480 resolution

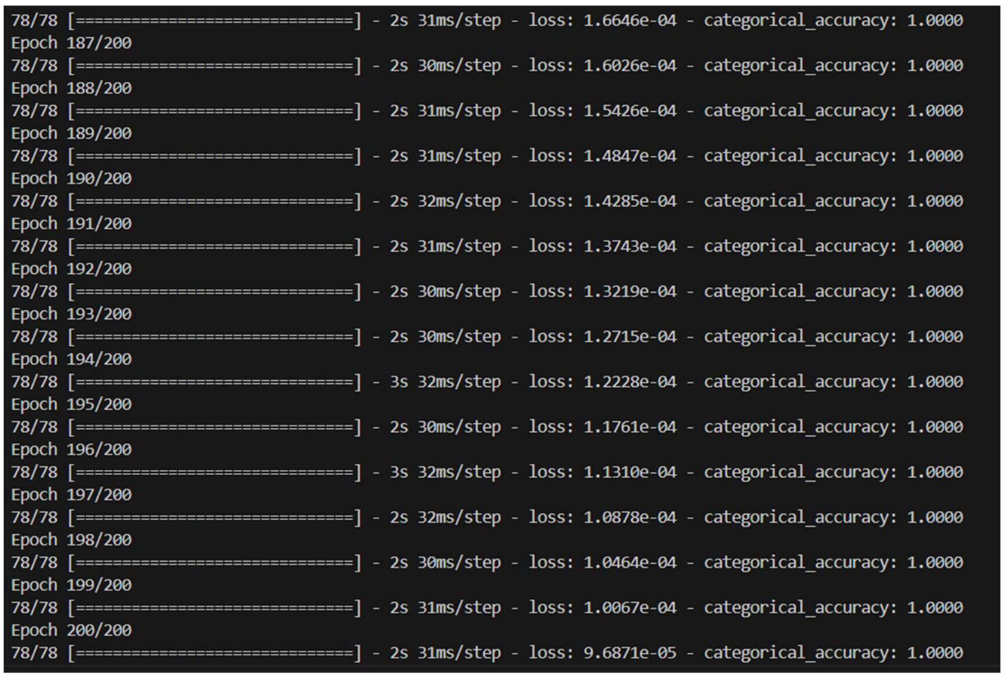
7.PROJECT SCHEDULING:

ID	Name	Color
1	Information Gathering and Research	
2	Data Collection	
3	Image Labeling	
4	Configuring SSD Model	
5	Training Model	
6	Evaluating and Updating Model Configuration	
7	Testing and Finalization	

Fig.7.1. Gantt Chart for Project Scheduling for Sign language recognition.

8.RESULT AND DISCUSSION

All the 26 alphabet gestures are collected one after the other and are preprocessed into Numpy arrays and stored. The dataset collected is then passed onto the LSTM model. The LSTM model will have all the 26 alphabet classes for each of the alphabets. The algorithm is made to run on 250 epochs until the accuracy becomes closer to 1 and training loss goes to nearly 0. The algorithm runs with the above specified features and generates the model which is stored for future use. The model can now be used for tracking the live hand gestures and for further operations.



```
78/78 [==============================] - 2s 31ms/step - loss: 1.6646e-04 - categorical_accuracy: 1.0000
Epoch 187/200
78/78 [==============================] - 2s 30ms/step - loss: 1.6026e-04 - categorical_accuracy: 1.0000
Epoch 188/200
78/78 [==============================] - 2s 31ms/step - loss: 1.5426e-04 - categorical_accuracy: 1.0000
Epoch 189/200
78/78 [==============================] - 2s 31ms/step - loss: 1.4847e-04 - categorical_accuracy: 1.0000
Epoch 190/200
78/78 [==============================] - 2s 32ms/step - loss: 1.4285e-04 - categorical_accuracy: 1.0000
Epoch 191/200
78/78 [==============================] - 2s 31ms/step - loss: 1.3743e-04 - categorical_accuracy: 1.0000
Epoch 192/200
78/78 [==============================] - 2s 30ms/step - loss: 1.3219e-04 - categorical_accuracy: 1.0000
Epoch 193/200
78/78 [==============================] - 2s 30ms/step - loss: 1.2715e-04 - categorical_accuracy: 1.0000
Epoch 194/200
78/78 [==============================] - 3s 32ms/step - loss: 1.2228e-04 - categorical_accuracy: 1.0000
Epoch 195/200
78/78 [==============================] - 2s 30ms/step - loss: 1.1761e-04 - categorical_accuracy: 1.0000
Epoch 196/200
78/78 [==============================] - 3s 32ms/step - loss: 1.1310e-04 - categorical_accuracy: 1.0000
Epoch 197/200
78/78 [==============================] - 2s 32ms/step - loss: 1.0878e-04 - categorical_accuracy: 1.0000
Epoch 198/200
78/78 [==============================] - 2s 30ms/step - loss: 1.0464e-04 - categorical_accuracy: 1.0000
Epoch 199/200
78/78 [==============================] - 2s 31ms/step - loss: 1.0067e-04 - categorical_accuracy: 1.0000
Epoch 200/200
78/78 [==============================] - 2s 31ms/step - loss: 9.6871e-05 - categorical_accuracy: 1.0000
```

Fig 8.1 Training of model with 250 epochs.

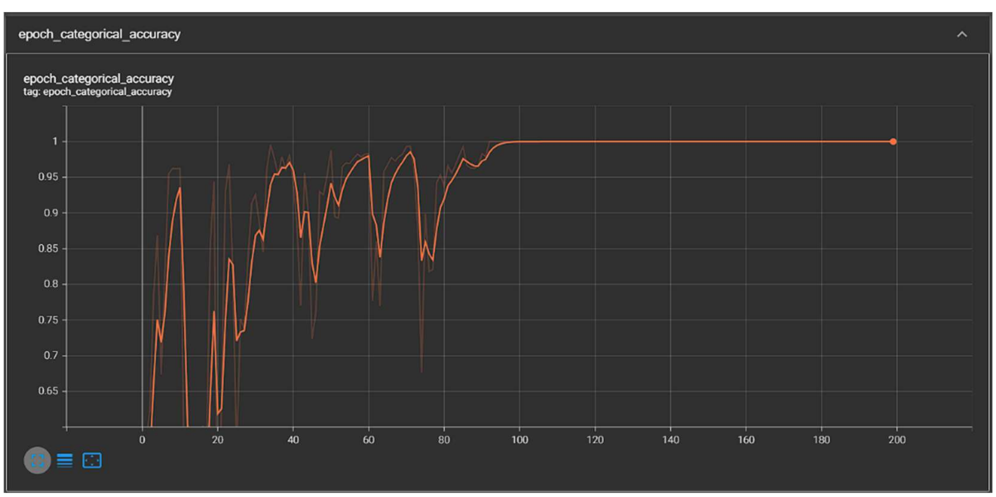


Fig 8.2. Accuracy Graph while Training.

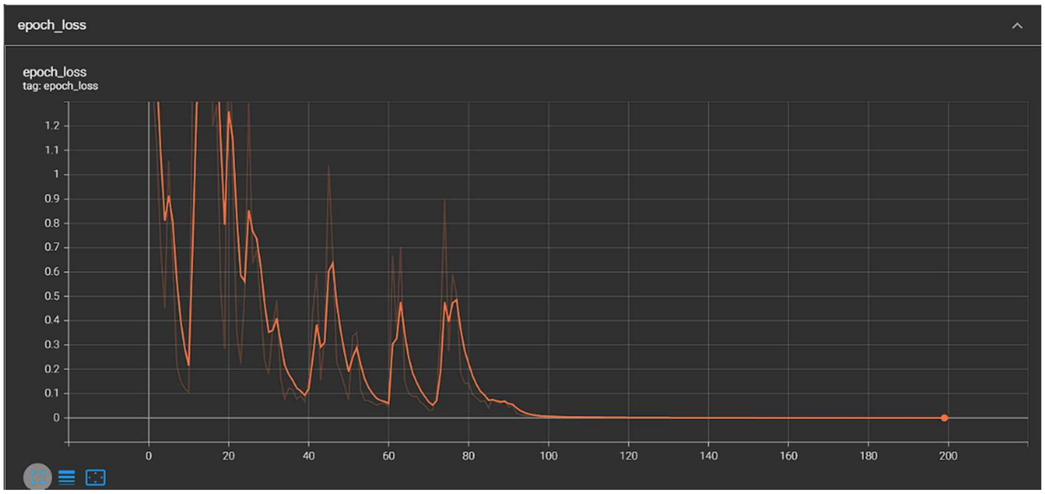


Fig 8.3. Accuracy Graph while Training.

The tanh activation function introduces nonlinearity to the network, allowing it to learn complex patterns and relationships within the data. It is advantageous in that it can handle both positive and negative inputs, producing values between -1 and 1. This property enables the model to capture a wider range of information and gradients during training. The system uses Adam Optimizer technique. In place of the conventional stochastic gradient descent method, Adam is an optimization technique that may be used to repeatedly update network weights depending on training data. For all weight updates, stochastic gradient descent maintains a constant learning rate, which does not change throughout training. As learning progresses, a learning rate is maintained and independently adjusted for each network weight (parameter).The main reason for using the LSTM model is that we can detect motion from hand gestures, and it can be used to detect hand gestures that have motion in them. The model that we are using is sequential and they are faster to train and have faster detections. The network implemented here consists of six layers three LSTM layers and three dense layers. In Figure 8.6 you can find the summary of the model with the layer type , output shape and params. Fig 8.. shows the sample image result of gesture recognition for the alphabet from A.

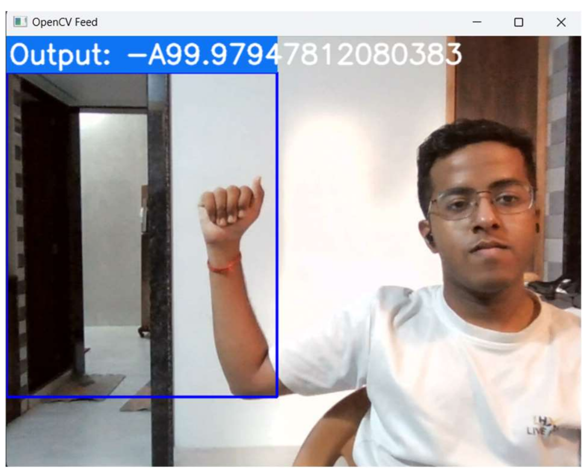


Fig 8.4. Live Detection using Webcam.

True Positive (Correct Detection with Input at Correct Region)	False Positive (Correct Detection with Input at Wrong Region)
True Negative (No Detection with No Input)	False Negative (No Detection with inputs)

The ratio of True Positives to All Positives is known as precision.

$$Precision = \frac{TruePositive(TP)}{TruePositive(TP) + FalsePositive(FP)}$$

The recall is the measure of the model correctly identifying True Positives.

$$Recall = \frac{TruePositive(TP)}{TruePositive(TP) + FalseNegative(FN)}$$

The F – score is calculated from the precision and recall of the test. The F – score or F – measure is a measure of a test's accuracy.

$$F1Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

Alphabet	Precision	Recall	Column A	Support
A	1	1	1	13
B	1	1	1	12
C	1	1	1	13
D	1	1	1	15
E	1	1	1	8
F	1	1	1	12
G	1	1	1	16
H	1	1	1	9
I	1	1	1	10
J	1	1	1	11
K	1	1	1	9
L	0.97	1	0.97	15
M	0.94	1	0.94	17
N	1	1	1	12
O	1	1	1	10
P	1	1	1	10
Q	0.92	1	0.96	12
R	1	0.9	0.95	10
S	1	1	1	10
T	1	1	1	11
U	0.95	1	0.97	18
V	1	0.83	0.91	12
W	1	0.92	0.96	13
X	1	1	1	9
Y	1	1	1	13
Z	1	1	1	12
Accuracy			0.99	312
Micro Average	0.99	0.99	0.99	312
Weighted Average	0.99	0.99	0.99	312

Fig 8.5. Precision and Recall Table.

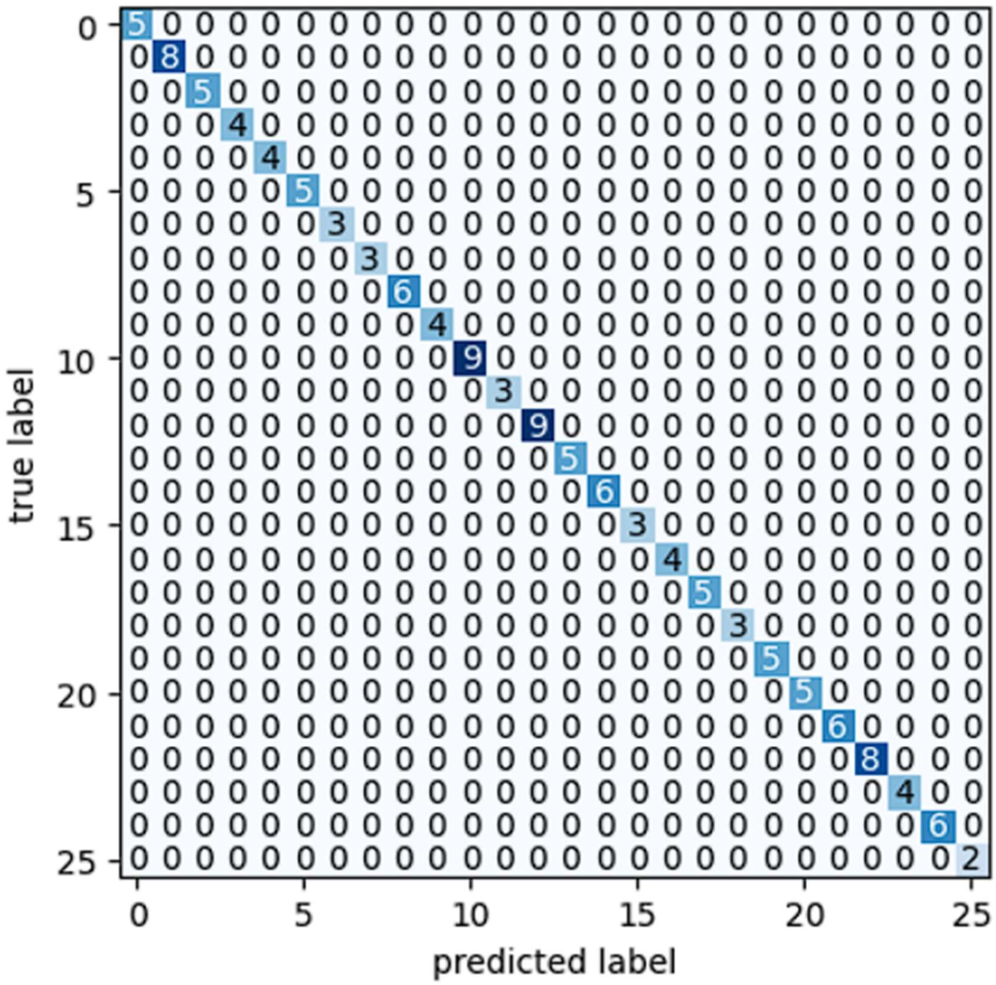


Fig. 8.6. Confusion Matrix.

9. CONCLUSION:

Our study successfully developed a system capable of recognizing American Sign Language (ASL) gestures. We achieved this using a combination of two key technologies: LSTM models and Mediapipe.

LSTMs, or Long Short-Term Memory models, are a type of artificial neural network designed to process sequences of data. In our case, they enabled the computer to learn the patterns inherent in ASL gestures. By training the LSTM model with examples of ASL gestures over 250 rounds, each with 78 examples, we reached optimal performance.

Mediapipe, on the other hand, provided the means to extract crucial information from video inputs. Specifically, we utilized Mediapipe's capabilities to identify and track key points on the hands, which are vital for understanding sign language gestures. These key points were then stored using NumPy arrays, a fundamental tool for numerical computing in Python.

In addition to LSTM models and Mediapipe, we leveraged OpenCV to interface with the camera, capturing video input and facilitating the creation of a dataset for training. TensorFlow, a popular open-source machine learning framework, served as our primary tool for building and training the ASL recognition model.

By combining these technologies, we established a solid foundation for further exploration in ASL recognition. Our project not only showcases the potential of LSTM models and Mediapipe in gesture recognition but also underscores the importance of interdisciplinary approaches in advancing assistive technologies.

10. FUTURE SCOPE:

Expanding the ASL Sign Language Detection system involves several key ways to make it better. First, adding more signs to the dataset will help the system recognize a wider range of gestures accurately. Also, trying out different models made for specific tasks or with different strengths could make it work even better.

Another way to improve it is by making it work with different sign languages. This means adding signs from other languages to the dataset, so people who use different sign languages can use it too.

Making the system faster and more efficient, especially on devices like phones and tablets, is really important too. This means making it recognize signs quickly and accurately in real-time.

Lastly, looking into ways to combine different types of data, like hand movements or depth data, could also help make the system better at recognizing signs. These are all important ways to make the ASL Sign Language Detection system better in the future.

REFERENCES

[1] “Hand gesture recognition of static letters American sign language (ASL) using deep learning,” Engineering and Technology Journal, Vol. 38, No. 06, A. A. Abdulhussein and F. A. Raheem, pp. 926-937, 2020.

[2] “Sign Language Recognition Using Leap Motion Controller with Machine Learning Approach, Department of Electronics Engineering”, Teak-Wei Chong and Boon-Giin Lee, American Keimyung University, Daegu 42601, Korea, October 2018.

[3] “Gesture Recognition of Sign Language Alphabet Using a Magnetic Positioning System” Matteo Rinalduzzi, Alessio De Angelis, Francesco Santoni, Emanuele Buchicchio, Antonio Moschitta, Paolo Carbone, Paolo Bellitti, Mauro Serpelloni, June 2021.

[4] “American Sign Language Alphabets Recognition using Hand Crafted and Deep Learning Features IEEE Xplore Part Number:CFP20F70-ART" Rajesh George Rajan, Dr.M.Judith Leo, published on 2020.

[5] “Hand Gesture Recognition of English Alphabets using Artificial Neural Network” Sourav Bhowmick, Sushant Kumar and Anurag Kumar, published on 2015.

[6] “Draft manuscript accepted for publication in Sign Language Studies, Volume 6, Number 3”, Ross E. Mitchell, Gallaudet Research Institute,2006.

[7] “A Novel Feature Extraction for American Sign Language Recognition Using Webcam” , Onamon Pinsanoh, Yuttana Kitjaidure, Ariya Thongtawee. Published in 2018.

[8] “Recognition of Dynamic Hand Gestures from 3D Motion Data using LSTM and CNN architectures.” Chinmaya R. Naguri, Razvan C. Bunescu , 2017.

[9] “.Hand Gesture Tracking and Recognition based Human-Computer Interaction System and Its Applications” , Kai Li, Qieshi Zhang, Jun Cheng, Jianming Liu , (2018).

[10] “A Method for Stochastic Optimization. Published as a conference paper at the 3rd International Conference for Learning Representations” , Diederik P. Kingma, Jimmy Ba. Adam , San Diego, 2015.

[11] “Linguistics Of American Sign Language, American National Standard for Information Sciences”, Clayton Valli, Ceil Lucas,

[12] “Hand Gesture Recognition for Human Computer Interaction. 7th International Conference on Advances in Computing Communications” , Aashni Hariaa , Archanasri Subramaniana , Nivedhitha Asokkumara, Shristi Poddara , Jyothi S Nayaka , ICACC- 2017, 22- 24 August 2017, Cochin, India.

[13] “Hand Gesture Detection for American SignLanguage using K- Nearest Neighbor with Mediapipe” , Arsheldy Alvin1, Nabila Husna Shabrina2, Aurelius Ryo3, Edgar Christian4 Fakultas Teknik dan Informatika, Universitas Multimedia Nusantara, Teknik Komputer Tangerang, Indonesia.

[14] “A Real-time Hand Gesture Recognition and Human- Computer Interaction System” , Pei Xu, Department of Electrical and Computer Engineering, University of Minnesota, Twin Cities.

[15] “Google Research 1600 Amphitheatre Pkwy, Mountain View, CA 94043, USA” , Fan Zhang ,Valentin Bazarevsky ,Andrey Vakunov ,Andrei Tkachenka ,George Sung, Chuo-Ling Chang, Matthias Grundmann.

[16] “Hand gesture recognition using machine learning algorithms. Computer Science and Information Technologies” , Abhishek B, Kanya Krishi, Meghana M, Mohammed Daaniyaal, Anupama H S , BMS Institute of Technology, Bangalore, India.

[17] “Hand gesture recognition based on convolution neural network. part of Springer Nature” , Gongfa Li, · Heng Tang, · Ying Sun, · Jianyi Kong, · Guozhang Jiang, · D u Jiang · Bo Tao, ·Shuang Xu, · Honghai Liu , (2017)

[18] “Dynamic Hand Gesture Recognition Using 3DCNN and LSTM with FSM Context-Aware Model”, Noorkholis Luthfil Hakim , Timothy K. Shih , Sandeli Priyanwada Kasthuri Arachchi ,Wisnu Aditya , Yi-Cheng Chen and Chih-Yang Lin.

[19] “LONG SHORT-TERM MEMORY, Neural Computation 9(8):1735-1780" ,Sepp Hochreiter ,Fakultat fur Informatik, Technische Universitat Munchen, 80290 Munchen, Germany, 1997.

[20] “The Swiss AI Lab IDSIA,Istituto Dalle Molle di Studi sull’Intelligenza Artificiale,University of Lugano SUPSI,Galleria 2, 6928 Manno-Lugano,Switzerland” , Deep Learning in Neural Networks: An Overview.Jurgen Schmidhuber,8 October 2014.

[21] Deep Learning using Rectified Linear Units (ReLU). Abien Fred M. Agarap . arXiv:1803.08375v2 [cs.NE] 7 Feb 2019.

[22] “Deep learning based part-of-speech tagging for Malayalam Twitter data (Special issue: deep learning techniques for natural language processing).” Journal of Intelligent Systems 28, no. 3 : 423-435, Kumar, S., M. Anand Kumar, and K. P. Soman. (2019)

[23] “A deep learning approach for Malayalam morphological analysis at character level.” Procedia computer science 132 : 47-54" , Premjith, B., K. P. Soman, and M. Anand Kumar. (2018)

[24] “LSTM based paraphrase identification using combined word embedding features.” In Soft computing and signal processing, pp. 385-394 , Aravinda Reddy, D., M. Anand Kumar, and K. P. Soman. Springer, Singapore, 2019.4

[25] “Real Time Sign Language Recognition and Speech Generation.” Journal of Innovative Image Processing 2, no. 2: 65-76 Thakur, Amrita, Pujan Budhathoki, Sarmila Upreti, Shirish Shrestha, and Subarna Shakya. (2020)