Vision-Based Real Time Bangla Sign Language Recognition System Using MediaPipe Holistic and LSTM

By

Sk. Estiaque Ahmed Sajal
ID: 191-15-2380
And
Md. Walid Foysol
ID: 191-15-2660

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Mohammad Jahangir Alam
Senior Lecturer
Department of CSE
Daffodil International University

Co-Supervised By

Amit Chakraborty Chhoton
Senior Lecturer
Department of CSE
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY
DHAKA, BANGLADESH
February, 2023

APPROVAL

This Project titled "**Vision-based real time bangla sign language recognition system using mediapipe holistic and LSTM**", submitted by Sk. Estiaque Ahmed Sajal, ID: 191-15-2380, Md. Walid Foysol, ID: 191-15-2660 to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 1 February, 2023.

BOARD OF EXAMINERS

4/2/23

Chairman

Dr. Touhid Bhuiyan
Professor and Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Tania Khatun (TK)
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

1.2.23

Ms. Lamia Rukhsara(L R)
Senior Lecturer
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

External Examiner

Dr. Mohammad Shorif Uddin
Professor
Department of Computer Science and Engineering
Jahangirnagar University

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Mohammad Jahangir Alam, Senior Lecturer, Department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:

Mohammad Jahangir Alam
Senior Lecturer
Department of CSE
Daffodil International University

Co-Supervised by:

Amit Chakraborty Chhoton
Senior Lecturer
Department of CSE
Daffodil International University

Submitted by:

Sajal

Sk. Estiaque Ahmed Sajal
ID: 191-15-2380
Department of CSE
Daffodil International University

Walid

Md. Walid Foysol
ID: 191-15-2660
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes it possible for us to complete the final year internship successfully.

We are really grateful and wish our profound indebtedness to **Mohammad Jahangir Alam**, **Lecturer (Senior Scale)**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of "Deep Learning" to carry out this project. His endless patience, scholarly guidance ,continual encouragement , constant and energetic supervision, constructive criticism , valuable advice ,reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

We would like to express our heartiest gratitude to **Dr. Touhid Bhuiyan,** Professor and Head**,** Department of CSE, for his kind help to finish our project and also to other faculty members and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discussion while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

ABSTRACT

Bangla Sign Language Detection system converts the Bangla sign language into text, so that deaf-mute people can communicate with ordinary people. Deaf-mute people are quite detached from society because there is a communication gap between normal people and a deaf-mute people. On the strength of technological welfare, now it is possible to capture any deaf-mute people's gesture and with the help of machine learning, it can be converted into text. In this research, we adopt a development model for recognizing gestures that accommodates MediaPipe for extracting hands and posing landmarks and long short-term memory (LSTM) to train and recognize the gesture. This will convert Bangla sign language gestures into readable text. The requirements analysis served as the foundation for our proposed model, which will be carried out in four stages: data collection and preprocessing; training and testing of the suggested neural network; and finally, testing in real time. A gesture model is taught to recognize gestures with the help of a self-created dataset of Bangla sign language. The trained model successfully identifies the gesture, and the text equivalent of the gesture is displayed on screen. The purpose of this model is to help create a medium to communicate between normal people and the hearing impaired.

TABLE OF CONTENTS

CONTENTS

PAGE

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Introduction

On the subject of expressing inner voice, deaf and mute people face more difficulties than normal people which creates communication gaps in the society. In the case of helping them communicate smoothly, proper application of technology is the key to make these people feel part of the society. Deafness in Bangladesh is one of the major health issues. The estimated result from the 2011 census shows that about 9.6% of the population of Bangladesh is deaf or hard of hearing. Most of them had to live their life as a burden in society. Because they can’t communicate with other people with words. Normally a deaf or mute people communicate using hand gestures or sign language creating different movements and positions of hands that indicate different meanings. According to WHO, in this world about 15% of people have a disability in speaking and it is estimated that up to five out of every 1000 babies are born with hearing loss or acquire it soon after birth. A child's growth and academic success can be significantly impacted by hearing loss. Around the World today, a good number of sign languages are in use. The Centre for Disability in Development (CDD) in Bangladesh has standardized a sign language for the Bengali deaf and mute community that is used in all of the country's schools for the speech and hearing impaired.[4] A visual language, sign language makes use of body language, gestures, face emotion, and hand forms. People who are hard of hearing communicate with others via sign language.[6] It is the basic method of communication for them, but it is difficult to interpret for most of the people, so the communication needs to be developed, and that's where machine learning may become handy.[6] Pre-trained machine translators can be helpful to ease the interpretation of what a person is trying to express, therefore easing the communication.[6] To recognize indications of numbers, alphabets, words, or other signs, such as hand gestures at traffic signals, sign language recognition systems are utilized.[5] While some scientists are concentrating on static images, others are working on real-time symbol identification. [5] In recent years, many works have been executed on image classification tasks incorporating embedded systems and machine learning algorithms.[3] More often, researchers preferred to use deep learning techniques in their work which eventually outperformed conventional feature extraction-based algorithms and showed better precision and accuracy.[3] hand motion Humans can communicate with machines and computers using only their hands and a recognition system. Communication with computers may take on new dimensions thanks to hand gestures. The primary goal of this simultaneous hand gesture detection system is to quickly and accurately identify hand gestures.[12] In order to determine how a hand moves and then identify patterns of the same movement, hand recognition uses algorithms and neural networks. [12] Our desire is to construct a hand gesture identification system which will only require a camera to take input and to provide data to the model to identify the sign language.

1.2 Motivation

The idea that sign language is universal is a frequent fallacy, yet this is untrue. In actuality, much like spoken languages, sign languages are specific to a society and have developed with time. Additionally, they have their own grammar and vocabulary and are typically learned as the mother tongue of deaf children. Around the World today, a good number of sign languages are in use. In Bangladesh, the Centre for Disability in Development (CDD) has developed a formal sign language for the Bengali deaf and mute community, which is followed by schools for the speech and hearing impaired countrywide.

The general public's lack of understanding of sign language makes it challenging to integrate the deaf and dumb into society. As a result, scientists have been working on strategies to construct automated sign language recognition systems in order to bridge this communication gap. This subject of study is currently behind and suffering. Furthermore, research on the recognition of Bangla sign language has not been as successful as it has been for other sign languages. As a result, our purpose is to do research on recognizing Bangla sign language. A variety of technologies may and have been used in sign language or gesture recognition. To detect signs, we are utilizing a real time vision-based technique with Long short-term memory (LSTM) based neural network for our thesis.

1.3 Rationale of the Study

This study seeks to investigate the feasibility and accuracy of using sign language detection to enable people who are deaf or hard-of-hearing to communicate more easily. Sign language is a vital form of communication for many people in the deaf and hard-of-hearing community, yet it is not widely used due to the difficulty of recognizing gestures. This study aims to explore the potential of using the machine learning algorithm, LSTM, to accurately detect and interpret sign language, thus providing a valuable tool for communication between people with different hearing abilities. This research will also help to further develop the area of computer vision technology, which has the potential to revolutionize the way people interact with machines and technology.

1.4 Research Questions

- What is the situation of deaf and mute people in Bangladesh?
- What is the status of Bangla Sign Language?
- How does MediaPipe and LSTM work?
- How does MediaPipe and LSTM work in this detection?
- How can a normal person communicate with a deaf people using technology?

1.5 Expected Outcome

Researchers hoping to learn more about sign language and how it is understood and communicated with hope to find answers in studies of sign-language identification. This study will also shed light on the similarities and distinctions between the sign languages of other countries. People with communication challenges, such as deafness, will benefit from this study since it is expected that this research will reveal patterns in sign language that can be used to improve comprehension and expression. The findings of this study may also inform the creation of innovative tools to facilitate communication between speakers of different languages.

1.6 Research Objective:

- To find the situation of deaf and mute people in Bangladesh.
- To develop a Bangla Sign Language detection system prototype.
- Try to reduce the communication gap between a deaf people and normal people.
- Try to create a dataset of Bangla Sign Language with MediaPipe holistic.

1.7 Report Layout

This research paper’s contents are:

I. In the first chapter we discuss motivation, rational study and objectives.

II. In the second chapter we discuss the related work and research summary.

III. Research methodology, data collection and preparation, Research Subject and Instrumentation and discuss the applied model in Chapter 3.

IV. The experimental evaluation and numerical result of the study are discussed in Chapter 4.

V. The fifth chapter offers the summary, conclusion and future work.

CHAPTER 2
BACKGROUND

2.1 Preliminaries/Terminologies

- Collect data: The first step in developing a sign language detection system is to collect data. This can be done through video recordings of people signing or through photographs of people making different signs.
- Pre-process data: The data should then be pre-processed to remove any noise or outliers. This can be done by applying various filters and algorithms to the data.
- Feature extraction: After pre-processing, the data should be analyzed to extract features that can be used as inputs for machine learning algorithms. This can include extracting shapes, edges, and other features from the images.
- Train models: Once the features have been extracted, the data can be used to train machine learning models. This can include using supervised or unsupervised algorithms to recognize the different signs.
- Evaluate models: The models should then be evaluated to measure their accuracy and performance. This can be done by testing the models on a test set of data or by running experiments with real users.
- Deploy system: Once the models are trained and evaluated, they can then be deployed in a system to detect sign language. This can include a web or mobile app, or a hardware device such as a camera

2.2 Related Works

M. Hasan et al. [1] proposed a detection and recognition of Bangla Sign Language by hand gesture recognition using HOG (Histogram of Oriented Gradients) for extraction of features from the gesture image and SVM (Support Vector Machine) as classifier. Their testing result on their testing dataset gave an average recognition rate of 86.53%.

M. S. Islam et al. [2] conducted to create a potent model of Recognize Bangla Sign Language Digits using CNN (Convolutional Neural Network). Their model is trained and tested with 1075 images and the training model accuracy rate is 95%.

S. A. Khan et al. [3] aimed to create an efficient sign language translator device using CNN (Convolutional Neural Network) and customized ROI (Region of Interest). By using a custom dataset with 5 sign gestures, they trained the dataset and implemented it in Raspberry Pi for portability.

F. M. J. M. Shamrat et al. [5] proposed a new method in their article for Bengali sign language recognition based on deep convolutional neural networks. Their dataset has 10 indications for ten numbers (০-৯) and every indications has 31 images. Total 310 images. Their overall accuracy after using CNN is 92.88.

S. Rayeed et al. [6] presented a complete dataset for Bangla sign digits from Zero (Shunno in Bangla) to Nine (Noy in Bangla) using MediaPipe using MediaPipe, a cross-platform depth-map estimation framework. They had run MediaPipe on a benchmark American Sign Language dataset. They runned different classifiers in their proposed dataset and god 98.65% accuracy by using Support Vector Machine.

P. Roy et al. [7] made a project about a sign language conversation interpreter system where different methods were used and created Convolutional Neural Network(CNN) model using 100 images from 15 datasets. The proposed model successfully performed with the accuracy of 81%.

S.A. Shurid et al. [8] have proposed a machine-based approach for training and detecting the Bangla Sign Language aiming to create a multi model system for recognizing Bangla signs using the Convolutional Neural Network (CNN). The dataset here was built by using Kinect sensor for accurate depth tracking which provided with the 3D tracking improving detection accuracy by 90% outperforming some pretrained model.

T. Abedin et al. [9] conducted a noble architecture “Concatenated BdSL Network” which consists of CNN based image network and a pose estimation network. They achieved 91.51% of accuracy on their approach.

A. Abraham et al. [10] demonstrates a device that helps to bridge a gap between mute persons and other people forms which make use of an Arduino Uno board, a few flex sensors and an android application to enable effective communication amongst the users. Global System for Mobile (GSM) is used to transfer text messages which are then converted to speech by the application.

2.3 Comparative Analysis and Summary

Sign language detection systems are systems that are used to detect and recognize sign language. These systems are used to facilitate communication between deaf and hearing people, providing a bridge between spoken language and sign language. Sign language detection systems have been developed over the past few decades, and they come in a variety of forms, from computer vision-based systems that use cameras to detect sign language, to AI-based systems that use machine learning algorithms to recognize signs.

The most common type of sign language detection system is the computer vision-based system, which uses cameras to detect and recognize signs. These systems typically use a combination of motion capture technology and facial recognition algorithms to detect and recognize signs. Motion capture technology captures the motion of the user’s hands and body to detect the signs, while facial recognition algorithms are used to identify the user. These systems can be used to translate sign language into text or speech.

AI-based systems use machine learning algorithms to recognize signs. These systems are typically trained on datasets of sign language videos or images, and are able to accurately recognize signs with a high degree of accuracy. However, these systems are more expensive than computer vision-based systems, and require a large dataset of sign language videos or images for training.

2.4 Scope of the Problem

- Lack of training data: Identification of sign language requires a huge quantity of training data that accurately represents the language.
- Variation of hand gestures: Different individuals may use different hand gestures to communicate the same information using sign language.
- Cross-linguistic differences: Sign languages might have different structures and customs than spoken languages, making it challenging to reliably identify signs from one language to another.
- Lack of resources: Sign language identification systems frequently require specialized gear and software that may be difficult to acquire or prohibitively expensive for many users.
- Lack of accuracy: Due to the complexity of the language and the difficulty of recording tiny variations in hand motions, sign language recognition algorithms have difficulties accurately detecting signs.

2.5 Challenges

- Developing a comprehensive dataset of sign language gestures: Creating a comprehensive dataset of sign language gestures is essential for training a sign language recognition system.
- Designing an effective algorithm: An effective algorithm must be able to accurately identify the different signs and their meaning, while also taking into account variations in hand gestures.
- Ensuring the accuracy of the system: The accuracy of the system must be tested and evaluated to ensure that it can accurately identify signs.
- Creating a user-friendly interface: The user interface of the system must be intuitive and easy to use for users who are unfamiliar with sign language.
- Integrating the system with other technologies: The system must be able to integrate with other technologies, such as voice recognition, to provide a more comprehensive experience.

CHAPTER 3
RESEARCH METHODOLOGY

3.1 Introduction

This research methodology focuses on the development of a sign language identification system which will use image processing techniques to recognize signs from a video sequence and classify them into their respective categories. The proposed system will use a combination of Holistic Pipeline and Long Short-term Memory (LSTM) network in order to extract features from the images and recognize the signs. The system will be tested and evaluated on a dataset of sign language images. The results of the experiment will be analyzed to determine the performance of the system. The proposed system has potential applications in the areas of healthcare and education, where it can be used to detect and interpret sign language in real-time.

3.2 Data Collection

The first stage of data collection is to create the necessary folders to store them. We have gathered and constructed our own collection of data. Folders were created for the dataset using: os.path.join (‘folder_name’) method. Then we take a array called ‘action’ of 15 gestures (['আপনার কোন কিছুর প্রয়োজন?', 'আবার দেখা হবে।', 'তুমি কেমন আছ?', 'আপনাকে কিভাবে সাহায্য করতে পারি?', 'আমি ভালো আছি', 'আপনাকে আমার ভালো লেগেছে।', 'আজকের দিনটা খুব সুন্দর।', 'আপনি কি কাজ করেন?', 'তোমার নাম কি?', 'আপনি কোথায় থাকেন?', 'আপনি দেখতে অনেক সুন্দর।', 'এখন কয়টা বাজে?', 'শুভ জন্মদিন।', 'আমি দুঃখিত।', 'ধন্যবাদ।']) that will be utilized to train our recognition model. For every gesture, we have collected 30 videos. We had built a separate folder for each video in order to store the video's frames. Every video folder for each gesture folder began with index 0 and ended with index 29. After creating the gesture folders, the following step is to gather data in folders. A for loop is then executed to build 30 video files for each move or gestures.

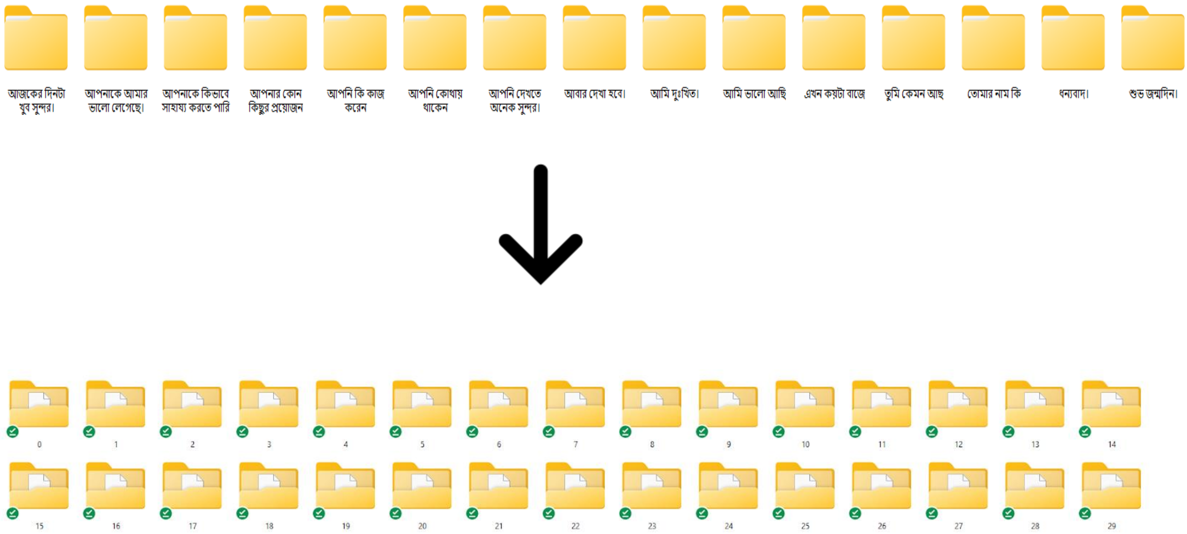

Figure 3.1: Folder of signs and subfolders for each sign

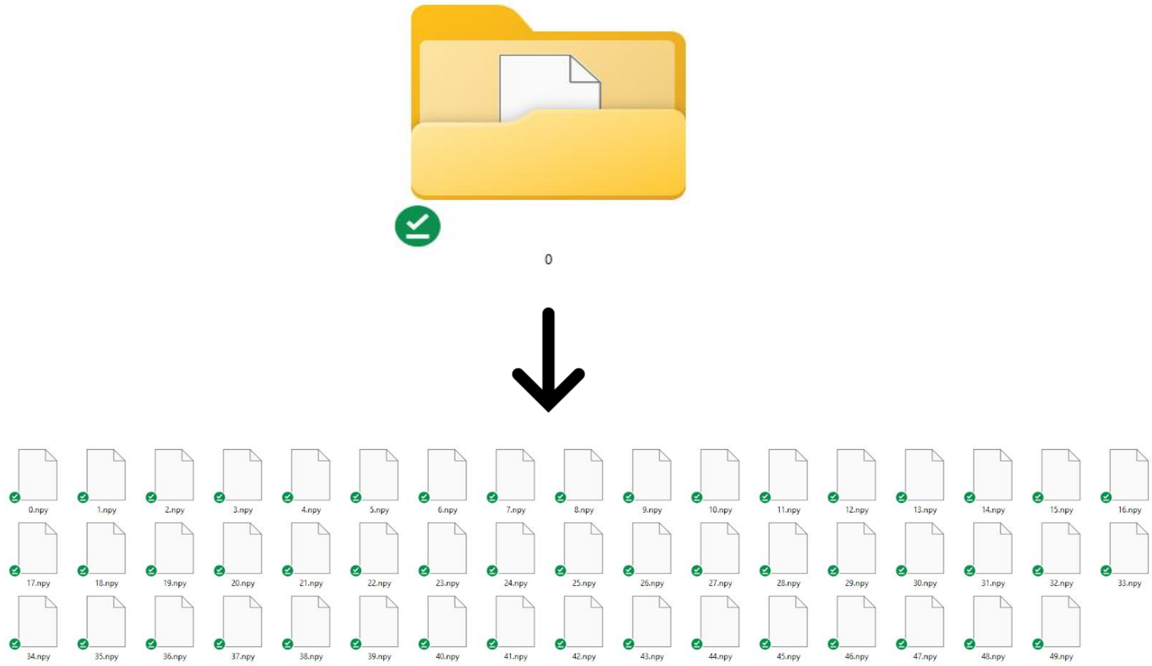

Figure 3.2: Collection of 50 frames inside each subfolder

On the next stage, we gathered the video data for each gesture and collected the array of key-points for each frame using the key-points extraction method. In the end, the NumPy array of collected key points is recorded for each frame within each video folder. However, the array really extends to 49.npy because we are collecting 50 NumPy arrays for every video. We captured 50 frames for each video and 30 videos for every sign sentence in this manner. In total, for 15 sentences, 22500 pieces of data were gathered.

3.3 Data preparation

In the data collection process, we collected data for each gesture. To prepare the data for train and test, we will label all ta gestures with a single index where the first gesture will be labeled with 0, the second one will be 1 and so on. Then the key points retrieved from the data are formatted using data preprocessing. The data is organized so that all arrays of key points of each motion are stored as one NumPy array (X), which is then mapped to another NumPy array of labels (Y). Then, we use the categorical function to transform Y into a binary class matrix. After preprocessing is complete, the train test split function is used to divide the data into training and testing sets. We used 80% of data for training and the rest of 20% was used for testing data.

3.4 Statistical Analysis

Analyzing the reliability of a sign language recognition system is possible through the statistical analysis of a dataset in sign language detection. By analyzing the data, we can learn how well the system recognizes different types of sign language, how quickly it can pick them up, and how many different kinds of sign language it can pick up. The information can be examined using several statistical techniques, including t-tests, chi-square tests, and linear regression. By comparing system accuracy to ground truth data, these techniques can be utilized to draw conclusions about the system's performance. Furthermore, the study can be used to examine how variables like sign language, user load, and processing speed affect the precision of the system.

3.4.1 Flow Model

First, we collect data of 15 sentences where for each sentence data taken 30 times and each time after pre-processing key points of 50 different frames have been collected. Then, the model is trained and with that model, we examine if our model can correctly predict exact gestures for specific sign language or not.

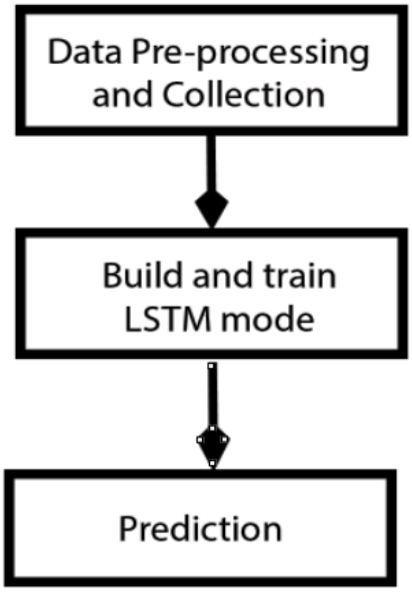

Figure 3.3: Flow Chart Model to Predict sign language

3.4.2 Training Model

In this study, we identified sign language by considering the recurrent processing using Long Short-Term Memory (LSTM) neural network.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 50, 64)            82688

 lstm_1 (LSTM)               (None, 50, 128)           98816

 lstm_2 (LSTM)               (None, 64)                49408

 dense (Dense)               (None, 64)                4160

 dense_1 (Dense)             (None, 32)                2080

 dense_2 (Dense)             (None, 15)                495

=================================================================
Total params: 237,647
Trainable params: 237,647
Non-trainable params: 0
_________________________________________________________________
```

Figure 3.4: Model Summary

3.5 Implementation Procedure:

The strategy that has been recommended takes into account the challenges that were met by past models and makes an effort to alleviate such challenges. We constructed a system with no tradeoffs between performance and efficiency. The researchers were hampered greatly by the segmentation of the hands in the background and the non-uniform backdrop. Using Google's Mediapipe solution and the open CV library, we built a dataset to ensure accurate landmark identification from the user's hands and body, and we continued to apply these methods in the real-time detection phase of the gesture recognition system to ensure accuracy. The suggested system is made up of main components seen in Fig. 3.5. First, in the dataset module, landmarks are extracted and the dataset creation process is executed. Next, in the preprocessing module, the data is processed and sent into the LSTM module, where the training for detecting gestures takes place. After tweaking the model's hyperparameters, the real-time feed is generated by a camera and processed by the models, with the findings shown as text next to the relevant gesture.

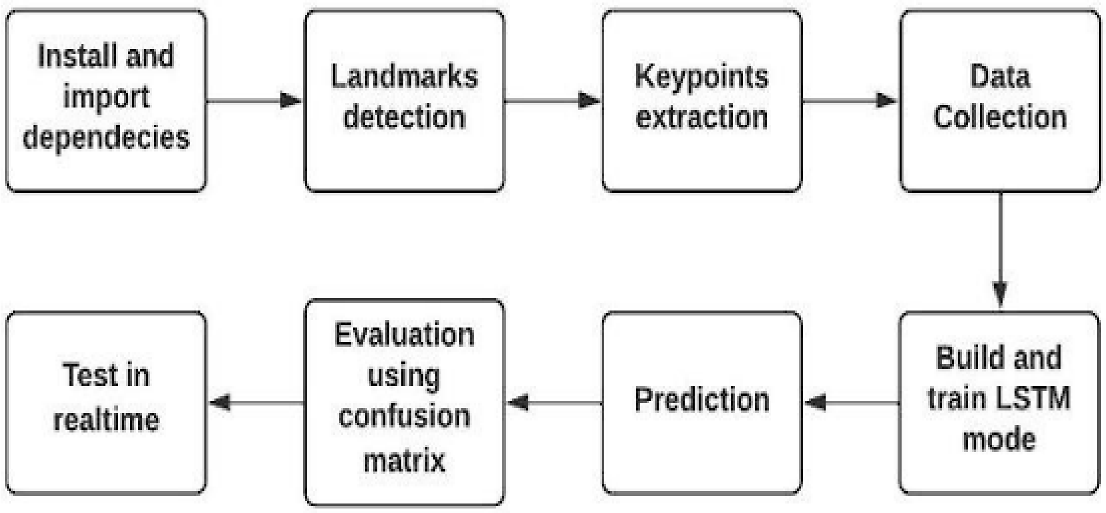

Figure 3.5: Proposed Method to Predict Depression

3.5.1 Landmarks Detection

MediaPipe Landmarks Detection is an open-source library which enables developers to quickly and easily create real-time facial, hand and pose landmark detection pipelines. It uses a combination of machine learning and computer vision techniques to accurately detect hand, face and pose in images, videos, and real-time feeds from cameras.

3.5.2 Keypoint Extraction

Keypoints extraction in sign language is the process of identifying and extracting key points from videos of sign language. Keypoints are specific points in a sign that are considered important for recognizing and interpreting the meaning of the sign. They provide a way of representing the overall form of a sign and are used to compare different signs. Examples of keypoints include the starting and ending points of a sign, the position of the hands and arms, the location of the head and shoulders, and the direction of the fingers.

In order to extract keypoints from sign language videos, researchers use a combination of computer vision techniques such as facial recognition, hand tracking, and motion capture. First, the video is analyzed to detect the signer's hands and face. Then, the hands and face are tracked to determine the location of each keypoint. Finally, the extracted keypoints are used to create a representation of the sign language.

Here, we used Holistic Pipeline to both detect the landmarks and extract their positions by saving those numeric values as numpy. Keypoint extraction can be used to improve the accuracy of sign language recognition and interpretation. By extracting keypoints, researchers can more accurately compare different signs and better recognize and interpret them.

3.5.3 Train Test Split

In order to train and test the model, we divided the data into a training set comprising 80% of the total and a testing set comprising 20%. Once our model has been trained, we can next evaluate its efficacy by testing it on data it has never seen. Table 3.1 displays the number of data separated for the split.

Table 3.1: DATASET TRAIN-TEST SPLIT

Dataset	Full	Training Split	Testing Split
Dataset 1 (15 sentences)	22,500	18000	4,500

3.5.4 Model Generation

Once the model has been created, this part will be reviewed to ensure its accuracy. Adam Optimizer was employed to provide evidence for this hypothesis in the research. Then we used Adam Optimizer to justify the verification of the model.

3.5.5 Model Testing

After the development of the model, it will be evaluated for precision by being fitted to the test data set in the data-splitting phase. After that is done, the model can make an accurate prediction of sign language.

3.6 Algorithms Details

This study examines the combined use of holistic pipeline and Long Short-Term Memory (LSTM). A holistic pipeline typically consists of several stages of pre-processing, such as tokenization, part-of-speech tagging, and semantic tagging. This pipeline provides a rich set of features that can be used by LSTM networks to better understand the context of a given text. By utilizing a holistic pipeline, LSTM networks can better identify and classify words, as well as recognize patterns in the text.

3.6.1 Holistic Pipeline

Holistic pipeline can be utilized in a variety of ways to detect motion. It can identify regular motions, such as the movement of a person, by recognizing patterns in a scene. A holistic pipeline can also detect large and complicated motions, such as a person walking across a room or a vehicle driving down a street. Even external forces, such as wind or earthquakes, can be detected by the integrated pipeline. This contributes to the creation of a comprehensive picture of the environment and facilitates a greater comprehension of the dynamics that occur in a given area.

Mediapipe as a whole provides a vast array of options, and its pipeline is comprised of three components: face, pose, hand, and face. The comprehensive model extracts a total of 543 distinct landmarks (468 faces, 33 poses, and 21 hands). Only 75 unique landmarks (33-pose, 21-left hand, and 21-right hand) are extracted here utilizing the Mediapipe holistic model supplied in the Mediapipe Python module. We developed a function (MP detect) that requires two parameters. Picture: The image on which detection is performed. Model: The mediapipe detection model (MP holistic; holistic model). The function changes the picture from BGR to RGB format before passing it to model.process(). The result is then saved. The function then transforms the picture back to BGR format and returns both the image and the result that was stored. We construct another function (draw_polished_landmarks) that accepts as arguments the output provided by the preceding procedures. Using this function, we illustrate the real-time hand detection by drawing the landmarks. After real-time hand detection has been successfully accomplished, the X, Y, and Z coordinates of the key points are extracted from the detection results (the saved result supplied by the mediapipe detection function discussed above). To extract the coordinates, we build another function (extract keypoints) that accepts the detection results as a parameter. This method returns the concatenated array of all arrays holding the coordinates of the holistic model's key points. This function is invoked during the collection of data. While we utilized the holistic model and retrieved critical points for hand and position, we only displayed hand landmarks (see Figure 2) and pose landmarks (see figure 1) on the screen during the identification phase to prevent clusters of landmarks.

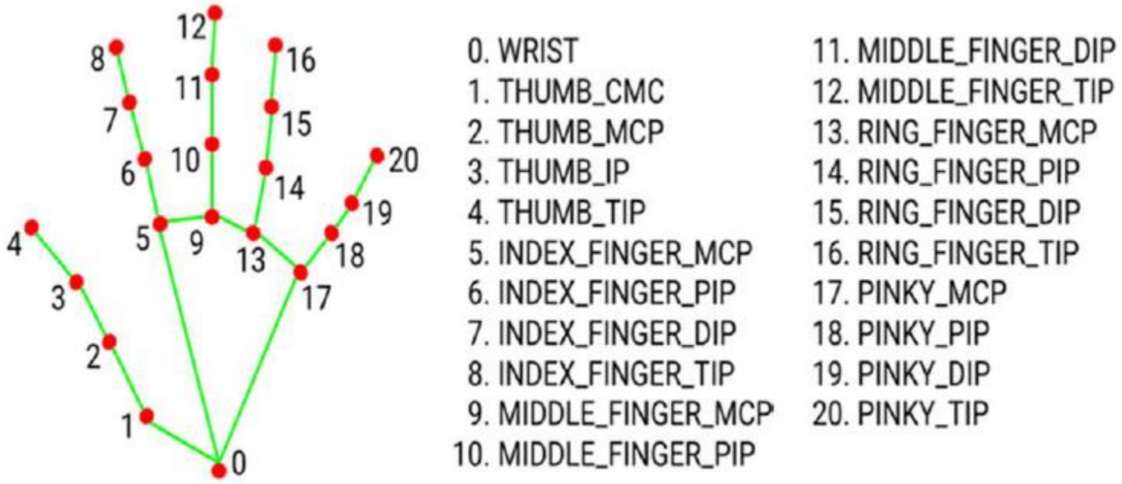

Figure 3.6: The hand_landmarks model of Mediapipe

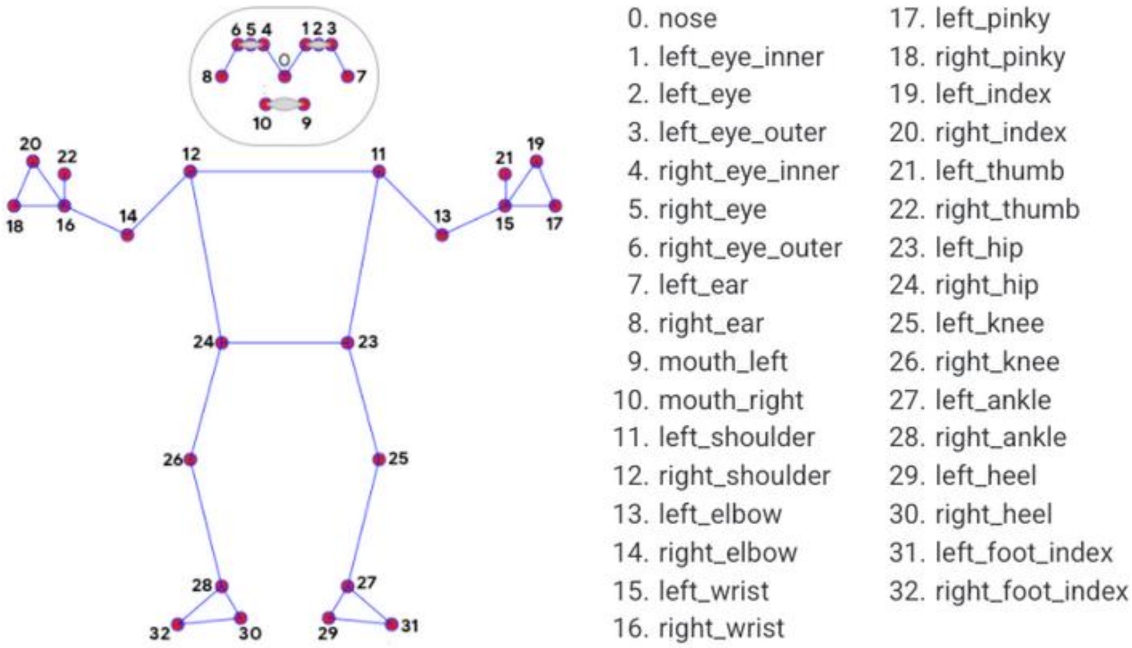

Figure 3.7: The pose_landmarks model of Mediapipe

3.6.2 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) that is capable of learning long-term dependencies. It was introduced by Hochreiter & Schmidhuber in 1997. Unlike traditional RNNs, LSTMs have memory cells that can remember information for long periods of time. This enables them to learn and make predictions based on longer sequences of data.

The core idea behind LSTMs is the concept of an "internal state" which is maintained within the memory cells. This internal state is updated each time a new input is received. The memory cells also have three "gates" that control how information flows through the network. The input gate controls what information enters the memory cell, the forget gate determines what information is removed, and the output gate decides what information is output.

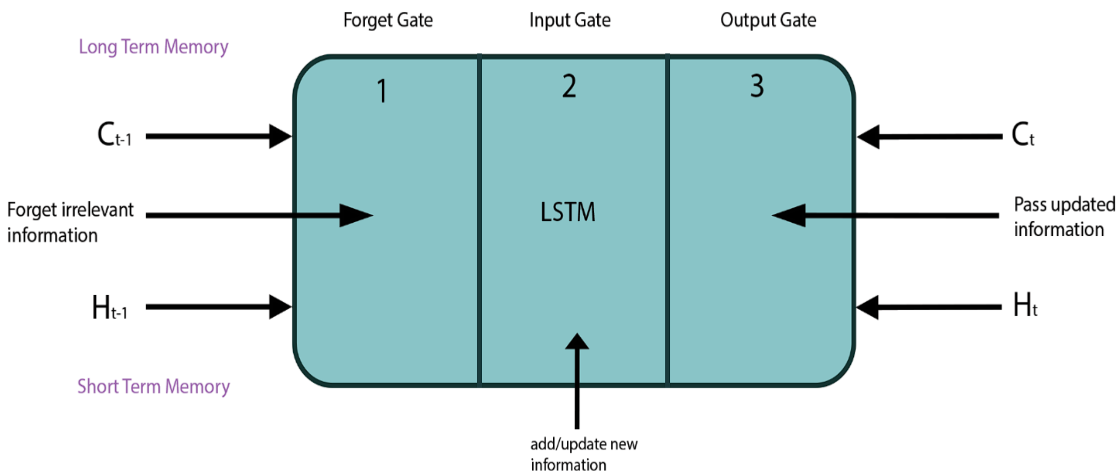

Figure 3.8: The repeating module in an LSTM containing four interacting layers

When an input is received, the input gate determines whether or not the information is relevant and should be stored in the memory cell. If it is deemed relevant, the forget gate then decides which stored information should be removed. Finally, the output gate decides which pieces of information should be passed on to the next layer.

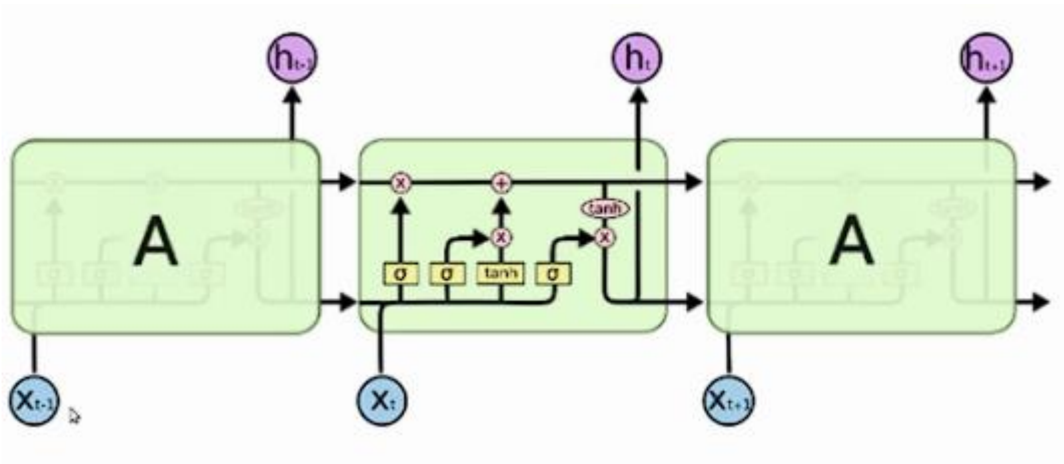

Figure 3.9: The repeating module in an LSTM

By controlling the flow of information through the network, LSTMs are able to learn long-term dependencies. This makes them well-suited for tasks such as language translation and image recognition. In addition, they are more resilient to vanishing gradients than traditional RNNs, allowing them to learn more complex patterns.

3.6.3 Libraries of Python

Table 3.2: LIBRARIES OF PYTHON

Keras	Model	Sequential
	Layer	LSTM
		Dense
		Softmax
	Preprocessing	Recurrent Processing
	Optimizers	Adam

3.6.4 Python Libraries Discussion

Keras: Keras is a powerful and user-friendly open-source deep learning library. It is written in Python and is designed to enable fast experimentation with deep neural networks. With an intuitive API, it is easy to build models with Keras and customize them as needed. It has a large range of pre-trained models that can be used for a wide variety of tasks, including image classification, natural language processing, and object detection. Keras also supports distributed and parallel training, allowing users to scale their models without having to rewrite code. By abstracting away the low-level details of the underlying hardware, Keras allows developers to focus on developing the most effective deep learning models. With its easy-to-use API, it is quickly becoming one of the most popular frameworks for deep learning.

Model: Only Sequential Model is the method used here to create Keras models here.

- **Sequential Model:** Sequential models are a type of deep learning architecture used to solve complex tasks such as image classification, natural language processing, and time series forecasting. This type of model is composed of layers, each with a specific purpose. The layers are "stacked" on top of each other, with each layer taking the output of the previous layer as input. The final layer is the output layer, and this layer produces the desired output. Each layer in the model can contain different types of neurons or nodes, and each node can contain different types of activation functions. These different layers and neurons are used to extract features from the input data and build a model that can make predictions. advantages of this model are that it is relatively simple and efficient to train, and can produce good accuracy when properly configured. It can also be used to model complex non-linear relationships. Disadvantages include more difficulty in interpreting the model and potential overfitting.

Layers: In Keras, a layer is a class that accepts a set of input tensors and returns a set of output tensors, which can be used to construct a neural network architecture. Layers can perform a variety of computations, such as convolutions, pooling, activation functions, and linear transformations. Layers can also be used to add regularization, such as dropout and batch normalization, to the model.

- **LSTMs Layer:** Long Short Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) layer that has been developed to learn long-term dependencies in a time series. It is a type of memory unit which is capable of storing information over long periods of time, and is able to remember information even after many time steps. The LSTM layer has a number of advantages over traditional RNNs, as it is better able to capture long-term dependencies in a time series. This makes it particularly useful for tasks such as speech recognition and natural language processing, where understanding and remembering the context of words over long periods of time is important. The LSTM layer is also able to learn from its own previous mistakes, and so can be used to improve the accuracy of predictions over time.

- **Dense Layer:** A dense layer in recurrent processing refers to a layer of neurons in a recurrent neural network that is fully connected to the previous layer. This means that each neuron in the dense layer is connected to every neuron in the previous layer, allowing for more efficient information processing. This type of layer is typically used for classification tasks, as it allows for more accurate predictions when dealing with complex data. The dense layer can also be used for feature extraction and dimensionality reduction, as each neuron in the layer has access to all of the information from the previous layer. By using this type of layer, a recurrent neural network can more accurately represent complex data.
- **Softmax:** Softmax is a popular activation function used in recurrent processing, especially when Long Short-Term Memory (LSTM) networks are used. Softmax is a non-linear activation function that takes a vector of real numbers as an input and outputs a vector of real numbers, typically in the range [0, 1], such that each output number is the probability associated with the corresponding input vector element.

  Softmax is used in recurrent processing to convert the output of an LSTM network into a probability distribution. This is important for tasks such as language modeling, where the output of the network is a probability distribution over the possible words in the language. Softmax is also used in other tasks, such as classification, where it is used to convert the output of the network into a probability distribution over the possible classes. Softmax is also used to generate the output of an LSTM network when it is used in reinforcement learning tasks. In this case, the output of the network is used as the action probability distribution, and the output of the Softmax activation is used to determine the action that is taken.

3.7 Implementation Requirements

To perform the complete job, we need a high-end PC with high GPU, processor and RAM. As we mount our task on Google Colab, it will provide some extra RAM and Space. So, an average PC can perform our task with the help of Google Colab. All required tools are given below:

Hardware & Accessories

- Intel core i3 8 gen or higher
- 8 GB (+4GB by Colab) RAM or Higher
- 512 GB HDD
- High-Speed Internet connection
- Camera/Webcam

Software, Language & Tools

- Windows 11
- Python 3.9
- Google Colab
- Browser (Edge/Chrome)

CHAPTER 4
EXPERIMENTAL RESULTS & DISCUSSION

4.1 Experimental Result & Analysis

After making the deep learning model, the expectation was met by obtaining a classification accuracy that is satisfactory compared to the baseline score. As a consequence, the section entitled "Experiment findings" is a scientific one in which any and all possible scores for any and all algorithmic applications and methods may be analyzed.

4.1.1 LSTM Neural Network Model Performance

While fitting the model, we used 200 epochs in our model. Last 50 epochs are given below:

Table 4.1: PERFORMANCE OF THE TRAINING MODEL

Epoch Stage	Training Accuracy	Training Loss
151	0.9778	0.0704
152	0.9861	0.0506
153	0.9861	0.0569
154	0.9750	0.0812
155	0.8889	0.2623
156	0.9444	0.1516
157	0.9667	0.0896
158	0.9806	0.0620
159	0.9806	0.0643
160	0.9889	0.0624
161	0.9722	0.0716

Epoch Stage	Training Accuracy	Training Loss
162	0.9750	0.0804
163	0.9861	0.0484
164	0.9667	0.0728
165	0.9833	0.0491
166	0.9861	0.0346
167	0.9889	0.0272
168	0.9889	0.0303
169	0.9889	0.0270
170	0.9917	0.0314
171	0.9861	0.0299
172	0.9944	0.0278
173	0.9750	0.0741
174	0.9806	0.0718
175	0.9861	0.0497
176	0.9833	0.0402
177	0.9861	0.0331
178	0.9944	0.0201
179	: 0.9944	0.0202
180	0.9972	0.0159
181	0.9944	0.0138
182	0.9972	0.0131
183	0.9972	0.0113

Epoch Stage	Training Accuracy	Training Loss
184	0.9889	0.0273
185	0.9917	0.0166
186	0.9944	0.0134
187	0.9972	0.0120
188	0.9944	0.0155
189	0.9944	0.0152
190	0.9944	0.0196
191	0.9778	0.0861
192	0.9833	0.0454
193	0.9833	0.0369
194	0.9861	0.0291
195	0.9972	0.0173
196	0.9944	0.0130
197	0.9944	0.0127
198	0.9944	0.0119
199	0.9972	0.0098
200	0.9972	0.0101

4.1.2 Training Accuracy and Training Loss

The training accuracy graph shows that our model can correctly identify sign language. After training, the model must be kept intact. The "h5" file format was used to save our model.

Figure 4.1: Training Loss

The model loss is the difference between the predicted values and the actual values of the output. The lower the model loss, the better the model's performance. Here, model loss is decreasing as the method progresses.

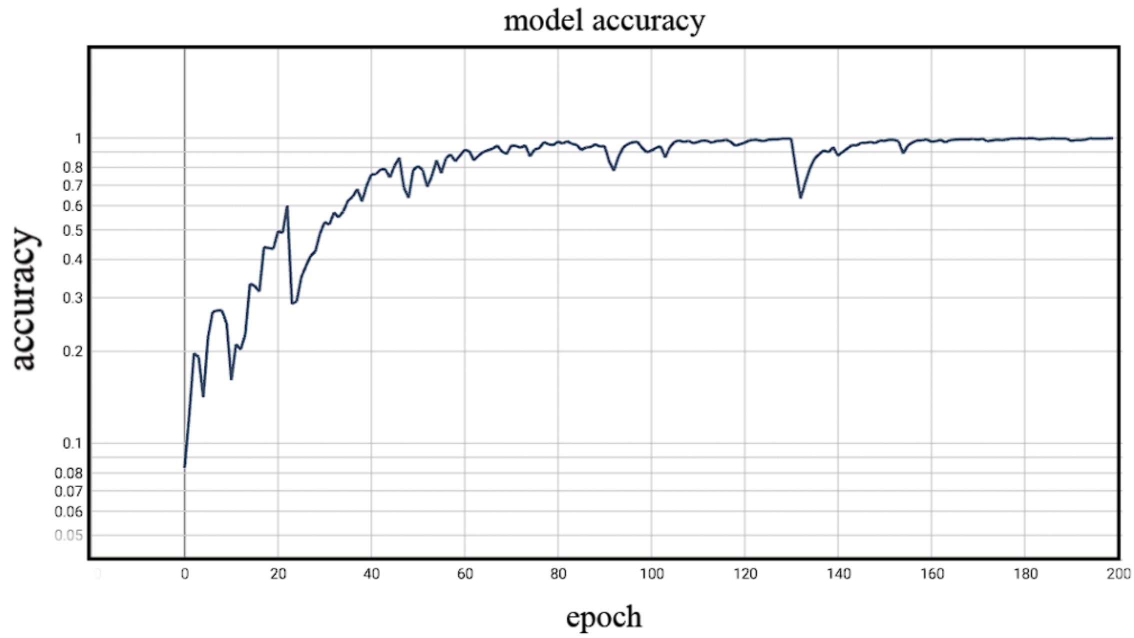

Figure 4.2: Training Accuracy

The figure 4.1 shows that there was some fluctuation in the accuracy graph but generally the accuracy was increasing in each epoch while decreasing the training loss. Finally, the training accuracy stopped at 99.7%.

4.1.3 Predicting Sign Language

We have evaluated the accuracy for both the test and training datasets and it can be seen that our model gives quality results with great performance.

```
In [42]: res = model.predict(X_test)

         3/3 [==============================] - 0s 21ms/step

In [22]: train_loss, train_accuracy = model.evaluate(X_train, y_train)

         12/12 [==============================] - 0s 16ms/step - loss: 0.0789 - categorical_accuracy: 0.9889

In [25]: print("Training loss: ", train_loss)
         print("Training accuracy: ", train_accuracy)

         Training loss:  0.07886236160993576
         Training accuracy:  0.9888888597488403

In [28]: from sklearn.metrics import accuracy_score
         yhat = model.predict(X_test)
         ytrue = np.argmax(y_test, axis=1).tolist()
         yhat = np.argmax(yhat, axis=1).tolist()
         accuracy_score(ytrue, yhat)

         3/3 [==============================] - 0s 16ms/step

Out[28]: 0.9888888888888889
```

Figure 4.3: Accuracy Evaluation

4.1.4 Confusion Matrix

In predictive modeling and machine learning, a confusion matrix is used to evaluate the performance of a classification method. It is a table frequently used to represent the performance of a classification model on a set of test data with known true values. The confusion matrix summarizes the performance of the model. A confusion matrix gives, at its most fundamental level, a tally of the number of accurate and wrong predictions made by a model. It is a graphical depiction of the model's performance, enabling the user to rapidly see where the model is performing or failing. The confusion matrix is also useful for detecting sources of bias or improvement areas for the model.

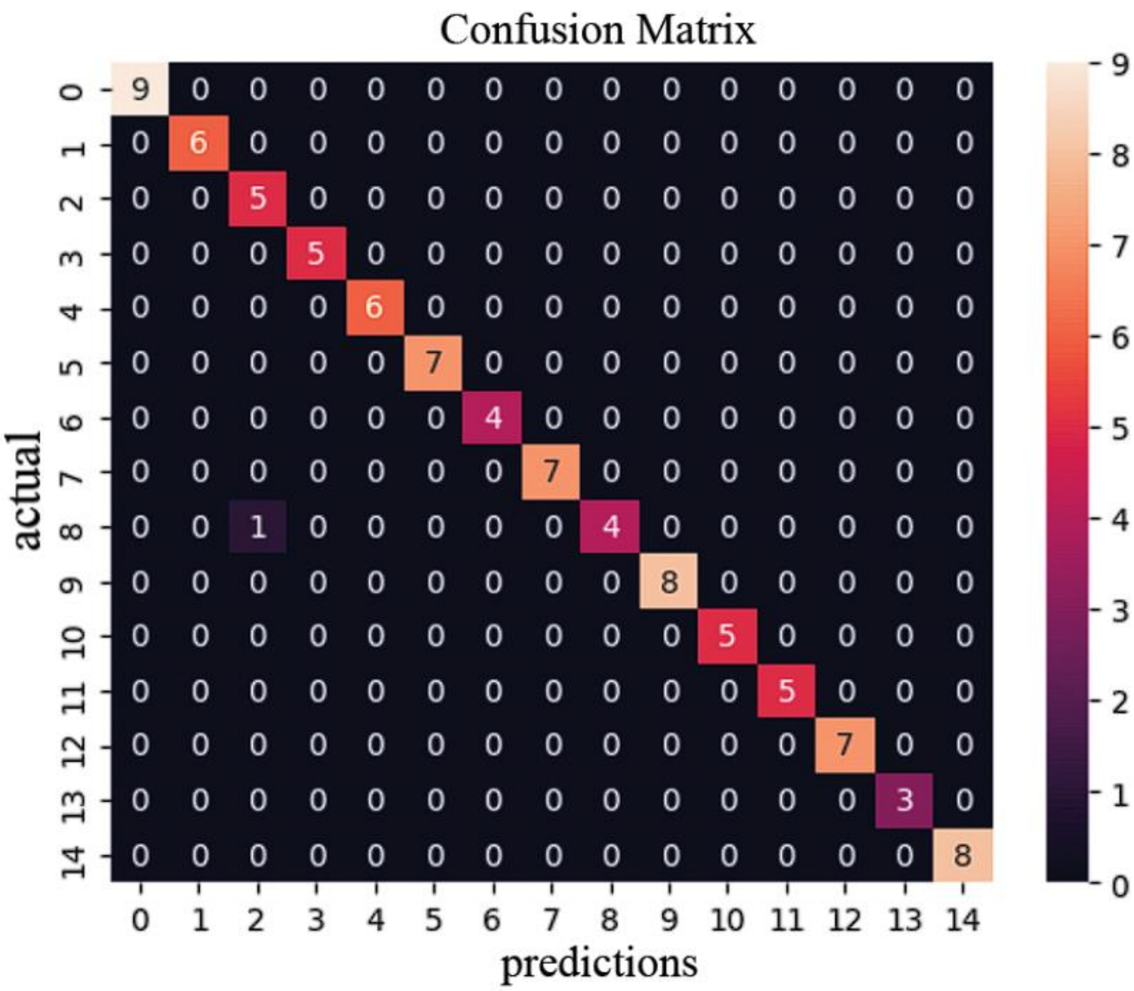

Figure 4.4: Confusion Matrix

Each row in the confusion matrix represents examples belonging to a predicted class, whereas each column represents occurrences belonging to an actual class. The confusion matrix displays the number of accurate and inaccurate guesses made for each class. In addition, it permits the calculation of a range of measures, including accuracy, recall, and F1 score. Consider, for instance, a model that predicts whether or not a patient has cancer. The confusion matrix will provide the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

4.2 Classification Report

Table 4.2 showcases that the model is an excellent way to evaluate the performance of the model. The classification report provides a detailed breakdown of the model's accuracy, precision, recall, and F1 score. The report also provides a visualization of the confusion matrix, allowing you to easily see where the model is succeeding or failing. Overall, the classification report of my model is a great tool for evaluating the model's performance and helps to make informed decisions about how to improve the model.

TABLE 4.2: CLASSIFICATION REPORT FOR MODELS TRAINED WITH CHARACTERS DATASETS

Labels	Precision	Recall	f1-score	Support
0	1.00	1.00	1.00	9
1	1.00	1.00	1.00	6
2	0.83	1.00	0.91	5
3	1.00	1.00	1.00	5
4	1.00	1.00	1.00	6
5	1.00	1.00	1.00	7
6	1.00	1.00	1.00	4
7	1.00	1.00	1.00	7
8	1.00	0.80	0.89	5
9	1.00	1.00	1.00	8
10	1.00	1.00	1.00	5
11	1.00	1.00	1.00	5

12	1.00	1.00	1.00	7
13	1.00	1.00	1.00	3
14	1.00	1.00	1.00	8
accuracy			0.99	90
macro avg	0.99	0.99	0.99	90
weighted avg	0.99	0.99	0.99	90

4.3 Result & Discussion

After application to both training and test data, our proposed model achieved 99.97% and 98.88% accuracy, respectively. Dataset and split-specific accuracy results for our model are displayed in Table 5.1.

TABLE 4.3: MODEL ACCURACY

Training Accuracy	Testing Accuracy
99.97%	98.88%

We determined that the predicted output was accurate after evaluating our model's accuracy with more input. Consequently, we can assert that our model is capable of detecting sign language distinguishing gestures.

CHAPTER 5
IMPACT ON SOCIETY, ENVIRONMENT AND SUSTAINABILITY

5.1 Impact on Society

A Bangla sign language prediction system has the potential to transform how hard of hearing and deaf individuals in Bangladesh communicate with the outside world. Through this technology, deaf and hard-of-hearing individuals can communicate more simply and efficiently with hearing individuals, allowing them to engage more fully in society. Additionally, the adoption of a sign language prediction tool would increase access to resources, opportunities, and education for the deaf and hard of hearing. Additionally, this technology could assist in bridging the communication gap between persons who are deaf or hard of hearing and those who are hearing, thereby reducing misunderstandings and fostering greater comprehension. Ultimately, the usage of a sign language prediction tool in Bangladesh would contribute to the development of a more inclusive and accessible society.

Sign language is a means of communication that is not reliant on spoken language, hence different ways exist in each region. Instead, it is based on gestures, facial expressions, and body language that are culturally specific. These distinctions cannot be ignored by developing a sign language identification system, because each language must be trained for the system to recognize and interpret it effectively. However, a Bangla sign language identification system could be beneficial to the deaf and hearing people of other nations, allowing them to communicate more effectively with persons who are hard of hearing or deaf in Bangladesh.

5.2 Impact on Environment

The Bangla Sign Language Identification System has no direct environmental impact. However, it could reduce the amount of paper necessary for communication between deaf individuals and those who do not understand sign language. It could also increase deaf people's ability to communicate with others, reducing feelings of isolation and enhancing their quality of life.

5.3 Ethical Aspects

Sign language detection system is generally seen as an ethical development as it promotes inclusion and equality for those who are deaf or hard of hearing. It allows them to communicate more easily with those who do not understand sign language and gives them access to more resources. In addition, the system also provides a safer environment for deaf people as they can communicate more easily with those around them. The system could facilitate deaf individuals' access to resources and services, allowing them to be more independent. This could improve the quality of life and social inclusion of individuals with hearing impairments.

The use of the Bangla Sign Language Identification System is seen as a positive ethical development as it promotes inclusion and equality for those who are deaf or hard of hearing. It allows them to communicate more easily with those who do not understand sign language and gives them access to more resources. In addition, the system also provides a safer environment for deaf people as they can communicate more easily with those around them.

5.4 Sustainability Plan

- ❖ Develop a freely available open-source environment for the sign language detection system to increase its usability and promote teamwork.
- ❖ Utilize renewable energy sources to power the system as much as possible.
- ❖ Incorporate artificial intelligence algorithms to improve accuracy and reduce misclassification.
- ❖ Utilize an automated testing platform to validate accuracy and reliability.
- ❖ Design data collection protocols to ensure the system is not collecting sensitive data without user consent.
- ❖ Ensure the system is secure and resilient to attack.
- ❖ Incorporate feedback loops to ensure the system is always learning and improving.
- ❖ Develop a comprehensive user interface to ensure ease of use for all users.
- ❖ Establish a maintenance plan to ensure the system is regularly updated.
- ❖ Regularly audit the system for any potential vulnerabilities or errors.

CHAPTER 6

CONCLUSION AND IMPLICATION FOR FUTURE RESEARCH

6.1 Conclusion

This paper explores the use of Long Short-Term Memory (LSTM) networks for Bangla sign language detection. The results of their experiments showed that the LSTM model achieved satisfactory accuracy. Furthermore, we discussed possible applications of the LSTM model in practical settings, such as real-time sign language detection and translation. The paper concluded that LSTM networks are an effective tool for sign language detection and can be used to develop practical applications.

6.2 Implication for Further Study

In the future, with further development, this system of Bangla sign language identification could become a useful aid for those with hearing impairments, allowing them to more easily access and understand information. This system could also be used to help teach sign language to non-speakers, as well as provide a platform for further research into the field. The system could be used to develop a library of Bangla sign language teaching materials as well as innovative methods for helping people with hearing impairments access and understand information. Furthermore, the system could be used to develop computer-based Bangla sign language translators, allowing for easier communication between speakers of different languages. Finally, the system could be further developed to allow for more sophisticated recognition of signs, making it more accurate and reliable.

REFERENCES

[1] Thasin Abedin, Khondokar SS Prottoy, Ayana Moshruba, and Safayat Bin Hakim. Bangla sign language recognition using concatenated bdsl network. arXiv preprint arXiv:2107.11818, 2021.

[2] Abey Abraham and V Rohini. Real time conversion of sign language to speech and prediction of gestures using artificial neural network. Procedia computer science, 143:587–594, 2018.

[3] A Sachin Agrawal, Agnishrota Chakraborty, and M Rajalakshmi. Real-time hand gesture recognition system using mediapipe and lstm. Journal homepage: www. ijrpr. com ISSN , 2582:7421, 2022.

[4] Md Alam, Mahib Tanvir, Dip Kumar Saha, Sajal K Das, et al. Two di-mensional convolutional neural network approach for real-time bangla signlanguage characters recognition and translation. SN Computer Science, 2(5):1–13, 2021.

[5] Najeefa Nikhat Choudhury, Golam Kayas, et al. Automatic recognition of Bangla sign language. PhD thesis, BRAC University, 2012.

[6] Muttaki Hasan, Tanvir Hossain Sajib, and Mrinmoy Dey. A machine learn-ing based approach for the detection and recognition of bangla sign lan-guage. In 2016 International Conference on Medical Engineering, Health Informatics and Technology (MediTec), pages 1–5. IEEE, 2016.

[7] Md Sanzidul Islam, Sadia Sultana Sharmin Mousumi, Nazmul A Jessan, AKM Shahariar Azad Rabby, and Sayed Akhter Hossain. Ishara-lipi: The first complete multipurposeopen access dataset of isolated characters for bangla sign language. In 2018 International Conference on Bangla Speech and Language Processing (ICBSLP), pages 1–4. IEEE, 2018.

[8] Sanzidul Islam, Sadia Sultana Sharmin Mousumi, AKM Shahariar Azad Rabby, Sayed Akhter Hossain, and Sheikh Abujar. A potent model to recognize bangla sign language digits using convolutional neural network. Procedia computer science, 143:611–618, 2018.

[9] Saleh Ahmad Khan, Amit Debnath Joy, SM Asaduzzaman, and Morsalin Hossain. An efficient sign language translator device using convolutional neural network and customized roi segmentation. In 2019 2nd Interna-tional Conference on Communication Engineering and Technology (IC-CET), pages 152–156. IEEE, 2019.

[10] Liang Liang, Wen Wang, Yunjian Jia, and Shu Fu. A cluster-based energy- efficient resource management scheme for ultra-dense networks. IEEE Ac-cess, 4:6823–6832, 2016.

[11] Prosenjit Roy, SM Miraj Uddin, Md Arifur Rahman, Md Musfiqur Rah-man, Md Shahin Alam, and Md Saidur Rashid Mahin. Bangla sign language conversation interpreter using image processing. In 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), pages 1–5. IEEE, 2019.

[12] FJM Shamrat, Sovon Chakraborty, Md Masum Billah, Moumita Kabir, Nazmus Shakib Shadin, and Silvia Sanjana. Bangla numerical sign language recognition using convolutional neural networks. Indonesian Journal of Electrical Engineering and Computer Science, 23(1):405–413, 2021.

[13] Safayet Anowar Shurid, Khandaker Habibul Amin, Md Shahnawaz Mirba-har, Dolan Karmaker, Mohammad Tanvir Mahtab, Farhan Tanvir Khan, Md Golam Rabiul Alam, and Md Ashraful Alam. Bangla sign language recognition and sentence building using deep learning. In 2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), pages 1–9. IEEE, 2020.

[14] Farhad Yasir, PW Chandana Prasad, Abeer Alsadoon, and Amr El-chouemi. Sift based approach on bangla sign language recognition. In 2015 IEEE 8th international workshop on computational intelligence and applications (IWCIA), pages 35–39. IEEE, 2015.

[15] Rahat Yasir and Riasat Azim Khan. Two-handed hand gesture recognition for bangla sign language using lda and ann. In The 8th international con-ference on software, knowledge, information management and applications (SKIMA 2014), pages 1–5. IEEE, 2014.

Thesis report

ORIGINALITY REPORT

24%	20%	11%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	ijrpr.com Internet Source	4%
2	dspace.daffodilvarsity.edu.bd:8080 Internet Source	3%
3	Submitted to Daffodil International University Student Paper	2%
4	www.researchgate.net Internet Source	2%
5	www.coursehero.com Internet Source	1%
6	www.pyimagesearch.com Internet Source	<1%
7	Submitted to Intercollege Student Paper	<1%
8	ijramr.com Internet Source	<1%
9	www.scien.cx Internet Source	<1%