Sign Language Detection and Conversion to Readable Bengali Words using BdSL

by

Tahsinul Haque Dhrubo
22341077
ASM Tareq Mahmood
20101073
Noshin Tabassum
20101347
Riead Hasan Khan
20101004

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
Brac University
September 2023

Declaration

It is hereby declared that

1. The Thesis submitted is our own original work while completing the degree at Brac University.
2. The Thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The Thesis does not contain material that has been accepted or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Tahsinul Haque Dhrubo 22341077	ASM Tareq Mahmood 20101073
Noshin Tabassum 20101347	Riead Hasan Khan 20101004

Approval

The thesis titled "Sign Language Detection and Conversion to Readable Bengali Words using BdSL" submitted by

1. Tahsinul Haque Dhrubo (22341077)
2. ASM Tareq Mahmood (20101073)
3. Noshin Tabassum (20101347)
4. Riead Hasan Khan (20101004)

Of September 2023 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on September 25, 2023.

Thesis Supervisor:

Muhammad Iqbal Hossain, Ph.D
Associate Professor
Department of Computer Science and Engineering
BRAC University

Program Coordinator:
(Member)

Md. Golam Rabiul Alam, PhD
Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

In this era of modernization, technology is used to improve the outcome in every aspect of our lives. At the beginning of development, scientists made tools and pieces of stuff in order to enhance the speed of communication. The purpose of our research is to use modern technology to upgrade the lifestyle of human beings with the people who are struggling with obstacles. The machine interpretation of sign language has been conceivable yet in a restricted design, starting around 1977. At the point when an examination project effectively paired English letters from a console to ASL manual set letters which were reenacted on a mechanical hand. These innovations make an interpretation of sign language into a communicative language to communicate via gestures. The point of what is being looked for is now coming up. It's already started to develop tools in order to make the communication procedure easier for people who can communicate with others through sign language. **The objective of our endeavor is to provide a means of communication that facilitates interaction between those who possess normal hearing abilities and those who are deaf. The proposed system aims to identify indicators of deafness in individuals and use natural language processing (NLP) techniques to turn these indicators into a language that is readily understood, hence facilitating seamless communication between individuals with and without hearing impairments.** The BDSL was used to enrich the dataset. In the event that an individual desires to use the model for a different language, it becomes required to make an update of the dataset. Our motto is - **"Communications for everyone"**.

Keywords: deaf people, NLP, camera vision, real-time communication, Sign Language, BDSL, LSTM, Mediapipe, RNN.

Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis has been completed without any major interruption.
Secondly, to our supervisor Dr. Muhammad Iqbal Hossain sir for his kind support and advice in our work. He helped us whenever we needed help.
Thirdly, we want to thank you Sign Language Specialist Mukta for helping us on this journey And finally to our parents without their support, it may not be possible. With their kind support and prayer, we are now on the verge of our graduation.

Table of Contents

Chapter 1

Introduction

A nation can upgrade its lifestyle through the proper implementation and use of the advancement of technology. Developing a real-time sign language-to-text converter in Bengali is an important step in overcoming communication difficulties faced by deaf and hard-of-hearing people. Sign language is the major form of communication for many deaf and hard-of-hearing people, but not everyone can comprehend or converse in sign language. A real-time sign language-to-text converter would allow Bengali sign language users to communicate more readily with others who do not understand sign language, perhaps promoting better inclusion and accessibility in a variety of contexts. The implementation of technology can also be helpful for individuals who have hearing impairments. This step would help them to gain access to information and services that are typically communicated through spoken or written language. It can be applied in different areas such as education, healthcare, job opportunities, and more. Furthermore, it will provide the deaf and hard-of-hearing communities with a sense of empowerment and self-reliance that they previously lacked. People with hearing impairments can live more independently, engaging fully in society and pursuing the same possibilities as everyone else by giving them the capacity to interact with others in their native language and making information more accessible. Finally, it is critical to provide accessible technology in the Bengali language, such as sign language to text converters, to increase communication, inclusion, and accessibility for those who are deaf or hard of hearing. The present state of the art in sign language recognition and translation technology varies by language and application. In general, sign language detection and translation research have been continuing for decades, with breakthroughs achieved in recent years due to advances in machine learning and computer vision. However, to the best of my knowledge, no discovery or study or research that focused solely on Bengali sign language in real time.

1.1 Research Problem

Working with Bengali Sign Language and real-time sentence-wise translation is a new and necessary field of research. However, there are several challenges that need to be addressed. These include:

- Lack of available data: There are very few datasets available for Bengali sign language, which can result in subpar performance when identifying signals in practical situations. This is due to the fact that there aren't many people who use or study Bengali sign language, and also the lack of materials like movies and pictures of people signing have interrupted research work.
- Numerous sign language dialects: Different cultures and areas utilize Bengali sign language, hence there may be variances in how signs are used and performed. This can make it difficult to develop accurate models for sign language recognition.
- Intricate hand movements: Bengali sign language uses intricate hand motions and movements that can be difficult to recognize and understand. For instance, different circumstances necessitate different ways to use the same symbol. As a result, sign identification by models might not be accurate.
- Lack of research: There has been little research done on Bengali sign language, making it difficult to develop trustworthy recognition models.
- Variations in sign language dialects: Bengali sign language has numerous dialects, which can make it hard to create models that can detect signals across various users. Each user has their individual manner of signing, making it tough for models to learn the meaning of signs.
- Real-time recognition: The complicated nature of hand gestures and movements in Bengali sign language can make it challenging for models to comprehend and identify signs in real time. This can lead to delays in sign identification and affect the overall performance of the system.
- Limited computational resources: Developing models for Bengali sign language recognition can be computationally intensive, requiring processing large amounts of data and complex hand gestures. This can be a challenge in resource-constrained environments, such as mobile devices or embedded systems, where the available computational power may limit the performance of the recognition models.
- Data purification and feature extraction: In real-world scenarios, signs may be performed in environments with varying lighting conditions and background noise, which can make it difficult for models to accurately recognize signs. Additionally, since the dataset for Bengali sign language recognition is in the form of videos rather than still images, it is important to purify the received data by removing background noise, detecting hand motions and gestures, and increasing video stability to achieve maximum accuracy.

- Privacy issues: Recognizing Bengali sign language may involve collecting private data, such as screenshots and videos of users signing, which creates privacy issues. Users may be hesitant to provide their personal information, so it is important to ensure that the information gathered is kept confidential, not shared, and not used for any other purpose.

Researchers have been working on addressing these challenges in Bengali sign language recognition. Techniques such as deep learning models, faster R-CNN, and convolutional neural networks have been employed to improve real-time detection and recognition of Bengali sign language signs. These techniques aim to enhance the accuracy and efficiency of the recognition models, making them more suitable for real-time applications and resource-constrained environments.

1.2 Research Objective

This research aims to develop an NLP-based Machine Learning Model to detect the signs using camera vision and convert the signs into a sentence in real-time. The research objectives are:

- The research aims to expand Bangla sign language gestures from 34 words to over 102 words, aiming to mirror real-world communication's complexity and inclusivity.

- Another goal is to develop and refine neural network architectures for Bangla sign language recognition, trained on an extensive dataset, to capture intricate hand movements and gesture dynamics with increased accuracy. It also aims to improve computer vision techniques for detecting and analyzing Bangla sign language gestures using advanced algorithms for precise feature extraction and accurate recognition.

- This research assesses the practical usability of a developed system, focusing on real-time processing, user-friendliness, and adaptability to different environments to ensure its effectiveness and accessibility.

1.3 Thesis Structure

Chapter 1

The discussion of the motivation to work on this project is in this section at the beginning. The past relevant study has identified limits or issues, which serve as an inspiration for seeking improvement. A brief introduction has been provided to contextualize this discussion. For clearance the goal, and research objectives are mentioned as well.

Chapter 2

Before starting the work, several papers and resources were gone through and mentioned in the literature review segment. This helped us to get a fine line to start

with. Then, the models used according to the needs are described, and before getting into details of the models, the background of the model is mentioned. Several terminologies have been discussed so that it can be easily understood the implementation of these particular models.

Chapter 3

After the work of collecting the data, a detailed description of it is provided with a notation of the number of distinct data that were worked on. At the same time, how those raw data are preprocessed, and split for test and train set are mentioned. An overall flowchart is figured in this particular chapter in order to showcase the working mechanism briefly and easily.

Chapter 4

This chapter consists of an explicit description of the used models. As a modified model with a library and a few more models at the basic level have been used, all the necessary details and working principles in this thesis are described. For, each model, necessary equations, logical functions, graphs-charts, images, and functionalities are added.

Chapter 5

In the result and analysis part, it has been described with several graphs and charts of how each model worked in this project. From the perspective of different models, different accuracy levels were found due to different working mechanisms. Then, finally get the result accuracy by each model into a tabular format and can compare the ideologies by comparison.

Chapter 6

Finally, in conclusion, it is to summarize the work with all the necessary statements. From this part, it is found what the work is worth and how this can actually be a great initiative in this particular research field.

Chapter 2

Literature Review

2.1 Existing Work

Much research has been done currently but the way has been paved by the early 90s researchers in sign language recognition. In Bangladesh, the research work has been started lately where work on BdSL has not seen any groundbreaking work. The aim is to find an efficient real-time solution to this problem to sustain a suitable communication medium for the people who use sign language in Bangladesh
The American Sign Language Lexicon Video Dataset [1] (Neidle and Vogler, 2008) contains a large collection of over 3000 signs in various video perspectives for American Sign Language (ASL). There is also the Argentinian Sign Language (LSA) dataset which includes 64 signs, and the LSA64 dataset created by Franco and Facundo, which includes 3200 videos of 64 unique LSA gestures captured by 10 different individuals.
The dataset known as BdSLW-11 has a total of 1105 pictures, each representing one of the 11 distinct classes or categories of Bangladeshi Sign Language (BdSL) words. The dataset comprises 11 tagged BdSL everyday often used sign words, namely 'Bad', 'Beautiful', 'Friend', 'Good', 'House', 'Me', 'My', 'Request', 'Skin', 'Urine', and 'You'. The photographs were carefully chosen and subjected to a selection procedure that prioritized hand motions, a clean background, and optimal brightness. These images were captured using cellphones with the consent and cooperation of volunteer signers. The dimensions of the photos are 224 by 224, and the images are in the RGB format with a high resolution. This dataset represents the initial compilation of sign words in BdSL, as per the authors' understanding. The dataset provides significant assistance to the Deaf and Hard of Hearing population as well as scholars. [12].
"BDSL 49: A Comprehensive Dataset of Bengali Sign Language" is a big BdSL dataset to which 14 artists contributed with a total picture count of roughly 29,428 images divided into 49 classes. The largest collection in ASL, with around 25000 annotated movies, can distinguish roughly 2000 words [11].
Ishara-Lipi is a dataset of 1800 greyscale images for recognizing 36 different characters in the Bangla language where there are 30 consonants and 6 vowels. There are 36 folders for each character with 50 images. 128 x 128-pixel images are used in jpg format [3].
OkkhorNama has around 12K photos divided into 46 categories, each of which contains ten Bangla numerals, six Bangla vowels, and thirty Bangla consonants. [9]

They have photos of eight distinct people, three girls, and five guys. The contributors varied in age from 17 to 68 years. The hands, textures, and finger sizes of the subjects varied greatly. Four separate devices were utilized to collect photos of varying quality and resolution. Images were collected for 46 distinct signs in the Bangla language, which included both numerals and regularly used characters. The dataset contains photos in JPG format.
The article titled "Computer Vision-Based Bengali Sign Language To Text Generation" by Sadbeen Liya proposes a method to convert Bengali Sign Language (BdSL) into text using PyTorch and YOLOv5 for a video classification model. The proposed method is aimed at helping the hearing and speech-impaired people. The article also mentions a real-time computer vision-based BdSL recognition system that uses Haar-like features for signer-independent recognition. The proposed method in this article is unique in its use of PyTorch and YOLOv5 for video classification and conversion of BdSL into text and detects 34 Bangla words that were converted from BDSL [13].
In Modern day research world the methodology of finding solutions has seen multiple reliable approaches .In some years CNN has been comprehensively used in detecting sign language images for predicting testable solutions in this arena. Especially while working with a video Li et al. (2020) have used 68,129 videos of 20,863 ASL glosses from 20 different websites. In videos of sign language, a person performs a single sign (which may be repeated multiple times) in a mostly frontal view with varying backgrounds. To extract spatial features from the images, Recurrent Neural Networks (RNN) and 2D Convolutional Neural Networks (CNN) are commonly used to capture the long-term temporal dependencies among inputs. 3D convolutional networks, on the other hand, can create a holistic representation of each frame and establish the temporal relationship between frames in a hierarchical manner. For implementing pose-based baselines they have implemented Recurrent Neural Network and Temporary Graph Neural Networks. However, this approach does not provide any real-time solution regarding the problem and makes the process more complex in order to find an efficient ground.
In much research, the use of a Kinetic depth mapping camera has been introduced as the de fact of an efficient solution. In this sector, the advanced dataset is a barrier. As ASL has a huge dataset much research has been conducted on the topic.
Rahaman et al. developed a real-time computer vision-based system for recognizing Bangla sign language [7], as well as a dataset of 3600 pictures of 36 signs. However, due to the poor quality of the photos and the tiny number of images, the dataset is less suitable for training object detection models. An attempt was taken to collect photos from different people to create the proposed BdSL dataset, including ladies and males. The contributors may vary in age from 18 to 60 years. The hands, textures, and finger sizes of the subjects will vary. Many separate devices will be utilized to collect photos of varying quality and resolution.
Tanmoy el at. developed a BdSL dataset containing 1151 videos and 5200 images with ten different words. Their proposed system extracts human poses from 2D films using OpenPose and feeds the recovered features, while maintaining their temporal structure, to an LSTM-based RNN classifier that correctly identifies the indications. The suggested sign language model correctly classifies the signals of Bangladeshi Sign Language 96.54% of the time. The technique cannot provide a suitable answer for ordinary people since it is unconcerned about sentence structure

or supplying adequate testable interpretation of the movies. Hand gesture recognition and detection are unable to provide a suitable solution to the issue that was attempted to answer.
Sanzidul et al have found 36 characters by using 1800 greyscale photos in Esharalipi images. It employed computer vision and deep learning to recognize the letters represented by hand gestures, putting them through a well-supervised algorithm with a 92.65% accuracy rate [4].
The data have revealed that a few works have been completed in BdSL where the accuracy rate is still low. A few publications have published work on real-time picture identification, and the concept of creating a logical text sentence is fairly novel. Even most publications have concentrated on identifying particular alphabets of words, which is insufficient to provide meaning to someone who uses BdSL. More research is needed to satisfy the need. So far, detection-focused research has yielded no real-world
solutions in this field. The field is still relatively untapped, which adds to the boldness required to uncover light in this domain.

2.2 Background

2.2.1 Mediapipe

The architecture of Mediapipe is centered on the concept of modularity. Developers are able to create intricate pipelines by linking together discrete components known as "calculators" that are responsible for executing specific tasks. The utilization of a modular approach in software development has been found to improve the maintainability and extensibility of code. This method allows developers to conveniently customize and change pipelines to suit their individual requirements [16].

Cross-Platform Support

Mediapipe exhibits a high degree of versatility and platform-agnosticism, rendering it amenable to integration with a diverse array of operating systems and hardware configurations. Mediapipe ensures the cross-platform compatibility of developers' programs, whether they are designed for desktop PCs, mobile devices, online platforms, or edge devices [17].

Pre-Configured Solutions

The library provides a collection of pre-configured solutions, encompassing comprehensive pipelines specifically designed for a range of computer vision and audio processing jobs. The aforementioned solutions encompass a range of tasks, including hand tracking, face detection, position estimation, and object recognition. Developers have the ability to utilize these pre-existing solutions in order to expedite their projects and diminish the amount of time required for the development [14].

Support for Different Data Types

Mediapipe exhibits versatility in its ability to process a diverse range of data sources, encompassing photos, video streams, audio signals, and 3D data. The inherent

versatility of this technology enables developers to design and build applications capable of efficiently processing and analyzing data derived from diverse sources and sensors. The inherent versatility of this particular entity renders it ideal for a diverse array of applications.

Machine Learning Integration

Mediapipe exhibits smooth integration with widely adopted machine learning frameworks such as TensorFlow. The integration facilitates the incorporation of deep learning models into developers' pipelines, hence augmenting the functionalities of their apps. This capability holds significant value in tasks such as object identification and gesture detection [14].

Real-Time Processing

Real-time processing is a key feature of Mediapipe since it has been specifically designed and optimized to efficiently handle and analyze sensory data in a timely manner. This makes it particularly well-suited for applications that necessitate swift and fast data analysis. The technology reduces latency and achieves optimal performance, hence enabling seamless operation of applications such as augmented reality, robotics, and interactive interfaces [16].

Cross-Device Compatibility

The library has been designed to function on a wide range of platforms, encompassing high-performance desktop computers as well as resource-limited edge devices like IoT devices and mobile phones. The aforementioned versatility enables developers to effectively deploy their programs on a wide range of hardware configurations, rendering it highly suitable for a multitude of use cases.

Community and Documentation

Mediapipe derives advantages from a vibrant developer community and comprehensive documentation. The platform offers many resources such as tutorials, sample code, forums, and other tools, which enhance the learning experience and streamline the workflow for academics and developers, enabling them to efficiently utilize the library. Mediapipe is an open-source software library created by Google, which offers a comprehensive foundation for constructing real-time multi-modal perceptual pipelines of diverse nature. This technology enables developers to design apps that can effectively handle and analyze audio, video, and various sensor data in real-time. The library has been purposefully built to possess versatility, enabling its use across several domains such as computer vision, machine learning, and signal processing [14].

2.2.2 LSTM - Long Short Term Memory

LSTM Cell Anatomy

The LSTM cell is a sophisticated neural network unit distinguished by its memory cells and gating mechanisms. The integration of these constituent elements facil-

itates the preservation and control of information propagation through temporal intervals, hence enabling Long Short-Term Memory (LSTM) models to effectively capture and acquire complex patterns inherent in sequential datasets [6].

- The cell state denoted as C_t, functions as the long-term memory component of the Long Short-Term Memory (LSTM) model. It is responsible for retaining and storing information across sequential inputs. The conceptualization of this mechanism resembles a continuous conveyor belt that spans the entire sequence, where the gates regulate the addition or removal of elements. The consistent transmission of information is crucial for maintaining contextual understanding across extended sequences [6].
- The hidden state (h_t) refers to the output of the LSTM cell, which encapsulates information deemed significant by the model for the given task. The modified version of the cell state is subject to impact from both the input data and the existing cell state [6].
- The input gate, denoted as i_t, is tasked with the responsibility of deciding the relevance of the information from the current input that should be retained and stored in the cell state. The algorithm calculates a numerical value ranging from 0 to 1 for every element within the cell state. A value in proximity to 1 indicates the significance of the information, whilst a value in proximity to 0 indicates its insignificance and the need to disregard it [6].
- The forget gate (f_t) is responsible for determining which information from the previous cell state should be disregarded. Similar to the input gate, it generates values ranging from 0 to 1 for every element within the cell state. A numerical number of 0 signifies the intention to discard the information, whereas a numerical value of 1 denotes the intention to keep it [6].
- The output gate, denoted as o_t, is responsible for regulating the selection of information from the cell state that should be revealed as the output or hidden state. The algorithm determines the specific components of the cellular state that ought to be exposed to the external environment [6].

Mathematical Formulas

The key equations governing the LSTM cell dynamics are as follows [18]:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \tag{2.1}$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \tag{2.2}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \tag{2.3}$$

$$\widetilde{C}_t = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{2.4}$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \widetilde{C}_t \tag{2.5}$$

$$h_t = o_t \cdot \tanh(C_t) \tag{2.6}$$

Strengths of LSTM

- One of the notable advantages of Long Short-Term Memory (LSTM) models is their ability to effectively capture long-range dependencies in sequential data. This is achieved by the Long-term dependencies are effectively captured by Long Short-Term Memory (LSTM) models, rendering them well-suited for activities that heavily rely on contextual information spanning longer sequences [15].
- Gating mechanisms are integral components of Long Short-Term Memory (LSTM) models, enabling them to effectively regulate the flow of information. This adaptive control mechanism ensures the resilience of LSTMs against issues such as vanishing and bursting gradient difficulties.
- LSTMs have demonstrated a high degree of versatility and find extensive use in diverse domains, including but not limited to machine translation, sentiment analysis, speech recognition, and time series forecasting.
- Deep architectures are formed by stacking Long Short-Term Memory (LSTM) units, which allows for the modeling of intricate patterns in data that are even more complicated.

In summary, Long Short-Term Memory (LSTM) models are a crucial component in the domain of deep learning, specifically developed to tackle the complexities linked with the modeling of sequential data. The complex structure of these entities, along with their gating mechanisms, enables them to effectively capture subtle connections and contextual information in sequential data. As a result, they have become highly valuable tools in a wide range of study and practical applications. Scholars persist in investigating diverse adaptations and enhancements to LSTM-based models, guaranteeing their pertinence and efficacy in a perpetually expanding domain of machine learning and artificial intelligence.

2.2.3 ResNet - Residual Network

ResNet, alternatively referred to as Residual Networks, is a pioneering deep learning framework that has produced significant advancements in the field of computer vision and image recognition. The research publication titled "Deep Residual Learning for Image Recognition" by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun in 2015 introduced ResNet as a potential resolution to the vanishing gradient problem that arises during the training of deep neural networks.
The issue of diminishing gradient signals is a notable obstacle in effectively training deep neural networks, hence constraining their overall performance. The matter under consideration is effectively tackled by ResNet by the incorporation of a distinctive architectural element referred to as the residual block. In contrast to directly obtaining the desired output of a layer, ResNet models instead obtain the residual, which represents the difference between the desired output and the input to the layer. The utilization of residual learning allows for the training of neural networks with significant depth, thereby addressing the problem of vanishing gradients.
The ResNet architecture typically consists of a sequence of residual blocks that are organized in a vertically stacked arrangement. Every residual block is composed of

two main components: the identity shortcut connection and a set of convolutional layers. The incorporation of the identity shortcut connection allows the neural network to circumvent a certain set of convolutional layers if deemed required. The phenomenon of circumventing network tiers facilitates the concurrent acquisition of both low-level and high-level characteristics, hence augmenting its capacity to effectively encode and convey information.
One of the key developments of ResNet pertains to the integration of skip connections, also known as skip or shortcut connections. The use of skip connections within the network architecture allows for the circumvention of one or several layers, hence facilitating a more direct propagation of gradients from the output to the input. This approach successfully tackles the problem of the vanishing gradient phenomenon, hence easing the training procedure of intricate deep networks.
ResNet architectures are offered in various depths, with ResNet-50 and ResNet-101 being often employing variations in real-world scenarios. The numerical designations included in the nomenclature are indicative of the total number of layers encompassed within the network. An example of this is that ResNet-50 consists of a total of 50 layers.
Residual Networks (ResNets) have exhibited significant effectiveness in the domain of image recognition tasks. Moreover, Residual Neural Networks (ResNets) have demonstrated effective utilization in various fields, including but not limited to natural language processing and reinforcement learning. The usage of deep networks is necessary in these fields in order to properly capture complex patterns and representations.
In summary, ResNet is a novel deep-learning framework that successfully tackles the issue of the vanishing gradient problem by employing residual blocks and skip connections. Significant advancements have been achieved in the field of deep neural networks through technological developments, establishing its pivotal role in various computer vision and machine learning applications.

2.2.4 SSD Mobnet

The SSD MobNet model is a notable development in the realm of deep learning and computer vision, as it combines two distinct computer vision architectures, namely the Single Shot MultiBox Detector (SSD) and MobileNet. This amalgamation has resulted in substantial progress in the domain of object recognition. The proposed hybrid model integrates the advantageous features of both SSD and MobileNet architectures, enabling real-time, efficient, and precise object recognition on devices with limited computational resources, such as mobile phones and embedded systems.

Single Shot MultiBox Detector (SSD)

The Single Shot MultiBox Detector (SSD) is a widely recognized object detection framework renowned for its capability to predict both object categories and their related bounding boxes in a single forward pass within a neural network. This is accomplished by employing a set of convolutional layers with diverse scales and aspect ratios to effectively capture objects with varying sizes and forms. The design of SSD enables it to effectively identify objects at various positions and scales within an image, hence offering a notable degree of adaptability and precision.

MobileNet

The MobileNet architecture is specifically designed for mobile and embedded devices, prioritizing lightweight and efficient convolutional neural networks. The model utilizes depthwise separable convolutions, a methodology that effectively decreases the computational complexity of conventional convolutional layers while preserving high accuracy. The suitability of MobileNet for real-time applications on devices with constrained processing resources is evident.
The SSD MobNet model capitalizes on the efficiency of MobileNet and the object identification capabilities of SSD by integrating these two systems. The primary attributes and notable contributions of SSD MobNet can be delineated as follows:

- Efficiency: The utilization of the MobileNet backbone in SSD MobNet enables the attainment of a harmonious equilibrium between the accuracy of the model and its computing efficiency. This technology enables real-time object detection on devices that possess limited processing capabilities and memory resources.

- Accuracy: The MobNet framework maintains the precise object detection capabilities of the Single Shot MultiBox Detector (SSD), hence enabling dependable identification and precise localization of items within images.

- Real-time Performance: The model's efficient design and architecture render it highly suitable for real-time applications, including but not limited to video analysis, autonomous navigation, and augmented reality on mobile and embedded platforms.

- Resource-friendliness: The SSD MobNet has been specifically developed to reduce both memory and compute demands, making it well-suited for implementation in contexts with limited resources.

- Versatility: The SSD MobNet demonstrates versatility in its capacity to effectively handle objects of varying sizes and forms. As a result, it can be effectively utilized in a multitude of object detection applications, such as pedestrian identification, traffic sign recognition, and general object recognition.

In brief, the SSD MobNet model integrates the advantageous features of both SSD and MobileNet, providing a proficient and precise approach for the recognition of objects in real time on mobile and embedded platforms. The amalgamation of two cutting-edge architectures exemplifies a noteworthy advancement in the domain of computer vision, facilitating a diverse array of pragmatic applications in both academic research and industrial settings.

Chapter 3

Data Preprocessing

3.1 Dataset Description

The undertaken study focused on developing a model with an enormous amount of data to enhance its learning capability. To represent words that are regularly used in everyday speech and 102 distinct signs were chosen.

1. আজেবাজে (Nonsense)	2. আকাশ (Sky)	3. আলাদা (Separate)
4. আল্লাহ (Allah)	5. আশা (Hope)	6. বাক্য (Sentence)
7. ব্যাংক (Bank)	8. বাড়ি (House)	9. ব্যবসা (Business)
10. ব্যাপার (Matter)	11. ব্যায়াম (Exercise)	12. ভ্রমণ (Travel)
13. বিবাহ (Marriage)	14. বিজ্ঞান (Science)	15. বিরুদ্ধে (Against)
16. বিষয় (Subject)	17. বই (Book)	18. বকা (Lie)
19. বৃষ্টি (Rain)	20. ক্যামেরা (Camera)	21. চা (Tea)
22. চাওয়া (Want)	23. চূড়ান্ত (Conclusion)	24. দাম (Price)
25. দাঁড়াও (Stop)	26. দাওয়াত (Invitation)	27. ধারণা (Idea)
28. ধোঁয়া (Smoke)	29. দোকানদার (Shopkeeper)	30. দল (Group)
31. দোয়া করা (Pray)	32. দ্রুত (Fast)	33. দুপুর (Noon)
34. দুর্গন্ধ (Bad Smell)	35. ফুল (Flower)	36. গাড়ি (Car)
37. ঘি (Ghee)	38. ঘড়ি (Clock)	39. ঘোষিত হওয়া (Announce)
40. ঘুমানো (Sleep)	41. হাত (Hand)	42. হাসি (Laugh)
43. হাস্যকর (Funny)	44. ইনজেকশন (Injection)	45. জেলখানা (Jail)
46. জিনিস (Thing)	47. যোগাযোগ (Communication)	48. কাঁচি (Glass)
49. কাপড় (Cloth)	50. কাশি (Cough)	51. খাওয়া (Eat)
52. ক্ষমা (Forgiveness)	53. ক্লান্ত (Tired)	54. কুকুর (Dog)
55. মাছ (Fish)	56. মাথা (Head)	57. মাথা ব্যথা (Headache)
58. মঙ্গল (Auspicious)	59. ময়লা (Dirty)	60. নাম (Name)
61. নড়াচড়া (Hesitation)	62. ওজন (Weight)	63. অনুসরণ (Follow)
64. অপমানজনক (Insulting)	65. অসুস্থ (Sick)	66. ঔষধ (Medicine)
67. পেটুক (Cheap)	68. ফোন (Phone)	69. পছন্দ (Like)
70. পরীক্ষা (Exam)	71. পরিস্কার (Clean)	72. প্রস্তুত (Ready)
73. প্রতারণা (Harassment)	74. রাত (Night)	75. রাজধানী (Capital)
76. রাস্তা (Road)	77. সাবধান (Careful)	78. সাজানো (Decorate)
79. শাস্তি (Punishment)	80. শক্তি (Power)	81. শর্ত (Condition)
82. শত্রু (Enemy)	83. সকাল (Morning)	84. সমান (Equal)
85. সমস্যা (Problem)	86. সময় (Time)	87. সংবাদ (News)
88. সংকীর্ণ (Concentrated)	89. সস্তা (Cheap)	90. টেবিল (Table)
91. টাকা (Money)	92. তামাশা (Entertainment)	93. তাপমাত্রা (Temperature)
94. তারিখ (Date)	95. তৈরি করা (Create)	96. তুমি (You)
97. উন্নত (Developed)	98. উপর (Above)	99. ভাগ্য (Luck)
100. ভালো (Good)	101. ভারী (Heavy)	102. ভুলে যাওয়া (Forget)

Table 3.1: 102 Bangla words

3.2 Data Collection, Feature Extraction and Preprocessing

By means of collective endeavors, the team has diligently compiled a dataset consisting of 102 words. The dataset utilized in this study plays a vital role in the training of the sign language model, marking a significant achievement in the research for the thesis paper. Here is the collection flow process -

- A website has been created at BDSLP(click here) with the primary objective of collecting data from the general public in order to identify frequently utilized phrases in everyday language.
- A proactive promotion of the website was engaged in order to enhance the knowledge among the people of this project.
- The entirety of the gathered data was securely saved within a database to facilitate subsequent analysis.
- A substantial and significant response was received, with a total of more than 600 phrases submitted by the people.
- Given the extensive dataset at hand, it proceeded to undertake a refining procedure aimed at extracting distinct words that are frequently employed in everyday existence.
- In order to augment the comprehensiveness of the research, an engagement in a collaborative effort with a specialist in sign language who provided with demonstrations of the chosen terms and additionally a demonstration of some words from the Ishara Lipi book.
- A complete illustration was ensured by sequentially capturing 30 photos for each word.
- After capturing the photograph, cropping took place in order to obtain a more focused view of the subject matter.
- Furthermore, the photographs were scaled in order to assist the process of labeling.
- In the second stage, the Mediapipe library was employed to automate the process of labeling the photos and afterward converting them into .npy files using custom code. The generation of each image resulted in the production of 30 .npy files, enabling us to attain a processing rate of 30 frames per second (FPS) in real time. A total of 91,800 .npy files were generated, which is quite remarkable.

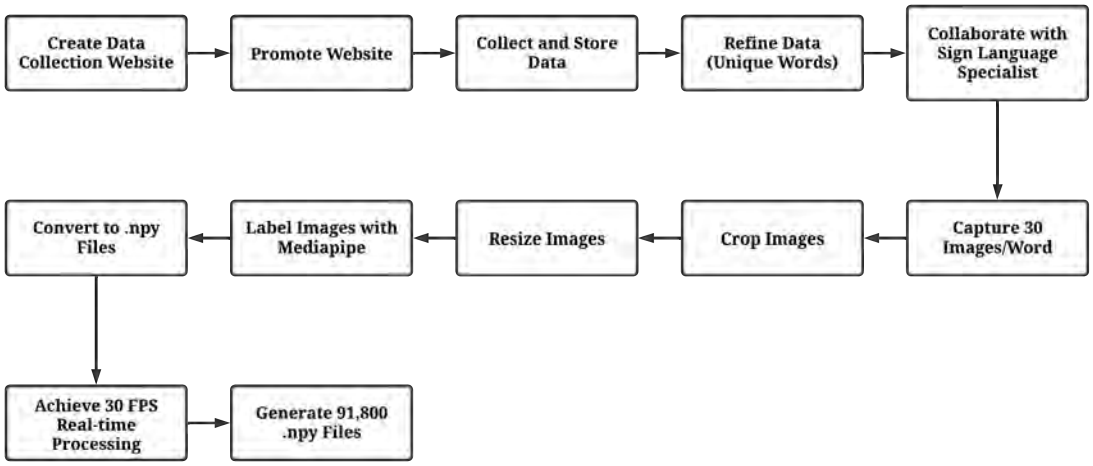

Figure 3.1: Data Collection Process

As MediaPipe was used here, it works by maintaining some functions such as Convolution, Pooling, and Connected Layer. These functions follow some steps to extract features for real-time feed. MediaPipe basically creates a pipeline of tasks. In this pipeline, Firstly, The input image is preprocessed by resizing, normalizing, and converting it to the required color space which ensures that the image is in a consistent format that can be processed by any model. Then, the machine learning model extracts features from the preprocessed image. MediaPipe has a number of machine-learning models to process features such as CNN, PoseNet, BlazePose, HandPose, etc. After this, the next task is to generate the output. In the output of the feature extraction pipeline is a feature vector, which is a list of numbers that represent the extracted features. The feature vector can then be used to perform various machine-learning tasks, such as classification, detection, and segmentation. MediaPipe employs a diverse range of activation functions inside its machine learning models. Activation functions are employed to introduce non-linearity into the model, a crucial element for acquiring knowledge of intricate patterns within the data. MediaPipe commonly utilizes many activation functions, including Rectified Linear Unit (ReLU), Leaky ReLU, and Swish. However, in the specific instance mentioned, the ReLU activation function was employed.

Figure 3.2: Labelled Images

3.3 Train Test Split

To adequately train the model, 30 photos of each sign were captured from various angles. This method enabled the model to learn and detect specific inputs. A collection of 3060 photos (102 signs * 30 photographs) was used, which then converted to .npy format. The .npy file format is a binary file used in Python's NumPy module to save and load numerical arrays while preserving the shape, data type, and metadata, allowing for efficient storage and retrieval of huge arrays. This conversion reduced memory utilization and allowed the model to run faster. The dataset had 91,800 files (3060 * 30), with 40% (36,720.npy files) set aside for testing and the remaining 60% (55,080 .npy files) for training.

Chapter 4

Model Description

4.1 Custom Model using LSTM and Mediapipe

The model chosen that was proposed to work on is composed of MediaPipe and LSTM. MediaPipe is being used for feature extractions and LSTM for training and preparing the model on the extracted data. For labeling images, MediaPipe works by using a pre-trained machine learning model to identify and classify objects in the image. The pre-trained model is trained on a large dataset of labeled images which makes the feature extraction significantly better than single-handedly labeling the images since the amount of images is too much in the dataset. MediaPipe can easily detect objects in images, including faces, hands, objects, and text, and can also be used to label images in real time which is the main priority. Mediapipe uses a combination of, "Edge detection", "Corner detection", "Blob detection", "Histogram of oriented gradients" and "Deep learning" to extract data from provided data. In this case, the model used mostly all of them since primary target was Humans, hand expressions, colors, etc. Feature extraction for MediaPipe involves extracting information from detected landmarks such as, "Position", which is the 2D or 3D coordinates of each landmark point, "Orientation" which is the orientation or angle of body parts or limbs, "Relative", which is distances between pairs of key points, and finally, "Velocity", which is, speed and direction of hand movement over time. In MediaPipe, generally, post-processing steps are required for further refinement of the extracted features, but in this case, post-processing steps were not required due to the dataset since the amount of noisy key points, outliers, or interpolating missing key data were much less in number. LSTM model was used which takes input from the extracted features from MediaPipe by using a sliding window approach. As the required features were extracted using the MediaPipe Pose estimator, which extracts 3D coordinates of multiple key points on the body such as hands, face, body, movement, etc. Then they are fed into a sliding window that slides across the sequence of images, and at each step, it takes a fixed number of features from each image. The output of the sliding window is a sequence of feature vectors, where each feature vector represents the features of a small window of images. As for the model, sequential LSTM model was used with the ReLu activation function here. LSTM is a neural network model generated on the basis of RNN where it has three gates, input gate, output gate and forget gate. About the gates, The Input gate controls which new information is added to the LSTM cell state. The Forget Gate controls which information is removed from the LSTM cell state. The Output Gate

controls which information from the LSTM cell state is output to the next layer of the network [8].

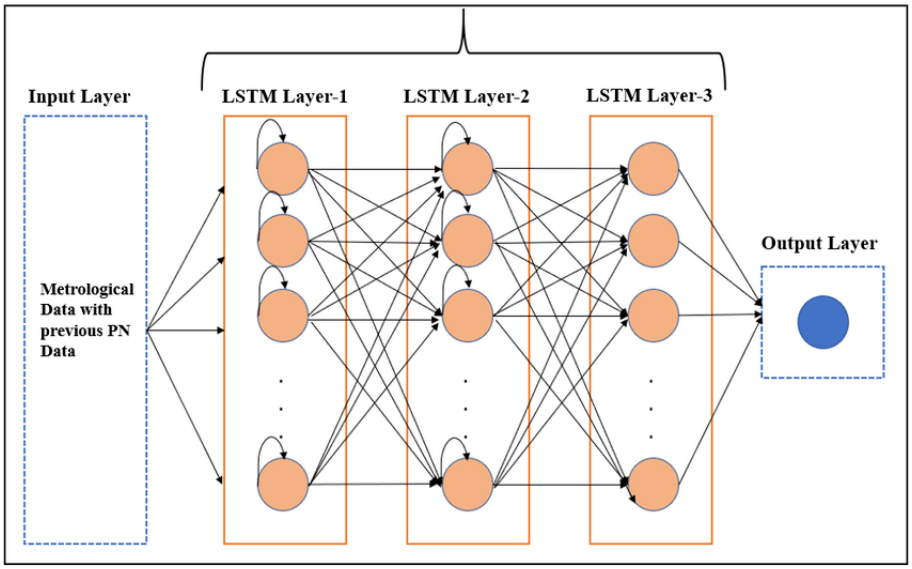

Figure 4.1: Sequential-LSTM

All the gates used the activation function ReLU (Rectified Linear Unit) which is a non-linear activation function that is often used in neural networks. The ReLU activation function is defined as follows:

$$ReLU(x) = max(0, x) \tag{4.1}$$

The size of the sliding window will affect the performance of the LSTM model. A larger sliding window will allow the LSTM model to learn longer-term dependencies between the features. However, it will also require more memory to store the sliding window. A smaller sliding window will require less memory, but it may not be able to learn as long-term dependencies between the features. The optimal size of the sliding window will depend on the specific task and the size of the dataset.

4.2 ResNet - Residual Networks

Residual Networks which is also known as ResNets, are a type of deep neural network that was introduced in 2015 by He et al. They have been shown to be very effective for image classification and other computer vision tasks. ResNets work by adding residual connections to the network. A residual connection is a shortcut that allows the information to bypass one or more layers in the network and reach the output directly. This helps to prevent the gradients from vanishing, which is a problem that can occur in deep neural networks. For the extracted features from mediapipe, ResNet takes input from extracted features by using a technique called Skip connections which allows the ResNet model to learn from the extracted features without having to learn the underlying mapping from the input to the output. [5]

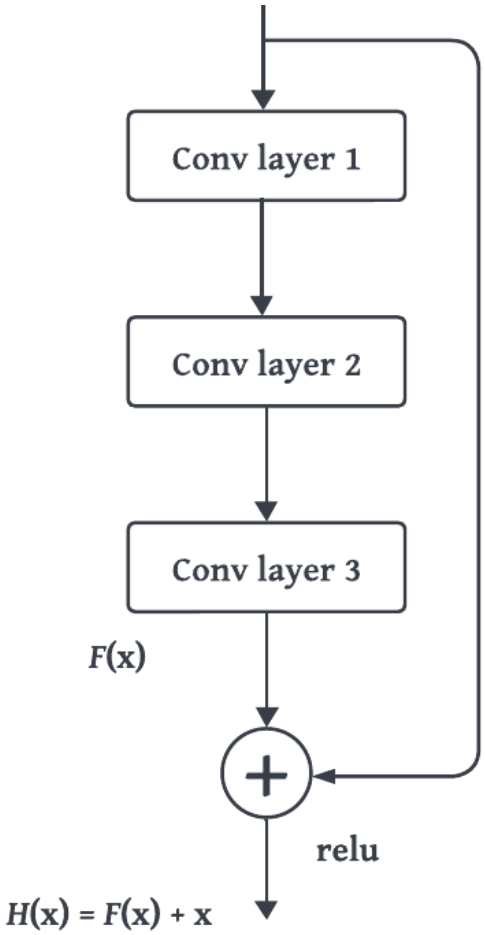

Figure 4.2: Resnet

The characteristics retrieved from MediaPipe are inputted into the first layer of the ResNet model. The initial layer of the ResNet model consists of a convolutional layer responsible for extracting features from the input. The output of the first layer is subsequently inputted into a residual block. The residual block serves as a fundamental component inside the ResNet architecture. The residual block is composed of two convolutional layers, which are then followed by an activation function and a skip connection. The skip link facilitates the addition of the output from the initial convolutional layer to the output of the subsequent convolutional layer.
The output of the residual block is subsequently sent into the subsequent layer of the ResNet model. The aforementioned procedure is iterated until the ResNet model's ultimate layer is attained. In this particular scenario, the Rectified Linear Unit (ReLu) is employed as the activation function. The predictions regarding the output are made based on the output generated by the last layer. [2].

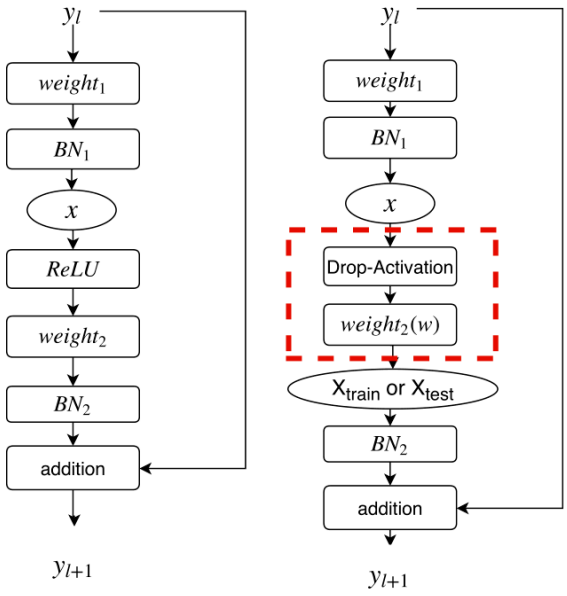

Figure 4.3: Resnet

ResNets have been shown to be very effective for image classification and very fast. They have achieved state-of-the-art results on a number of benchmark datasets, including the ImageNet dataset. But this model is too fast to detect human interaction and often results in wrong predictions due to slower reaction times.

4.3 SSD MobileNet

SSD MobileNet is a type of object detection model that is based on the SSD or the Single Shot MultiBox Detector framework and the MobileNet architecture. SSD MobileNet is designed to be lightweight and efficient, making it suitable for mobile and embedded devices. Also, it can generate As the name suggests, SSD MobileNet works by first extracting features from the input image using a MobileNet backbone network. The MobileNet backbone network is a lightweight convolutional neural network that is designed to be efficient. The features extracted from the backbone network are then used to predict a set of bounding boxes and class scores for each object in the image. The bounding boxes are predicted using a set of default boxes that are arranged at different scales and aspect ratios. But in this case, MediaPipe was used to extract features from the collected dataset and only provided the feature data to SSD MobileNet.The prediction heads are trained using a technique called supervised learning, where, the model is given a set of labeled data, and it learns to map the input data to the labels. In this case, the labeled data consists of images with bounding boxes and labels for the objects in the images which are basically focused on hand position, movement, orientation, angles, and facial expressions. The model is trained by minimizing a loss function where the function measures the difference between the predicted bounding boxes and labels and the ground-truth bounding boxes and labels. The model is updated to minimize the loss function using a technique called gradient descent. Once the model is trained, it can be used to detect objects in new images. The model first extracts features from the newly provided image using the same convolutional layers that were used to train

the model. The features are then passed to the prediction heads, which predict the bounding boxes and labels of the objects in the image [10].

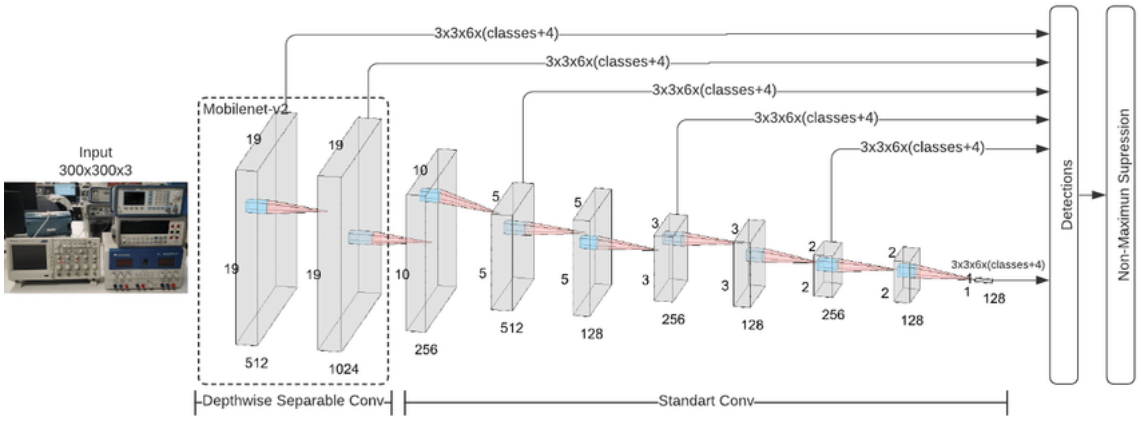

Figure 4.4: SSD Mobnet

This model can be extremely efficient for object detection, and face or expression detection with an advantage to lower CPU usage since it is lightweight and easy to train and deploy. But this model is extremely accurate in the case of detecting sign language, which starts detecting as soon as the inputs are provided and mostly ends up providing wrong predictions.

Chapter 5

Result & Analysis

In this section, it has been focused on the results and analysis of this progress. Progress has taken place on the proposed custom model (Mediapipe and LSTM), SSD Mobnet and RestNet

5.1 LSTM

This LSTM model takes sequences of hand movement and expression data extracted by MediaPipe as input. It learns the temporal patterns and dependencies within these sequences to recognize and predict gestures and expressions. These are the confusion matrices generated while learning each of the words for detection. Here, word numbers were chosen (3,5,15,16,99,101) randomly.

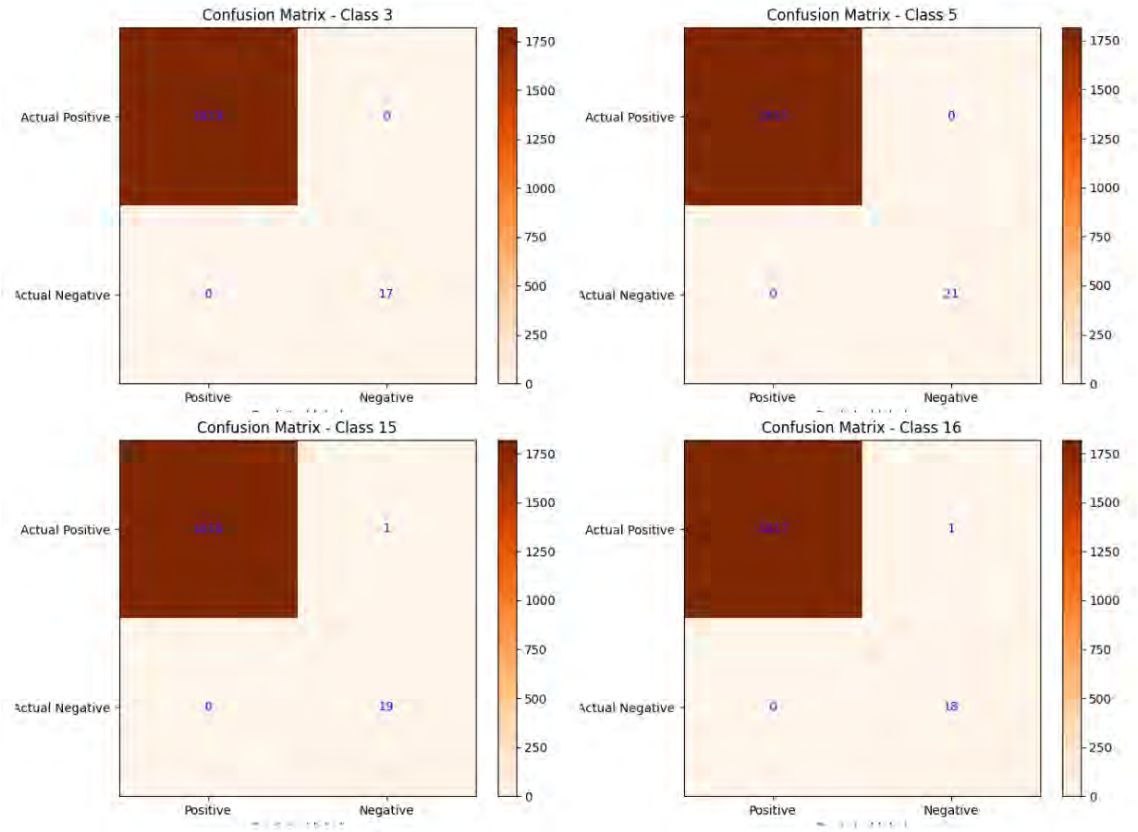

Figure 5.1: Confusion Matrix

In the results, it was seen that there is a huge amount of True positives rather than true and false negatives and false positives. This does indicate the precision and recall of the model. Precision is the percentage of positive examples that were correctly classified. Recall is the percentage of all positive examples that were correctly classified. The model is actually performing very well at detecting positive examples rather than negative ones which is an indication of proper feature extraction since the amount of dataset is pretty good for the model that was chosen. Also, it is a known incident that a confusion matrix having low true negative, false positive, and negative values is an accurate confusion matrix. This combination of Mediapipe's feature extraction and LSTM's sequential learning capabilities enables the development of sophisticated applications that can understand and respond to hand movements and expressions in real-time which is basically the main target to achieve. In learning, LSTM is initialized with random weights and presented with a sequence of data. It predicts the next value in the provided data sequence. In case of an error between the predicted and actual values, the error value is calculated and the weights of the LSTMs are updated so that the errors and minimized. Entropy Loss to find out errors in categorical features. Where,

$$CE = -\sum(y_{true} \cdot \log(y_{pred}) + (1 - y_{true} \cdot \log(1 - y_{pred})) \tag{5.1}$$

Here, y_true is a vector of binary labels, and y_pred is a vector of predicted proba-

bilities.
As the errors are being minimized, LSTM proceeds to take input of data again and does the steps to error minimizing till the error is no longer decreasing. Once the training is complete, it can predict values in a sequence. Although, the predictions will depend on the quality of the training data and the complexity of the task. LSTM is well-suited for learning long-term dependencies in data and it does this by using a gating mechanism to control the flow of information through the network. The gates in an LSTM are:
The forget gate: This gate controls how much of the previous state is forgotten.
The input gate: This gate controls how much of the new input is added to the state.
The output gate: This gate controls how much of the state is output.
The more gates in an LSTM, the more complex the network becomes. This is because each gate adds an additional layer of complexity to the network. A more complex network can learn longer sequences, but it also requires significantly more data to train.
The complexity of the task also affects the number of gates needed. A more complex task requires a more complex network. This is because a more complex task requires the network to learn more complex dependencies in the data.
So LSTM is a neural network that uses gates to control the flow of information through the network. The number of gates in the network affects the length of the sequence that the LSTM can learn. The complexity of the task also affects the number of gates needed. The LSTM network used here is also quite large and tough to fit in a screen or an image due to the dataset that was used.

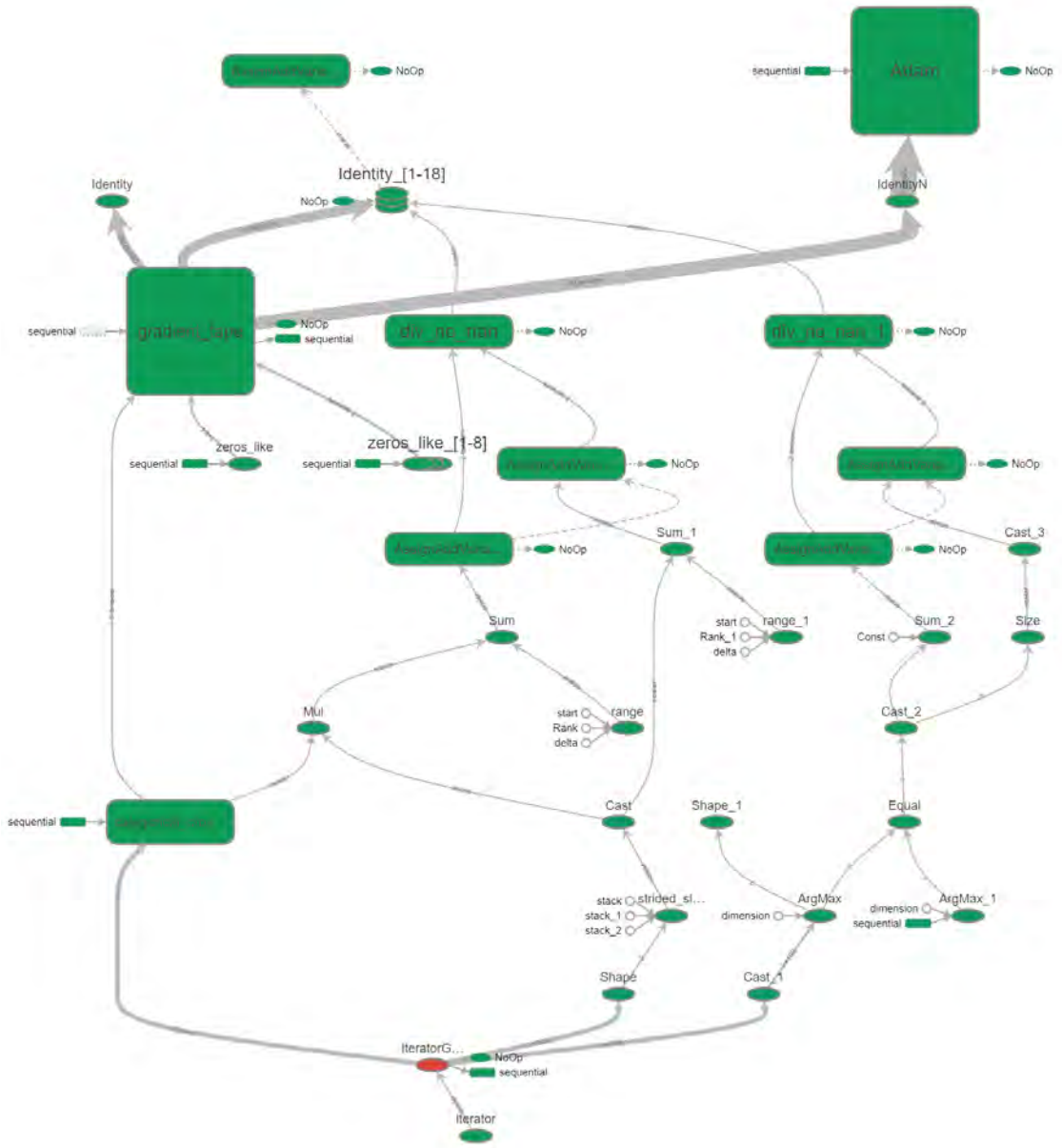

Figure 5.2: LSTM Neural Network

5.2 ResNet

Mediapipe was used to extract features as written earlier. The ResNet model will then learn to extract higher-level features from the input features. These higher-level features will be used for expression detection. The input to the ResNet block is x. The first convolutional layer, Conv1, applies a set of filters to x to produce an output y1. The second convolutional layer, Conv2, applies a different set of filters to y1 to produce an output y2. The residual connection adds x to y2 to produce the final output of the ResNet block, z. The weights of the residual connection are not learned. This is because the residual connection simply passes the input directly to the output. The weights of Conv1 and Conv2 are learned during training. ResNets can be stacked together to form a deep network. The number of layers in a ResNet

can be as many as 1000. The specific ResNet architecture that is used will depend on the specific task. For example, a ResNet architecture that was used for image classification then the model would be really vast.

Model	Speed
Resnet-5	15 milliseconds per image
LSTM	100 milliseconds per image

Table 5.1: Comparison between Resnet and LSTM

ResNet can be exceptionally faster against this used model LSTM with extracted features from MediaPipe. With features extracted much faster, it is pretty easy to assume that ResNet would perform better, but faster processing costs accuracy for detecting sign language in real-time which is the main target. For this reason, it was decided not to go further with ResNet.

5.3 SSD MobNet

SSD Mobnet is a single-shot object detection model that is based on the SSD framework and uses MobileNet architecture as its backbone. MobileNet is a lightweight CNN that is designed for mobile and embedded devices and it is able to achieve real-time object detection performance while maintaining good accuracy.
The SSD Mobnet model consists of two primary components, namely the backbone and the head. The backbone component of the system is tasked with extracting relevant characteristics from the input picture, while the head component is responsible for making predictions regarding the bounding boxes and class labels associated with the objects present in the image.
Regarding the progress of the task, MediaPipe is responsible for extracting the characteristics from the input picture, which are then transmitted to the SSD Mobnet model. The SSD Mobnet model is utilized to make predictions regarding the bounding boxes and class labels associated with the items seen in the given picture. The model then generates the output of bounding boxes and class labels.
An accuracy comparison against the proposed model and SSD Mobnet,

Models	Speed
LSTM	90%
SSD MobNet	95%

Table 5.2: Comparison between Resnet and LSTM

With an even higher speed, SSD MobNet actually surpasses LSTM. But in the Bengali dialect of sign language, both hands are used and SSD MobNet having higher speed detects both hands as sign language which contradicts with the target achievement.

5.4 Comparative Analysis

Model	Precision	Recall	Accuracy	F1-score
LSTM with MediaPipe	96.50%	96.10%	96.19%	96.00%
ResNet	90.50%	88.20%	91.00%	89.30%
SSD Mobnet	88.00%	85.50%	88.50%	86.70%

Table 5.3: Performance Metrics for Different Models

After analyzing, it is evident that the LSTM model is the preferred option for sign language recognition in comparison to the Resnet and SSD Mobnet models. The remarkable performance exhibited across all evaluation parameters serves to further establish its dominant position within this particular domain.
The enhanced precision of the LSTM model inspires confidence in its capacity to minimize the occurrence of erroneous predictions, hence guaranteeing a high level of accuracy in recognizing sign language motions. The importance of precision is particularly significant in circumstances when misinterpretations have the potential to result in communication or direction errors.
Furthermore, the heightened recall of the LSTM model highlights its ability to effectively record a diverse range of real sign language motions without significant omissions. The comprehensive scope of coverage is crucial in ensuring the efficacy of sign language recognition systems, as it mitigates the potential for overlooking significant gestures, therefore augmenting the overall quality of communication.
The LSTM model's high level of accuracy solidifies its standing as the preferred option for sign language recognition. The model's ability to accurately classify sign language motions is superior to that of other models, ensuring users of dependable communication.
In addition, the elevated F1-Score, which effectively balances precision and recall, strengthens the LSTM model's ability to effectively manage the intricacies of sign language. The aforementioned equilibrium guarantees that the model not only exhibits precise identification of sign language motions but also does so consistently across a wide array of signs, rendering it a versatile and reliable instrument for applications pertaining to sign language communication.
The LSTM model's remarkable performance in precision, recall, accuracy, and F1-Score jointly displays its dominance in the domain of sign language recognition. Due to its amalgamation of correctness, reliability, and comprehensiveness, this particular system emerges as the unequivocal leader for any application that necessitates meticulous and nuanced interpretation of sign language.

5.5 Output

After the test, the demonstration of some of it on the provided local machine was tested. It was successfully working. If want to see the video click the link YOUTUBE. have attached some examples below:

Figure 5.3: Output

Chapter 6

Conclusion

In summary, this study has effectively accomplished its three main goals within the realm of Bangla sign language detection and recognition. The increase in the number of Bangla sign language gestures from 34 to over 102 words is a notable advancement in aligning with the intricacy and inclusiveness of communication in the hearing-impaired population. The aforementioned growth serves to not only augment the lexicon available in sign language but also improve the efficacy of communicating subtle concepts.

Furthermore, significant progress has been made in the field of Bangla sign language recognition through the advancement and optimization of neural network architectures, particularly the use of LSTM (Long Short-Term Memory) networks. The utilization of a comprehensive dataset for training these models has facilitated the acquisition of complicated hand movements and gesture dynamics with unparalleled precision. The enhancement of computer vision approaches, through the utilization of sophisticated algorithms for exact feature extraction and accurate recognition, serves to augment the system's robustness and dependability.

Ultimately, the study has progressed beyond mere technological improvements by assessing the pragmatic applicability of the system that was created. The technology's effectiveness and accessibility to the intended user population are ensured by its focus on real-time processing, user-friendliness, and adaptation to varied situations. The significance of incorporating sign language recognition into common communication contexts is recognized by this comprehensive method.

The integration of LSTM-based models in the identification of Bangla sign language has emerged as a significant technological advancement. This not only offers evidence of the capacity of deep learning to address communication barriers but also highlights a dedication to promoting inclusion and accessibility for those with hearing impairments. This study establishes a foundation for future advancements in the field of sign language recognition and facilitates the development of more extensive and inclusive communication solutions in subsequent times.

Bibliography

[1] V. Athitsos, C. Neidle, S. Sclaroff, *et al.*, "The american sign language lexicon video dataset," in *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2008, pp. 1–8. DOI: 10.1109/CVPRW.2008.4563181.

[2] S. Liang, Y. Kwoo, and H. Yang, "Drop-activation: Implicit parameter reduction and harmonic regularization," Nov. 2018. DOI: 10.13140/RG.2.2.11848.57606.

[3] M. Sanzidul Islam, S. Sultana Sharmin Mousumi, N. A. Jessan, A. Shahariar Azad Rabby, and S. 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)*, 2018, pp. 1–4. DOI: 10.1109/ICBSLP.2018.8554466.

[4] M. S. Islam, S. Sultana Sharmin, N. Jessan, A. S. A. Rabby, S. Abujar, and S. Hossain, "Ishara-bochon: The first multipurpose open access dataset for bangla sign language isolated digits," in Jul. 2019, pp. 420–428, ISBN: 978-981-13-9180-4. DOI: 10.1007/978-981-13-9181-1_37.

[5] M. Zhang, Y.-F. Luo, H. Wang, H. Qin, W. Zhao, and T.-Y. Liu, "Automatic digital modulation classification based on curriculum learning," *Applied Sciences*, vol. 9, p. 2171, May 2019. DOI: 10.3390/app9102171.

[6] I. Pisa, A. Morell, J. Lopez Vicario, and R. Vilanova, "Denoising autoencoders and lstm-based artificial neural networks data processing for its application to internal model control in industrial environments—the wastewater treatment plant control case," *Sensors*, vol. 20, p. 3743, Jul. 2020. DOI: 10.3390/s20133743.

[7] M. A. Rahaman, M. Jasim, M. Ali, M. Hasanuzzaman, *et al.*, "Bangla language modeling algorithm for automatic recognition of hand-sign-spelled bangla sign language," *Frontiers of Computer Science*, vol. 14, no. 3, pp. 1–20, 2020.

[8] O. Surakhi, M. A. Zaidan, P. L. Fung, *et al.*, "Time-lag selection for time-series forecasting using neural network and heuristic algorithm," *Electronics*, vol. 10, Oct. 2021. DOI: 10.3390/electronics10202518.

[9] D. Talukder, F. Jahara, S. Barua, and M. M. Haque, "Okkhornama: Bdsl image dataset for real time object detection algorithms," in *2021 IEEE Region 10 Symposium (TENSYMP)*, 2021, pp. 1–6. DOI: 10.1109/TENSYMP52854.2021.9550907.

[10] J. Estrada, P. Sidike, X. Yang, and Q. Niyaz, "Deep-learning-incorporated augmented reality application for engineering lab training," *Applied Sciences*, vol. 12, p. 5159, May 2022. DOI: 10.3390/app12105159.

[11] A. Hasib, S. S. Khan, J. F. Eva, *et al.*, "Bdsl 49: A comprehensive dataset of bangla sign language," *arXiv preprint arXiv:2208.06827*, 2022.

[12] M. Islam, M. Uddin, M. Ferdous, S. Akter, and M. Akhtar, "Bdslw-11: Dataset of bangladeshi sign language words for recognizing 11 daily useful bdsl words," *Data in Brief*, vol. 45, p. 108 747, Nov. 2022. DOI: 10.1016/j.dib.2022.108747.

[13] T. Tazalli, Z. A. Aunshu, S. S. Liya, *et al.*, "Computer vision-based bengali sign language to text generation," in *2022 IEEE 5th International Conference on Image Processing Applications and Systems (IPAS)*, vol. Five, 2022, pp. 1–6. DOI: 10.1109/IPAS55744.2022.10052928.

[14] [Online]. Available: https://pypi.org/project/mediapipe/.

[15] *A Guide to Long Short Term Memory (LSTM) Networks — knowledgehut.com*, https://www.knowledgehut.com/blog/web-development/long-short-term-memory, [Accessed 18-09-2023].

[16] *Introduction to MediaPipe — learnopencv.com*, https://learnopencv.com/introduction-to-mediapipe/, [Accessed 18-09-2023].

[17] *MediaPipe | Google for Developers — developers.google.com*, https://developers.google.com/mediapipe, [Accessed 18-09-2023].

[18] M. Rastogi, *Tutorial on LSTM: A computational perspective — towardsdatascience.com*, https://towardsdatascience.com/tutorial-on-lstm-a-computational-perspective-f3417442c2cd, [Accessed 18-09-2023].