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

Conference Paper · March 2023

DOI: 10.1109/ICCIKE58312.2023.10131801

3 authors:

Md. Walid Foysol
Daffodil International University

Sk. Estiaque Ahmed Sajal
Daffodil International University

Mohammad Jahangir Alam
Daffodil International University

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

1st Md. Walid Foysol
Dept. of CSE
Daffodil International University
Dhaka, Bangladesh
walid15-2660@diu.edu.bd

2nd Sk. Estiaque Ahmed Sajal
Dept. of CSE
Daffodil International University
Dhaka, Bangladesh
ahmed15-2380@diu.edu.bd

3rd Mohammad Jahangir Alam
Dept. of CSE
Daffodil International University
Dhaka, Bangladesh
jahangir.cse@diu.edu.bd

***Abstract*—The 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 normal individuals and deaf-mute individuals have a communication gap. On the strength of technological welfare, it is now possible to capture any deaf-mute person'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: collecting data and processing it, training and testing the chosen neural network, and finally, real-time testing. 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.**

***Index Terms*—Bangla Sign Language Detection System, MediaPipe Holistic, LSTM, Bangla Sentence Gestures, Real-time Gestures Deteciton.**

I. Introduction

On the subject of expressing inner voice, deaf and mute people face more difficulties than normal people, which creates communication gaps in society. In the case of helping them communicate smoothly, proper application of technology is the key to make these people feel part of 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 lives as a burden on society. Because they can't communicate with other people with words. Normally, deaf or mute people communicate using hand gestures or sign language, creating different movements and positions of hands that indicate different meanings. According to the 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. For the benefit of Bengali deaf and mute people, the Center for Disability in Development (CDD) in Bangladesh has standardized a sign language 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, facial emotion, and hand forms. People who are hard of hearing communicate with others via sign language [6]. It is their only way of communicating with others that most people can't understand, but AI research suggests that this could change. Machine translators that have already been taught to interpret human speech can help individuals communicate more effectively [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, significant efforts have been made to solve picture classification problems using embedded devices and machine learning techniques [3]. When deep learning approaches finally beat traditional feature extraction-based algorithms, they became the favored method of analysis among academics [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. This simultaneous hand gesture detection system's main objective is to swiftly and precisely recognize hand movements [12]. Hand recognition uses algorithms and neural networks to analyze a hand's motion before identifying patterns of that motion. [12] Our desire is to construct a hand gesture identification system that will only require a camera to take input and provide data to the model to identify the sign language.

II. Motivation

The idea that sign language is universal is a frequent fallacy, yet this is untrue. In actuality, much such as 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. A formal sign language for the Bengali deaf and mute people was established by the Center for Disability in Development (CDD) in Bangladesh, and it is now used in all schools for the speech and hearing impaired. 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 field of research is behind and suffering at the moment. On top of that, unlike other sign languages, Bengali sign language has not benefited as much from study regarding its recognition. 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.

III. Related Work

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 the extraction of features from the gesture image and SVM (Support Vector Machine) as a classifier. Their testing result on their testing dataset gave an average recognition rate of 86.53%. M. S. Islam et al. [2] conducted experiments to create a potent model of recognizing 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 five sign gestures, they prepared the data set for use, then built a portable version on Raspberry Pi. 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 (0–9), and every indication has 31 images. Total: 310 images. Their overall accuracy after using CNN is 92.88%. S. Rayeed et al. [6] used MediaPipe, a cross-platform depth-map estimation tool, to offer a full dataset for Bangla sign digits from zero to nine in Bangla. MediaPipe had been tested on an ASL benchmark dataset. Many classifiers were tested on the proposed dataset, but only Support Vector Machine achieved the desired 98.65% accuracy. 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 performed successfully with an 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 the Kinect sensor for accurate depth tracking, which provided 3D tracking and improved detection accuracy by 90% outperforming some pretrained models. T. Abedin et al. [9] worked on a noble architecture called "Concatenated BdSL Network" which is a CNN-based image network combined with a pose estimation network. They achieved 91.51% of accuracy with their approach. A. Abraham et al. [10] demonstrate a gadget that allows silent people to enable an unique form of communication using an Arduino Uno board, some flex sensors, and an android app. The Global System for Mobile (GSM) is used to transfer text messages, which are then converted to speech by the application.

IV. Research Methodology

The proposed system will use image processing techniques to recognize signs from a video sequence and classify them into their respective categories. 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. 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.

A. 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 (['do you need something', 'hope to see you again', 'how are you', 'how can i help you', 'i am fine', 'i like you', 'it is a beautiful day', 'what do you do', 'what is your name', 'where do you live', 'you are so beautiful', 'what time is it', 'happy Birthday', 'i am sorry', 'thank you']) 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. The next stage is to collect data in folders after generating the gesture folders. After that, 30 video files are created for each movement or gesture using a for loop.

The key-points extraction approach was used to collect the array of key-points for each frame and the video data for each move in the following step. The next step is to record the NumPy array of gathered key points for each frame into each movie folder. The array really goes to 49.npy, however, since we are compiling 50 NumPy arrays for each movie. 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.

B. Data preparation

In the data collection process, we collected data for each gesture. To prepare the data for training and testing, we will label all the gestures with a single index, where the first gesture

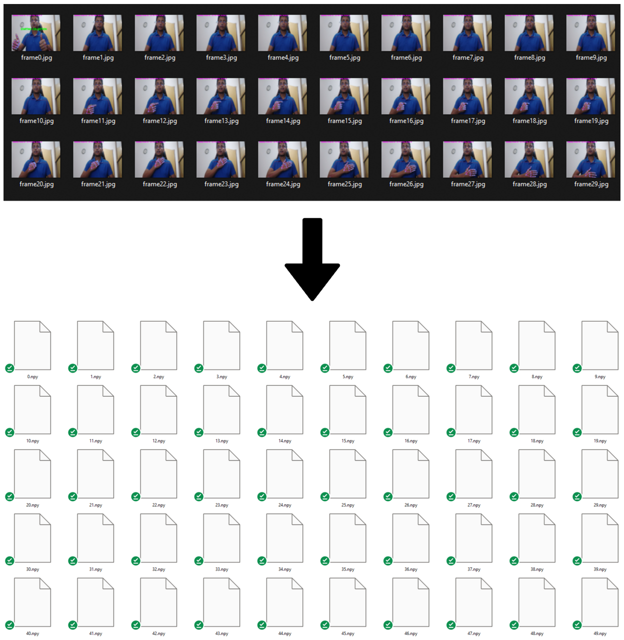

Fig. 1. Collection of 50 frames

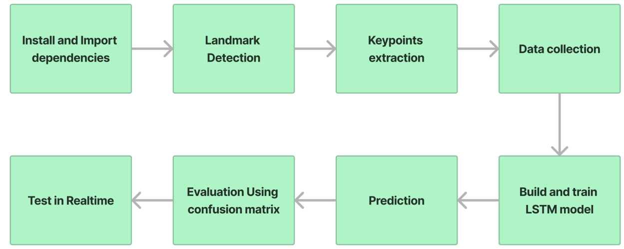

Fig. 2. Proposed Method to Predict Depression

will be labeled with 0, the second one will be 1, and so on. After that, data preparation is used to format the main points that were extracted from the data. All key point arrays for each motion are saved as one NumPy array (X) in the data structure, which is then mapped to another NumPy array of labels (Y). The category function is then used to convert Y into a binary class matrix. The data is divided into training and testing sets using the train-test split function when preprocessing is finished. The remaining 20% of the data were utilized for testing, while the remaining 80% were used for training.

C. 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. We created a dataset to guarantee precise landmark identification from the user's hands and body using the MediaPipe solution of Google and Python's openCV library, and we continued to employ these techniques in the real-time detection stage of the gesture recognition system to guarantee accuracy. The key parts of the proposed system may be seen in 2. First, the dataset construction procedure is carried out, and landmarks are extracted in the dataset module. The data is then processed in the preprocessing module and transmitted into the LSTM module, where training for gesture detection takes place. The real-time feed is produced by a camera and analyzed by the models once the model's hyperparameters have been adjusted. The results are shown as text next to the appropriate gesture.

D. Landmarks Detection and Keypoint Extraction

MediaPipe Landmarks Detection is an open-source library that 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.

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 picking out key points, researchers can compare different signs more accurately and see and understand them better.

E. Model Generation and Model Testing

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.

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.

V. Algorithms Details

This study examines the combined use of a 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.

A. Holistic Pipeline

Motion may be detected in a number of ways using a holistic pipeline. By identifying patterns in a scene, it may recognize regular movements, such as the movement of a human.

The pipeline of MediaPipe, which offers a wide range of alternatives, is made up of three parts: face, stance, and hand. The full model retrieves 543 unique landmarks in total (468 faces, 33 poses, and 21 hands). Using the MediaPipe holistic model provided in the MediaPipe Python package, only 75 distinct landmarks (33 poses, 21 left hand, and 21 right hand) are retrieved here. We created a function called MP detect that needs two inputs. Picture: The picture that detection is being done on. The model for mediaPipe detection (MP holistic; holistic model). Before sending the image to the model, the function converts it from BGR to RGB (). The outcome is then stored. After converting the image back to BGR format, the method returns the image together with the result that was previously kept. We create a different function (draw polished landmarks) that takes the output from the methods before as parameters. By sketching the landmarks with the help of this function, we demonstrate real-time hand detection. The X, Y, and Z coordinates of the key points are derived from the detection results when real-time hand detection has been successfully completed (the saved result supplied by the mediapipe detection function discussed above). We create a different method (extract keypoints) that takes the detection results as an argument and extracts the coordinates. This function produces the concatenated array of all arrays containing the coordinates of the central points of the holistic model. During data collection, this function is used. In order to avoid landmark clusters, we only presented hand landmarks (see in 3) and posture landmarks (see in 4) on the screen during the identification phase while still using the holistic model to extract essential points for hand and position [16].

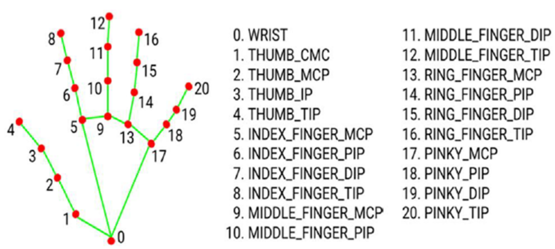

Fig. 3. The hand_landmarks model of MediaPipe

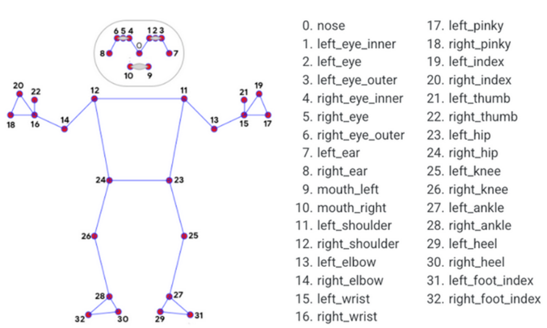

Fig. 4. The pose_landmarks model of MediaPipe

B. Long Short-Term Memory (LSTM)

Recurrent neural networks (RNNs) with Long Short-Term Memory (LSTM) may learn long-term dependencies. LSTMs feature memory cells that can retain data for a long time, in contrast to conventional RNNs. They are then able to learn from and forecast the future using larger data sequences.

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. 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 [17].

VI. Results and Discussion

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.

A. 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. The model’s loss is the difference between what was expected and what was seen. The lower the model’s loss, the better the model’s performance. Here, the model loss is decreasing as the method progresses. The figure 6 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%.

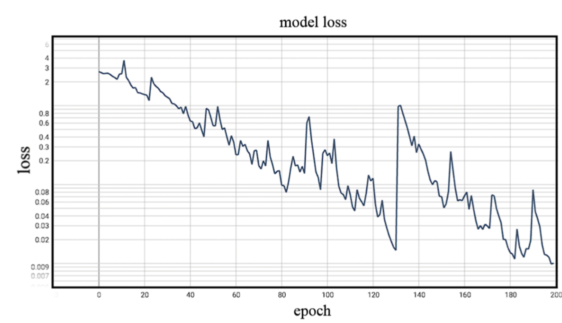

Fig. 5. Training Loss

B. 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

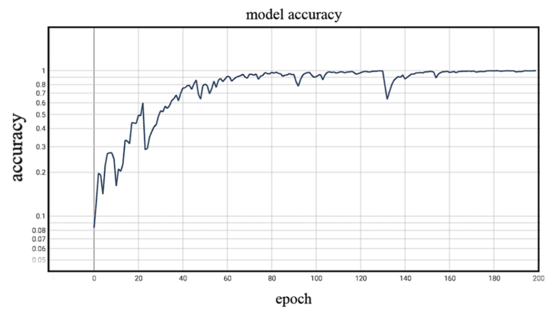

Fig. 6. Training Accuracy

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. 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).

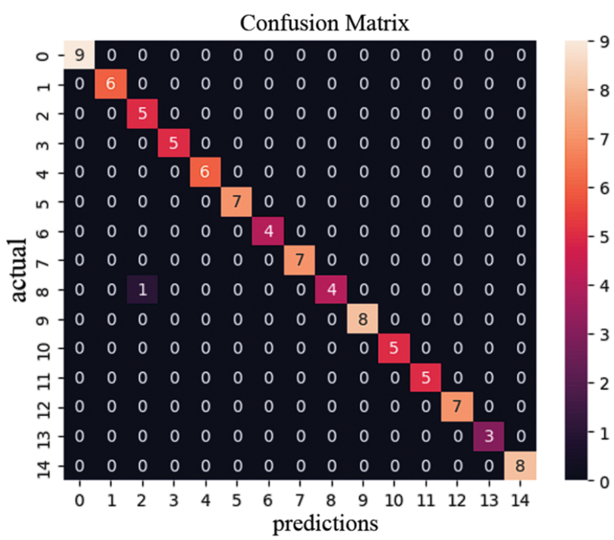

Fig. 7. Confusion Matrix

C. Classification Report

Table I 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 helping to make informed decisions about how to improve it.

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

TABLE I
CLASSIFICATION REPORT FOR MODELS TRAINED WITH CHARACTERS DATASETS

D. Result

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. 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 and distinguishing gestures.

VII. Conclusion and Future Scope

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.

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. Finally, the system could be further developed to allow for more sophisticated recognition of signs, making it more accurate and reliable.

VIII. System Prototype

The purpose of our work is to create a system that can recognize Bengali sign language motions. So we created a prototype, which is a web-based application, with the help of the model that we have built with LSTM. Our prototype website has two tabs. One is "Home," and the other is "Sign Tutorial." On the "Sign Tutorial" tab, a user can learn all fifteen gestures to examine our prototype.

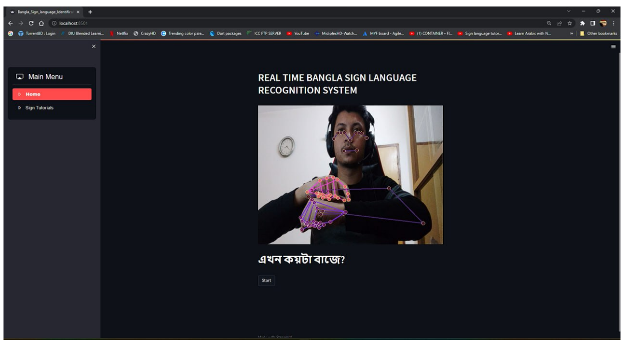

Fig. 8. Home Page

Then on the "Home" page, there is a button named "Start." After pressing the button, our prototype will start capturing 50 frames with the help of the camera. The prototype will capture the gesture and then compare it with the pre-trained LSTM model. Then it will show the output in Bangla language. The start button will appear again, and we can run another prediction by pressing it again.

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 dimensional 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. A convolutional neural network and user-defined ROI segmentation are used in this effective sign language translation system. The second annual international conference on communication engineering and technology will take place in 2019. (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 Elchouemi. 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.

[16] Google. (n.d.). MediaPipe Holistic. Retrieved from MediaPipe: https://google.github.io/mediapipe/solutions/holistic.html

[17] Olah, C. (n.d.). Understanding LSTM Networks. Retrieved from Colah's blog: https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/

[18] Malti Bansal, Apoorva Goyal, and Apoorva Choudhary. A comparative analysis of k-nearest neighbour, genetic, support vector machine, decision tree, and long short term memory algorithms in machine learning. Decision Analytics Journal, page 100071, 2022.

[19] Arpita Halder and Akshit Tayade. Real-time vernacular sign language recog- nition using mediapipe and machine learning. Journal homepage: www. ijrpr. com ISSN, 2582:7421, 2021

[20] Moh Harris, Ali Suryaperdana Agoes, et al. Applying hand gesture recogni- tion for user guide application using mediapipe. In 2nd International Sem- inar of Science and Applied Technology (ISSAT 2021), pages 101–108. At- lantis Press, 2021.

[21] Andre Jallen S Ong, Melvin Cabatuan, Janos Lance L Tiberio, and John An- thony Jose. Lstm-based traffic gesture recognition using mediapipe pose. In TENCON 2022-2022 IEEE Region 10 Conference (TENCON), pages 1–5. IEEE, 2022.

[22] Kunal Sharma, Kunwar Abhayjeet Aaryan, Urvashi Dhangar, Richa Sharma, and Shweta Taneja. Automated indian sign language recognition system using lstm models. In 2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), pages 461–466. IEEE, 2022.