



Daffodil
International
University

Breaking Language Barriers: A Multimodal Approach to Bangla and English Sign Language Detection

Submitted by

ISTIYAK AHMED

201-35-2967

Department of Software Engineering

Daffodil International University

Supervised by

MD. SHOHEL ARMAN

Assistant Professor

Department of Software Engineering

Daffodil International University

This Thesis report has been submitted in fulfillment of the requirements for the Degree of
Bachelor of Science in Software Engineering.


Fall-2023

© All right Reserved by Daffodil International University

Approval

This thesis titled on “**Breaking Language Barriers: A Multimodal Approach To Bangla And English Sign Language Detection**”, submitted by **Istiaq Ahmed (ID: 201-35-2967)** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

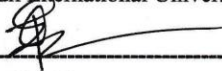
BOARD OF EXAMINERS



Afsana Begum
Assistant Professor

Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Chairman



Md Rajib Mia

Lecturer

Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 1



Musabbir Hasan Sammak

Lecturer

Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 2



Dr. Md. Manowarul Islam

Associate Professor

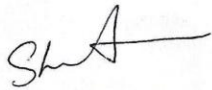
Department of Computer Science & Engineering
Jagannath University

External Examiner

Declaration

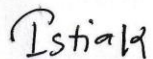
I hereby declare that, this thesis report is done by me under the supervision of Md. Shohel Arman, Assistant Professor. Department of Software Engineering, Daffodil International University, in partial fulfillment my original work. I am also declaring that neither this thesis nor any part therefore has been submitted else here for the award of Bachelor or any degree.

Supervised By



Md. Shohel Arman
Assistant Professor,
Department of Software Engineering,
Daffodil International University.

Submitted By



Istiak Ahmed
ID: 201-35-2967
Department of Software Engineering,
Daffodil International University.

Acknowledgement

I want to start by expressing my gratitude to Almighty God for bestowing His divine favor and allowing me to complete my undergraduate thesis.

I would like to express my gratitude and deep amount of respect to my supervisor **Md. Shohel Arman, Assistant Professor** in the "Department of Software Engineering" at "Daffodil International University in Dhaka. His profound knowledge and guidance in the segment on "Machine Learning" help me a lot to complete this entire thesis work. It has been made possible by his never-ending empathy, academic leadership, continuous motivation, regular and vigorous monitoring, constructive criticism, helpful counsel, reviewing numerous subpar manuscripts, and fixing them at every level.

I wish to extend my sincere appreciation to **Dr. Imran Mahmud, Head of the "Software Engineering Department,** Faculty of Science and Information Technology, as well as to the other professors, faculties, and personnel of the SWE Department of "Daffodil International University" for their considerate assistance in accomplishing my work.

Last but not least, I must respectfully thank my parents for their unwavering love and patience.

Abstract

Hearing loss is a barrier to living a normal life for the deaf. Approximately 2.6 million people in Bangladesh are suffering from hearing loss. For these people, sign language plays a very important role in communicating with others. Traditional methods of sign language are limited by availability, cost, and accessibility. This paper proposes an approach to sign language detection using MediaPipe, a cross-platform framework for building pipelines of machine learning and computer vision algorithms. The proposed model can process different machine learning algorithms to detect Bengali and English letters, words, and numbers from sign language gestures captured by a webcam with high efficiency. The model is trained on a dataset of over 135,000 hand gesture images and it has achieved more than 98% recognition accuracy. It can also process data in variations of lighting, background, and hand posture making it suitable for real-world applications. The proposed system provides a low-cost, accessible, and real-time sign language interpretation tool that has great potential to revolutionize communication problems among hearing and deaf people in our country.

Keywords- MediaPipe, deaf community, ensemble learning, machine learning

Table of Contents

Approval.....	i
Declaration	ii
Acknowledgement	iii
Abstract	iv
CHAPTER 1	1
INTRODUCTION:.....	1
BACKGROUND:	2
MOTIVATION:.....	3
PROBLEM STATEMENT:	3
RESEARCH GAP:.....	3
RESEARCH QUESTIONS:.....	4
RESEARCH OBJECTIVE:.....	4
RESEARCH SCOPE:	5
SUMMARY:	5
CHAPTER 2	6
LITERATURE REVIEW:	6
INTRODUCTION:	6
PREVIOUS LITERATURE:.....	6
SUMMARY:	12
CHAPTER 3	13
RESEARCH METHODOLOGY:.....	13
INTRODUCTION:	13
DATASET:.....	13
DATA COLLECTION:.....	14

DATA CLEANING:	15
DATA PREPROCESSING:	15
MACHINE LEARNING:	18
ENSEMBLE LEARNING:	19
BAGGING:.....	20
BOOSTING:.....	20
XGB CLASSIFIER:.....	21
RANDOM FOREST CLASSIFIER:	22
K-NEAREST NEIGHBORS CLASSIFIER:	23
FLASK:	27
EVALUATION METHODS:.....	27
SUMMARY:	28
CHAPTER 4	29
RESULTS AND DISCUSSION:	29
INTRODUCTION:	29
RESULT:.....	29
DISCUSSION:.....	35
SUMMARY:	35
CHAPTER 5	36
CONCLUSION:	36
LIMITATIONS:.....	36
FUTURE SCOPE:	37
REFERENCES	38

List of Figures

Figure 1: Raw Data	13
Figure 2: Datasets	15
Figure 3: Data Preprocessing.....	15
Figure 4: Hand Landmark in MediaPipe (MediaPipe github).....	17
Figure 5: Data process and prediction process	25
Figure 6: Accuracy (%) per classification model	29
Figure 7: Accuracy (%) per classification model	30
Figure 8: precision, recall and F1 score for ensemble	31
Figure 9: precision, recall and F1 score for XGBoost	32
Figure 10: precision, recall and F1 score for Random Forest	32
Figure 11: precision, recall and F1 score for KNN.....	33
Figure 12: Error per model.....	34
Figure 13: Error per model.....	35

CHAPTER 1

INTRODUCTION:

Sign language is a process of expressing thoughts of deaf and hard-of-hearing communities.(Tayade & Halder, 2021) when it's difficult to interact with each other in that case sign language can be used. The world population is rising so the deaf community also growing up. In Bangladesh, around 2.6 million people face problems interacting using language.(Shamrat et al., 2021) Bangladesh Sign Language (BSL) is mainly used in Bangladesh for deaf and hard-of-hearing people.

Nowadays, Bangladeshi people use English in their day-to-day life. In diverse sign language groups, more inclusion is cultivated by the recognition of both the Bangla and English sign language. It is important to know English sign language also. There is a communication gap also with normal people. Users of sign language are not comfortable with existing services because existing methods are facing a lack of attention. Few methods are really good but few of them need extra devices or high-configuration computers. (Siddique et al., 2023)

The ML-based Sign Language Detection system aims to communicate with differently-abled people without the help of any expensive human interpreter. (Sanmitra Rishi et al., 2021) No doubt CNN performs better in the case of sign language detection.(Jāmi‘at al-Baḥrayn & Institute of Electrical and Electronics Engineers, n.d.) But most people are not able to buy high-configuration computers. In this paper, I have tried to find a way to build a CPU-based model with high accuracy that can be implemented in any device.

In addition to recognizing sign language, media pipe is a very useful framework. (Goyal & Velmathi, n.d.) MediaPipe can accurately identify key points of hands. It has real-time capabilities, multimodal integration, and an open-source nature MediaPipe has a good potential

for detecting sign language. By using MediaPipe it is easily possible to convert a hand picture into numeric data depending on various points on hands.

BACKGROUND:

For those who are deaf or hard of hearing, going about their everyday lives can be like scaling a mountain. Their route is blocked by walls of communication barriers caused by a lack of accessibility and an inadequate comprehension of sign language. Many people find it difficult to be understood and heard, which makes them feel alone and excluded. A glimmer of optimism does, though, as there is an increasing willingness to learn sign language. With this wave of understanding, obstacles to communication should vanish and inclusion should rule the world. Imagine lively signs resonating in classrooms, offices bustling with natural conversation, and public areas open to all. This vision, albeit distant, kindles a spark of hope. By removing barriers to communication and promoting proficiency in sign language, we can build a world where being deaf is not a barrier but rather a unique and vibrant culture that is accepted by all. This brief explanation captures the core of the opportunities and difficulties faced by the deaf and hard-of-hearing communities, highlighting the potential of sign language to bridge the gap and foster a more inclusive society. (Tayade & Halder, 2021) To reduce those people's problem sign language detection might be a good way. Where without any barrier by using a camera normal people can understand what deaf or hard-of-hearing people want to say.

Many people feel as though there is a barrier preventing them from connecting and understanding people who use sign language when they speak with them. Words flow freely, but without the comprehension required for sign language, discussions turn annoyingly one-sided. The only people who can begin to narrow the enormous gap between the hearing and deaf communities are those who have made the effort to understand the language of hands and expressions. In a society where many people speak multiple languages, miscommunication can result in misconceptions, which makes the deaf and hard of hearing feel alienated and alone. (Goyal & Velmathi, n.d.) When communication breaks down, doors of opportunity may close, preventing deaf and hard-of-hearing people from accessing social networks, necessities, and educational chances. Maximum normal people are not able to understand sign languages. It is

difficult for them to communicate with deaf people. That time talented deaf or hard-of-hearing people also can compete with everyone.

Many people want to learn sign language. But because of a lack of enough resources, and opportunities they cannot learn properly. Specially in Bengali, it is more difficult though there is no fully organized sign language method for this. (Siddique et al., 2023)

MOTIVATION:

Nowadays, English is a very essential language. In our day-to-day lives, we cannot ignore English. For that with the Bengali sign language model, I have added English sign language. By this, this system will be much more user-friendly and give a better feel to an end-to-end user. Multi-modal sign language detection always gives extra facilities to the user. Real-time sign language detection can give users a good experience. It can help everyone with interactive communications. Many people have an interest in learning sign language but there is a lack of resources. Or the resources are not enough. The main motive of this research is to mitigate language barriers with deaf or hard-of-hearing people and make it easier to learn sign language.

PROBLEM STATEMENT:

Bengali is one of the most widely spoken languages in the world, and it echoes in the busy tapestry of India. However, tucked away in its brightness is Bengali Sign Language (BSL), a hidden language. Communicating with members of the deaf and hard-of-hearing populations who use BSL can be a confusing task. There are few, expensive, and obscure learning resources available. In addition to taking a lot of time, the current approaches may discourage motivated learners from taking on this linguistic adventure.

RESEARCH GAP:

- Existing research primarily focuses on either Bangla or English sign language detection separately.

- There is a lack of algorithms that can effectively detect both Bangla and English sign languages within a single unified framework.
- Deep learning approaches are commonly used, but they can be computationally expensive.
- Many studies rely on relatively small datasets for training and evaluation.

Most research focuses on identifying either English or Bangla sign language separately because there are no general algorithms that can identify both in one cohesive system. Current techniques generally make use of deep learning techniques that need a lot of compute. However, because of their computing requirements, these might provide difficulties. Additionally, a lot of research uses small datasets for assessment and training, which may reduce the robustness and generalizability of the model. By filling up these gaps, accessibility in this field may be advanced and more effective and adaptable algorithms for concurrently identifying Bangla and English sign languages may be developed.

RESEARCH QUESTIONS:

- What are the most effective techniques for sign language gesture detection using machine learning and AI?
- Can the sign language detection system developed for Bangla and English be generalized to support other sign languages with minimal adjustments?
- How can a diverse dataset for sign language gestures be collected and annotated effectively to train the system?

RESEARCH OBJECTIVE:

To close the gap in current research that focuses only on individual languages, develop a complete algorithm that can recognize both Bangla and English sign languages within a single, cohesive framework. Look into and create methods for detecting sign language that are both economical and computationally efficient. In particular, look at alternatives to deep learning techniques that need a lot of resources. To increase the accuracy and generalizability of sign

language identification algorithms, investigate methods for dataset augmentation or make use of bigger datasets to strengthen the resilience of the model. Examine how sign language technology may boost job prospects, make communication simpler, and make services more accessible to Deaf people in order to determine how it will affect society.

RESEARCH SCOPE:

- Breaking Down Communication Barriers.
- Sign language will be easier for hearing people to learn and understand.
- The technology can expand career opportunities for Deaf individuals.
- Deaf individuals will have easier access to various services.

Encouraging inclusion by removing obstacles to communication, technology that makes it simpler for hearing people to learn and comprehend sign language is available. By creating a wider range of professional options, this innovation also broadens the Deaf community's career possibilities. Enhancing involvement and engagement across several domains, greater access to services also makes Deaf persons' experiences more seamless. All things considered, this technology advancement improves accessibility and communication, which benefits the Deaf community's abilities to integrate into society and pursue careers.

SUMMARY:

Communication challenges are faced by growing deaf groups, underscoring the necessity for accessible sign language solutions. The techniques now in use for identifying sign language in Bangla and English are frequently costly, device-dependent, or inaccurate. The goal of this research is to create a CPU-powered, dependable model that uses MediaPipe to close the communication gap between the two languages and provide a cost-effective, universal tool for all devices. In other words, the goal of this research is to enable everyone to use sign language detection.

CHAPTER 2

LITERATURE REVIEW:

INTRODUCTION:

As a researcher, I evaluated earlier work, and research in the literature review segmentation. By this I have learned what research has previously been performed and a general overview of this domain. I have focused on what methods previously used in this kind of study, datasets, and findings. Finally, after analysis, I focused on the limitations of previous studies and based on these limitations my goal is to improve results.

PREVIOUS LITERATURE:

Arpita Halder et al (2021) conducted machine learning models in different languages. MediaPipe framework had been used to detect hand gestures in that case. It can process American, Indian, Italian, and Turkish sign language. In this study, SVM, KNN, Random Forest, Decision Tree, Naïve Bayes, ANN, and MLP have been used. The research faced challenges in the case of detecting words.(Tayade & Halder, 2021)

Kaushal Goyal et al (2023) focused on Indian sign language recognition using mediapipe. CNN and LSTM are used in this case. Mainly this study showed a comparison between CNN and LSTM models. CNN outperforms here. LSTM shows only 85% accuracy. There can be some scope to increase the accuracy.(Goyal & Velmathi, n.d.)

Khuat Duy Bach et al (2021) introduced Vietnamese sign language detection using a modified mediapipe version. This study depends on the mediapipe framework. Better accuracy will come if the modified version recognizes hand gestures better. LSTM model performs here with 90% accuracy. Here, the preprocessing data was not as good as expected.(Duy Khuat et al., 2021)

Sumaya Siddique et al (2023) approach deep learning method for Bengali sign language detection. This paper uses two Bengali sign language datasets, a public database known as Okkhornama (Talukder et al., 2021) and a collected custom dataset. In this paper, Dectectron2, EfficientDet-D0, and YOLOv7 have been used. For detecting sign language this study requires extra hardware resources. Among all the models, The Detectron2 model accomplishes the best accuracy at 94.915%.(Siddique et al., 2023)

Md Shafiqul Islalm et al (2019) found that CNN shows 99.80% accuracy in case of detecting Bengali sign language. The dataset used in this study contains almost 30000 samples. The author mentioned that in the future, will compare the performance of the proposed method with existing methods on the same dataset on a large scale.(Jāmi‘at al-Baḥrayn & Institute of Electrical and Electronics Engineers, n.d.)

F. M. Javed Mehedi Shamrat et al (2021) focus on the CNN model for Bengali sign language detection. This study achieved 99.8% accuracy. Xiaomi Vidlok W77webcam has been used for collecting data. The dataset consists of ten hand signs (a total of 310 pictures), with each signing class consisting of 31 images captured at various lengths, orientations, and intensities. Bengali numerical sign language is depicted in the photos. This study is only about numerical data. Improvement must be needed in that dataset to apply that in real life.(Shamrat et al., 2021)

P. Rishi Sanmitra et al (2021) introduced the Single Shot Detection method with real-time sign language detection. The LabelImg software is used for graphically labeling the images which is further used when recognizing the images. The results have shown up to an average accuracy of 85% (Sanmitra Rishi et al., 2021)

N. Padmaja et al (2022) state that ResNet-50+Faster R-CNN can perform with 86% accuracy in the case of real-time sign language detection. This study only covered 4 signs- V, L, U, and I Love You. (“Real Time Sign Language Detection System Using Deep Learning Techniques,” 2022)

Dardina Tasmere et al (2020) used the CNN method for hand gesture recognition. CNN always performs well. 99.2% accuracy was shown in this case. The dataset consists of 3219 images from six different people. Hand detection process has been processed by color segmentation. This study was only for Bengali sign language detection.(Tasmere & Ahmed, 2020)

Shobhit Tyagi et al (2023) focused on YOLOv8 and YOLOv5 for American sign language detection. The dataset used images for testing, training, and validation in the ratio of 1: 21: 2 respectively (72 for testing, 1512 for training, 144 for validation) and the new model has been trained for 80 epochs. YOLOv8 performed better than YOLOv5 in the case of sign language detection.(Tyagi et al., 2023)

The dataset prepared in this study is the largest image database for BdSL Alphabets and Numerals, aiming to reduce interclass similarity while dealing with diverse image data including various backgrounds and skin tones. The best model in this study exceeded previous state-of-the-art efforts in the recognition of Bengali Sign Alphabets and Numerals. ResNet18 outperformed MobileNet_V2 and EfficientNet_B1 in terms of overall accuracy, precision, sensitivity, F1 score, and specificity after five-fold cross-validation and achieved the highest overall accuracy of 99.99%.(Podder et al., 2022)

The research culminated in the successful creation and deployment of an American Sign Language (ASL) fingerspelling translator. Leveraging convolutional neural networks (CNNs), the project obtained highly reliable models capable of accurately classifying ASL letters ranging from a to e and a to k. The achievement demonstrates strong performance in interpreting and translating sign language gestures into corresponding alphabetic characters. (B Garcia & SA Viesca, 2016)

Indian Sign Language identification using LSTM and GRU achieved around 97% accuracy across 11 distinct signs by using an IISL2020 dataset of various hand motions. The IISL2020 dataset is a customized dataset made up of 630 samples overall and 11 static indicators that were collected from 16 respondents, both male and female, who were between the ages of 20 and 25. With the suggested technique using the dataset it achieved a 97% accuracy rate in sign identification.(Kothadiya et al., 2022)

The method achieved great accuracy in detecting sign language from continuous video data by combining stacked LSTMs for automated classification with heuristics for video stream segmentation. When evaluated independently the approaches yield encouraging results and suggest that sign language can be used to facilitate ongoing natural communication between humans and robots. The study emphasizes how important it is to understand and identify sign language to promote genuine human-machine interaction.(Mocialov et al., n.d.)

The primary conclusions of the paper highlight the benefits of EnStimCTC in reducing WER, the superior short-term temporal modeling capabilities of 3D CNN approaches, the effectiveness of proximal transfer learning, the superior performance of 2D CNN-based models with an intermediate per gloss representation in CSLR datasets, and the effectiveness of 3D CNN-based architectures in isolated SLR. The "Dataset" used in the study is a brand-new RGB+D dataset for Greek sign language that includes sentence annotations, temporal gloss annotations, and translations into Modern Greek. It contains configurations for GSL isol, continuous GSL SI, and continuous GSL SD. Compared to other datasets, this one includes seven native GSL signers who record encounters with public services. It also has about three times as many unique gloss phrases and almost twice as much vocabulary, all with fewer training examples. The discrete subset of the dataset selects examples from the same distribution as the continuous subset. (Adaloglou et al., 2020)

The significant discoveries include the development of a hybrid artificial neural network design, a new technique for extracting features from an image using binary robust invariant scalable key points, and the use of numerical image features from a convolutional neural network using ensemble models. The ASL alphabet, ASL fingerspell, Cambridge hand gesture, NUS I, NUS II, and ISL digits are the datasets that were used in the study. These files cover Indian Sign Language (ISL) numbers, fingerspelling, hand gestures, and sign language alphabets. Through the linked URLs, these datasets are accessible to the general public. Shortcomings of the study include the requirement to improve the system's training process by including different datasets of sign images taken in different scenarios and adapting the system to record real-time sign language usage that incorporates natural language rules. (Jana & Krishnakumar, 2022)

The primary conclusion of the paper is that it provides a system that can automatically recognize Indian sign language numerical symbols with up to 97.10% accuracy using isolated images and a standard camera. The experimental results show that the system can be used as a "working system" for numerical recognition in Indian Sign Language. The methodology covered automatic recognition of Indian sign language numerical symbols using discrete images, feature extraction using hierarchical centroid and direct pixel value techniques and classification using neural network and KNN methods. (Sharma et al., 2014)

(Islam et al., 2023) have overcome two fundamental obstacles in Bangla sign language recognition: limited data and model constraints. This is a significant advancement in the field. The MVBSL-W50 dataset offers a solid basis for training and assessing recognition systems. It is a treasure mine of 4,000 high-resolution films covering 50 isolated words in 13 categories. In the meantime, the innovative attention-based Bi-GRU model solves the riddles of human pose information, recognizing not only hand gestures but also minute body movements and facial emotions with an astounding 85.64% accuracy rate. These ground-breaking accomplishments have the potential to greatly enhance communication, foster acceptance and inclusivity for the Deaf and hard-of-hearing community, and pave the path for a time when everyone will be able to enjoy the silent symphony of Bangla sign language.

The main findings of the paper are as follows: a dataset (BdSLImset) containing pictures of Bangladeshi signs in various lighting and background conditions was established; a real-time sign language detection method utilizing the Faster Region-based Convolutional Network method was developed; and a high accuracy of roughly 98.2 percent was attained in a short detection time of roughly 90.03 milliseconds. Restrictions on the available datasets, time-consuming data training, continuing research, and future upgrades to the system. The "Dataset" for this study is called BdSLImset, and it is made up of images of signs from Bangladesh with random lighting and backgrounds. Every gesture is depicted in more than 100 photos, each with ten different labeled sign languages. The images are smaller than 200 kb and have a maximum resolution of 700×1280 . The dataset is divided into training and testing sets using an 8:2 ratio. The authors also used bounding boxes to identify and choose the region of each hand gesture. (Hoque et al., 2018)

This ground-breaking method represents by (Talukder & Jahara, 2020) a significant advancement in the communication of the Deaf and hearing worlds. Consider the following scenario: you record a series of photos or a video stream in Bangladeshi Sign Language (BdSL), and you can hear the words or see them rendered as text right away! This state-of-the-art device does exactly that, interpreting BdSL in real-time with astounding speed and accuracy. It outperforms current BdSL identification methods with an astounding 97.95% object detection accuracy, demonstrating its proficiency in hand gesture and body position recognition. Finally, it guarantees lag-free, flawless communication with a response time of 33 milliseconds per frame.

Future interactions will be able to easily transition between the spoken and silent realms thanks to this ground-breaking approach, which will promote inclusivity and understanding like never before. 12.5k photos representing 49 distinct signs—10 numbers, 36 regular BdSL characters, and three suggested signs for sentence generation—make up the dataset utilized in the study. The dataset's suitability for real-time object detection tasks was ensured by taking it under a variety of lighting and background conditions.

In this study paper, a ground-breaking development by (Katoch et al., 2022) that can help close the communication gap between the hearing/speech impaired community and the general public is discussed: a real-time sign language recognition system. The system's main advantage is its extraordinary accuracy, which surpasses 99% when it comes to identifying Indian Sign Language (ISL) characters and alphabets. Imagine the effect: a silent conversation that quickly transitions into understanding between sign motions and spoken words. The potential for this real-time technology to transform daily encounters and improve communication in social, professional, and educational contexts is enormous. By removing barriers to communication and promoting an inclusive society, it provides the Deaf population with a potent weapon for active engagement and participation. This study is a major step toward a time when spoken and sign language coexist together, enhancing communication between people and dismantling barriers to quiet. The study made use of a manually generated dataset for Indian sign language that had 26 unique alphabets (A–Z) and 10 number signs (0–9) from three separate participants. With roughly 1000 images of each sign in each sign folder, there are roughly 36,000 total photos for the two image acquisition methods. The dataset was recorded using the webcam, and it was made to work well in a range of environments.

The study by (Rahman et al., 2019) primary conclusions are as follows: the population with hearing or listening impairments has been growing, with 466 million people living with them as of early 2018 and 400 million by 2050. The suggested CNN model achieves 100% accuracy in predicting every sign, which greatly increases the recognition accuracy of American Sign Language (ASL) reported by some prominent existing approaches. The Massey University Gesture dataset (MU HandImages ASL) comprising 2425 photos in PNG format from 5 persons was one among the datasets utilized in the study. Ten examples of each sign language digit were gathered from each of the 218 Turkey Ankara Ayranci Anadolu high school students in the

dataset. The Center for Vision, Speech and Signal Processing group at the University of Surrey, UK, gathered the ASL finger spelling dataset, which consists of more than 65,000 color photos of 24 static ASL alphabet signs. 87,000 samples over 29 classes (26 for the letters A–Z and three special characters: delete, nothing, and space) make up the ASL Alphabet dataset.

SUMMARY:

From the above studies, I can conclude that many processes and ways are used for sign language detection. However, very few studies mentioned multimodal approaches. Maximum work has been done in deep learning. In my research work, I tried to solve their shortcomings as much as I could.

CHAPTER 3

RESEARCH METHODOLOGY:

INTRODUCTION:

For object detection, MediaPipe has been used. Then I used an ensemble learning technique with a K-nearest neighbors' classifier, Random Forest classifier, and XGBoost classifier algorithms. And compare the ensemble learning result with individual algorithm results for the best outcome. Flask has been applied to my live sign language dataset.

DATASET:

There are lots of data sets available. I have collected data from secondary sources. I have collected different datasets about Bengali and English sign language and used those to train the model.

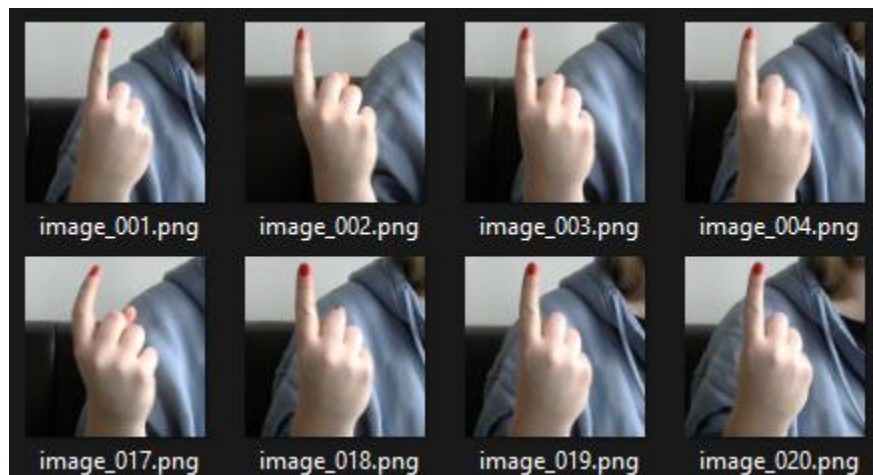


Figure 1: Raw Data

DATA COLLECTION:

I have collected data from secondary sources. Then make a custom data set with different types of data both Bengali and English. BdSL-D1500 (Podder et al., 2022) is one of the biggest sources of my data. Here for every Bengali letter, I have found 1500 images. For Bengali words, I have collected a dataset Bangla-Sign-Language. (Siddique et al., 2023) For Digits from 0 to 9 in American Sign Language I have collected an ASL dataset. (Jana & Krishnakumar, 2022) I have collected English letters from ASL Fingerspelling Images (RGB & Depth) where for every letter I have gotten approximately 5250 images. (B Garcia & SA Viesca, 2016) After collecting all images I combined all the images and made a custom dataset with 135115 images. After merging all data, it is converted into a big custom dataset.


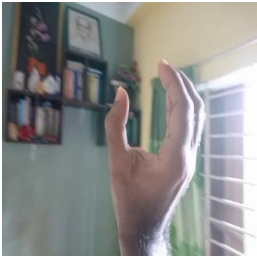
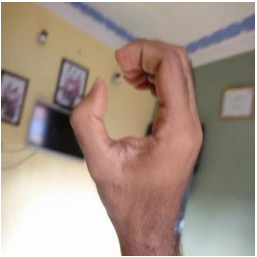






BdSL-D1500			
Bangla Word			
Digits from 0 to 9 in American Sign Language			



Figure 2: Datasets

DATA CLEANING:

After collecting all the different datasets, I have to organize all the images. I have created different folders for different letters and words. That it can be easier to handle datasets. For this task visual studio and Python have been used. Python can easily handle organizing all images for their specific folders. After completing this organization, I prepared the data by using Mediapipe and then again came to data cleaning to handle null values. Maximum data were good. However, a few of them were not specific or contained null values. So, I just drop those bias values or null values.

DATA PREPROCESSING:

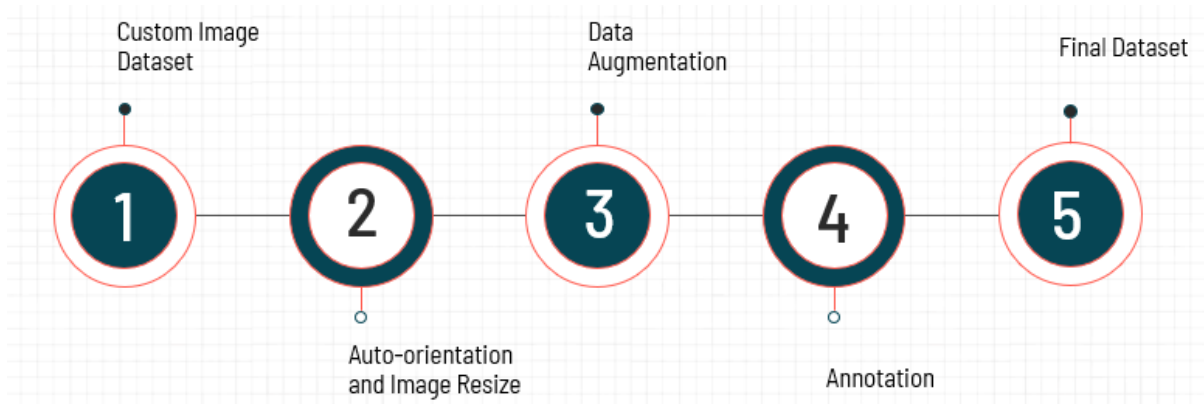


Figure 3: Data Preprocessing

This is the full process I have followed for data preprocessing. Custom image dataset is the first step in the process which I have mentioned in the data collection part how I have done this.

The second step involves automatically adjusting the orientation of the images and resizing them. The orientation (upright, rotated, flipped) of an image is determined by algorithms analyzing its metadata or pixel patterns. Photos are automatically flipped to a standard position, usually with the hands pointing up into the frame. Consistent feature extraction and model training are thus guaranteed. Better feature extraction and pattern identification are made possible by consistent orientation and size, which improves model performance. Because uniform image dimensions minimize computing overhead, they frequently speed up model inference and training. Model training becomes more effective and less biased when photos are resized to help standardize feature scales.

The third step is data augmentation. Though I have taken a large dataset it is always better to make dataset larger. Real-world datasets are sometimes quite little, particularly in specialist fields like sign language. By artificially expanding the range of training instances, augmenting them gives the model a more comprehensive learning experience. Different techniques are used to conduct signs, with minor differences in hand placement, size, and illumination. By simulating these variations, augmentation lessens overfitting and gets the model ready for practical situations. The more modifications the model is subjected to, the less vulnerable it is to particular noise or data distortions, which improves the model's performance on unknown samples.

Machine learning that is suitable for production may be achieved by building pipelines that do inference over any type of sensory input using the MediaPipe framework. The public has access to its code in order to facilitate research endeavors and the creation of technical prototypes. (Zhang et al., 2020a) Within the perception pipeline, functions such as the media processing model, the inference model, and data transformations are fed into the graph modular components of MediaPipe. (Lugaresi et al., n.d.)

Zhang has researched the usage of MediaPipe for hand gesture recognition (Zhang et al., 2020b) using a single RGB camera for AR/VR applications in a real-time system that can predict the human hand skeleton. The pipelines of two models for hand gesture detection may be realized by the MediaPipe in the following ways (Lugaresi et al., n.d.) (Zhang et al., 2020b)

After processing the collected picture, a palm detector model spins it to display the bounding box of the hand in its orientation. 3D hand key points on hand are generated using a hand landmark model using a cropped bounding box photo. A gesture recognizer classifies the essential points of a 3D hand and organizes them into a unique set of motions.

The framework has been built to detect initial palm detectors called BlazePalm.(Suryaperdana Agoes, 2021) Hand detection is a complex task. First, the non-maximum suppression method is applied to the palm, and the palm is trained instead of the hand detector. should stay away from other areas where square bounding box ratios are used to model it and where 3-5 fewer anchors are used. The next is a feature extraction encoder-decoder that can even be employed for small objects and is used for larger scene context awareness. Lastly, by supporting a large number of anchors arising from the high-scale variation, lowers the focus loss during training. (Zhang et al., 2020b) .

Obtains accurate key point localization of 21 key points using a 3D hand-knuckle coordinate that is operated inside the hand areas identified by regression, generating coordinate predictions straight from the model of hand landmarks in MediaPipe. (Zhang et al., 2020b).

see in Figure 4.

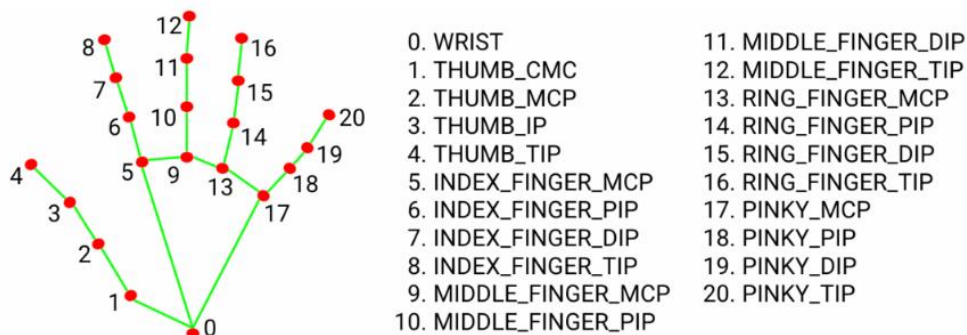


Figure 4: Hand Landmark in MediaPipe (MediaPipe github)

The image's width and height normalize x and y to [0.0, 1.0], while z indicates the depth of the landmark. The three variables that make up each hand-knuckle coordinate of the landmark are x,

y, and z. The precursor is situated at the landmark's wrist depth. The value decreases as the landmark approaches closer to the camera.

For hand gesture detection, a simple algorithm materialization is to calculate the joint state or gestures with a fixed accumulated angle of each finger such as a bent finger or straight finger. Then mapping the finger states obtained earlier to set pre-defined gesture labels according to English and Bengali sign languages.

MACHINE LEARNING:

By machine learning, computers may learn on their own and become more efficient by gathering experience, eliminating the need for traditional programming techniques.(Bi et al., 2019) From a theoretical area, machine learning has evolved into a ubiquitous tool with real-world applications in various scientific and industrial sectors. The ability to draw conclusions from data and make evidence-based judgments gives it strength and has led to important improvements in several industries. The prevalent paradigm in machine learning is still supervised learning, where models are trained on labeled data to do certain tasks like regression or classification. Deep learning has advanced significantly in recent years using sophisticated neural networks to reach previously unheard-of performance and accuracy in domains including autonomous systems, natural language processing, and image identification.(Horvitz & Mulligan, 2015)

supervised, unsupervised, and semi-supervised techniques are all included in machine learning. Supervised learning predicts outcomes (dependent variables) by utilizing labeled data, much like normal epidemiology procedures. Regression and other well-known machine learning algorithms fall under this category. Similar to finding subgroups in statistics, unsupervised learning finds intrinsic links within data without preconceived results. Common clustering algorithms include expectation-maximization and k-means. Semi-supervised learning reduces the need for intensive labeling by leveraging both labeled and unlabeled data to improve model performance. It uses techniques similar to those for incomplete outcome data, imputes incomplete outcomes to improve forecasts.

In machine learning, regression, and classification are included in supervised learning. Whereas classification predicts categorical outcomes, regression forecasts continuous ones. Algorithms

designed for one task can be adapted to handle another; for instance, regression can be handled by classification algorithms and vice versa. This plasticity makes the machine learning models more suitable for a wide range of prediction tasks. (Bi et al., 2019)

In my study, I have used the supervised classification method. Because I have labeled data with different classes.

ENSEMBLE LEARNING:

A machine learning technique called ensemble techniques combines multiple base models to create an ideal predictive model. The main objective of machine learning and model construction is to get better accuracy. It will start to make more sense as I go into more detail about the reasons behind adopting ensemble techniques. Any machine learning task seeks to choose a single model that best fits the desired outcome. Ensemble approaches use many models and average them to create a single final model, as opposed to building one model and hoping it is the most accurate prediction we can make. It is remarkable that Decision Trees are currently the most widely used and relevant ensemble approaches in data science; they are not the only kind.[19] That's why I cannot say which one I am using is the best. To solve this problem ensemble learning is the best way. Here I can use multiple algorithms at a time and get the best output.

Ensemble techniques in machine learning combine multiple models to increase prediction accuracy above that of a single model. They leverage individual proficiencies to improve overall performance by integrating many models. There are primarily two approaches. Model Variations and Training Diversification. The same model is trained using different dataset versions (such as bagging and boosting) in order to decrease overfitting and increase resilience. By training many models on the same dataset (e.g., Super Learner, Bayesian model averaging), this strategy takes advantage of the unique benefits of each model and then combines their outputs using preset procedures. Ensemble techniques eliminate the need to preselect a single model in advance and avoid overfitting by pooling the benefits of multiple models. These methods can be applied separately or in combination, giving predictive modeling a versatile and potent boost. (Bi et al., 2019)

BAGGING:

Using many bootstrapped copies of the original data, the same algorithm is trained through a process known as "bagging," or "bootstrap aggregating." Averaging forecasts yields the final prediction for quantitative outcomes; majority voting or averaging probabilities yields the final prediction for categorical outcomes. This technique greatly lowers model variance without affecting bias. By training models on arbitrary subsets of features, feature bagging reduces correlation amongst ensemble models and helps further decrease overfitting. As a result, splits on trees become "random forests," whereby they take into account random feature subsets, preventing an over-reliance on powerful predictors and improving overall performance. Using models that omitted some observations, out-of-bag error calculation evaluates mean prediction error in order to estimate prediction errors for random forests. Rankings of variable relevance provide an overview of the relative value of predictors for each tree in the forest, assisting in understanding predictors' impacts on outcomes and interactions between them. (Bi et al., 2019)

BOOSTING:

An incremental method called "boosting" refines classifiers by emphasizing prediction errors when training models on subsets of data. The weighting of observations and classifiers is done using the widely used AdaBoost algorithm. Iteratively, observations are prioritized in succeeding iterations based on their initial equal weights that increase if they are erroneously classified. In order to highlight models with higher accuracy, the final classifier is a weighted average of the classifiers from each iteration. Gradient boosting, a generalization of AdaBoost, boosts classifier performance above simple classification errors by optimizing any differentiable loss function by gradient descent. Programs like Stata, SAS, and sklearn. ensemble in Python, and gbm, adabag, fastAdaboost, xgboost, ada, and caret in R, make it easier to use these boosting techniques. Applying and assessing boosting techniques in various programming environments is made possible by the variety of functions these tools offer.

For sign language detection I am using 3 algorithms with soft voting technique. The soft voting technique provides output based on probability calculation. 3 algorithms are the XGB classifier,

Random Forest Classifier, and KNN. Random Forest is a bagging algorithm, XGB is a Boosting algorithm and KNN is a single base model.

XGB CLASSIFIER:

Regression trees serve as the foundational learners in the ensemble technique known as XGBoost, or extreme gradient boosting. Since XGBoost is a boosting algorithm, it is resilient on unbalanced datasets and provides higher precedence to misclassified instances during the next algorithm iteration. The model makes advantage of gradient boosting, whereby successive trees improve upon the faults of their predecessors. (Jana & Krishnakumar, 2022)

One kind of gradient boosted decision tree model is the XGBoost algorithm, which was first presented by Chen and Guestrin. Using the training data, this method trains decision trees in a sequential fashion. A new decision tree is added to the ensemble of earlier trees at each iteration in order to increase the value of the objective function. The loss term (abbreviated as "l") and the regularization term (Ω), which make up the objective function that the algorithm seeks to minimize, are the two fundamental components. The following components are included in the equation that defines the objective function at the t-th iteration (L_t):

1. " y_i " stands for instance i's actual class label.
2. Based on the ensemble of decision trees that have been built thus far, " \hat{y}_i " indicates the expected class label of instance i.
3. The function of the tree at K iteration is represented by " f_k ".
4. The number of instances in the training set is indicated by "n".
5. The regularization term (Ω) aids in limiting the model's complexity and preventing overfitting.

The regularization term, which maintains a balance between accurate prediction and model complexity, and the loss resulting from discrepancies between real and predicted labels combine to form the objective function (L_t). The goal of adding decision trees iteratively is to continuously enhance this objective function in order to produce a predictive model that is reliable and accurate.

$$L^t = \sum_{i=1}^n [l(y_i, \hat{y}_i^{t-1} + f_t(x_i))] + \Omega(f_t)$$

The regularization term (Ω) in the equation denotes the model complexity penalty applied by the XGBoost technique to reduce overfitting. In order to prevent overfitting of the model to the training set, this phrase is intended to strike a balance between the model's predicted accuracy and complexity.

The parts are broken out as follows:

T: The quantity of leaves within the tree. The depth or complexity of each decision tree in the ensemble is controlled by this parameter.

γ : Hyperparameter handling the complexity term penalty. More complex models are penalized by stricter regularization enforced by higher values of .

ω : The weight allotted to every leaf. The contribution of each leaf to the total complexity penalty is calculated by this term.

$$\Omega(f_t) = \gamma T_t + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2$$

Complex trees are penalized by the regularization term in the XGBoost objective function, which takes into account both the weights (ω) and the number of leaves (T). This penalty promotes generalizability by incentivizing the algorithm to prioritize simpler trees with fewer leaves and smaller leaf weights, hence averting the model from fitting noise or anomalies in the training data. Practitioners can fine-tune the trade-off between model complexity and forecast accuracy by adjusting the hyperparameters γ and ω .(Aydin & Ozturk, 2021)

RANDOM FOREST CLASSIFIER:

A random forest is an ensemble method in where every single model is a decision tree. Using bagging or bootstrap aggregation, random forest takes many decision trees that are each trained on different features. To obtain the final prediction, which is the average of the outcomes from each tree, it minimizes overfitting by eliminating the variance of a single decision tree. One

could consider the random forest algorithm to be an enhancement of the decision tree, another supervised machine learning approach. The decision tree solves regression and classification problems by utilizing the idea of the n-ary tree. Every leaf node in the tree has a corresponding class label. The internal nodes known as decision nodes, which assist in producing predictions from a sequence of feature-based splits, display features. In order to get around the decision tree algorithm's propensity to overfit training data, Random Forest builds the forest using a collection of decision trees. In order to increase the model's accuracy and decrease its sensitivity to training data, Random Forest constructs several random decision trees and combines them. The "Bagging method," which combines bootstrapping and aggregation, is used to train a random forest model.

Assume that the training set $X = \{x_1, x_2, \dots, x_n\}$, where x_i represents a single training sample and the matching label set, $Y = \{y_1, y_2, \dots, y_n\}$, where y_i is the label of the x_i training sample. With bootstrapping, a replacement where repeated selections from the original training set X are permitted creates m new random training sets, each of size n . As a result of this phase, the model is less susceptible to training data because we are utilizing distinct datasets for each tree. Once that is done, random subsets of all the features are taken into consideration while building m decision trees utilizing each of these random datasets independently. The process known as "feature bootstrapping" lowers the correlation between trees by selecting a random selection of features. The term "random forest" refers to this collection of random decision trees. By passing the data point through each of the random decision trees, the aggregation technique calculates the forecast for a new data point. In the case of regression, the output predictions made by each individual decision tree are averaged to determine the final prediction for a fresh test sample x following training. In contrast, the majority vote is used to determine the outcome prediction in the classification instance, where the largest number of random decision trees predicted is considered the outcome.(Das et al., 2023)

K-NEAREST NEIGHBORS CLASSIFIER:

A multi-dimensional feature space is used by the supervised learning algorithm KNN to classify objects based on how close they are to other labeled objects. It starts by learning labeled training data by heart. Based on similarity, KNN finds its k nearest neighbors in the training set when it encounters a new, unlabeled object. Later, it often uses majority voting to assign new objects to

the most prevalent class among these neighbors. This approach is predicated on the assumption that items in the feature space that are close to each other are likely members of the same class. The KNN algorithm classifies each row of a sample dataset into a group specified in a training dataset in the Matlab environment. To reflect the same features, the number of columns in the training and sample datasets must be equal. The groups in the training data are specified by a grouping variable, and the assigned category for each sample row is indicated by the class variable. If the majority vote fails to provide a clear winner when classifying more than two groups with the same value of k, random tie-breaking is employed.

KNN is a top-notch classification method that excels at solving regression issues. Their classification process is predicated on the feature vectors that are taken from the training images. KNN is the most effective method for classification if the input is a multimodal system. Reducing the size of the design set or utilizing fast-KNN can both lower the computational complexity of the KNN. A few distance measures are used to form the K-neighbors. The KNN method is the best option when the picture to be classified is represented using local features. The elements vote to form a group. The KNN algorithm's performance is contingent upon the kind of distance measures it employs. The Euclidean Distance is the most commonly used distance metric. Minkowsky, Chi-Square, and Cosine are other metrics for measuring distance.(Amrutha & Prabu, 2021)

The widely used Euclidean distance approach can be computed using the feature vectors $X = (x_1, x_2 \dots x_n)$ and $Y = (y_1, y_2 \dots y_n)$, indicating that they are n-dimensional vectors.

Then the Euclidean Distance

$$Dist(E) (X, Y) = \sqrt{\frac{\sum_{i=1}^n x_i - y_i}{n}}$$

The homogeneity of the elements is measured by the Cosine distance, which may be computed using

$$Dist(c) (X, Y) = \frac{\vec{x} \cdot \vec{y}}{|\vec{x}| |\vec{y}|}$$

The following formula can be used to calculate Minkowsky, the third widely used distance measure.

$$\text{Dis}_{(m)}(X, Y) = (\sum_{i=1}^n |x_i - y_i|^m)^{\frac{1}{m}}$$

When the Manhattan Distance is represented by $m=1$, and the Euclidean Distance is shown by $m=2$. The variable 'm' is always assigned to either 1 or 2, despite the fact that it can take on any value. (Amrutha & Prabu, 2021)

I have used KNN, XGBoost, and Random Forest with ensemble learning techniques. After that use these 3 algorithms individually to make a comparison to find out whether ensemble learning is better than an individual algorithm or not. For ensemble learning, I have to find out the best parameters, for that I have used the parameter tuning technique.

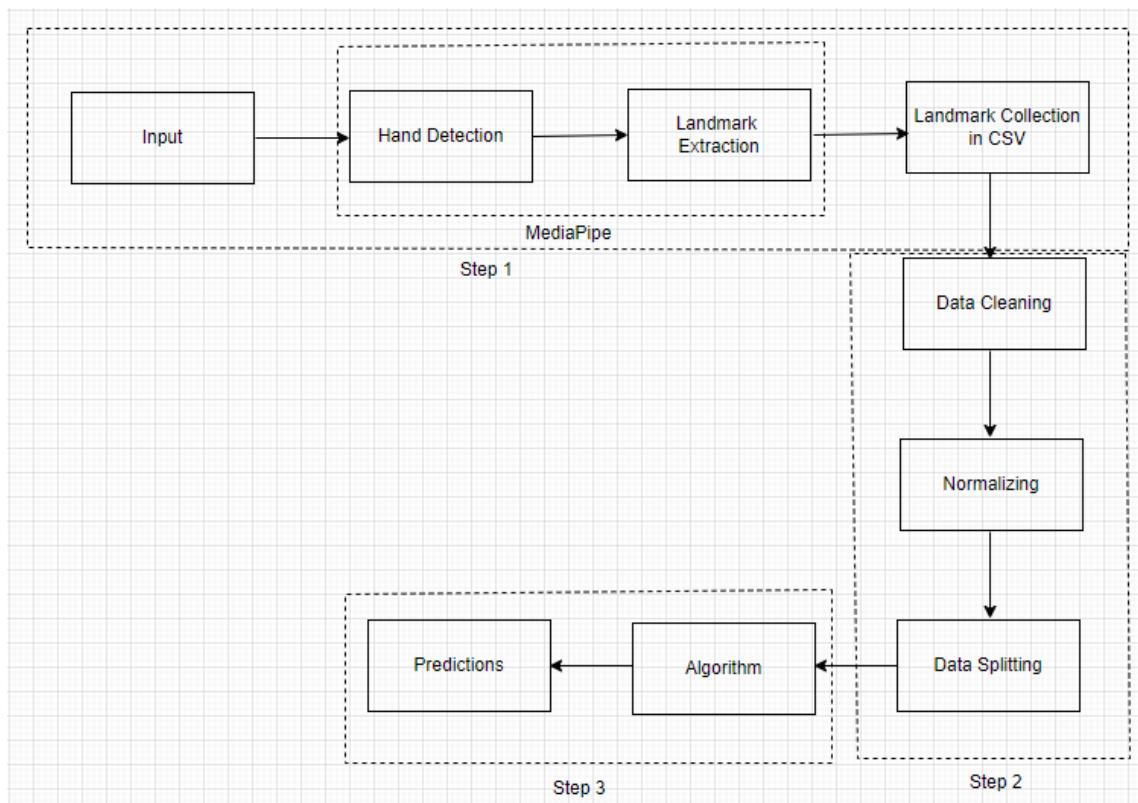


Figure 5: Data process and prediction process

This is the final diagram for my experiment. There are 3 steps.

Step 1: MediaPipe

input: An image delivered into the system serves as the process's first input.

Hand Detection: A hand detection algorithm subsequently processes the input. This mentions MediaPipe, a framework for creating multimodal applied machine learning pipelines (e.g., audio, video, any time-series data).

Landmark Extraction: The process of extracting landmarks comes after the hand has been identified. The position and orientation of the hand can be ascertained using these landmarks, which are particular areas of interest within the hand image.

Gathering and storing the extracted landmarks in a CSV (Comma-Separated Values) file format allows them to be accessed and utilized for additional processing. Result and Discussion This section interprets the result of my sign language detection study.

Step 2: Information Processing

Data Cleaning: Next, the CSV file's landmark data is cleaned. Addressing missing numbers, eliminating or fixing inaccurate data, and other tasks can all be included in data cleaning.

Normalization: The data is standardized following cleaning. In order to guarantee that the input data has a consistent scale, the process of normalization involves modifying the range of pixel intensity values. Prior to entering the data into a machine learning model, this is a crucial step.

Data Splitting: After the data has been standardized, it is usually divided into sets for the machine learning model's training and testing. This makes it possible to assess the model's effectiveness using hypothetical data.

Step 3: Prediction

Algorithm: To produce predictions, an algorithm—likely a machine learning model—uses the processed data as input. This can entail keeping track of hand movements or categorizing the hand motions.

Predictions: A collection of forecasts derived from the input data is the algorithm's output. Applications such as gesture control, sign language interpretation, and other types of human-computer interface might benefit from these predictions.

FLASK:

based on Python A microweb framework is called Flask. It is classified as a micro-framework as it doesn't require any modules or toolkits. It does not include any component that presently has comparable functionality provided by third-party libraries, such as custom post types or database intermediate layers. Flask provides helpful features and tools to develop a web service in only one Python file. It offers designers flexibility and gives novice developers a straightforward base.

I made an app.py file after the training session. Following the installation of the Flask library, I loaded my model that had been specially trained (weights file, "best.pt" or "last.pt") into that file. I then uploaded my model to the server using the index.html file as assistance.

EVALUATION METHODS:

The "Confusion Matrix" is a tool that anybody may use to evaluate any form of model. We have to have a look at a few of the elements to get that factor. Those are

True-Positive (TP): A value that has the status of "True Positive" indicates that it was accurately predicted.

True-Negative (TN): A true negative that results from the model correctly predicting the negative class.

False-Positive (FP): A "false-positive" is when a result is believed to be positive when it is not.

False-Negative (FN): By mistake, the term "False-negative" is eliminated.

Accuracy:

The accuracy of a machine's outcome prediction relies on the quality of the model. When each class is given equal weight, something is noteworthy. Every course is extremely important to my field of work. As a result, establishing the accuracy of the model is essential to its correctness.

$$Accuracy = \frac{TP + TN}{(TP + FP + TN + FN)}$$

Precision:

It serves as a metric for assessing the effectiveness of a classification model. Precision is calculated as the true positive-value divided by the overall positive-value.

$$Precision = \frac{TP}{TP + FP}$$

Recall:

Recall is the properly specified appraisal of the true positive. To calculate the recall, divide the actual positive value by the total number of related records that are presently in existence.

$$Recall = \frac{TP}{TP + FN}$$

F1 Score:

The F1 score is an index of test accuracy. The F1 value is computed using recall and precision.

$$F1\ Score = 2 * (Precision * Recall) / (Precision + Recall)$$

SUMMARY:

After data collection and preprocessing I have applied ensemble learning method with KNN, XVG, Random Forest Classification to identify Bengali and English sign language and Use FLASK to deploy the web services into my processed dataset. I have overviewed all the process and algorithm architectures. Also introduced few evaluation methods along with their formula by which I can evaluate my model.

CHAPTER 4

RESULTS AND DISCUSSION:

INTRODUCTION:

The results section provides a thorough summary of the findings and insights obtained from the study, serving as a testament to the conclusion of intensive research and analysis. This part presents and explains the results of the experiments, analysis, and models that were used to get knowledge of the information that was discovered throughout the course of the investigation.

RESULT:

Ensemble learning model with KNN, XGBoost, and Random Forest achieved exceptional accuracy of 98% in my dataset. Whether for individual models the accuracy is lower than 98%. Random Forest achieved 97%, KNN 97%, and XGBoost 96%. This statistically significant increase shows how well ensemble learning works to maximize each model's strengths while minimizing its faults.

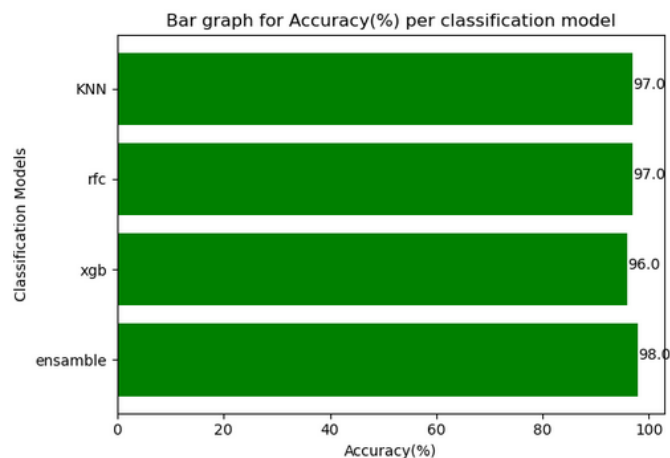


Figure 6: Accuracy (%) per classification model

Comparison of individual models

Individual models also perform well but not equal to ensemble learning accuracy. Random Forest (97%) and KNN (97%) slightly outperform XGBoost (96%). However, the ensemble model just makes a difference with an additional 1% improvement and shows the potential of combining diverse learning styles.

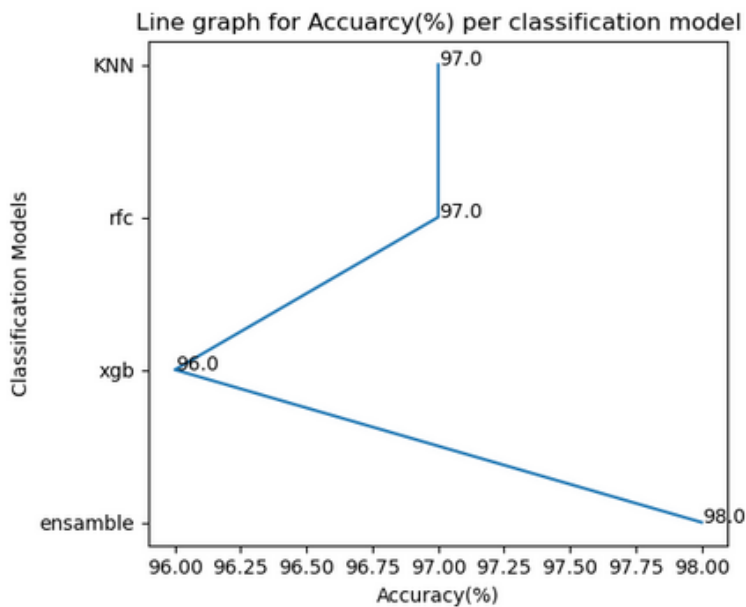


Figure 7: Accuracy (%) per classification model

After successfully running the ensemble learning method with XGB, KNN and Random Forest with best parameters the weighted average precision recall and F1 score all are same 0.98. This shows that 98% of the positive predictions produced by XGB, KNN, and Random Forest as a whole were correct. represents the ensemble model's capacity, taking into account the combined outcomes from XGB, KNN, and Random Forest, to identify 98% of all relevant cases. The harmonic mean of recall and precision, which displays a balanced performance of 98%, indicates that the ensemble approach successfully captured the pertinent occurrences and attained a high degree of accuracy. When the ensemble of models (XGB, KNN, and Random Forest) consistently achieve weighted average precision, recall, and an F1 score of 0.98, it is essentially

a sign of strong and dependable performance in the task at hand, demonstrating a high degree of accuracy and effectiveness in identifying relevant instances.

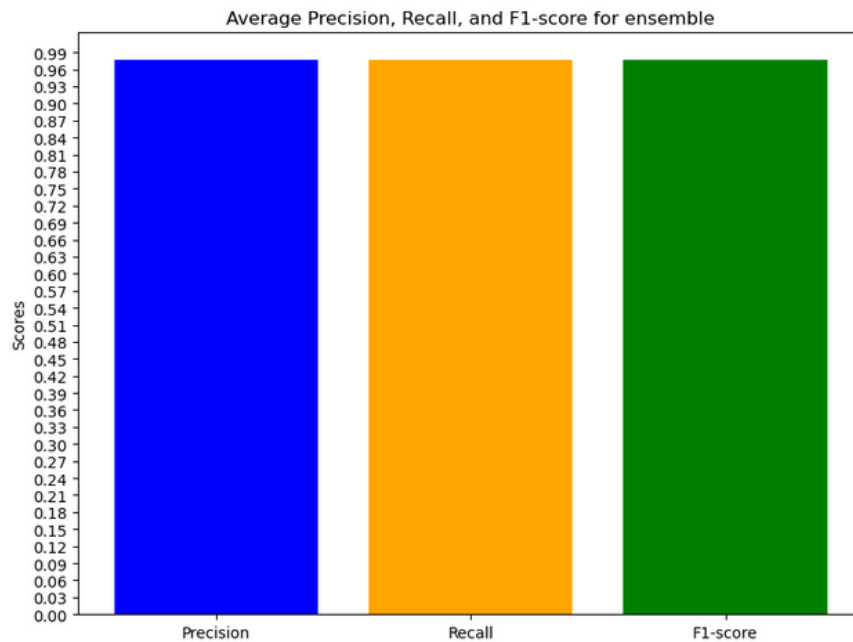


Figure 8: precision, recall and F1 score for ensemble

In case of individual model runs XGB gained 0.96 precision, recall, and F1 score. All three are the same. This means that 96% of the situations that the XGBoost model correctly predicted as positive were really so. It shows that 96% of all relevant cases can be accurately identified using the XGBoost model. A well-balanced performance between accuracy and recall is shown by the harmonic mean of precision and recall, which is also 96%. Achieving a precision, recall, and F1 score of 0.96 consistently throughout the XGBoost model tests points to a steady and dependable model performance, demonstrating a high degree of prediction accuracy while successfully catching pertinent occurrences.

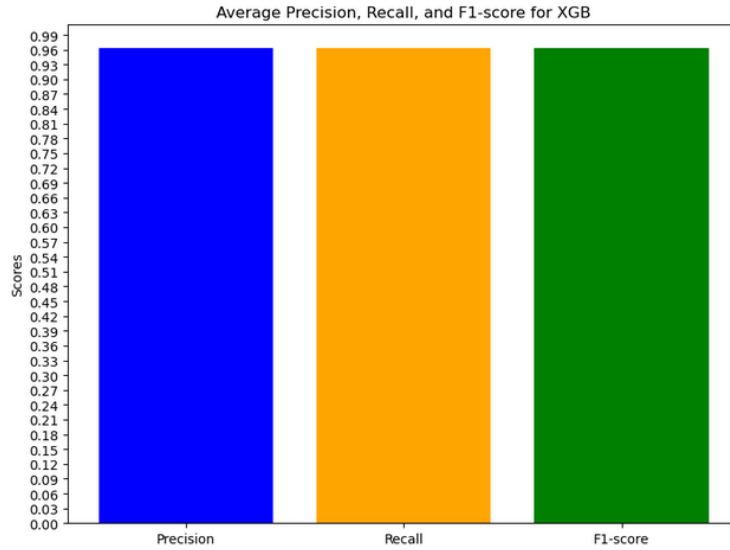


Figure 9: precision, recall and F1 score for XGBoost

Random Forest achieved 0.97 precision, recall, and F1 score. A precision score of 0.97 indicates that 97% of the model's positive predictions were really accurate. A recall score of 0.97 indicates that 97% of the pertinent episodes could be captured by the model. A score of 0.97 indicates a high performance overall and a well-balanced trade-off between recall and accuracy. In conclusion, the Random Forest model's 0.97 precision, recall, and F1 score shows a high degree of accuracy, efficacy in finding pertinent instances, and balanced performance in the job at hand.

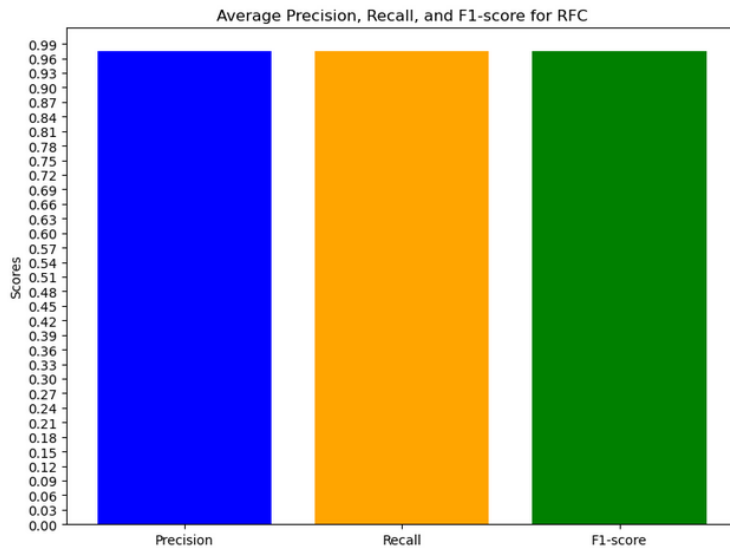


Figure 10: precision, recall and F1 score for Random Forest

My final algorithm KNN achieved 0.97 for all the matrices precision, recall, and F1 score.

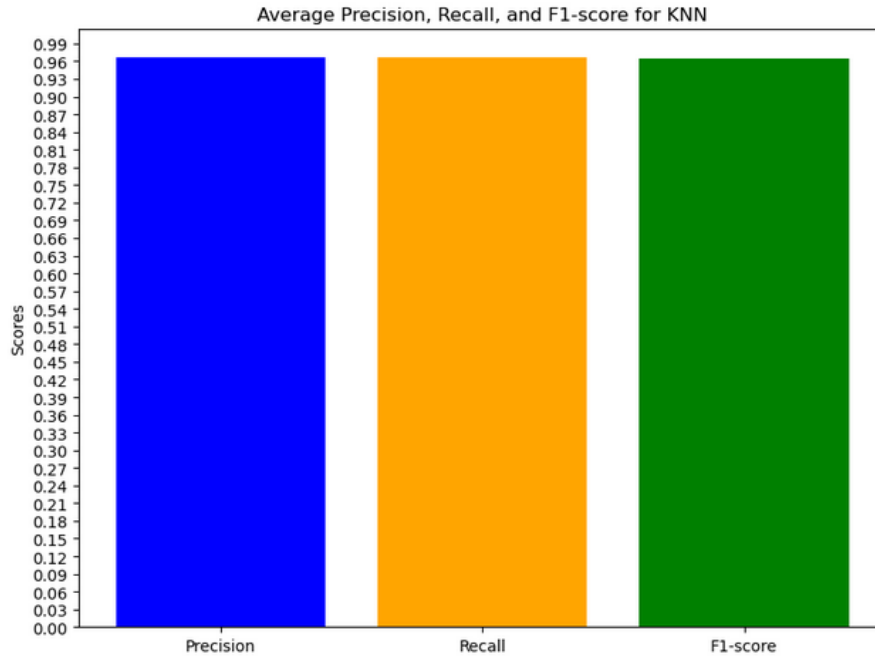


Figure 11: precision, recall and F1 score for KNN

Errors are also in control. Among all the model's ensemble learning shows minimum error scores across all three metrics mean absolute error, mean squared error and root mean squared error. Ensemble has a very potential to accurately fit with this dataset and make accurate predictions. It just combines the strengths of all three algorithms and mitigates the weaknesses of individual models. That ensures the best performance.

```
Mean Absolute Error: 0.39336861192317657
Mean Squared Error: 14.686378270362283
Root Mean Squared Error: 3.8322810792480086
```

Random Forest also performs well with very similar scores to the ensemble model. Its result shows that Random Forest has enough strength to predict accurately in this study.

```
Mean Absolute Error: 0.39655108611183065
Mean Squared Error: 14.441179735780631
Root Mean Squared Error: 3.8001552252218107
```

KNN shows a slightly higher error score as compared to the ensemble and Random Forest. Though it shows higher accuracy in prediction equal to Random Forest but error is also higher.

```
Mean Absolute Error: 0.5477556155867224
Mean Squared Error: 18.92617399992599
Root Mean Squared Error: 4.350422278345631
```

Finally, XGBoost. XGBoost shows the highest error scores among all the models. More careful hyperparameter tuning or investigation of potential overfitting issues might improve its performance.

```
Mean Absolute Error: 0.6198793620249418
Mean Squared Error: 23.317433297561337
Root Mean Squared Error: 4.828812824862995
```

After analyzing all error metrics and visualizing all three-

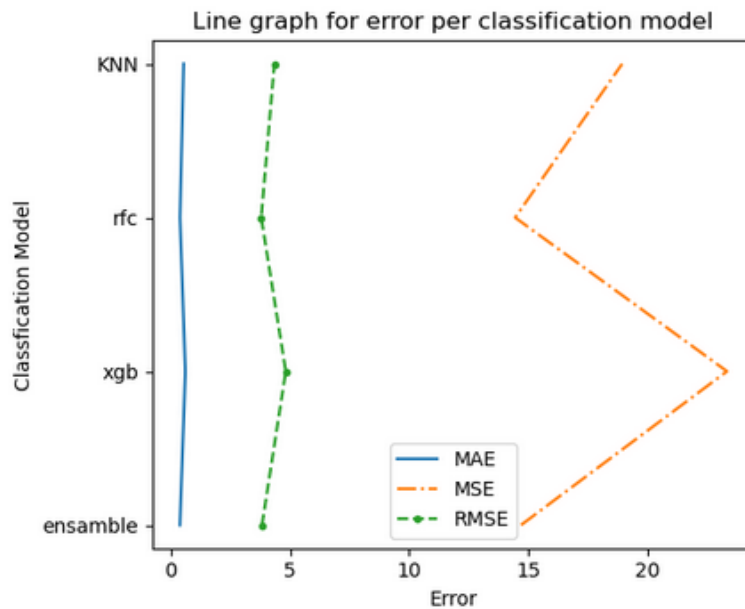


Figure 12: Error per model

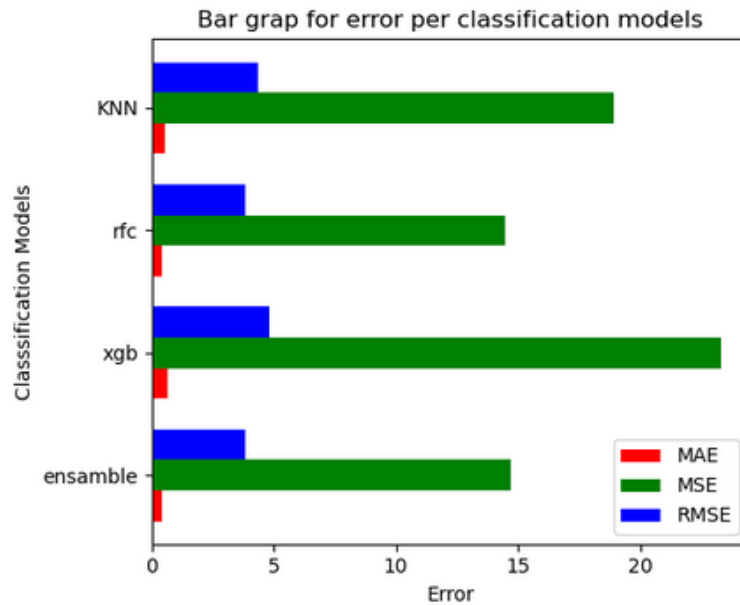


Figure 13: Error per model

DISCUSSION:

Though all the models perform well ensemble learning is just a few ahead of all the models. When XGBoost, KNN, and Random Forest classifiers are integrated into ensemble learning, the predicted accuracy outperforms that of individual models. This result is consistent with ensemble techniques' intrinsic ability to combine many models to provide forecasts that are more reliable and accurate. By maximizing each model's unique strengths, the ensemble reduced overfitting and increased the overall stability of forecasts. Combining models like a team boosted knowledge and performance, proving the power of collective intelligence.. This demonstrates that ensemble learning is effective in obtaining higher accuracy and resilience when compared to single models, demonstrating its potential to improve prediction in machine learning applications.

SUMMARY:

This segment just shows the final results of all the models and errors. Also I have shown the visualizations of the outcome and model performances and comparison.

CHAPTER 5

CONCLUSION:

The main purpose of this study is to prove the effectiveness of the ensemble learning method for sign language detection, focusing on Bengali and English signs. The ensemble learning combining KNN, XGBoost, and Random Forest achieved an outstanding 98% accuracy. A large dataset has been used in this study. After collecting all the images, I just made a custom dataset with proper cleaning. Data preprocessing has a huge effect on better performance. This study focused on English and Bengali letters, though a few Bengali words were also added. In the future, it can be more realistic, if it is possible to add more words. Accuracy might be increased more by more efficient parameter tuning. Through the utilization of the complementing features of these various algorithms inside the ensemble framework, the study illustrated the effectiveness and efficiency of merging several models. This method not only outperformed individual models, but it also demonstrated the possibility of greatly improving accuracy and dependability in sign language detection—a crucial area where accuracy is essential for clear interpretation and communication.

LIMITATIONS:

Though there have been notable advancements in both Bangla and English sign language recognition, there are still certain restrictions that need be taken into account. First and foremost, depending too much on prefabricated datasets limits the range of original data collection, which may have an effect on the diversity and inclusivity of sign language representations. With a primary focus on letter identification, the model's word detection capacity in Bangla is proven, while English word detection is still unexplored, indicating a possible avenue for further research. Additionally, recognizing other languages beyond Bangla and English remains a challenge, highlighting the model's language limitations. A larger dataset would improve the

research's model's adaptability and resilience. Hardware limitations caused training and parameter tuning times to increase, indicating the influence of computational resources on research productivity. Because of these drawbacks, larger datasets, multilingual recognition, and enhanced computing infrastructure are all necessary to increase the model's precision and usefulness.

FUTURE SCOPE:

Future advancements in Bangla and English sign language identification have a great deal of promise, even with the current constraints. Adding more sign language movements to the dataset—words and phrases in addition to letters—would significantly improve the model's performance. The model's inclusiveness and accuracy in identifying different sign languages may be improved by including data from primary sources and linguistic backgrounds.

Additionally, investigating and incorporating cutting-edge methods like domain adaptation or transfer learning may improve the model's ability to recognize languages other than English and Bangla. This extension would make it easier to use sign language recognition in a wider range of verbal contexts.

Training and parameter tuning durations might be greatly shortened by investing in computer resources or using cloud-based solutions, allowing for quicker iterations and model improvements. It is also possible to promote more inclusion and communication for those with hearing impairments by emphasizing user-friendly interfaces or mobile applications, which might democratize access to sign language translation technology.

Additionally, working with linguists, sign language specialists, and community members would guarantee the creation of accurate and culturally sensitive models that meet the unique requirements of a variety of sign language users.

In order to develop more inclusive and reliable sign language detection systems with a wider range of applications, the future scope primarily entails utilizing cutting edge technology, broadening the diversity of datasets, and encouraging multidisciplinary partnerships.

REFERENCES

- Tayade, A., & Halder, A. (2021). Real-time Vernacular Sign Language Recognition using MediaPipe and Machine Learning. *International Journal of Research Publication and Reviews*, 2(5). <https://doi.org/10.13140/RG.2.2.32364.03203>
- Adaloglou, N., Chatzis, T., Papastratis, I., Stergioulas, A., Papadopoulos, G. Th., Zacharopoulou, V., Xydopoulos, G. J., Atzakas, K., Papazachariou, D., & Daras, P. (2020). A Comprehensive Study on Deep Learning-based Methods for Sign Language Recognition. <https://doi.org/10.1109/TMM.2021.3070438>
- Amrutha, K., & Prabu, P. (2021, February 11). ML based sign language recognition system. 2021 International Conference on Innovative Trends in Information Technology, ICITIIT 2021. <https://doi.org/10.1109/ICITIIT51526.2021.9399594>
- Aydin, Z. E., & Ozturk, Z. K. (2021). Performance Analysis of XGBoost Classifier with Missing Data. <https://www.researchgate.net/publication/350135431>
- Bi, Q., Goodman, K. E., Kaminsky, J., & Lessler, J. (2019). What is machine learning? A primer for the epidemiologist. *American Journal of Epidemiology*, 188(12), 2222–2239. <https://doi.org/10.1093/aje/kwz189>
- B Garcia & SA Viesca. (2016). Real-time American Sign Language Recognition with Convolutional Neural Networks
- Das, S., Imtiaz, M. S., Neom, N. H., Siddique, N., & Wang, H. (2023). A hybrid approach for Bangla sign language recognition using deep transfer learning model with random forest classifier. *Expert Systems with Applications*, 213. <https://doi.org/10.1016/j.eswa.2022.118914>
- Duy Khuat, B., Thai Phung, D., Thi Thu Pham, H., Ngoc Bui, A., & Tung Ngo, S. (2021). Vietnamese sign language detection using Mediapipe. *ACM International Conference Proceeding Series*, 162–165. <https://doi.org/10.1145/3457784.3457810>
- Goyal, K., & Velmathi, G. (n.d.). INDIAN SIGN LANGUAGE RECOGNITION USING MEDIAPIPE HOLISTIC.
- Hoque, O. B., Jubair, M. I., Islam, Md. S., Akash, A.-F., & Paulson, A. S. (2018). Real Time Bangladeshi Sign Language Detection using Faster R-CNN. <https://doi.org/10.1109/CIET.2018.8660780>
- Horvitz, E., & Mulligan, D. (2015). Data, privacy, and the greater good. *Science*, 349(6245), 253–255. <https://doi.org/10.1126/science.aac4520>

- Islam, M. S., Joha, A. J. M. A., Hossain, M. N., Abdullah, S., Elwarfalli, I., & Hasan, M. M. (2023). Word level Bangla Sign Language Dataset for Continuous BSL Recognition. <http://arxiv.org/abs/2302.11559>
- Jāmi‘at al-Baḥrayn, & Institute of Electrical and Electronics Engineers. (n.d.). 2019 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT).
- Jana, A., & Krishnakumar, S. S. (2022). Sign Language Gesture Recognition with Convolutional-Type Features on Ensemble Classifiers and Hybrid Artificial Neural Network. *Applied Sciences (Switzerland)*, 12(14). <https://doi.org/10.3390/app12147303>
- Katoch, S., Singh, V., & Tiwary, U. S. (2022). Indian Sign Language recognition system using SURF with SVM and CNN. *Array*, 14. <https://doi.org/10.1016/j.array.2022.100141>
- Kothadiya, D., Bhatt, C., Sapariya, K., Patel, K., Gil-González, A. B., & Corchado, J. M. (2022). Deepsign: Sign Language Detection and Recognition Using Deep Learning. *Electronics (Switzerland)*, 11(11). <https://doi.org/10.3390/electronics11111780>
- Lugaresi, C., Tang, J., Nash, H., Mcclanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C.-L., Yong, M. G., Lee, J., Chang, W.-T., Hua, W., Georg, M., Grundmann, M., & Research, G. (n.d.). MediaPipe: A Framework for Perceiving and Processing Reality. <https://github.com/google/mediapipe>.
- Mocialov, B., Turner, G., Lohan, K., & Hastie, H. (n.d.). Towards Continuous Sign Language Recognition with Deep Learning. <https://github.com/CMU-Perceptual-Computing-Lab/openpose/>
- MediaPipe Github: <https://google.github.io/mediapipe/solutions/hands>. Access 2021.
- Medium <https://towardsdatascience.com/ensemble-methods-in-machine-learning-what-are-they-and-why-use-them-68ec3f9fef5f>
- Podder, K. K., Chowdhury, M. E. H., Tahir, A. M., Mahbub, Z. Bin, Khandakar, A., Hossain, M. S., & Kadir, M. A. (2022). Bangla Sign Language (BdSL) Alphabets and Numerals Classification Using a Deep Learning Model. *Sensors*, 22(2). <https://doi.org/10.3390/s22020574>
- Rahman, M. M., Islam, M. S., Rahman, M. H., Sassi, R., Rivolta, M. W., & Aktaruzzaman, M. (2019, December 1). A new benchmark on american sign language recognition using convolutional neural network. 2019 International Conference on Sustainable Technologies for Industry 4.0, STI 2019. <https://doi.org/10.1109/STI47673.2019.9067974>
- Real time sign language detection system using deep learning techniques. (2022). *Journal of Pharmaceutical Negative Results*, 13(S01). <https://doi.org/10.47750/pnr.2022.13.s01.126>

- Shamrat, F. M. J. M., Chakraborty, S., Billah, M. M., Kabir, M., Shadin, N. S., & Sanjana, S. (2021). Bangla numerical sign language recognition using convolutional neural networks. *Indonesian Journal of Electrical Engineering and Computer Science*, 23(1), 405–413. <https://doi.org/10.11591/ijeecs.v23.i1.pp405-413>
- Sharma, M., Pal, R., & Kumar Sahoo, A. (2014). INDIAN SIGN LANGUAGE RECOGNITION USING NEURAL NETWORKS AND KNN CLASSIFIERS. 9(8). www.arpnjournals.com
- Siddique, S., Islam, S., Neon, E. E., Sabbir, T., Naheen, I. T., & Khan, R. (2023). Deep Learning-based Bangla Sign Language Detection with an Edge Device. *Intelligent Systems with Applications*, 18. <https://doi.org/10.1016/j.iswa.2023.200224>
- Suryaperdana Agoes, A. (2021). Applying Hand Gesture Recognition for User Guide Application Using MediaPipe.
- Talukder, D., & Jahara, F. (2020, December 19). Real-Time Bangla Sign Language Detection with Sentence and Speech Generation. *ICCIT 2020 - 23rd International Conference on Computer and Information Technology, Proceedings*. <https://doi.org/10.1109/ICCIT51783.2020.9392693>
- Tasmere, D., & Ahmed, B. (2020, December 19). Hand Gesture Recognition for Bangla Sign Language Using Deep Convolution Neural Network. *2020 2nd International Conference on Sustainable Technologies for Industry 4.0, STI 2020*. <https://doi.org/10.1109/STI50764.2020.9350484>
- Tyagi, S., Upadhyay, P., Hoor Fatima, I., & Kumar Sharma, A. (2023). American Sign Language Detection using YOLOv5 and YOLOv8. <https://doi.org/10.21203/rs.3.rs-3126918/v1>
- Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C.-L., & Grundmann, M. (2020a). MediaPipe Hands: On-device Real-time Hand Tracking. <http://arxiv.org/abs/2006.10214>
- Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C.-L., & Grundmann, M. (2020b). MediaPipe Hands: On-device Real-time Hand Tracking. <http://arxiv.org/abs/2006.10214>