Final Year Project - Semester VII Report

Identification of Sign Language

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF THE DEGREE OF

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY

BY

Odrin Rodrigues
Anushka Shinde
Kaustubh Desle
Shivam Yadav

UNDER THE GUIDANCE OF

PROF. MEENA UGALE
(Department of Information Technology)

INFORMATION TECHNOLOGY DEPARTMENT

XAVIER INSTITUTE OF ENGINEERING

UNIVERSITY OF MUMBAI

2022 - 2023

XAVIER INSTITUTE OF ENGINEERING
MAHIM CAUSEWAY, MAHIM,MUMBAI - 400016.

CERTIFICATE

This to certify that,

Odrin Rodrigues	XIEIT201903039
Anushka Shinde	XIEIT2020032006
Kaustubh Desle	XIEIT20180307
Shivam Yadav	XIEIT201903054

Have satisfactorily carried out the MINI-PROJECT work titled "***IDENTIFICATION OF SIGN LANGUAGE*** " in partial fulfillment of the degree of Bachelor of Engineering as laid down by the University of Mumbai during the academic year 2022-2023.

Prof. Meena Ugale
Supervisor / Guide

Prof. Meena Ugale
Head of Department

Dr. Y. D. Venkatesh
Principal

PROJECT REPORT APPROVAL FOR B.E.

This project entitled "Identification of Sign Language "
By

Name	Roll No.
Odrin Rodrigues	(XIEIT201903039)
Anushka Shinde	(XIEIT2020032006)
Kaustubh Desle	(XIEIT20180307)
Shivam Yadav	(XIEIT201903054)

is approved for the degree of BACHELOR OF ENGINEERING.

Examiners

1.

2.

Date:

Place: MAHIM, MUMBAI

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources.

We also declare that I have adhered to all the principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission.

We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which thus have not been properly cited or from whom proper permission have not been taken when needed.

Name	**Signature**
Odrin Rodrigues (XIEIT201903039)	
Anushka Shinde (XIEIT2020032006)	
Kaustubh Desle (XIEIT20180307)	
Shivam Yadav (XIEIT201903054)	

Date:

Contents

List of Figures

List of Tables

Acknowledgement

We would like to thank Fr. Dr. John Rose S.J (Director of XIE) for providing us with such an environment so as to achieve goals of our project and supporting us constantly.

We express our sincere gratitude to our Honorable Principal Dr. Y.D.Venkatesh for encouragement and facilities provided to us.

We would like to place on record our deep sense of gratitude to Prof. Meena Ugale ,Head of Deptartment Of Information Technology, Xavier Institute of Engineering, Mahim, Mumbai, for his generous guidance help and useful suggestions.

With deep sense of gratitude we acknowledge the guidance of our project guide Prof. Suvarna Aranjo. The time-to-time assistance and encouragement by her has played an important role in the development of our project.

We would also like to thank our entire Information Technology staff who have willingly cooperated with us in resolving our queries and providing us all the required facilities on time.

Odrin Rodrigues (XIEIT201903039)	____________________
Anushka Shinde (XIEIT2020032006)	____________________
Kaustubh Desle (XIEIT20180307)	____________________
Shivam Yadav (XIEIT201903054)	____________________

Chapter 1

Introduction

1.1 Problem Definition

Sign Languages are the native language and primary mode of communication. So normal people have difficulty in understanding their hand signs. Hearing or speech-disabled people who know Sign Language require a translator who also knows Sign Language to effectively communicate their thoughts to others. In order to communicate information to regular people, systems that can identify various signs are required. To overcome these issues, this system translates sign language.

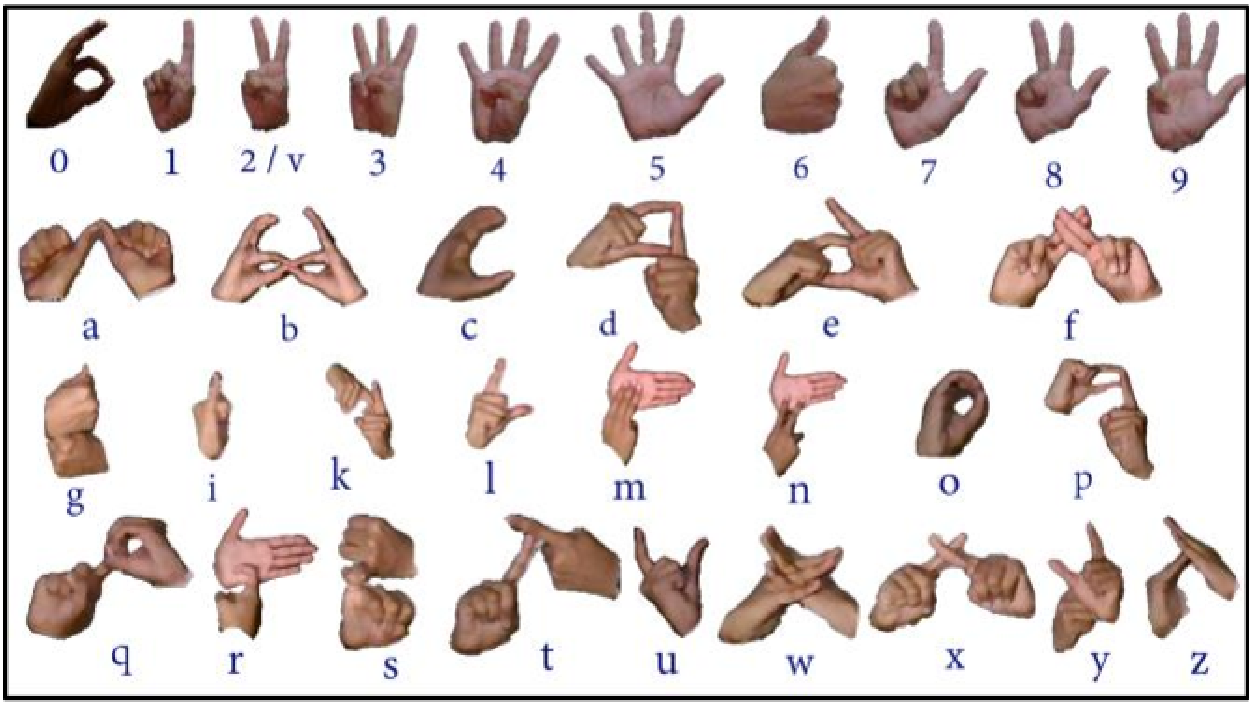


Figure 1.1: Indian Sign Language Alphabets [11].

In the Fig.1 hand sign shows the Indian Sign Language (ISL) , which includes signs for the numbers 0 to 9 and the alphabet. ISL is a two-handed sign language, in contrast to other sign languages like American Sign Language (ASL) and British Sign Language (BSL), which use one hand only. It makes it challenging to grasp and learn.

1.2 Project Scope

1. To build an effective sign language recognition (ISL) system.
2. The main aim is to detect hand sign movement using Convolutional Neural Network (CNN). A CNN is a network architecture for deep learning which learns directly from data, eliminating the need for manual feature extraction..
3. To create a system that generates words based on hand gestures.
4. Develop a Web application providing an interface for sign language communication.

Chapter 2

Review of Literature

2.1 Literature Survey

Table 2.1: Summary of CNN Based Approaches Applied for Sign Language Recognition.

Year	Authors	Application	Algorithm Architecture	Performance Results
2018	G.Anantha Rao K.Syamala, P.V.V.Kishore, A.S.C.S.Sastry [1].	Deep Convolutional Neural Networks for Sign Language Recognition.	CNN	Deep Convolutional Neural Networks for Sign Language Recognition. CNN Comparing the proposed CNN model to existing state-of-the-art classifiers, its average recognition rate was higher at 92.88
Apr 2018	Jie Huang, Wengang Zhou, Qilin Zhang, Houqiang Li, Weiping Li [2].	Video-Based Sign Language Recognition without Temporal Segmentation.	Two-stream 3D CNN	Proposed method LS-HAN achieves 82.7 % accuracy which is more than LSTM-E 76.8%.
June 2018	M.A Hossen, Arun Govindaiah, Sadia Sultana, Alauddin Bhuiyan [3].	Bengali Sign Language Recognition Using Deep Convolutional Neural Network	Pre-trained VGG16, CNN	Proposed recognition system results - validation loss of 0.3523 (categorical cross-entropy) and validation accuracy of 84.68%.
2018	Kshitij Bantupalli, Ying Xie [4]	American Sign Language Recognition using Deep Learning and Computer Vision.	CNN, RNN, Machine learning, HMM.	Instead of using Pool Layer, the system was able to attain 93% accuracy with SoftMax Layer.
Apr 2019	Sruthi C. J and Lijiya A [5].	Signet: A Deep Learning based Indian Sign Language Recognition System.	CNN	The proposed method gave training accuracy of 99.93% and validation accuracy of 98.64%.

Year	Authors	Application	Algorithm Architecture	Performance Results
Dec 2019	Lean Karlo S. Tolentino, Ronnie O. SerfaJuan, August C.Thio-ac, Maria Abigail B.Pamahoy, Joni Rose R. Fortezaz and Xavier Jet O. Garcia [6].	Sign language identification using Deep Learning.	CNN	The system obtained an average of 93.667%. The testing accuracy of 90.04% in letter recognition, 93.44% in number recognition, and 97.52% in static word identification.
Jan 2020	Ankita Wadhawan, Parteek Kumar [7].	Deep learning-based sign language recognition system for static signs.	CNN	A performance accuracy rate of 99.90%was calculated by the authors.
2020	Necati Cihan Camg,Oscar Koller, SimonHadfifield and Richard Bowden [8].	Sign Language Transformers: Joint End-to-end Sign Language Recognition and Translation.	RNN-based attention architectures, Connectionist Temporal Classification (CTC).	Compared to earlier approaches, the authors Language Transformers outperform both their recognition and their translation effectiveness with a 2% reduction in word error rate.
13 Apr 2022	Razieh Rastgoo, Kourosh Kiani, Sergio Escalera [9].	Word separation in Continuous sign language using isolated signs and post-processing	CNN, LSTM	The proposed model obtains an average of recognized Softmax outputs equal to 0.98 and 0.59.
30 Apr 2022	Abdul Mannan, Ahmed Abbasi, Abdul Rehman Javed, Anam Ahsan, Thippa Reddy Gadekallu, Qin Xin [10].	Hypertuned Deep Convolutional Neural Network for Sign Language Recognition	CNN	The proposed Deep CNN model can recognize the ASL alphabets with an accuracy rate of 99.67% on unseen test data.

a. Deep Convolutional Neural Networks for Sign Language Recognition [1]

According to paper [1], the model is constructed with input layer, four convolutional layers, five rectified linear units (ReLu), two stochastic pooling layers, one dense and one SoftMax output layer. The CNN architecture uses four convolutional layers with different window sizes followed by an activation function, and a rectified linear unit for non-linearities. The network is trained to learn the features of each sign by means of a supervised learning. The recognition accuracy is further improved by replacing ANN with deep ANN and reported an increase in recognition rate by 5%. Hence, CNN's are a suitable tool for simulating sign language recognition on mobile platforms [1].

b. Video-Based Sign Language Recognition without Temporal Segmentation [2]

In the proposed paper [2], Hierarchical Attention Network with Latent Space (LS-HAN) is used to translate signing videos sentence-by-sentence. Each video is divided into clips with a sliding window algorithm. The alignment information needs to be reconstructed. Empirical experiments verify that the strategy out performs others, therefore it is employed in the testing phase. After encoding, the start symbol "Start" is fed to Hierarchical Attention Network (HAN) indicating the beginning of sentence prediction. During each decoding timestamp, the word with the highest probability after the soft max is chosen as the predicted word, with its representation in the latent space fed to (HAN) for the next timestamp, until the emission of the end symbol "End" [2].

c. Bengali Sign Language Recognition Using Deep Convolutional Neural Network [3]

The author [3] approach was designed to function with one-handed gestures. Convolution, Max-Pooling, ReLU, Dropout, Fully Connected, and SoftMax layers form up the deep learning network. The neural network employs a stack of layers to perform classification using the features that are collected from the convolution layers. The pre-trained network is run once on the dataset to extract the features required for classification. The validation accuracy of the authors' proposed model was 84.68%, with a validation loss of 0.3523 [3].

d. American Sign Language Recognition using Deep Learning and Computer Vision [4]

Kshitij Bantupalli, Ying Xie [4] mentioned in the paper about CNN model extracted temporal features from the frames which was used further to predict gestures based on sequence of frames. Used two different approaches to classification: a. Using the predictions from the Softmax layer and b. Using the output of the global pool layer. The global pool layer gives a 2048 sized vector, which possibly allows for more features to be analyzed by the RNN. The feature sequence is then fed to a Long-Short Term Memory (LSTM) allowing longer time dependencies. RNN's suffer from the vanishing/ exploding gradient problem, LSTM's deal with the problem allowing for higher accuracies on longer sequences of data. The model reported an incredibly high 99% accuracy on the training set. The gesture segments identified and processed by the CNN are classified by the LSTM into one of the gesture classes using sequence data. The system was able to achieve 93% accuracy with Softmax Layer rather than Pool Layer [4].

e. Signet: A Deep Learning based Indian Sign Language Recognition System [5]

The paper [5] presented a vision-based deep learning architecture for signer independent Indian sign language recognition system. The system was successfully trained on all 24 ISL static alphabets with a training accuracy of 99.93% and with testing and validation accuracy of 98.64%. The recognition accuracy obtained is better than most of the current state of art methods [5].

f. Sign language identification using Deep Learning [6]

The author of the paper [6] proposed a model where the network uses a Stochastic gradient descent optimizer as its optimizer to train the network having a learning rate of 1×102. The total number of epochs used to train the network is 50epochs with a batch size of 500. The images were resized to (50, 50, 1) for training and testing. To attain specific results, Keras and CNN architecture containing a set of different layers for processing of training of data used. The number of filters in the CNN layers is increased to 64. The fully connected layer is being specified by the dense layer along with rectified linear activation [6].

g. Deep learning-based sign language recognition system for static signs [7]

Ankita Wadhawan, Parteek Kumar [7] proposed sign language recognition system includes four major phases that are data acquisition, image preprocessing, training and testing of the CNN classifier. The model training is based on convolutional neural networks. The classifier takes the preprocessed sign images and classifies it into the corresponding category. The classifier is trained on the dataset of different ISL signs. The system achieved training and validation accuracy of 99.76% and 98.35%, respectively, using RMSProp and it has been found that the SGDoptimizer outperformed Adam, RMSProp and other optimizers with training and validation accuracy of 99.90% and 98.70%, respectively, on grayscale image dataset [7].

h.Sign Language Transformers: Joint End-to-end Sign Language Recognition and Translation [8]

The paper [8] used a modified version of JoeyNMT to implement the Sign Language Transformers. All the components of network were built using the PyTorch. Transformers are built using 512 hidden units and 8 heads in each layer. Used the Adam optimizer to train the networks using a batch size of 32. Network is evaluated at every100 iterations [8].

i.Word separation in continuous sign language using isolated signs and post-processing [9]

Boundary identification of the sign presented the biggest hurdle in CSLR. Words in a video of continuous signs. The paper indicates Use of transfer learning to solve this issue. The authors developed a two-step solution predictor model and a post- processing methodology. The hand-crafted SVD utilized to feed features to LSTM Network extracted from the features using the prediction model. Application of the post-processing algorithm on the Predictor Model's outputs from the Softmax. Both the SVD feature extractor and the hand pose estimator received a window size of 50 frames. Then, the many-to-one LSTM Network maps the matching SVD feature sequence. Into a single vector up to 50 frames. After that, a Fully Connected (FC) layer receives this vector. Finally, the FC outputs are covered with a Softmax layer. The suggested methodology employed a specified threshold, 0.51, to approve or reject a recognized class for the separation of the isolated signs in a continuous sign video. There are various difficulties with the comparable signs. In the sliding window that is open. There were some difficulties in the placards that said "Congratulation," "Excuse," "Upset," "Blame," "Fight," and "Competition.". In order to learn more powerful features to better describe sign categories and decrease miss-classifications as a result, adding more samples could to title interclass variation [9].

j.Hypertuned Deep Convolutional Neural Network for Sign Language Recognition [10]

Paper suggested an architecture based on CNN, containing dense, max-pooling, dropout, and many convolutional layers (fully connected). Which performs convolution on input with various filter and kernel sizes to map feature. Model correctly learns features from three main blocks, each of which has a different parameter configuration and uses ReLU as an activation function. By employing the flatten layer, which turns data into a vector before connecting to a group of the fully connected layer, these features were made flat. Authors evaluated the model on new data in a later stage and reported accuracy of 99.67% [10].

Chapter 3

Proposed System

3.1 Features

The proposed system has 2 modules- Hand signs to Text/Audio and Text/Audio to Hand signs

a Hand signs to Text/Audio

- Takes input images from the camera source.
- Processes image signs.
- Predicts the class of the given input image signs.
- Generate the text of image signs.

b Text/Audio to Hand signs

- Audio inputs are taken
- Converted to text format
- Converted text is processed with model
- System gives the output as images

3.2 Block Diagram / Architecture

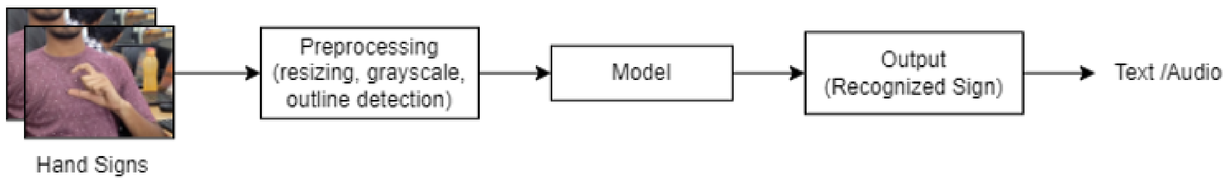


Figure 3.1: Hand Sign to Text/Audio Block diagram

Raw hand sign images captured by cameras are preprocessed to match the specific requirement for model processing. The CNN model is provided with features that are extracted from the output of preprocessing phase. These features are processed by a model, which ultimately transforms and shows the recognized sign language and converts into the text/audio.

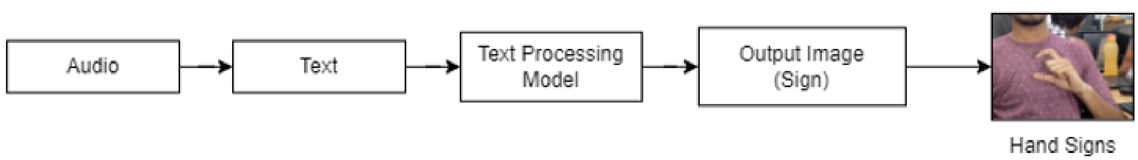


Figure 3.2: Audio to Hand Sign Block diagram

The audio input is used as raw data, which gets converted to text and subjected to model-based analysis. The model encodes the words into vectors provided text as input, creating the relation between words. These vectors allow for the representation of associated hand sign images.

Chapter 4

Proposed Methodology

4.1 Design

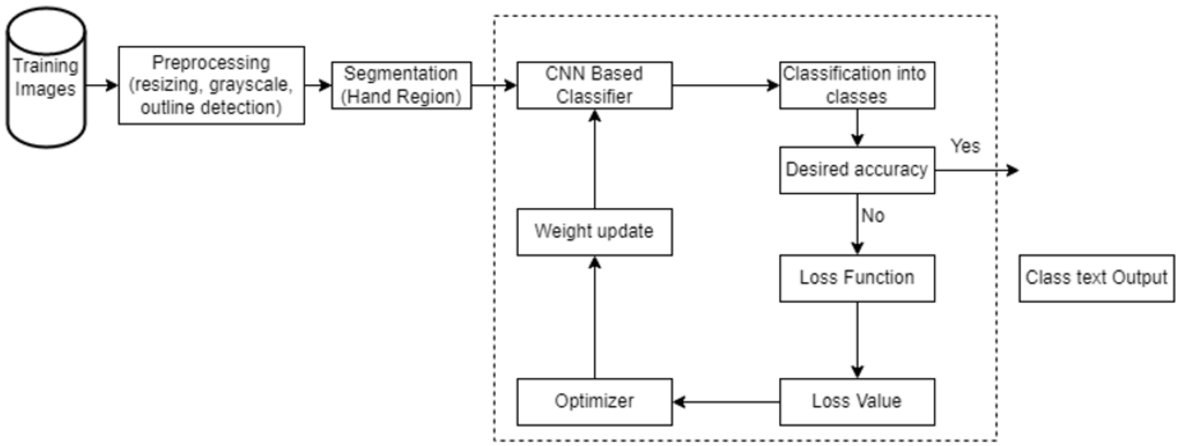


Figure 4.1: Model Training

1. Training Images: The block consists of system's core resource, and it includes the raw dataset of various images of (ISL) which is used for training and testing the model.

2. Preprocessing: Training images from the dataset are downsized to 128x128. Further converted to greyscale using the Gaussian Blur function. This improves edge recognition by minimizing extraneous pixel processing in RGB fashion. Then, a model is trained using the training data from the transformed image.

3. Hand Segmentation: After preprocessing, the portion of the frame where the hand signs are performed is segmented out.

4. CNN Based classifier: Fundamental characteristics are taken from the preprocessed image. The classifier uses these feature vectors to identify sign classes

5. Loss Function: The decision to retrain the model is made based on a threshold value. For retraining, the loss function determines the loss value. A loss function compares the target and anticipated output values. To determine how well the neural network mimics the training data.

6. Optimizer: During training to minimize the output discrepancy between the expected and the target output depends upon the optimizer. Considering the worth of the prior loss, optimizers modifies the weights and learning rates of our neural network in order to minimize losses. Updating the weights causes the procedure to be repeated until the desired precision is attained.

Chapter 5

Requirement Analysis

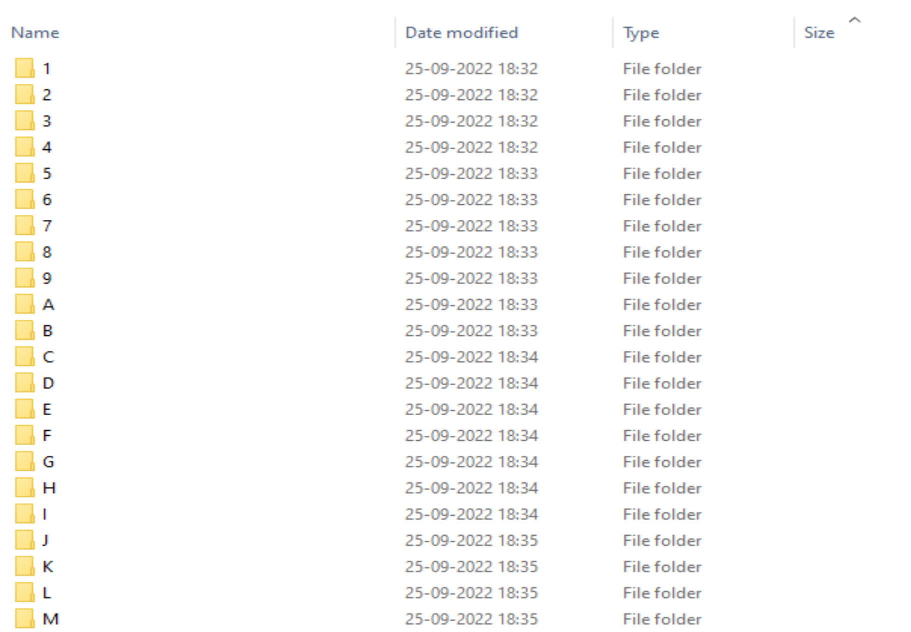
5.1 Frontend

HTML, CSS, JavaScript

- The boundary structures for web elements, such as the frame structure for camera input, and submit buttons will be created using HTML.
- CSS stylesheets will be used to define the size of buttons, frame structures, and the color of various text elements, as well as to align HTML elements in pages.
- JavaScript can be used to increase the web application's capabilities. It is flexible and can be used to dynamically define web page elements.

5.2 Dataset

- Data is collected from the Kaggle community.
- Include 35 classes 26 letters and 0–9 number symbols
- There are 1200 colored sample photos for each class.
- Each of the sample images is in the JPG format and ranges in size from 6.40 kb to 8kb.
- Each sample is 128 $\times$ 128 in size.



Name	Date modified	Type	Size
1	25-09-2022 18:32	File folder	
2	25-09-2022 18:32	File folder	
3	25-09-2022 18:32	File folder	
4	25-09-2022 18:32	File folder	
5	25-09-2022 18:33	File folder	
6	25-09-2022 18:33	File folder	
7	25-09-2022 18:33	File folder	
8	25-09-2022 18:33	File folder	
9	25-09-2022 18:33	File folder	
A	25-09-2022 18:33	File folder	
B	25-09-2022 18:33	File folder	
C	25-09-2022 18:34	File folder	
D	25-09-2022 18:34	File folder	
E	25-09-2022 18:34	File folder	
F	25-09-2022 18:34	File folder	
G	25-09-2022 18:34	File folder	
H	25-09-2022 18:34	File folder	
I	25-09-2022 18:34	File folder	
J	25-09-2022 18:35	File folder	
K	25-09-2022 18:35	File folder	
L	25-09-2022 18:35	File folder	
M	25-09-2022 18:35	File folder	

Figure 5.1: Different classes in the dataset

35 classes, including the A-Z alphabet and numbers 1 through 9, make up the Indian Sign Language dataset. Different folder structures are used to hold various class pictures.

Figure 5.2: Dataset sample images

Each sample contains a total of 1200 colored samples, which are used for the model's training and validation.

Chapter 6

Modern Tools

6.1 Hardware

- CPU Core i5 10th Gen 4 core.
- RAM 8 GB.
- Graphic Nvidia GeForce MX330.

6.2 Software

1. Python
   - Python programming language
   - Python is a general-purpose high-level programming language that is widely used in data science and for producing deep learning algorithms.
2. Keras
   - Keras is a high-level, deep-learning API developed by Google for implementing neural networks.
   - It is written in Python and is used to make the implementation of neural networks easy. It also supports multiple backend neural network computation
3. TensorFlow
   - TensorFlow is a Python library for fast numerical computing created and released by Google.
   - It is a foundation library that can be used to create Deep Learning models directly

   CNN
   - Machine learning includes convolutional neural networks, also known as ConvNet or CNNs. It is a subset of the several artificial neural network models that are employed for diverse purposes and data sets.
   - A CNN is a particular type of network design for deep learning algorithms that is utilized for tasks like image recognition and pixel data processing. The structure of a CNN is comparable to the connection structure of the human brain.
   - Similar to how the brain has billions of neurons, CNNs also have neurons, but they are structured differently. A CNN performs better with image inputs and voice or audio signal inputs compared to the earlier networks.

(a) CNN Layers
A convolutional layer, a pooling layer, and a fully connected (FC) layer make up a deep- learning CNN. The first layer is the convolutional layer, while the final layer is the FC layer. The complexity of the CNN grows from the convolutional layer to the FC layer. The CNN is able to identify increasingly larger and more intricate aspects of an image until it successfully recognizes the complete thing as a result of the rising complexity.

(b) Convolution Layer
The convolutional layer, the central component of a CNN, is where most computations take place. The first convolutional layer may be followed by a subsequent convolutional layer. A kernel or filter inside this layer moves over the image's receptive fields during the convolution process to determine whether a feature is present. The kernel traverses the entire image over a number of iterations. A dot product between the input pixels and the filter is calculated at the end of each iteration. A feature map or convolved feature is the result of the dots being connected in a certain pattern. In this layer, the image is ultimately transformed into numerical values that the CNN can understand and extract pertinent patterns from.

(c) Pooling Layer
The pooling layer similarly to the convolutional layer sweeps a kernel or filter across the input image. Contrary to the convolutional layer, the pooling layer has fewer input parameters but also causes some information to be lost. Positively, this layer simplifies the CNN and increases its effectiveness.

(d) Fully connected layer
Based on the features extracted in the preceding layers, picture categorization in the CNN takes place in the FC layer. Fully connected in this context means that every activation unit or node of the subsequent layer is connected to every input or node from the preceding layer. The CNN does not have all of its layers fully connected because that would create an excessively dense network. It would cost a lot to compute, increase losses, and have an impact on output quality.

Chapter 7

Implementation

As shown in Fig. 8, every color image has three channels: red, green, and blue. There are various color spaces in which a picture might exist, including grayscale CMYK and HSV. Multiple color channels in images present a computationally demanding problem when processing large amounts of data. A grayscale (or gray level) image is one that only uses different shades of grey as its colors. Red, green, and blue hues that are equal in intensity in RGB space are referred to as "grey" hues. Grayscale just has one dimension compared to the three of an RGB image, which requires less CPU power for processing. Each class is preprocessed and converted into grayscale image.

Figure 7.1: The dataset samples transformed into a grayscale format

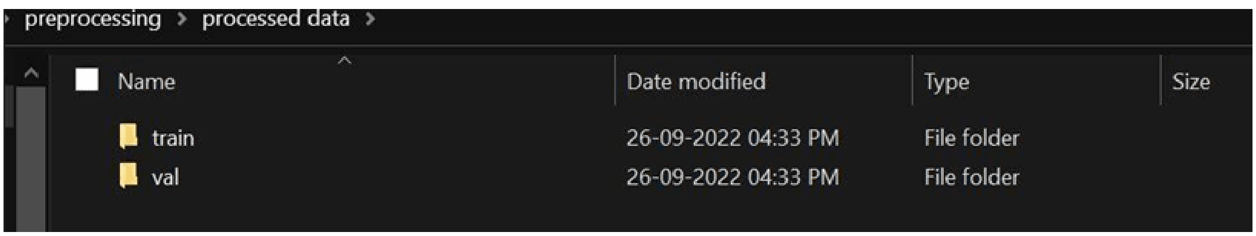


Figure 7.2: Dataset divided into Training and Validation set

Training set: The training set makes up the majority of the dataset in terms of size. The model will be trained (fit) using the training set. The training data is used to determine the values of the model parameters (rules or patterns). In other words, the training set is used to match the model's parameters to a set of hyperparameters.

Validation set: The process of training a model will be continuous. By experimenting with different combinations of hyperparameters, the model's accuracy will be improved. The validation set is then used to evaluate the model's performance. The validation test is therefore helpful for hyperparameter tuning or choosing the best model from a variety of models.

Chapter 8

Timeline & Plan for Next Semester

8.1 Timeline

Table 8.1: Timeline

2022 July 1-7	Group Formation.
July 8	Final Year Project Guidance to whole Class.
July 9-15	Project Selection and Idea Discussion .
July 16	Project Selection Presentation to Industry Experts.
July 17-28	Final Project Topic Selection based on Industry Experts Input and Guidance.
July 29	Discussion of the Project with the Guide.
August 5	Finalizing the Problem Statement and Objective of the Project.
August 26	Preparation of Literature Survey.
September 2- 11	Discussion of Literature Survey.
September 12	Project Review Presentation-1 .
September 13- 22	Preparation of review paper.
September 23	Discussion of Review paper with Guide.
September 30	Discussion of frontend implementation with guide.
October 7	Discussion of Project Synopsis.
October 21	Project Evaluation Presentation -2.
2022	November -December.

8.2 Plan for Next Semester

- Build a hand sign-to-text model
- Retraining model based on the loss values and accuracy
- Build audio/text to visual sign model

Chapter 9

Conclusion

The difficulties, developments, and potential future directions of the vision-based hand gesture recognition system were examined in this research. It seems that the publications we read emphasized the value of data collection, features, and the training data's context. Furthermore, it was observed that most of the databases used in hand gesture recognition studies came from a limited setting, highlighting the need for sign language databases that are less limited and contain data from other environments. Thus concludes that more attention needs to be paid to the uncontrolled environment setting in order to make the vision-based gesture recognition system ready for real-life application as it can provide researchers with the opportunity to improve the system's ability to recognize hand gestures in any form of environment. Data collection is a core procedure that has been stressed and placed in the spotlight of many vision-based hands gesture recognition studies.

Reference

[1] G.Anantha Rao, K.Syamala, P.V.V.Kishore, A.S.C.S.Sastry, “Deep Convolutional Neural Networks for Sign Language Recognition.” , 2018.

[2] Jie Huang, Wengang Zhou, Qilin Zhang, Houqiang Li, Weiping Li, “Video-Based Sign Language Recognition without Temporal Segmentation”, AAAI-18, April 2018

[3] M.A Hossen, Arun Govindaiah, Sadia Sultana, Alauddin Bhuiyan “Bengali Sign Language Recognition UsingDeep Convolutional Neural Network” June 2018

[4] Kshitij Bantupalli, Ying Xie, “Sign American language identification using Deep Learning and Computer Vision.”, IEEE,2018.

[5] Sruthi C. J and Lijiya A. Member, “Signet: A Deep Learning based Indian Sign Language Recognition System.”,IEEE, April 2019.

[6] Lean Karlo S. Tolentino, Ronnie O. Serfa Juan, August C. Thio-ac, Maria Abigail B. Pamahoy, Joni Rose R. Fortezaz and Xavier Jet O. Garcia. “Sign language identification using Deep Learning.”, IJMLC, December 2019.

[7] Ankita Wadhawan, Parteek Kumar, “Deep learning-based sign language recognition system for static signs” , Jan 2021

[8] Necati Cihan Camg, Oscar Koller, Simon Hadfifield and Richard Bowden, “Sign Language Transformers: JointEnd-to-end Sign Language Recognition and Translation”, 2020.

[9] Razieh Rastgoo, Kourosh Kiani, Sergio Escalera, “Word separation in continuous sign language using isolated signs and post-processing”, 13 Apr 2022.

[10] Abdul Mannan , Ahmed Abbasi , Abdul Rehman Javed , Anam Ahsan ,Thippa Reddy Gadekallu , and Qin Xin “Hypertuned Deep Convolutional Neural Network for Sign Language Recognition”,30 April 2022.

[11] K.Shenoy,Tejas Dastane,Varun Rao,Devendra Vyavaharkar,“Real-timeIndianSignLanguage(ISL)Recognition” ,1 July 2021.