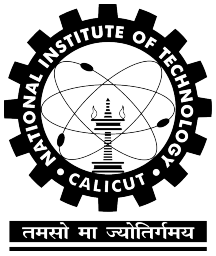
Indian Sign Language Dynamic Hand Gesture Recognition

CS4099 Project Final Report

Submitted by

Butukuri Peri Reddy	**B210578CS**
Bachu Venkata Dhanush	**B210536CS**

Under the Guidance of
LIJIYA A

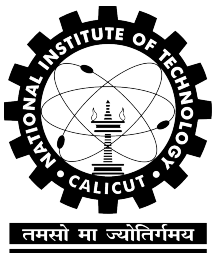


Department of Computer Science and Engineering
National Institute of Technology Calicut
Calicut, Kerala, India - 673 601

May 07, 2025

NATIONAL INSTITUTE OF TECHNOLOGY CALICUT, KERALA, INDIA - 673 601

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



2025

CERTIFICATE

Certified that this is a bonafide record of the project work titled

ISL DYNAMIC HAND GESTURE RECOGNITION

done by

Butukuri Peri Reddy
Bachu Venkata Dhanush

of eighth semester B. Tech in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering of the National Institute of Technology Calicut

Project Guide	**Head of Department**
Lijiya A	Dr. Subashini R
Associate Professor	Associate Professor

DECLARATION

We hereby declare that the project titled, **ISL DYNAMIC HAND GESTURE RECOGNITION**, is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or any other institute of higher learning, except where due acknowledgement and reference has been made in the text.

Place :NIT CALICUT
Date :07-05-2025

Signature :
Name : Butukuri Peri Reddy
Reg. No. :B210578CS
Name : Bachu Venkata Dhanush
Reg. No. :B210536CS

Abstract

Indian Sign Language (ISL) is the way of communication for deaf and dumb people in India to communicate with others. Almost 6.3% population of India are deaf people, which makes it difficult to communicate with other people. Even though ISL is available as standard but is still not very common among nonsigners. Sign language interpreters are very few and limited to few gestures, making it difficult for people with hearing and speaking disabilities to interact seamlessly in educational, professional, and social settings.

In this paper, we have developed a Sign Language interpreter that takes the input sign gesture and gives the name of the gesture as the output. We have used K-means clustering technique for key frames extraction and 2D Convolutional Neural Network (CNN) along with Long Short-Term Memory (LSTM) to train the system with a given database. For our work, we have used a dataset that consists of 10 classes with each class having approximately 100 video sequences.

Index Terms — CNN, Sign Language Recognition, Indian Sign Language, LSTM

ACKNOWLEDGEMENT

We would like to express our heartfelt gratitude to all those who supported and guided us throughout the course of this project.

First and foremost, we are immensely grateful to our project guide, Ms. Lijiya A, Associate Professor, Department of Computer Science and Engineering, NIT Calicut, for her valuable guidance, constant encouragement, and constructive feedback throughout the development of this project. Her insights and mentorship were instrumental in the successful completion of our work.

We would also like to extend our sincere thanks to Dr. Subashini R, Head of the Department, for providing us with the necessary infrastructure and academic environment to carry out our project effectively.

We are thankful to the Department of Computer Science and Engineering, NIT Calicut, for offering us the opportunity to explore and work on this meaningful problem related to Indian Sign Language and contribute, in a small way, to the betterment of assistive technology.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Indian Sign Language (ISL) is an essential communication medium for the deaf and mute community in India, consisting of gestures that include both static and dynamic hand movements. Dynamic gestures, involving continuous motion, are challenging to recognize due to their temporal complexity. An automated system for interpreting these gestures can help bridge the communication gap between ISL users and non-users, fostering inclusivity in various social settings and improving accessibility in education, workplaces, and public services.

Our project uses deep learning techniques to recognize dynamic ISL gestures. Convolutional Neural Networks (CNNs) extract spatial features from video frames, while Long Short-Term Memory (LSTM) networks model temporal dependencies across sequential frames. This combination enables accurate recognition of gestures from video input, making it suitable for real-time applications and contributing to advancements in assistive technologies for the hearing-impaired community.

Chapter 2

Literature Survey

ISLR has undergone notable advancements with the integration of deep learning and computer vision techniques.This literature survey examines few important research papers that lay the groundwork for the concept of ISLR. Each paper contribute unique insights into feature extraction, Gesture Recognition methodologies.

2.0.1 Real-Time Indian Sign Language Recognition System using YOLOv3 Model

[3] In this paper authors used YOLOv3 model which is an incremental change to the existing YOLOv2 architecture.YOLOv3 is a real-time object detection model based on YOLOv2 using the Darknet-53 network made up of 106 fully convolutional layers. The objects are also detected at three different scales, namely layer 82, 94, and 106 having downsampling factors of 32, 16, and 8 respectively. By using 1x1 detection kernels at all these varied layers, the model is thus capable of producing feature maps of many spatial dimensions. This makes YOLOv3 highly suited for real-time object detection tasks such as Indian Sign Language recognition.

In ISL recognition, gestures are assumed to be located in another spatial space away from the person's body, and the YOLOv3 model is trained on static images as well as dynamic video streams. The dataset contains 780 images of 16 classes plus 35 videos grouped into 7 classes. Images have been annotated pixel by pixel using LabelImg: a bounding box has been drawn around gestures, then saved in YOLO format. The annotations include class of the object, center of the bounding box and the size. DarkNet framework is employed in training the model where, for static recognition it requires 32,000 iterations while dynamic recognition has 14,000 iterations. Training is optimized through manual alteration in the configuration files and storing the weights after every 1,000 iterations.

2.0.2 Dynamic Two Hand Gesture Recognition using CNN-LSTM based network

[1] In this paper, the author presents a dynamic hand gesture recognition model for Indian Sign Language (ISL) using a combination of Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks. The CNN is employed to extract spatial features from gesture images, while the LSTM captures the temporal dynamics of sequential gestures. The model's architecture includes convolutional, pooling, and fully connected layers, followed by a softmax layer for classification. Preprocessing techniques such as resizing, color space conversion, and noise reduction are applied to optimize the input data. The extracted features are then fed into a stacked LSTM network for final gesture classification.

2.0.3 Continous Dynamic Indian Sign Language Gesture Recognition with Invariant Backgrounds

[2] In this paper, the authors present a system for continuous Indian Sign Language (ISL) gesture recognition with invariant backgrounds. The approach

focuses on recognizing continuous ISL gestures across various backgrounds using a vision-based approach.

The framework begins with pre-processing, where background elimination techniques are used to isolate the hand and upper body from the background. This step involves detecting foreground objects based on pixel differences and applying skin color segmentation to extract the hand region. After this, a gradient-based key frame extraction method is used to identify meaningful gestures from continuous sequences. By tracking changes in the hand's orientation across frames, the system isolates the start and end of each gesture, reducing the computational burden by focusing on frames with significant information.

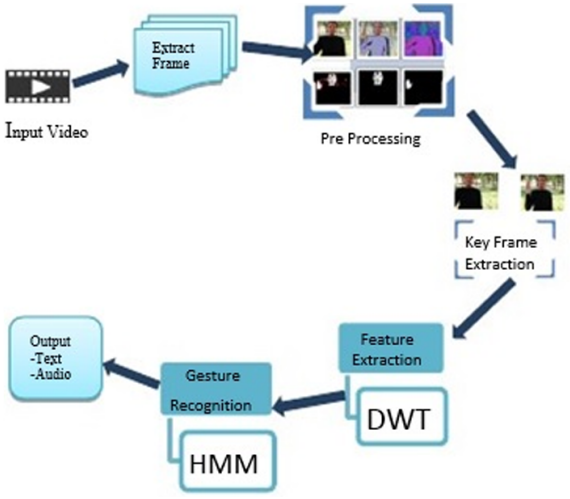


Figure 2.1: Framework for Continuous Dynamic ISL gesture Recognition

[2] Once key frames are extracted, Discrete Wavelet Transform (DWT) is applied to these frames for feature extraction. DWT decomposes the image into multiple frequency components, capturing important gesture details while discarding irrelevant noise. Finally, Hidden Markov Models (HMM) are used to recognize gestures. HMM models the temporal sequence of gestures,

where each state represents a gesture or part of it. The system is trained using a custom ISL dataset, and the Viterbi algorithm is employed to decode the most likely sequence of gestures during testing.

2.0.4 A Modified LSTM Model for Continuous Sign Language Recognition using Leap Motion

[5] This paper introduces a novel method for continuous sign language recognition using a combination of Leap Motion sensor data and a modified Long Short-Term Memory (LSTM) model. The Leap Motion sensor plays a crucial role by capturing detailed 3D hand and finger movement data, including the positions of fingertips and palms in real time. This data, which is collected at a high frame rate (up to 200 frames per second), ensures that even subtle finger and hand movements are accurately tracked. To prepare the data for the recognition model, preprocessing steps are applied to remove redundant information, such as wrist movements, and normalize the data across different users to account for variations in hand sizes and spans. The extracted dynamic features are used as inputs for further processing and analysis.

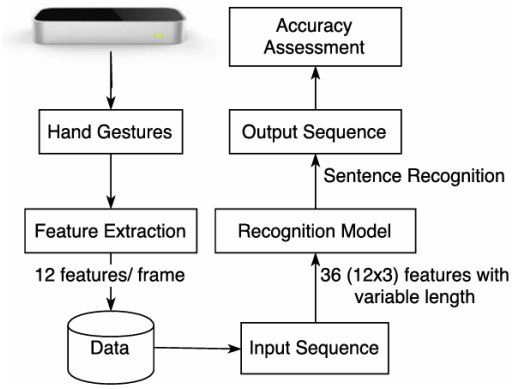


Figure 2.2: Framework for continuous SLR using Leap motion sensor

[5] The modified LSTM architecture introduces a reset gate, which allows

the system to segment continuous sign sequences by resetting the network's memory when transitioning between different sign gestures. This helps improve recognition accuracy in continuous sign language sequences, overcoming the challenge of handling transitions between connected gestures.

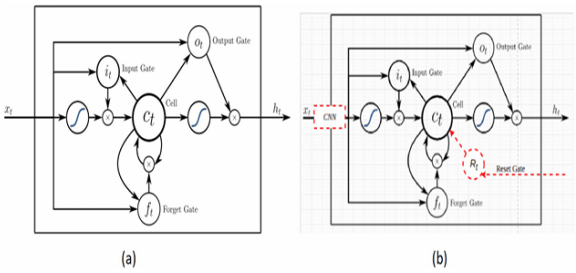


Figure 2.3: Pictorial representation of LSTM units: (a) basic LSTM (b) modified LSTM with Reset functionality.

[5] The system is trained in two stages: first on isolated sign gestures and then fine-tuned for continuous gestures. By adding a reset mechanism, the modified LSTM can better differentiate between gestures in a sequence, thereby improving performance compared to traditional LSTM models. The proposed system also uses a CNN to capture spatial relationships between hand movements, which are then modeled by the LSTM for temporal analysis.

[5] The experimental results demonstrate that the reset-enabled LSTM significantly improves recognition accuracy and segmentation of continuous gestures compared to traditional models.

2.0.5 Real-Time Dynamic Hand Gesture Recognition

[4] The paper Real-Time Dynamic Hand Gesture Recognition presents a system for recognizing hand gestures in real-time using dynamic video input, with applications in Human-Computer Interaction(HCI).

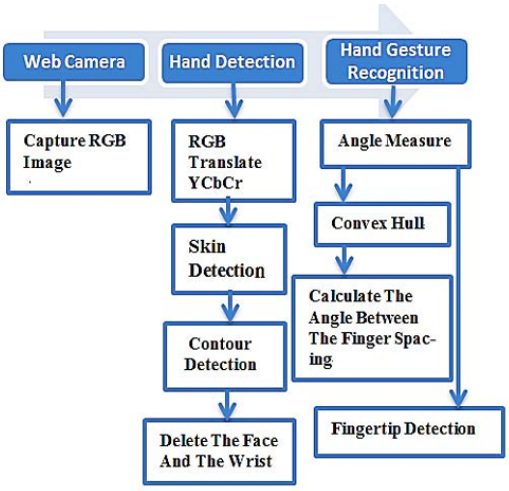


Figure 2.4: The framework of real-time dynamic hand gesture recognition

There are two main parts in this paper they are hand detection and hand gesture recognition.

[4] The hand detection process begins by converting the captured image from RGB to YCbCr color space for effective skin detection, generating a binary mask where skin areas are white. Morphological operations are applied to remove noise, and contours are extracted using OpenCV. Small or irrelevant contours (e.g., facial contours) are filtered out based on a size threshold. The hand's contour is identified by distinguishing the wrist from the face, and the contour center is computed to further refine the hand region.

[4] In hand gesture recognition, the system uses the convex hull algorithm to identify convex defect points (start, end, and depth points) along the hand contour, which represent the gaps between fingers. Angles between fingers are calculated using these points, helping to classify different gestures. Fingertips are identified as sharp points on the contour, with candidate points filtered

by removing those inside a defined radius or with obtuse angles.

2.0.6 Static Hand Gesture Recognition using Mixture of Features and SVM classifier

[9] The research paper presents a system for static hand gesture recognition that involves three main steps: preprocessing, feature extraction, and classification. In the preprocessing step, the images from the dataset, which have resolutions of 256×248 and 320×240, are first converted to grayscale to reduce computational complexity and focus on intensity variations. The pixel intensity values are then normalized to ensure uniformity across the images and mitigate any lighting inconsistencies. Noise removal techniques are applied to enhance image quality and ensure that only relevant information is retained for further analysis. Additionally, the hand regions are segmented using specific skin color thresholds in the YCbCr color space,isolating the hand and reducing the influence of the background.

[9] In the feature extraction step, multiple methods are employed to capture various aspects of the hand gestures. The Histogram of Oriented Gradients (HOG) technique is used to extract shape and edge information, while Local Binary Patterns (LBP) focuses on texture analysis. Geometric features are also extracted to provide structural details of the hand. These features from different methods are then fused into a single comprehensive feature vector, enhancing the system's ability to differentiate between gestures accurately. This step ensures that the system has a rich and varied set of characteristics to base its classification on.

[9] The classification step utilizes a Support Vector Machine (SVM) to categorize the hand gestures. The SVM classifier processes the fused feature vectors and efficiently classifies them into their respective gesture categories. The system achieves an impressive accuracy of 99.50%, demonstrating

the effectiveness of the preprocessing, feature extraction, and classification pipeline. By carefully combining these steps, the system ensures high recognition accuracy and robustness in static hand gesture recognition.

2.0.7 K-Nearest Correlated Neighbor Classification for ISL Gesture Recognition using Feature Fusion

[6] The dataset used in this study comprises manually created images of ISL gestures for alphabets A to Z, representing single-handed and double-handed gestures. For each gesture, the dataset includes 20 images, resulting in a total of 520 training images and 260 test images. To address the complexity of ISL gestures, the dataset was divided into two categories: single-handed and double-handed gestures. This categorization significantly reduced the computational cost and improved the model's efficiency by narrowing the classification focus within smaller subgroups. Each gesture in the dataset was preprocessed to extract HOG (Histogram of Oriented Gradients) and SIFT (Scale Invariant Feature Transform) features, which were then fused to provide a robust representation for classification.

[6] The K-Nearest Correlated Neighbor (KNCN) algorithm was used as the core classification technique in the proposed method. Unlike traditional K-Nearest Neighbor (KNN), KNCN calculates a correlation matrix to measure the similarity between features of test and training images. The algorithm combines the fused HOG and SIFT feature matrices, reducing dimensionality, and aligning feature vectors of varying dimensions. A high correlation coefficient indicates a strong similarity between test and training instances. The KNCN classifier identifies the gesture by finding the training image with the highest correlation to the test image. This approach leverages the combined power of HOG and SIFT features, yielding improved accuracy over single-feature-based methods.

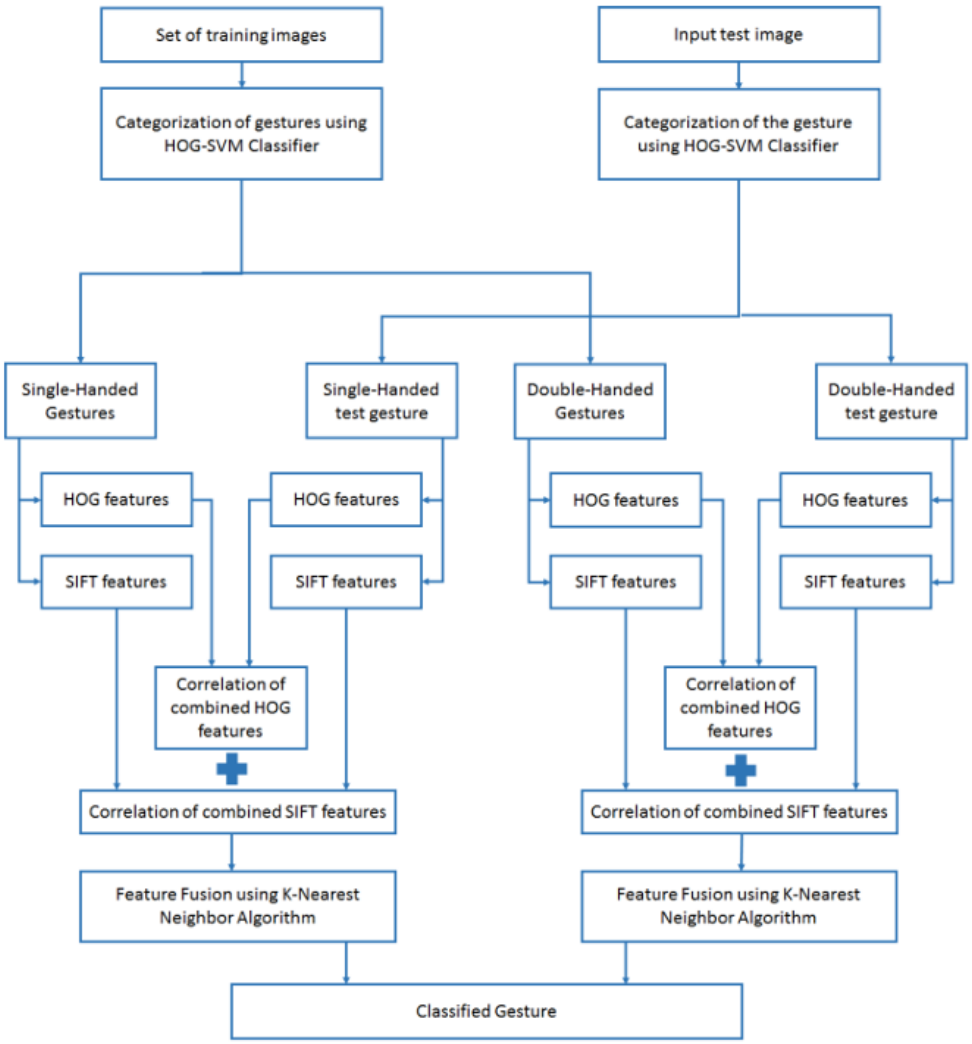


Figure 2.5: Model for ISL gesture Recognition using KCNN

Chapter 3

Problem Definition

The goal of this project is to develop a reliable Indian Sign Language Recognition (ISLR) system capable of accurately recognizing ISL Dynamic gestures and classifying them into their respective classes using hand shapes and motion patterns.

Chapter 4

Methodology

4.0.1 Dataset

The dataset contains ten different types of gestures which are Bye, Enjoy, H, Hello, J, Morning, Rain, Wind, Work, Y. Each category is represented by approximately 100 video sequences. To ensure compatibility with the proposed model, the size of the extracted keyframes was standardized to 224x224 pixels before concatenation. For model training, validation and testing, the dataset was partitioned into subsets comprising 70%, 15% and 15% of the total data. So, the training set consists of 668 video sequences, the validation set consists of 140 video sequences, and the testing set consists of 140 video sequences.

4.0.2 Data Preprocessing and Key Frames extraction

Each gesture is represented as a sequence of frames in a video. Since videos contain redundant information, processing every frame is computationally expensive. Instead, we extract key frames that best represent the gesture. This process involves:

1. Preprocessing includes reading frames, resizing frames, and preparing frames for VGG19.

2. The pretrained VGG19 model (without classification layers) is used to extract deep feature representations of video frames.

3. The frames are passed through the model, and feature vectors are obtained by global average pooling over spatial dimensions (reducing them to 1D feature vectors).

4. Principal Component Analysis (PCA) is applied to reduce the high-dimensional feature vectors obtained from VGG19.

5. The KMeans clustering algorithm is used to group similar frames into 10 clusters based on reduced PCA features.

6. The closest frame to each cluster center is chosen as a keyframe.

7. The keyframes along with label are saved as .pt files for using in model training

The approach analyzes frames using VGG19 to extract keyframes by grouping similar visual features and selecting representative frames. This method assumes frames with similar features represent similar content and requires tuning the number of clusters for desired granularity. By employing this keyframe extraction approach, the computational burden is reduced while preserving essential visual information for subsequent analysis.

4.0.3 Feature Extraction and Classification

Traditional 2D CNNs are effective in capturing spatial relationships within individual frames but struggle to incorporate temporal dependencies across consecutive frames in a video sequence. This limitation hinders their performance in dynamic gesture recognition tasks. To address this, we employ Long Short-Term Memory (LSTM) networks alongside 2D CNNs to explicitly model the temporal evolution of hand gestures in video sequences. The

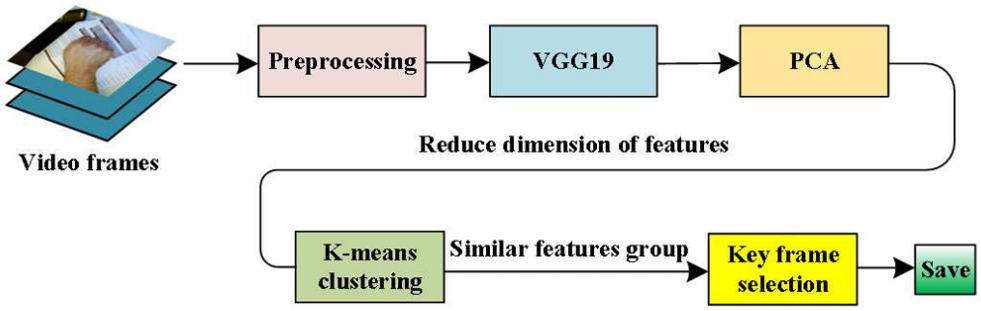


Figure 4.1: Workflow for Key Frames Selection in Video Analysis

hybrid approach significantly improves the model's ability to recognize a diverse range of hand gestures with high accuracy.

The model begins with a 2D CNN that processes each video frame individually to extract spatial features. The convolutional layers utilize 3×3 filters, followed by Batch Normalization and ReLU activation to stabilize learning and enhance feature extraction. Max Pooling layers are applied to progressively reduce the spatial dimensions while preserving crucial information. The final CNN output consists of 512 feature maps of size 7X7, which serve as input for the LSTM.

Since the LSTM network requires sequential data, the output feature maps from the CNN are reshaped to match its input requirements. This restructuring ensures that the CNN-extracted spatial features are appropriately formatted as a sequence, allowing the LSTM to process them over time. The LSTM network consists of three layers with a hidden size of 128, designed to capture temporal dependencies in hand gesture movements. The batch-first approach ensures compatibility with PyTorch's LSTM format, enabling efficient sequence processing.

The fully connected (FC) layers receive the LSTM's final time-step output

and perform classification. The FC architecture includes a linear layer

$$(128 -> 64\, neurons)$$

, Batch Normalization, ReLU activation, and a final output layer

$$(64 -> num - classes)$$

, where $num-classes = 10$. These layers map the extracted spatial-temporal features to probability distributions over the ten gesture classes, allowing the model to predict the most likely hand gesture.

By combining CNNs for spatial encoding and LSTMs for motion tracking, the model effectively recognizes hand gestures from video sequences. This hybrid approach ensures that both static spatial information and dynamic motion cues are captured, leading to more accurate and robust gesture classification.

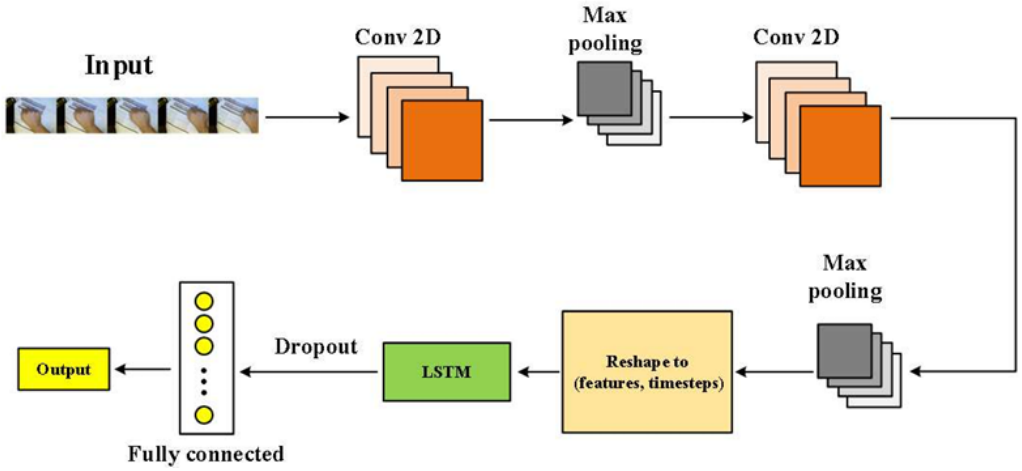


Figure 4.2: Proposed Model for ISL dynamic hand gesture recognition

Chapter 5

Results

5.1 Model Training

The training process of the CNN-LSTM model incorporates an early stopping mechanism to optimize performance and prevent overfitting. The model is trained using a supervised learning approach, where it learns to classify hand gestures from video sequences. The training loop involves iterating over the dataset for a defined number of epochs (up to 50), during which the model updates its parameters using backpropagation and the Adam optimizer. Each batch of input data is passed through the model, and the Cross-Entropy Loss function is used to compute the error between the predicted and actual labels. The gradients of the loss are calculated and propagated backward to adjust the model's weights. The training accuracy and loss are recorded at each epoch to monitor the learning progress.

To ensure optimal generalization, a validation set is used to evaluate the model's performance after each epoch. The EarlyStopping class is implemented to halt training if the validation loss does not improve over a predefined number of epochs (patience = 5). This prevents unnecessary computation and reduces the risk of overfitting. The best-performing model,

determined by the lowest validation loss, is saved to disk for future use. After training, the model is reloaded, and its final performance is assessed on the validation dataset.

The Loss vs epochs and accuracy vs epochs graphs are given below in Fig 5.1 and Fig 5.2

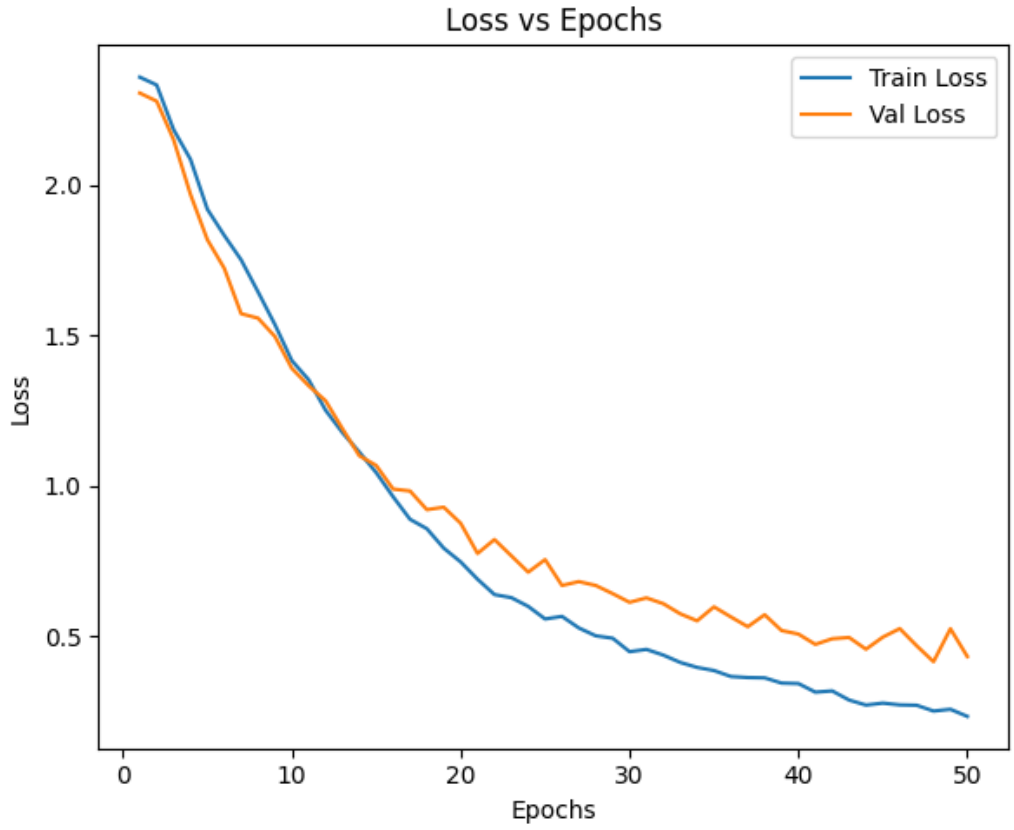


Figure 5.1: Loss vs epochs

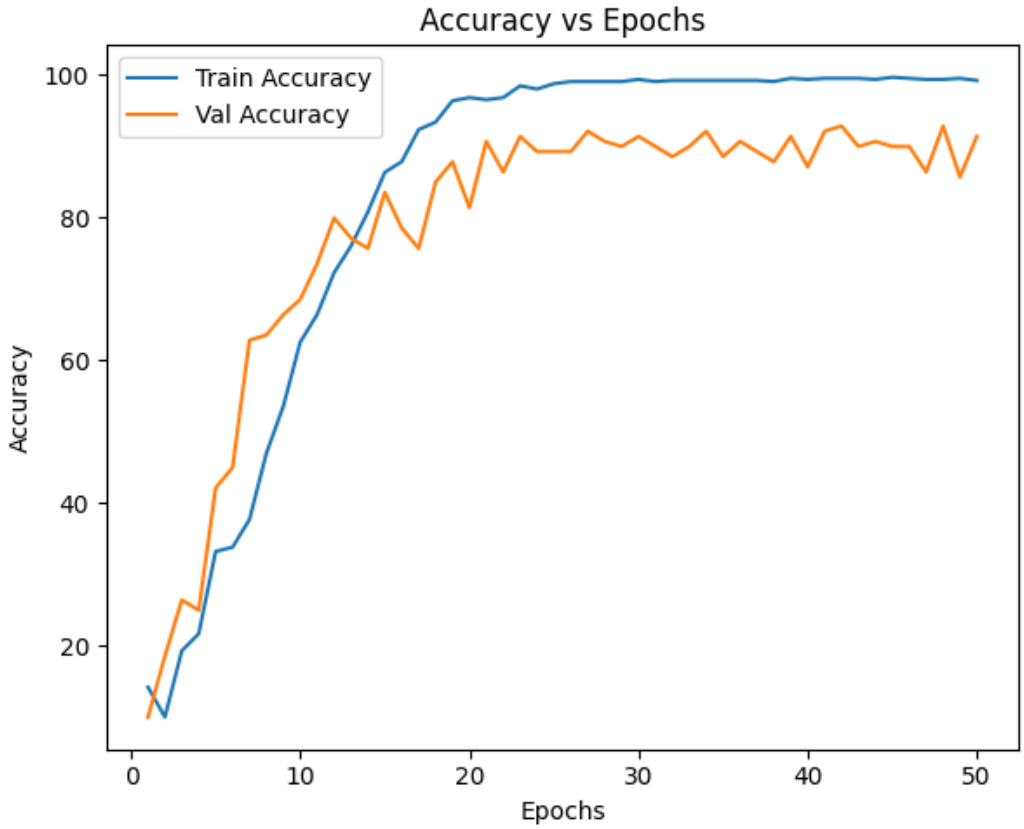


Figure 5.2: accuracy vs epochs

5.2 Experimental Results

Evaluating the performance of a classification model requires appropriate metrics. Accuracy, while a basic measure, often falls short in complex scenarios. A deeper understanding of metrics like precision, recall, F1-score, and confusion matrix is crucial for a comprehensive evaluation.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{5.1}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{5.2}$$

$$\text{F1-score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5.3}$$

These measures are characterized by TP(True Positive), FP(False positive), followed by TN(True negative) and FN(False negative). Using these mentioned equations the classification reports are created and the performance of the model is determined.

The confusion matrix and classification report is shown below in Fig 5.3 and Fig 5.4.

Class	**Label**
0	Bye
1	Enjoy
2	H
3	Hello
4	J
5	Morning
6	Rain
7	Wind
8	Work
9	Y

Table 5.1: Mapping of Classes to Labels

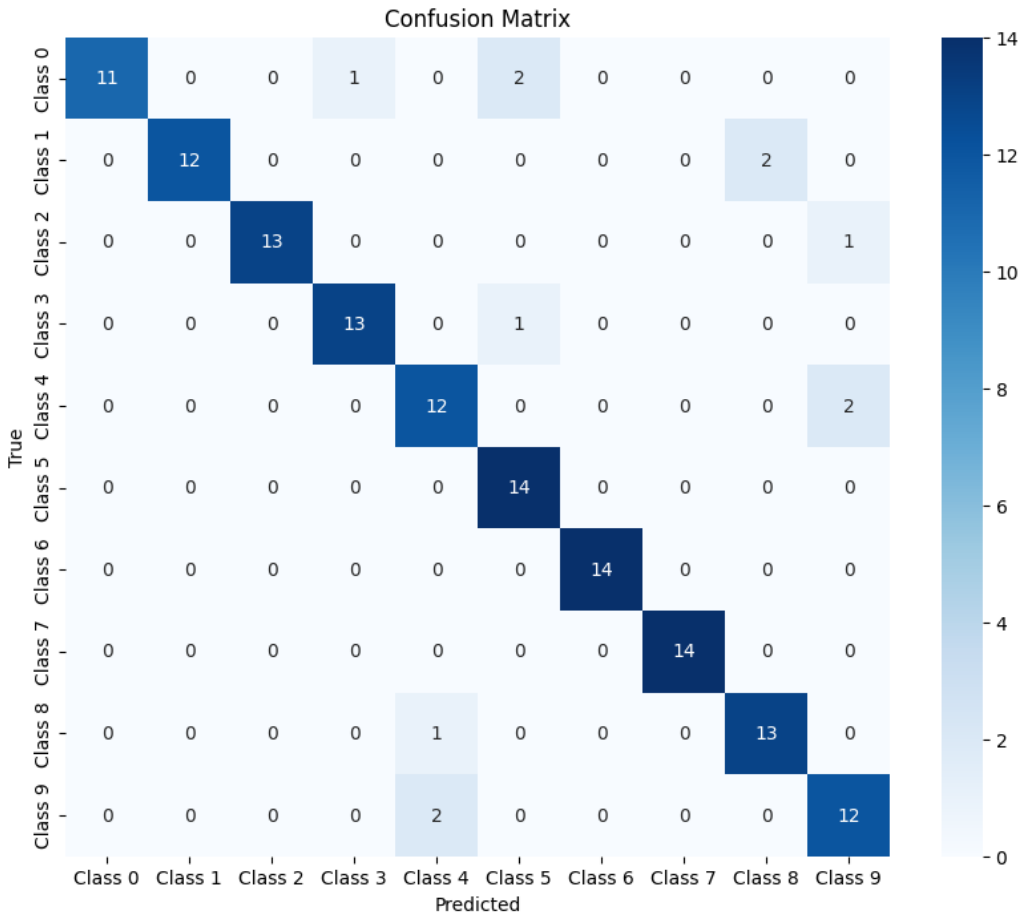


Figure 5.3: Confusion Matrix for our 2DCNN-LSTM Model

We have achieved an accuracy of 91% on test dataset.

Classification Report:

	precision	recall	f1-score	support
Class 0	1.00	0.79	0.88	14
Class 1	1.00	0.86	0.92	14
Class 2	1.00	0.93	0.96	14
Class 3	0.93	0.93	0.93	14
Class 4	0.80	0.86	0.83	14
Class 5	0.82	1.00	0.90	14
Class 6	1.00	1.00	1.00	14
Class 7	1.00	1.00	1.00	14
Class 8	0.87	0.93	0.90	14
Class 9	0.80	0.86	0.83	14
accuracy			0.91	140
macro avg	0.92	0.91	0.91	140
weighted avg	0.92	0.91	0.91	140

Figure 5.4: Classification Report

Table 5.2: Comparison of State of the Art Methods

Reference	**Methodology Used**	**Accuracy**
[17]	Key Frame extraction with SVM	79.3%
[18]	Inception V3 + LSTM	81%
—	Proposed Method (CNN+LSTM)	**91%**

Chapter 6

Conclusion and Future work

The paper presented a vision-based deep learning architecture for ISL dynamic gesture video classification using a hybrid CNN-LSTM model. The system was trained on a curated dataset of videos, each represented by a sequence of key frames extracted through feature clustering and dimensionality reduction techniques. The model was trained and evaluated with a high degree of accuracy, achieving 91% accuracy on the test dataset, demonstrating its effectiveness in capturing both spatial and temporal patterns from video data.

Sign language recognition is a broad research domain that includes complex challenges such as continuous gesture recognition, co-articulation handling, and sentence-level understanding. Although the proposed system effectively handles dynamic ISL gestures, it can be further extended with additional modules to recognize continuous sign sequences and eliminate transitional ambiguities.

References

[1] V. Sharma, M. Jaiswal, A. Sharma, S. Saini, and R. Tomar, "Dynamic two hand gesture recognition using cnn-lstm based networks," in *2021 IEEE International Symposium on Smart Electronic Systems (iSES)*, pp. 224–229, 2021.

[2] K. Tripathi, N. Baranwal, and G. Nandi, "Continuous dynamic indian sign language gesture recognition with invariant backgrounds," in *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 2211–2216, 2015.

[3] N. Sarma, A. K. Talukdar, and K. K. Sarma, "Real-time indian sign language recognition system using yolov3 model," in *2021 Sixth International Conference on Image Information Processing (ICIIP)*, vol. 6, pp. 445–449, 2021.

[4] H. Y. Lai and H. J. Lai, "Real-time dynamic hand gesture recognition," in *2014 International Symposium on Computer, Consumer and Control*, pp. 658–661, 2014.

[5] A. Mittal, P. Kumar, P. P. Roy, R. Balasubramanian, and B. B. Chaudhuri, "A modified lstm model for continuous sign language recognition using leap motion," *IEEE Sensors Journal*, vol. 19, no. 16, pp. 7056–7063, 2019.

[6] B. Gupta, P. Shukla, and A. Mittal, "K-nearest correlated neighbor classification for indian sign language gesture recognition using feature fusion," in *2016 International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1–5, 2016.

[7] N. Heidari, J. Norouzi, M. S. Helfroush, and H. Danyali, "Dynamic hand gesture recognition with 2dcnn-lstm and improved keyframe extraction," in *2024 14th International Conference on Computer and Knowledge Engineering (ICCKE)*, pp. 429–434, 2024.

[8] A. Sridhar, R. G. Ganesan, P. Kumar, and M. Khapra, "Include: A large scale dataset for indian sign language recognition," in *Proceedings of the 28th ACM International Conference on Multimedia*, MM '20, (New York, NY, USA), p. 1366–1375, Association for Computing Machinery, 2020.

[9] D. K. Ghosh and S. Ari, "Static hand gesture recognition using mixture of features and svm classifier," in *2015 Fifth International Conference on Communication Systems and Network Technologies*, pp. 1094–1099, 2015.

[10] A. Kolkur, A. Yattinmalgi, G. Korimath, S. Chikkamath, N. S. R, and S. Budihal, "Deep learning based indian sign language recognition for people with speech and hearing impairment," in *2024 IEEE International Conference on Contemporary Computing and Communications (InC4)*, vol. 1, pp. 1–5, 2024.

[11] S. C.J. and L. A., "Signet: A deep learning based indian sign language recognition system," in *2019 International Conference on Communication and Signal Processing (ICCSP)*, pp. 0596–0600, 2019.

[12] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 3, pp. 311–324, 2007.

[13] Rosalina, L. Yusnita, N. Hadisukmana, R. B. Wahyu, R. Roestam, and Y. Wahyu, "Implementation of real-time static hand gesture recognition using artificial neural network," in *2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT)*, pp. 1–6, 2017.

[14] L. Chen, F. Wang, H. Deng, and K. Ji, "A survey on hand gesture recognition," in *2013 International Conference on Computer Sciences and Applications*, pp. 313–316, 2013.

[15] P. Shukla, A. Garg, K. Sharma, and A. Mittal, "A dtw and fourier descriptor based approach for indian sign language recognition," in *2015 Third International Conference on Image Information Processing (ICIIP)*, pp. 113–118, 2015.

[16] X. Wang, Z. Chen, X. Wang, Q. Zhao, and B. Liang, "A comprehensive evaluation of moving static gesture recognition with convolutional networks," in *2018 3rd Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)*, pp. 7–11, 2018.

[17] A. Kaur and S. Bansal, "A machine learning based approach for dynamic hand gesture recognition in human-robot interaction," in *2024 12th International Conference on Intelligent Systems and Embedded Design (ISED)*, pp. 01–06, 2024.

[18] G. Jayadeep, N. Vishnupriya, V. Venugopal, S. Vishnu, and M. Geetha, "Mudra: Convolutional neural network based indian sign language translator for banks," in *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, pp. 1228–1232, 2020.