STM JOURNALS

Recent Trends in
Programming Languages

ISSN: 2455-1821
Volume 11, Issue 1, 2024
DOI (Journal): 10.37591/RTPL

https://journals.stmjournals.com/rtpl/

Research RTPL

An Automation Detection for Sign Language Using AI

Om Krishna Hankare[1,*], Ravikishor Bandla[1], Gangadri[1], Navaneeth Harivikas[1], Mahesh Sutar[2]

Abstract

Sign language recognition has attracted considerable interest because of its ability to facilitate communication between the deaf community and the public, thereby bridging communication divides. Traditional approaches to sign language recognition often face challenges in accurately interpreting the complex and nuanced gestures inherent in sign languages. However, recent advancements in deep learning techniques have shown promising results in improving the accuracy and robustness of sign language recognition systems. This study presents an enhanced sign language recognition system utilizing state-of-the-art deep learning architectures. Convolutional neural networks (CNNs) are utilized to extract spatial characteristics from images of sign language, while recurrent neural networks (RNNs) capture the temporal relationships present in sequences of sign language. The proposed system is trained on large-scale sign language datasets to learn discriminative features and improve generalization performance. Furthermore, to address the challenges posed by variations in lighting conditions, backgrounds, and signer characteristics, data augmentation techniques are employed to enhance the robustness of the model. Additionally, transfer learning is explored to leverage pre-trained models on large-scale visual recognition tasks for improved performance on sign language recognition. Experimental results demonstrate that the proposed approach achieves state-of-the-art performance on benchmark sign language recognition datasets, surpassing previous methods in terms of accuracy and generalization. The system's effectiveness is validated through extensive evaluation on diverse sign language datasets, showcasing its potential for real-world applications in facilitating communication for the hearing-impaired community.

Keywords: Sign language recognition, deep learning, convolutional neural networks, recurrent neural networks, data augmentation, transfer learning

***Author for Correspondence**
Om Krishna hankare
E-mail: omhankare29@gmail.com

[1]Student, Department of Computer Science and Engineering (Artificial Intelligence & Machine Learning), Vishwaniketan's Institute of Management Entrepreneurship and Engineering Technology, Khalapur Mumbai University, Mumbai, Maharashtra, India
[2]Professor, Department of Computer Science and Engineering (Artificial Intelligence & Machine Learning), Vishwaniketan's Institute of Management Entrepreneurship and Engineering Technology, Khalapur Mumbai University, Mumbai, Maharashtra, India

Received Date: March 14, 2024
Accepted Date: March 20, 2024
Published Date: April 4, 2024

Citation: Om Krishna Hankare, Ravikishor Bandla, Gangadri, Navaneeth Harivikas, Mahesh Sutar. An Automation Detection for Sign Language Using AI. Recent Trends in Programming Languages. 2024; 11(1): 1–14p.

INTRODUCTION

Sign language plays a vital role in communication for millions of people globally who experience deafness or hearing impairment. In contrast to spoken languages, sign languages utilize visual gestures, facial expressions, and body movements to convey messages and aid in communication. However, despite its importance, there exists a significant communication barrier between the deaf community and the general population, primarily due to the limited understanding and recognition of sign language among non-signers.

Traditional methods of sign language recognition often rely on handcrafted features and rule-based approaches, which struggle to effectively capture the complexity and variability inherent in sign language gestures. These methods are often limited in their ability to adapt to different signing styles, variations in hand shapes, and environmental factors such as lighting conditions and background clutter.

The swift progress in deep learning methodologies, notably convolutional neural networks (CNNs) and recurrent neural networks (RNNs), has led to a fundamental transformation in the field of sign language recognition research. Deep learning models have demonstrated remarkable capabilities in automatically learning hierarchical representations from raw sensory data, thereby overcoming the limitations of traditional handcrafted feature-based approaches.

In this context, this study proposes an enhanced sign language recognition system leveraging state-of-the-art deep learning architectures. By harnessing the power of CNNs for spatial feature extraction and RNNs for capturing temporal dependencies, the proposed system aims to achieve robust and accurate recognition of sign language gestures.

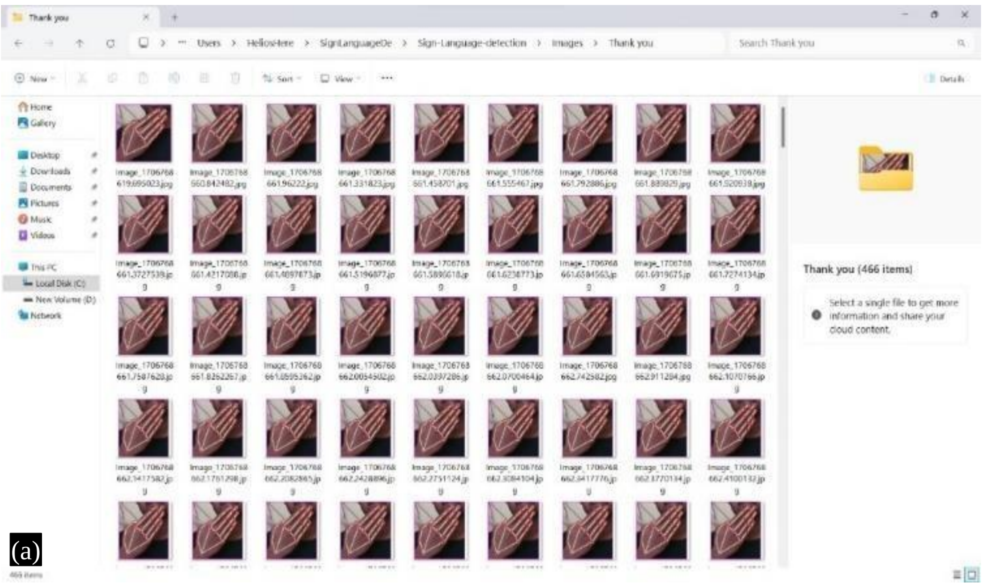
Moreover, recognizing the challenges posed by variations in signing styles, lighting conditions, and signer characteristics, this study investigates the use of data augmentation techniques and transfer learning to enhance the generalization performance of the model. By augmenting the training data with synthetic variations and leveraging knowledge transfer from pre-trained models on large-scale visual recognition tasks, the proposed system aims to improve its adaptability to diverse real-world scenarios.

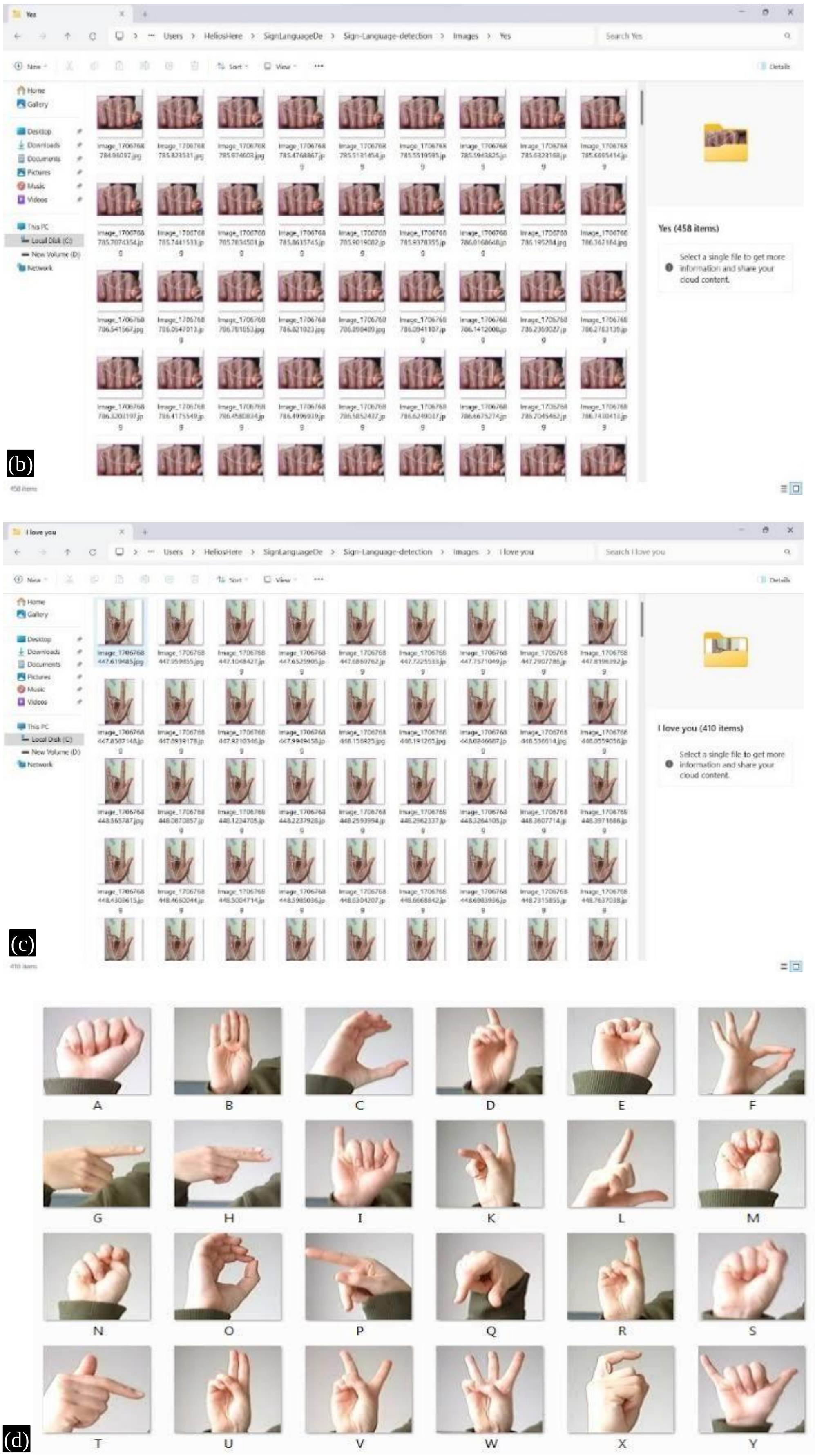
The contributions of this study are two-fold: first, it presents a comprehensive framework for sign language recognition that integrates cutting-edge deep learning techniques, including CNNs, RNNs, data augmentation, and transfer learning. Second, it provides empirical evidence demonstrating the effectiveness of the proposed approach through extensive experimentation on benchmark sign language datasets.

Overall, this study aims to advance the state-of-the-art in sign language recognition, with the ultimate goal of facilitating seamless communication and inclusion for the deaf and hard of hearing community in various societal contexts.

METHOD OF DATA COLLECTION

The first step in collecting hand gestures dataset is to gather a diverse collection of hand gestures images, including a variety of actions, symbols, and sizes. Ensuring a balanced dataset, where each hand gesture class has an equal quantity of images, is significant. One way to collect a hand gesture image dataset is to use publicly available datasets as shown in Figure 1.




(b)
Yes (458 items)
(c)
I love you (410 items)
(d)
A
B
C
D
E
F
G
H
I
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y

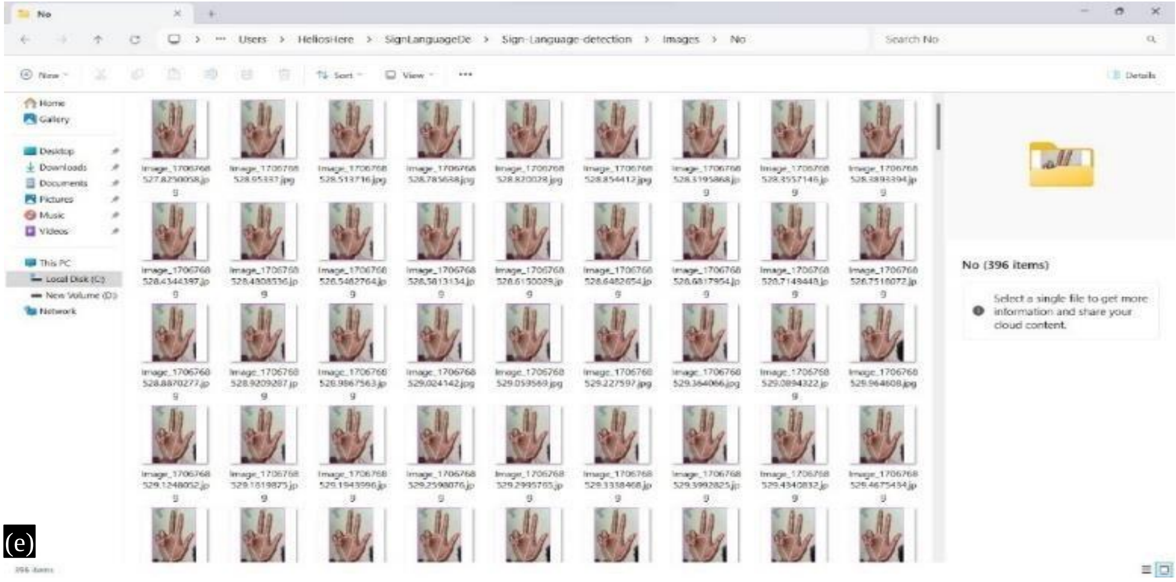

(e)

Figure 1. (a–e) Collection of hand gestures images.

DATA COLLECTION

Data collection for sign language recognition involves gathering a diverse and representative dataset comprising sign language gestures performed by individuals from different demographics, with variations in signing styles, hand shapes, facial expressions, and environmental conditions.

The dataset forms the basis for training and assessing the sign language recognition system. Below is an overview of how the data collection process was conducted:

Selection of Signers

Identify individuals proficient in sign language from different regions and cultural backgrounds to ensure diversity in signing styles and gestures. Consider including signers with varying levels of proficiency to capture a wide range of signing nuances.

Annotation and Labelling

Annotate each video or image in the dataset with corresponding sign language labels, indicating the gesture being performed. This annotation process requires collaboration with sign language experts or native signers to ensure accurate labelling of gestures.

Recording Setup

Set up recording equipment in a controlled environment to capture high-quality videos or images of sign language gestures. Ensure adequate lighting and minimal background distractions to facilitate accurate gesture recognition.

Gesture Variation

Capture a wide range of sign language gestures, including common vocabulary, phrases, and sentences, as well as non-manual components such as facial expressions and body movements. Record gestures from multiple viewpoints to account for variations in hand orientation and movement trajectories.

Data Augmentation

To increase the robustness of the model and improve its generalization performance, augment the dataset by introducing variations in lighting conditions, backgrounds, and signer characteristics. This can include flipping, rotation, scaling, and adding noise to the images or videos.

Ethical Considerations

Obtain informed consent from participants involved in data collection, ensuring that they understand the purpose of the study and how their data will be used. Respect participants' privacy and confidentiality by adhering to data protection regulations and guidelines.

Dataset Splitting

Partition the gathered dataset into training, validation, and testing subsets to effectively assess the performance of the sign language recognition system. Ensure that each set contains a balanced representation of gestures and signers to prevent bias during model training and evaluation.

Dataset Documentation

Document detailed information about the dataset, including the number of samples, gesture labels, signer demographics, recording conditions, and any pre-processing steps applied. This documentation is crucial for ensuring reproducibility and distributing the dataset among the research community.

By following these steps, researchers can collect a comprehensive and well-curated dataset for training and evaluating sign language recognition systems, enabling the development of robust and accurate models that can facilitate communication for the deaf and hard of hearing community as shown in Figure 2.

IMAGE ACQUISITION

In sign language recognition, image acquisition pertains to the procedure of obtaining visual data, which includes images or videos, depicting sign language gestures performed by individuals. This visual data serves as input to the sign language recognition system, which analyses the gestures and interprets their meaning. Here is how image acquisition is specifically applied in sign language recognition:

Camera Setup

Choose appropriate cameras capable of capturing high-resolution images or videos with sufficient frame rates to capture fast hand movements accurately. The cameras must be strategically placed to capture the signer's hand gestures clearly, ensuring there are no obstructions or distortions in the images.

Recording Environment

Create a controlled environment with adequate lighting to ensure optimal visibility of the signer's hand gestures. Avoid harsh shadows, glare, or reflections that could obscure the gestures or introduce noise into the captured images.

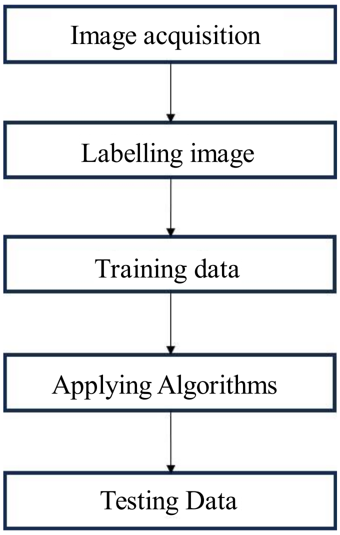


Figure 2. Sign language recognition systems Flow chart.

Signer Positioning

Instruct signers to position themselves within the camera's field of view, ensuring that their hand movements are captured effectively. Signers should be positioned facing the camera to provide a clear view of their hand gestures and facial expressions.

Variability Consideration

Capture a diverse range of sign language gestures performed by different signers to account for variations in signing styles, hand shapes, and facial expressions. Include gestures from multiple viewpoints to capture variations in hand orientation and movement trajectories.

Data Annotation

Annotate the captured images or videos with corresponding sign language labels, indicating the gestures being performed. This annotation process may involve collaboration with sign language experts or native signers to ensure accurate labelling of gestures.

Data Augmentation

Optionally, augment the dataset by introducing variations in lighting conditions, backgrounds, and signer characteristics. This can help improve the robustness of the sign language recognition system by exposing it to a wider range of visual conditions.

Data Pre-processing

Before further analysis, refine the captured images or videos to improve their quality and suitability for sign language recognition. This may include noise reduction, contrast enhancement, and image stabilization to improve the clarity of hand gestures and reduce artifacts.

Dataset Splitting

Separate the annotated dataset into training, validation, and testing subsets to train and assess the sign language recognition system effectively. Ensure that each set contains a balanced representation of gestures and signers to prevent bias during model training and evaluation.

LABELLING IMAGE

Labelling images in the context of sign language recognition involves associating each image with the corresponding sign language gesture it represents. This labelling process is essential for training supervised machine learning models to recognize and interpret sign language gestures accurately. Here is how you can label images effectively:

Annotation Tool Selection

Choose a suitable annotation tool or software that allows you to label images efficiently. There are various annotation tools available, including labelling, VGG Image Annotator, and LabelMe, among others.

Define Labelling Scheme

Establish a labelling scheme that defines the set of sign language gestures or classes to be recognized by the model. This may include common vocabulary words, phrases, or specific gestures relevant to the application domain.

Image Review

Review each image and identify the sign language gesture(s) being performed by the signer. Ensure that the gestures are clear and distinguishable in the image, with minimal occlusions or distractions.

Assign Labels

Assign the appropriate label or class to each image based on the sign language gesture it represents. Use the predefined labelling scheme to select the correct label for each gesture.

Multiple Labels

In cases where an image contains multiple sign language gestures or complex hand movements, assign multiple labels to capture all relevant gestures present in the image.

Consistency

Maintain consistency in labelling conventions across all images to ensure uniformity and accuracy in the annotated dataset. Use clear and descriptive labels that accurately represent the corresponding sign language gestures.

TRAINING DATA

Training data for sign language recognition consists of a collection of images or videos paired with corresponding labels indicating the sign language gestures they represent. This training data is used to train machine learning or deep learning models to accurately recognize and interpret sign language gestures. Here is how you can prepare training data for sign language recognition:

Data Collection

Gather a diverse and representative dataset of sign language gestures performed by individuals from different demographics, with variations in signing styles, hand shapes, and facial expressions. Ensure that the dataset covers a wide range of vocabulary words, phrases, and sentences commonly used in sign language.

Labelling

Label each image or video in the dataset with the corresponding sign language gesture(s) it represents. Use a consistent and descriptive labelling scheme to annotate the dataset accurately. Consider multiple annotators or expert review to ensure the quality and accuracy of the labels.

Data Augmentation

Optionally, augment the training data to increase its diversity and improve the robustness of the trained models. Utilize transformations such as rotation, scaling, flipping, and noise addition on the images or videos. Data augmentation aids in improving the model's ability to generalize across variations in lighting conditions, backgrounds, and signer characteristics.

Data Pre-Processing

Pre-process the images or videos in the training dataset to enhance their quality and suitability for training. This may include resizing, normalization, and cropping to ensure uniformity in image dimensions and intensity values. Pre-processing may also entail extracting pertinent features or representations from the raw data.

Dataset Splitting

Partition the annotated dataset into training, validation, and testing subsets for model training, validation, and evaluation purposes, respectively. Employ a stratified splitting approach to ensure each set contains a well-balanced representation of sign language gestures and signers. Typically, the training set encompasses the majority of the data, while the validation and testing sets constitute smaller subsets utilized for model assessment.

APPLYING ALGORITHMS

Applying algorithms in sign language recognition involves selecting appropriate machine learning or deep learning algorithms, implementing them, and training them on labelled sign language data to recognize and interpret sign language gestures effectively. Here is how you can apply algorithms in sign language recognition:

Algorithm Selection: Choose an algorithm or model architecture suitable for sign language recognition tasks. Common choices include:

Convolutional Neural Networks (CNNs)

CNNs are effective for learning spatial features from images or video frames, making them well-suited for sign language recognition tasks involving static gestures.

Recurrent Neural Networks (RNNs)

RNNs, including variants like Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU), are suitable for modelling sequential data, making them ideal for capturing temporal dependencies in sign language sequences.

Convolutional-Recurrent Neural Networks (CRNNs)

CRNNs combine the strengths of CNNs and RNNs, allowing them to capture both spatial and temporal information from sign language videos.

Transformer-based Models

Transformer architectures, such as the Transformer or its variants like BERT (Bidirectional Encoder Representations from Transformers), have shown promise in sequence modelling tasks and can be adapted for sign language recognition.

TESTING DATA

Testing data in sign language recognition refers to a separate dataset used to evaluate the performance of trained models. It consists of a collection of images or videos, each paired with the corresponding sign language gesture(s) it represents. Here is how testing data is typically utilized in sign language recognition:

Dataset Preparation

Prepare a testing dataset by selecting a subset of labelled images or videos from the overall dataset. Ensure that the testing data covers a diverse range of sign language gestures and signers, representative of the variability encountered in real-world scenarios.

Data Splitting

Separate the testing dataset from the training and validation datasets used during model development. To guarantee an impartial evaluation, it is important to avoid any overlap between the testing data and the data utilized for both model training and validation.

Evaluation Metrics

Define appropriate evaluation metrics to assess the performance of sign language recognition models on the testing data. Standard evaluation metrics comprise accuracy, precision, recall, F1-score, and confusion matrices.

Model Inference

Apply trained models to the testing dataset to perform inference on the unseen sign language gestures. Generate predictions for each sample in the testing data based on the learned representations and parameters of the model.

Performance Evaluation

Evaluate the model's predictions against the ground truth labels in the testing dataset to assess its accuracy and performance. Calculate evaluation metrics using standard formulas to quantify the model's ability to correctly recognize sign language gestures.

Error Analysis

Conduct a detailed analysis of the model's performance on the testing data to identify common errors, patterns, and areas for improvement. Examine misclassified samples to understand the reasons behind prediction failures and potential sources of ambiguity.

Cross-validation

Optionally, perform cross-validation or other validation techniques on the testing data to obtain more robust estimates of the model's performance. Partition the testing data into several subsets and assess the model's performance across these subsets to gauge its capability for generalization.

Reporting Results

Report the results of the model evaluation on the testing data in a clear and concise manner. Present performance metrics, including accuracy and other relevant evaluation metrics, along with any observations or insights gained from the analysis.

Model Evaluation: The given Figure 3 represents the modal evaluation details.

LITERATURE REVIEW

Deaf Mute Communication Interpreter-A Review [1]

The objective of this paper is to discuss the different existing approaches to deaf-mute communication interpreter systems. Deaf-mute communication methodologies are broadly categorized into two groups: Wearable Communication Devices and Online Learning Systems. Within the Wearable communication method, there are subcategories including Glove-based systems, Keypad methods, and Handicom Touch-screen devices.

Each of the aforementioned three subdivided methods incorporates a range of components including sensors, accelerometers, a compatible microcontroller, a text-to-speech conversion module, a keypad, and a touch-screen. The necessity for an external device to interpret messages between deaf-mute and non-deaf-mute individuals can be circumvented by the second method, namely the online learning system. The Online Learning System encompasses various approaches, including the SLIM module, TESSA, Wi-See Technology, SWI_PELE System, and Web-Sign Technology.

An Efficient Framework for Indian Sign Language Recognition Using Wavelet Transform [2]

The proposed ISLR system is regarded as a pattern recognition approach comprising two key modules: feature extraction and classification. Sign language recognition is accomplished through the combined utilization of Discrete Wavelet Transform (DWT) for feature extraction and nearest neighbor classification. Experimental findings demonstrate that the proposed hand gesture recognition system achieves a peak classification accuracy of 99.23% when employing the cosine distance classifier.

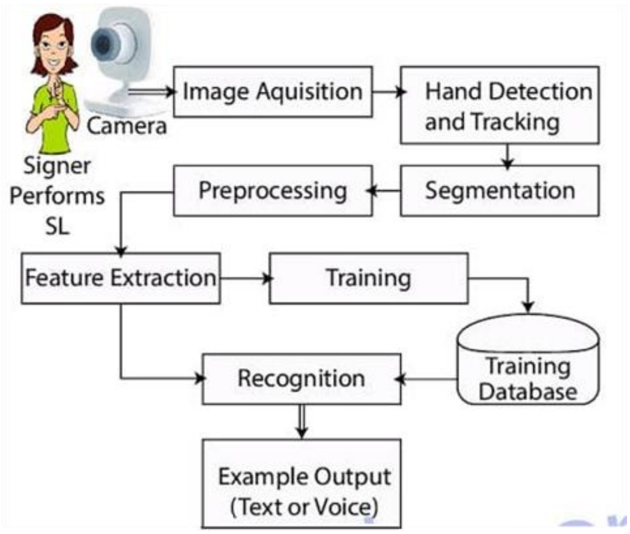


Figure 3. Modal evaluation Chart.

Hand Gesture Recognition Using PCA [3]

The authors of this paper introduced a methodology for database-driven hand gesture recognition, employing a skin color model approach and thresholding technique, combined with an efficient template matching method applicable to human robotics and similar applications. Initially, the hand region is delineated through the application of a skin color model in the YCbCr color space. Subsequently, thresholding is utilized to distinguish between foreground and background elements. Finally, a template-based matching method is devised using Principal Component Analysis (PCA) for recognition purposes.

Hand Gesture Recognition System for Dumb People [4]

The authors introduced a static hand gesture recognition system employing digital image processing. The SIFT algorithm is utilized to generate the hand gesture feature vector, with SIFT features computed at the edges to ensure invariance to scaling, rotation, and the addition of noise.

An Automated System for Sign Language Recognition [5]

This paper introduces a technique for automatically recognizing signs based on shape-based features. Otsu’s thresholding algorithm is employed to segment the hand region from the images, selecting an optimal threshold to minimize the within-class variance of thresholded black and white pixels. The features of the segmented hand region are computed using Hu’s invariant moments, which are then input into an Artificial Neural Network for classification. The system's performance is assessed based on Accuracy, Sensitivity, and Specificity.

Hand Gesture Recognition for Sign Language Recognition: A Review [6]

The authors discussed different approaches to hand gesture and sign language recognition previously proposed by various researchers. Sign language serves as the sole means of communication for individuals who are deaf or mute. Through sign language, these individuals convey their emotions and thoughts to others.

Design Issue and Proposed Implementation of Communication Aid for Deaf and Dumb People [7]

In this paper, the author introduced a system aimed at facilitating communication between deaf and mute individuals using the Indian Sign Language (ISL) and the general population, where hand gestures are converted into corresponding text messages. The primary goal is to develop an algorithm capable of converting dynamic gestures into text in real-time. Once testing is completed, the system will be deployed on the Android platform and made accessible as a smartphone and tablet application.

Real Time Detection and Recognition of Indian and American Sign Language Using Sift [8]

The author introduced a real-time vision-based system for hand gesture recognition in human-computer interaction across various applications. This system is capable of swiftly recognizing 35 different hand gestures from both Indian Sign Language (ISL) and American Sign Language (ASL) with high accuracy. An RGB-to-GRAY segmentation technique was employed to reduce the likelihood of false detections. Additionally, the authors proposed an improved method for Scale Invariant Feature Transform (SIFT) and utilized it to extract features. The system was modeled using MATLAB and includes a graphical user interface (GUI) for designing an efficient and user-friendly hand gesture recognition system.

A Review on Feature Extraction for Indian and American Sign Language [9]

The paper discussed recent advancements in research and development concerning sign language, focusing on manual communication and body language. Sign language recognition systems typically involve three main steps: preprocessing, feature extraction, and classification. Various classification methods utilized for recognition include Neural Network (NN), Support Vector Machine (SVM), Hidden Markov Models (HMM), Scale Invariant Feature Transform (SIFT), among others.

An Application Suite for Deaf and Dumb [10]

The author introduced an application aimed at assisting individuals who are deaf and mute to communicate with others using sign language. The primary feature of this system is the real-time conversion of gestures to text. The processing stages include extracting gestures, matching gestures, and converting them to speech. Gesture extraction utilizes a variety of image processing techniques such as histogram matching, computing bounding boxes, segmenting skin color, and region growing. Methods applicable for gesture matching include matching feature points and correlation-based matching. Additionally, other features of the application include verbalizing text and converting text to gestures.

RESULT

The results of a sign language recognition system implementation typically include the performance metrics obtained from testing the system on a separate dataset [11, 12]. These metrics offer valuable information regarding the system's accuracy, precision, recall, and other pertinent indicators of effectiveness. Here is how you can present the results of a sign language recognition system implementation:

Accuracy

Report the overall accuracy of the system in correctly recognizing sign language gestures. Accuracy is calculated as the proportion of correctly classified gestures to the total number of gestures in the testing dataset.

Precision and Recall

Provide precision and recall scores for each sign language gesture class. Precision quantifies the ratio of accurately classified instances of a class to the overall instances classified as that class. Meanwhile, recall gauges the ratio of accurately classified instances of a class to the total instances of that class present in the dataset.

Examples

Include examples of correctly and incorrectly recognized sign language gestures to illustrate the system's performance qualitatively. This helps stakeholders understand the strengths and limitations of the system in real-world scenarios as shown in Figure 4.

Comparison Between Existing Systems with Proposed System

The given Table 1 represents the comparison between existing systems and proposed systems.

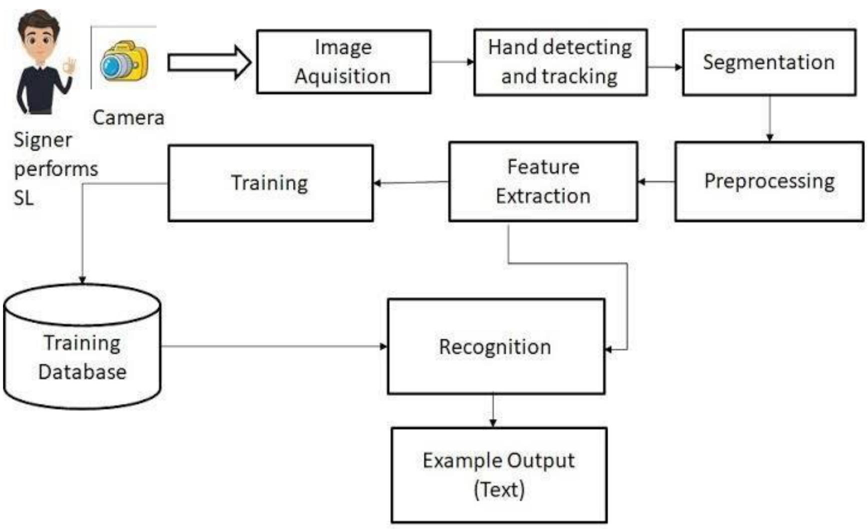


Figure 4. Recognized sign language gestures performance qualitatively.

Table 1. Comparison between existing systems with proposed system.

Parameters	Existing System	Proposed System
Real-Time	Less Accuracy	More Accuracy
Keras	No	Yes
Yolov5	No	Yes
Data analyzing	Yes	Yes
Data Acquisition	Yes	Yes
CN Network	No	Yes
Data Security	Yes	Yes

IMPLEMENTATION

Implementing a sign language recognition system involves translating research findings and algorithms into working software or hardware solutions capable of recognizing and interpreting sign language gestures [13–17].

Here is a broad outline of the implementation process.

Algorithm Selection: After conducting a literature review and defining your research goals, select suitable algorithms or models for sign language recognition. This could involve choosing deep learning frameworks such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), or hybrid architectures like convolutional-recurrent neural networks (CRNNs).

Data Collection and Preparation

Gather a dataset of sign language gestures, ensuring diversity in signing styles, hand shapes, and environmental conditions. Assign labels to the dataset according to the corresponding sign language gestures. Divide the dataset into training, validation, and testing subsets.

Data Pre-processing

Pre-process the images or videos in the dataset by resizing, normalization, and augmentation to improve model training and performance.

Model Development

Utilize deep learning frameworks like TensorFlow, PyTorch, or Keras to implement the chosen algorithms. Specify the model architecture, loss function, and optimization strategy. Train the model on the training dataset using the selected algorithms and optimization methods.

Model Evaluation

Assess the trained model's performance on the validation dataset to refine hyperparameters and enhance model effectiveness. Evaluate the model's accuracy, precision, recall, and other pertinent metrics using established evaluation techniques.

Testing and Validation

Evaluate the final trained model on the testing dataset to assess its generalization performance and validate its effectiveness in recognizing sign language gestures.

Integration and Deployment

Integrate the trained model into a software application, mobile app, or embedded system capable of capturing and recognizing sign language gestures in real-time. Ensure seamless integration with user interfaces and accessibility features to facilitate interaction with users.

User Testing and Feedback

Organize user testing sessions with individuals from the deaf and hard of hearing community to collect feedback on the usability, accuracy, and efficacy of the sign language recognition system. Integrate user input to enhance and optimize the system's performance and user interface [18–20].

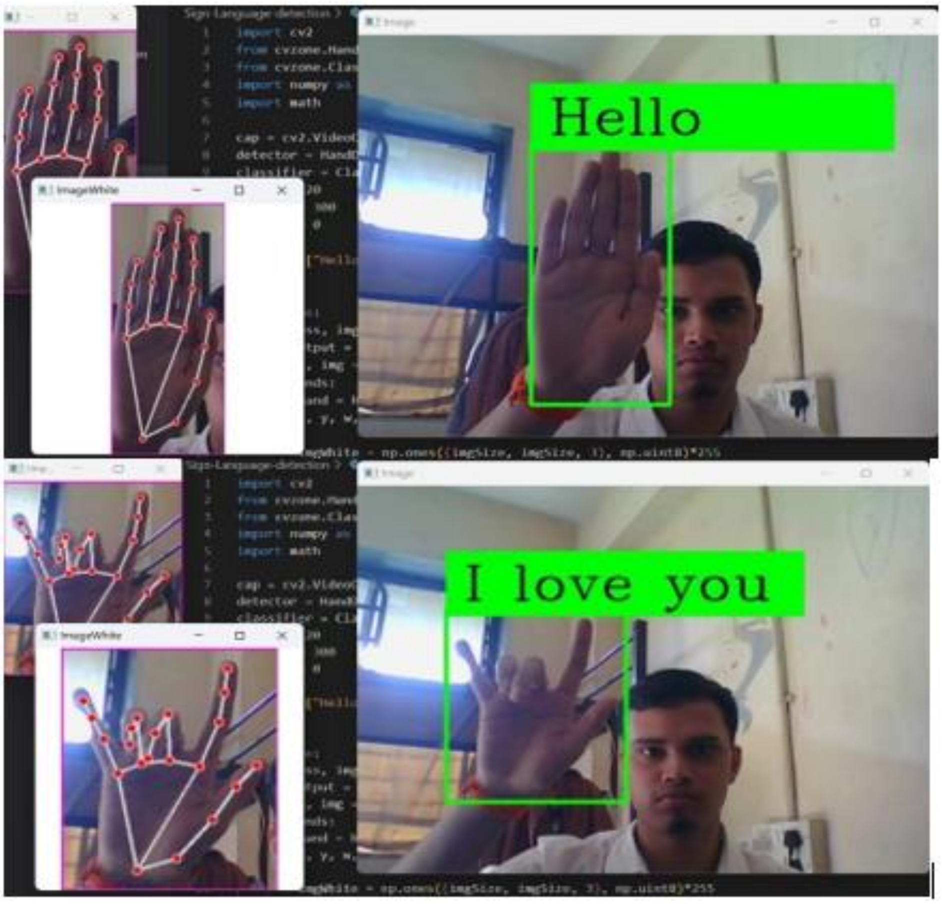


Figure 5. Result of the proposed model.

Continuous Improvement

Monitor the performance of the deployed system and collect additional data to address any shortcomings or limitations. Explore advanced techniques, such as transfer learning, multi-modal fusion, or attention mechanisms, to enhance the system's capabilities and robustness.

Documentation and Maintenance

Document the implementation process, including algorithms used, data pre-processing steps, model architecture, and evaluation results. Maintain the system by updating algorithms, addressing bug fixes, and incorporating new research findings to ensure continued effectiveness and relevance.

RESULT

The given Figure 5 is the resultant of the proposed model.

CONCLUSION

We are going to implement sign language recognition using concepts of artificial intelligence and image processing. After our applications is implemented, it would benefit the NGOs and various organizations which involve people with special needs as it will be a real-time system for hand gesture identification. The proposed system can also be further carried forward to implement Sign Language recognition which is more complex as it involves two hand gestures. Also, the response time of the system can be reducing with better camera and graphics support.

REFERENCES

1. Oudah M, Al-Naji A, Chahl J. Hand gesture recognition based on computer vision: a review of techniques. J Imaging. 2020 Jul 23; 6(8): 73.

2. James TG, Varnes JR, Sullivan MK, Cheong J, Pearson TA, Yurasek AM, Miller MD, McKee MM. Conceptual model of emergency department utilization among deaf and hard-of-hearing patients: a critical review. Int J Environ Res Public Health. 2021 Dec 7; 18(24): 12901.
3. Ewe EL, Lee CP, Kwek LC, Lim KM. Hand gesture recognition via lightweight VGG16 and ensemble classifier. Appl Sci. 2022 Jul 29; 12(15): 7643.
4. Foltz A, Cuffin H, Shank C. Deaf-Accessible Parenting Classes: Insights from Deaf Parents in North Wales. Societies. 2022 Jun 30; 12(4): 99.
5. Chong TW, Lee BG. American sign language recognition using leap motion controller with machine learning approach. Sensors. 2018 Oct 19; 18(10): 3554.
6. Vaitkevičius A, Taroza M, Blažauskas T, Damaševičius R, Maskeliūnas R, Woźniak M. Recognition of American sign language gestures in a virtual reality using leap motion. Appl Sci. 2019 Jan 28; 9(3): 445.
7. Su R, Chen X, Cao S, Zhang X. Random forest-based recognition of isolated sign language subwords using data from accelerometers and surface electromyographic sensors. Sensors. 2016 Jan 14; 16(1): 100.
8. Amin MS, Rizvi ST, Hossain MM. A comparative review on applications of different sensors for sign language recognition. J Imaging. 2022 Apr 2; 8(4): 98.
9. Lee BG, Chong TW, Chung WY. Sensor fusion of motion-based sign language interpretation with deep learning. Sensors. 2020 Nov 2; 20(21): 6256.
10. Yu H, Zheng D, Liu Y, Chen S, Wang X, Peng W. Low-Cost Self-Calibration Data Glove Based on Space-Division Multiplexed Flexible Optical Fiber Sensor. Polymers. 2022 Sep 20; 14(19): 3935.
11. Ahmed MA, Zaidan BB, Zaidan AA, Salih MM, Lakulu MM. A review on systems-based sensory gloves for sign language recognition state of the art between 2007 and 2017. Sensors. 2018 Jul 9; 18(7): 2208.
12. Mummadi CK, Philips Peter Leo F, Deep Verma K, Kasireddy S, Scholl PM, Kempfle J, Van Laerhoven K. Real-time and embedded detection of hand gestures with an IMU-based glove. Informatics. 2018 Jun 11; 5(2): 28.
13. Bird JJ, Ekárt A, Faria DR. British sign language recognition via late fusion of computer vision and leap motion with transfer learning to American sign language. Sensors. 2020 Sep 9; 20(18): 5151.
14. Santos DG, Fernandes BJ, Bezerra BL. HAGR-D: a novel approach for gesture recognition with depth maps. Sensors. 2015 Nov 12; 15(11): 28646–64.
15. Xia K, Lu W, Fan H, Zhao Q. A sign language recognition system applied to deaf-mute medical consultation. Sensors. 2022 Nov 24; 22(23): 9107.
16. Zhu Y, Zhang J, Zhang Z, Clepper G, Jia J, Liu W. Designing an Interactive Communication Assistance System for Hearing-Impaired College Students Based on Gesture Recognition and Representation. Future Internet. 2022 Jun 29; 14(7): 198.
17. Pagliari D, Pinto L. Calibration of kinect for xbox one and comparison between the two generations of Microsoft sensors. Sensors. 2015 Oct 30; 15(11): 27569–89.
18. Guzsvinecz T, Szucs V, Sik-Lanyi C. Suitability of the Kinect sensor and Leap Motion controller—A literature review. Sensors. 2019 Mar 2; 19(5): 1072.
19. Mujahid A, Awan MJ, Yasin A, Mohammed MA, Damaševičius R, Maskeliūnas R, Abdulkareem KH. Real-time hand gesture recognition based on deep learning YOLOv3 model. Appl Sci. 2021 May 2; 11(9): 4164.
20. Tscholl DW, Rössler J, Said S, Kaserer A, Spahn DR, Nöthiger CB. Situation awareness-oriented patient monitoring with visual patient technology: a qualitative review of the primary research. Sensors. 2020 Apr 9; 20(7): 2112.

