International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 01 | Jan 2026 www.irjet.net p-ISSN: 2395-0072

CNN based Multilingual Sign Language Recognition System

A. Dhanush[1], G. Amulya[2], D. Madhu[3], S. Jakeerhussen[4] , Madhumathi Thakur[5]

[1234]*Department of Information Technology, TKR College of Engineering and Technology, Telangana, India*
[5] *Assistant Professor, dept. of Information Technology*

Abstract - *While contemporary assistive technologies have made strides in bridging the gap between sign language users and the hearing world, most systems are strictly scoped to a single national sign language. This narrow focus overlooks a critical, often neglected communication barrier: the linguistic divide among sign language users of different nationalities and ethnicities. As sign languages like* ***Indian Sign Language (ISL)****,* ***American Sign Language (ASL)****, and* ***Russian Sign Language (RSL)*** *are distinct and mutually unintelligible, there is a pressing need for a unified environment that facilitates cross-lingual interaction. This project presents a* ***proof-of-concept*** *for an integrated sign language translation system designed to bridge this multi-ethnic divide. Unlike conventional models, this system incorporates a multi-lingual framework featuring* ***ISL, ASL, and RSL*** *within a single interface. By providing a platform where diverse sign languages are recognized and translated into a common medium, the project demonstrates that technology can facilitate direct communication between sign language users from different parts of the world.*

KeyWords:Proof-of-Concept,Cross-lingual communication, Indian Sign Language (ISL), American Sign Language (ASL), Russian Sign Language (RSL), Linguistic Diversity, Communication Barriers, Inclusive Design, Multi-ethnic Systems.

1. INTRODUCTION

Communication is the fundamental bridge that connects individuals, yet for the deaf and hard-of-hearing community, this bridge is often broken by the language barrier. Sign language is the primary mode of communication for millions of people worldwide, but the vast majority of the hearing population remains unfamiliar with it. This creates a significant gap in social inclusion, and involvement with the larger population.

While there have been advancements in sign language recognition, most existing solutions focus on a single language, typically American Sign Language (ASL). However, sign languages are not universal; they vary significantly by region. For instance, Indian Sign Language (ISL) uses two-handed gestures for many alphabets, whereas ASL is predominantly single-handed. Russian Sign Language (RSL) has its own unique syntax and gestures.

This project, the **Multilingual Sign Language Recognizer**, addresses this limitation by developing a deep learning-based system capable of interpreting gestures from three distinct sign languages: **ASL, ISL, and RSL**. By leveraging computer vision and neural networks, this system translates hand gestures into text in real-time, facilitating smoother communication across different linguistic regions.

1.1 Objectives

The primary objective is to develop a multilingual system by creating a unified platform that supports multiple sign languages (ASL, ISL, and RSL) within a single interface, thereby removing the need for separate applications for different regions. The project aims to implement real-time hand tracking using a robust mechanism based on Media Pipe, capable of detecting hand landmarks with high precision even in varying lighting conditions and backgrounds. Additionally, the system focuses on accurate classification by training and deploying specialized Convolutional Neural Network (CNN) models for each language to classify static hand gestures, such as alphabets and numbers, into their corresponding text output. Finally, the goal is to provide a user-friendly interface through a simple, intuitive Graphical User Interface (GUI) that allows users to easily switch between languages and view the translated text instantly.

1.2 Problem Statement

The problem of communication between the hearing and the hearing-impaired communities is a significant societal challenge. A primary issue is the language barrier, as the majority of the hearing population does not understand sign language, leading to social isolation for the deaf community. Furthermore, fragmentation in current technology is evident, as existing automated solutions focus almost exclusively on American Sign Language (ASL), resulting in a distinct lack of accessible tools for Indian Sign Language (ISL) and Russian Sign Language (RSL). Technical limitations also hinder progress; many traditional recognition systems rely on intrusive or expensive equipment like colored gloves or depth sensors (such as Kinect). Consequently, there is a critical need for a solution that operates on standard consumer hardware, specifically webcams, without extra equipment.

Lastly, scalability poses a problem, as current systems often struggle to accommodate multiple languages due to the

complexity of managing different gesture sets such as single-handed versus double-handed signs, within a single architecture.

2. Proposed System

The proposed **Multilingual Sign Language Recognizer** is a comprehensive software solution that uses a standard webcam to recognize gestures from ASL, ISL, and RSL.
The system captures video frames, processes them to isolate the hands, and uses a specialized Deep Learning model to predict the gesture. It employs **MediaPipe** for efficient hand tracking, ensuring that the background is ignored and only the hand geometry is analyzed. The system includes a language selector, allowing the user to switch models dynamically.

2.1 System Architecture

The proposed system functions as a continuous, real-time processing loop that begins by initializing the webcam and loading dedicated deep learning models for American, Indian, and Russian Sign Languages. Once operational, the system captures individual video frames and checks for the presence of a hand; if a hand is detected, the architecture utilizes MediaPipe to extract specific landmarks, which are then used to define a bounding box and crop the region of interest into a standardized 224x224 pixel image. This preprocessed input is subsequently routed to a specific Convolutional Neural Network (CNN) corresponding to the user's selected language to ensure accurate classification. The resulting prediction is immediately generated and overlaid as text onto the live video feed, providing instant translation before the system loops back to process the next frame or terminates upon a stop signal.

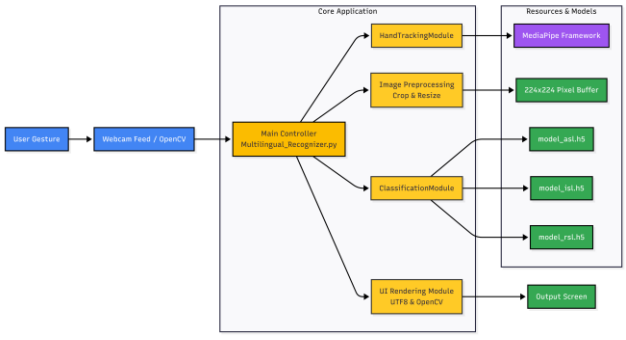

Fig -1: System Architecture of the mulit-sign language recognition system

2.2 Convolutional Neural Network (CNN) Design

The core intelligence of the system relies on specialized Convolutional Neural Networks (CNN) trained for each language. The architecture is designed with a specific sequence of layers to optimize feature extraction and classification. It begins with Convolutional Layers (Conv2D) to extract spatial features from the hand images, followed by Pooling Layers (MaxPooling2D) that reduce the dimensionality of the feature maps and computational load. These are succeeded by Flatten layers that convert the 2D arrays into 1D vector, which are then passed through fully connected Dense layers for classification. To ensure the model generalizes well to new, unseen gestures, a Dropout layer of 0.5 is included to prevent overfitting during the training phase.

Sample paragraph Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, sc, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper.

2.3 Recognition Workflow

The recognition process begins when the webcam captures a frame. Upon system startup, the **Initialization Phase** triggers the webcam and loads the pre-trained Convolutional Neural Network (CNN) models for ASL, ISL, and RSL into memory. The system then enters its main execution loop, where it captures a raw video frame.

For every frame, a conditional check is performed to determine if a hand is present. If the hand detection algorithm (powered by Media Pipe) returns a negative result, the system discards the frame and loops back to capture the next one. However, if a hand is detected, the **landmark extraction process** begins, identifying key points on the hand to calculate a bounding box. This Region of Interest (ROI) is then cropped and resized to a standard 224x224 pixel format to match the input layer requirements of the CNNs.

A critical routing step follows, where the system checks the user's currently selected language. Based on this selection, the pre-processed image is routed to the specific model (e.g., passing to the ASL CNN Model or RSL CNN Model). The selected model generates a prediction index, which is mapped to a character label. Finally, this text is overlaid onto the live video feed, providing instant feedback. The loop continues until a "Stop" signal is received, at which point the resources are released and the program terminates.

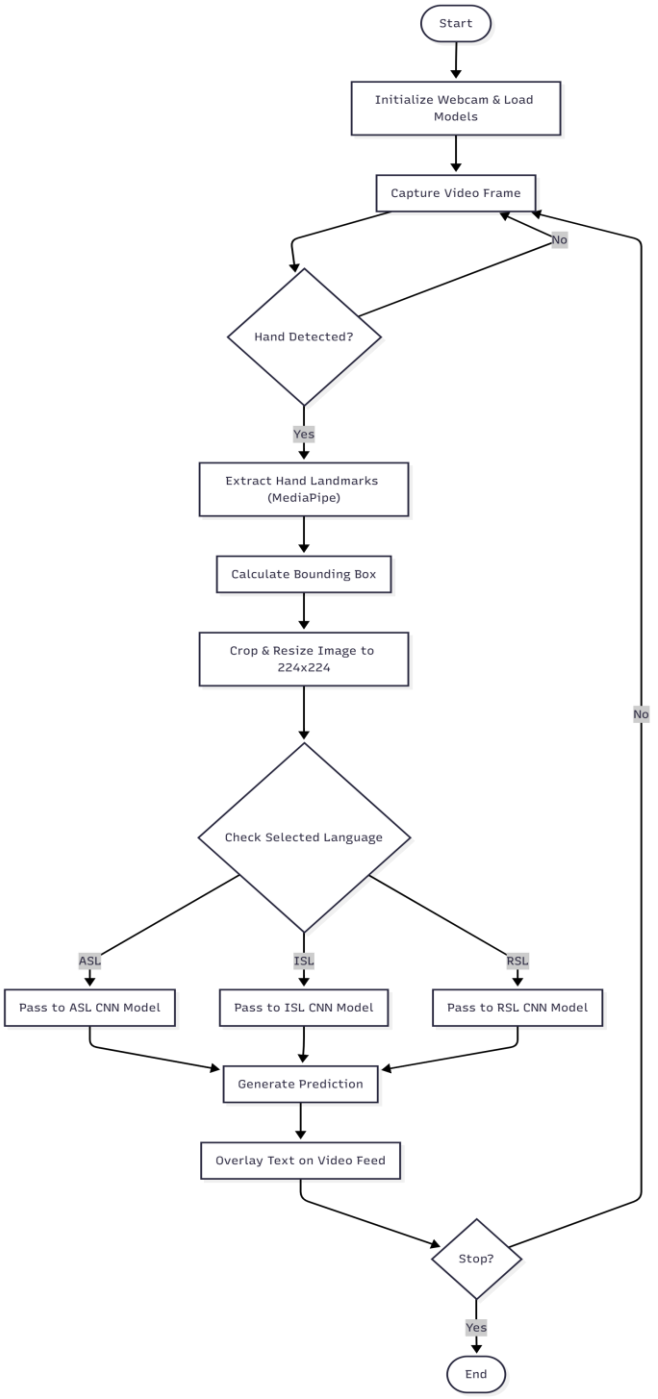

Fig -2: Process Workflow of the proposed Sign Language recognition system

3. IMPLEMENTATION DETAILS

The implementation of the proposed multilingual system is carried out using a combination of Python, Computer Vision libraries, and Deep Learning frameworks. The implementation is divided into three main modules: Data Collection, Model Training, and Real-Time Recognition.

3.1 Tracking Layer Implementation

Tracking layer is implemented using Google's Media Pipe framework. Specifically, the HandTrackingModule.py wraps Media Pipe's functionality to track 21 3D landmarks of the hand. This module provides the critical bounding box coordinates required to crop the hand from the video frame effectively, ensuring that the subsequent classification model receives only relevant data. The use of skeletal tracking rather than raw pixel processing allows the system to remain robust against complex backgrounds.

3.2 Preprocessing and Logic Layer

The logic layer is handled by Python scripts that manage the data flow. The ClassificationModule.py is responsible for loading the pre-trained Keras models (h5 files) and performing inference.
Before inference, the image undergoes strict preprocessing where the hand region is isolated with a margin to ensure the full gesture is captured, resized to 224x224 pixels, and normalized to a range of 0 to 1 to improve model convergence.

3.3 Frontend and Interface Implementation

The frontend is developed using OpenCV for rendering the visual output. The interface allows users to view the live feed, with bounding boxes and predicted text overlaid directly on the screen. For Russian Sign Language (RSL), a specific challenge arises as standard OpenCV functions do not support Cyrillic characters well. To address this, a UTF8ClassificationModule was implemented using the Python Imaging Library (PIL) to render Unicode text correctly before converting it back to an OpenCV array for display.

3.4 Data Collection and Model Storage

To ensure high accuracy, custom datasets were collected using the datacollection.py script. This script automates the process of capturing and cropping hand gestures, saving them into labeled folders for training. The trained models are stored as model_asl.h5, model_isl.h5, and model_rsl.h5, which are loaded into memory during system initialization. This modular storage allows for easy updates or the addition of new languages without rewriting the core application logic.

3.5 System Robustness and Usability

Robustness is a critical aspect of the system implementation. By using MediaPipe landmarks, the system handles slight variations in lighting and hand orientation effectively. The interface is designed to be minimal and intuitive, requiring no technical knowledge to operate. Additionally, the system

provides real-time feedback, processing frames at a sufficient rate to ensure a smooth user experience.

4. RESULTS AND PERFORMANCE ANALYSIS

This section discusses the results obtained after implementing and testing the proposed Multilingual Sign Language Recognizer. The system was tested under various conditions to validate the correctness of gesture classification and the responsiveness of the interface.

4.1 Classification Accuracy and Validation

The classification performance was evaluated for each language module to ensure distinct accuracy across the linguistic spectrum. The ASL model demonstrates high accuracy, exceeding 95% for distinct static signs like 'A', 'B', 'C', 'L', and 'V'. For ISL, the system successfully detects double-handed gestures, a unique feature of the language, provided the user is positioned slightly further back to capture both hands within the frame.
Furthermore, the RSL model successfully identifies Cyrillic gestures, distinguishing the system from standard recognizers that fail to process non-Latin scripts.

4.2 Real-Time Performance Analysis

The performance of the system was evaluated based on latency and frame rate. The system responds with low latency, updating the character instantly as the user changes their hand shape. The lightweight nature of the MediaPipe framework allows the system to run effectively on standard CPUs without requiring high-end GPUs, ensuring it meets the non-functional requirement of processing frames at 15-20 FPS for a smooth experience.

4.3 Computational Efficiency and Hardware Independence

The cost and resource analysis highlights the economic benefits of the proposed solution. Unlike sensor-based systems that rely on expensive hardware like Microsoft Kinect or Cyber Gloves, this system requires only a standard USB webcam or integrated laptop camera. This drastic reduction in hardware dependency makes the technology accessible to a wider demographic. Furthermore, the modular architecture ensures that adding new languages does not exponentially increase computational overhead, as only the relevant model is loaded during execution.

5. CONCLUSION

The proposed Multilingual Sign Language Recognizer demonstrates an effective solution for bridging the communication gap across different linguistic regions. By integrating ASL, ISL, and RSL into a single platform, the project moves beyond the fragmentation seen in existing systems. The implementation validates that computer vision techniques, specifically Media Pipe and CNNs, can effectively interpret diverse gesture sets—from single-handed ASL to double-handed ISL and Cyrillic RSL—without specialized hardware. This proof-of-concept serves as a foundation for a more inclusive global communication tool.

6. FUTURE WORK

Although the current implementation successfully demonstrates multilingual recognition, several enhancements can be considered in future work. Future iterations could move from static alphabet recognition to dynamic words and sentence recognition to enable fluid conversation. Additionally, developing a mobile application would increase portability and accessibility for users on the go. There is also potential for integrating Large Language Models (LLMs) to refine the recognized text and provide context-aware sentence construction, as suggested by recent research. Finally, developing a multilingual output engine that translates recognized gestures into localized spoken text would further bridge the gap between deaf and hearing communities.

ACKNOWLEDGEMENT

The authors would like to express their sincere gratitude to Mrs. T. Madhumathi, Assistant Professor, for her constant guidance and moral support throughout the project. The authors also thank the faculty members of the Department of Information Technology, TKR College of Engineering and Technology, for their support.

REFERENCES

[1] MediaPipe & CNN for ASL: "Enhancing Sign Language Detection through Mediapipe and Convolutional Neural Networks (CNN)," ResearchGate.

[2] ISL Recognition: "Indian Sign Language Recognition using Convolutional Neural Network," ITM Web of Conferences.

[3] RSL Datasets: "Slovo: Russian Sign Language Dataset," SberLabs.

[4] General Methodology: "Real-Time Sign Language Recognition Using MediaPipe and Deep Learning Approaches," JETIR.

[5] Kulkarni, A., Kariyal, A. V., Dhanush, V., & Singh, P. N. (2021). Speech to Indian sign language translator. Atlantis Highlights in Computer Sciences/Atlantis Highlights in Computer Sciences.

[6] Najib, F. M. (2024). Sign language interpretation using machine learning and artificial intelligence. Neural Computing and Applications.

[7] Indian sign language translator. (2022, December 1). IEEE Conference Publication | IEEE Xplore.

[8] Papastratis, I., Chatzikonstantinou, C., Konstantinidis, D., Dimitropoulos, K., & Daras, P. (2021). Artificial intelligence technologies for sign language. Sensors, 21(17), 5843.