# Real-Time Sign Language Translation Using Machine Learning

Muhammad Tayyab
*Department of Computer Science*
*University of Engineering and Technology*
Lahore, Pakistan
mu.tayyab001@gmail.com

***Abstract*—Abstract—Sign language serves as a critical medium of com- munication for the hearing and speech-impaired community. However, its widespread use is limited by the lack of real-time translation systems that can bridge the communication gap between sign language users and non-users. This study presents a Real-Time Sign Language Translation System leveraging machine learning techniques and computer vision to recognize and translate sign language gestures into textual or spoken language. The proposed system aims to provide a seamless, accurate, and real-time communication experience, enhancing inclusivity and accessibility. The system employs seven models—Random Forest, SVM, KNN, Decision Tree, Na¨ıve Bayes, XGB Classifier and Logistic Regression — and achieves a high accuracy rate of 99.50 Index Terms—Sign Language Translation, Real-Time Systems, Computer Vision, Machine Learning, Accessibility, Communication Technology**

***Index Terms*—Sign Language Translation, Real-Time Systems, Computer Vision, Machine Learning, Accessibility, Communication Technology**

## I. INTRODUCTION

Technology is rapidly advancing, and the lives of humans are greatly changing. A powerful tool that has come out to assist people in their daily life is Artificial Intelligence (AI). Among these AI innovations, Generative Pre-trained Transformers (GPT) and Large Language Models (LLMs) have demonstrated remarkable capabilities in processing extensive textual data and generating human-like responses. In this respect, several fields are being revolutionized leading to a new era of progress. But these models mainly work with textual languages only as far communication is concerned. This makes it necessary to find innovative ways for users of visual languages like sign language to interact with GPT models by live webcam gestures. Current methods for recognizing sign language, whether it is discrete or continuous, tend to focus on offline detection where the recorded gesture videos are analyzed in controlled environments. In this paper, we present a novel approach aimed at the development of real-time sign language recognition models interpreting visual language gestures into understandable text messages. The aim here is to link the hearing challenged society with the latest developments in GPT models so that they can both communicate effectively. Sign language is one of the most essential visual languages to the hearing-impaired community. As a visual and gestural language, it consists of several hand shapes, body movements, and facial expressions. Like spoken languages, it has its grammar and vocabulary, too. Over the last few years, there have been several studies in sign language processing [1]. Two major fields in this area are sign language production and sign language recognition. This paper focuses on sign language recognition. Sign language is composed of various hand gestures and body movements, which categorizes sign language recognition into a gesture recognition problem. Gesture recognition has taken human-computer interaction to another level where virtual environments have emerged. Most studies on gesture recognition have been sensor-based, glove- based, computer vision-based, or hybrid. Devices are attached with several sensors in a sensor-based approach that collects and transmits the signs data. These methods are primarily cumbersome due to the heavy sensors and devices like gloves. Computer vision-based methods record 2D images of hand and body motions by recognizing the position, trajectory, and movement of the gestures captured in the image frames. Sign language recognition can be broken down into two types: continuous sign language recognition (CSLR) and isolated sign language recognition (ISLR). CSLR refers to the sentence level sign language recognition, while ISLR refers to the word level [2]. ISLR is very similar to gesture recognition tasks done using a computer vision-based approach. Recent advances in pose estimation have mitigated the complexities of gesture recognition tasks to be background-agnostic. We can extract pose key points from RGB videos and store them for model training. These key points, because of their sequential nature, have found their utility in RNN, CNN, and LSTM models [3] [4]. ISLR depends quite a lot on hand shapes and movements, which require more distinctive hand features. Although many depend on the hand shape, pose- based approaches still face problems recognizing different hand shapes. This is inconsistent with the identification of minute movement of hand key points. Methods based on poses also find it challenging to obtain dense information about hand 82 10.21437/AVSEC.2024-18 shapes. In contrast, methods that depend on COCO's structure for topology in pose estimation also find it difficult to recognize different hand shapes. To address this, previous works have applied normalization to the key points or implemented additional models separately trained on the hands [5] [6]. The challenge of addressing this factor increases further since noisy key points result from the failure of hand detection by the pose estimation model.

The Mediapipe pipeline of pose estimation helps solve this problem by extending the basic topology of pose estimation to capture detailed hand data [7]. The Pose Estimation Mediapipe pipeline can accurately detect hand and upper body poses in video frames.

## II. Related Work

In the study conducted by R Sreemathy et al[1], a novel approach for continuous word-level sign language recognition has been proposed. The research involves the classification of 80 sign language terms utilizing You Only Look Once version 4 (YOLOv4) and Support Vector Machine (SVM) with media-pipe. The study introduces a Python-based system that utilizes a self-generated image dataset containing 80 static signs, amounting to 676 images. The chosen methodology, YOLOv4 and SVM with media pipe, achieves impressive accuracy rates of 98.8

Shivanarayna Dhulipala et al[2] present an innovative research endeavour aimed at bridging the communication gap between speech-impaired and non-speech-impaired individuals. The focus is on creating an efficient deep learning model capable of predicting British sign language. The researchers work with a pre-processed input dataset sourced from Kaggle, and employ Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) architectures. Notably, the CNN model achieves an accuracy rate of 97.4

B. Natarajan et al[4] contribute to the field by offering a comprehensive framework capable of managing sign language recognition, translation, and video generation tasks in real-time scenarios. This framework combines Neural Machine Translation (NMT), MediaPipe, and a Dynamic Generative Adversarial Network (GAN) model. Through experimentation, the researchers achieve a notable 95

Kurre et al[5] embark on a study focused on providing a robust recognition model for Indian Sign Language (ISL). By comparing feature detectors and descriptors, the researchers implement the Bag of Visual Words technique along with various models to predict ISL in real-time. Impressively, the proposed CNN achieves a perfect 100

Uyyala et al[6] delve into the realm of sign language recognition by proposing a unique convolutional neural network (CNN) that automatically extracts spatial-temporal characteristics from raw video streams. Their study focuses on American hand gestures, utilizing a Convolutional Neural Network architecture. The model achieves an impressive accuracy rate of 100

Sang-Ki Ko et al[7] present a novel neural network model that leverages human keypoint locations retrieved from hands, face, and other body parts to translate sign videos into phrases in natural English. Using the KETI sign language dataset, the researchers employ Recurrent Neural Network Models to achieve an accuracy rate of 93.28

Akshit J Dhruv et al[8] aim to address the communication needs of the deaf by creating a real-time sign language converter. The proposed system translates voice into text, which is then further translated into universal sign language for better comprehension. Employing long short-term memory (LSTM) techniques, the model achieves a high accuracy of 100

Aldhahri et al[9] embark on recognizing Arabic alphabet signs to facilitate communication for the deaf and hearing impaired. The researchers utilize the Arabic Alphabets Sign Language Dataset (ArASL2018) and deploy convolutional neural networks to achieve a commendable recognition accuracy of 94.46

Utpal Nandi et al[10] contribute by developing a convolutional neural network-based fingerspelling recognition system for the Indian sign language. They employ techniques such as data augmentation, batch normalization, dropout, stochastic pooling, and diffGrad optimizer to achieve an impressive accuracy of 99.64

Jyotishman Bora et al[11] aim to establish a technical method for understanding Assamese Sign Language using the MediaPipe framework and neural network models. Through their work, they achieve a remarkable accuracy of 99

Navroz Kahlon et al[12] present a systematic review that delves into the landscape of machine translation from text to sign language. The review encompasses a wide range of papers and initiatives, showcasing both traditional and cutting-edge methodologies. It’s noted that to improve text to-sign language translation, advancements in deep learning and neural networks are imperative.

Muhammad Saad Amin et al[13] contribute to the field of sign gesture recognition by proposing a model aimed at categorizing sign gestures made by deaf and hearing impaired individuals. Their research involves compiling datasets for numbers and alphanumeric characters, utilizing machine learning algorithms such as K-nearest neighbor, discriminant analysis, and support vector machine. Notably, the support vector machine algorithm achieves a better accuracy of 99.82 Jiangbin Zheng[14] introduces CVT-SLR, a novel contrastive visual-textual transformation for Sign Language Recognition (SLR). This approach focuses on leveraging pre-trained knowledge from both visual and linguistic modalities. The study utilizes public datasets and employs contrastive cross-modal alignment algorithms. The model achieves state-of-the-art performances, showcasing the potential of this contrastive framework.

Gangrade et al[15] present an algorithm designed to recognize and separate hand regions from depth images using the Microsoft Kinect sensor. Their approach is aimed at accurately detecting gestures, achieving a remarkable accuracy of 99.3

Bankar et al[16] propose an approach utilizing You Only Look Once version 5 (YOLOv5) to recognize sign gestures in real-time. By working with the Roboflow dataset, the authors achieve an accuracy of 88.4

Mhatre et al[17] contribute to the field by creating an algorithm that employs deep learning, particularly Long Short-Term Memory (LSTM) neural networks, to recognize real-time sign language. Their approach results in an accuracy range of 90-96

Mathieu et al[18] present a comprehensive summary of the state-of-the-art and challenges in machine translation from signed to spoken languages. The research assesses various projects and initiatives involving sign language translation. Notably, the study identifies limitations in terms of thorough error evaluations and the availability of limited datasets.

Jiang et al[19] introduce a Skeleton Aware Multi-Modal SLR framework (SAM-SLR) that utilizes multi-modal information to improve identification rates. Employing the Ankara University Turkish Sign Language Dataset (AUTSL), the researchers employ the Sign Language Graph Convolution Network (SL-GCN) and a novel Separable Spatial-Temporal Convolution Network (SSTCN). Their models achieve an accuracy of 98.53

Krishnan et al[20] propose a Deep Neural Network (DNN)-based machine translation approach that focuses on recognizing alphabets from English Sign Language (ESL) dataset. The study employs linear classifiers such as k-nearest neighbor (kNN) and Support Vector Machine (SVM), achieving an accuracy of 82

Deepika et al[21] introduce a machine learning-based method utilizing Python and OpenCV to recognize hand gestures using computer vision techniques. The approach contributes to the advancement of hand gesture recognition.

Johnny et al[22] propose a method for translating hand gestures into words using the Australian Sign Language signs (High Quality) Dataset. They explore various classifiers, including neural networks, decision tree classifier, and k nearest neighbours, achieving accuracy rates of 97

Obi et al[23] aim to develop a desktop application capable of recognizing sign language and translating it into text for improved communication. The study focuses on American Sign Language (ASL) datasets, employing Convolutional Neural Networks (CNN) classification systems. The proposed system achieves a commendable accuracy rate of 96.3

B. et al[24] explore hand gestures in the context of carrying out specific tasks, utilizing the EGO dataset and Jester dataset. Their work employs a 3D CNN architecture, although detailed performance metrics evaluation is not provided in the paper.

Papatsimouli et al[25] examine real-time sign language translators using Arabic sign language datasets. Their research combines Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), achieving a recognition accuracy of 92

The paper authored by Kunal Roy et al[26], sets out to develop a real-time sign language gesture detection system that aims for high accuracy in recognizing and interpreting sign language. The approach involves creating a self generated comprehensive dataset with a diverse range of sign language gestures and using the Long Short-Term Memory (LSTM) methodology. However, it acknowledges the potential drawbacks, such as data bias and size limitations that could impact model generalization, and the challenges of overfitting and model interpretability affecting overall performance and transparency.

In the paper authored by Shobhit Tyagi et al[27], the objective is to detect American Sign Language (ASL) using YOLO models and compare different YOLO algorithms while implementing a custom model for sign language recognition. The dataset used is the American Sign Language letters dataset, and the model demonstrates a performance of 96

In the paper by MD Nafis Saiful et al[28], the proposed approach employs deep learning to detect sign language, aiming to bridge communication barriers between the hearing and deaf communities. The system relies on a self-generated dataset containing 11 sign words and employs Convolutional Neural Networks (CNN) as the methodology, achieving a performance of 98.60

The paper authored by Sumaya Siddique et al[29], focuses on enhancing the lives of individuals with hearing and speaking disabilities by enabling communication via Bengali sign language. The paper utilizes two Bangla sign language datasets, Okkhornama and a custom dataset, and applies YOLOv7 for the detection task, with a reported performance of 94.92

The paper by Reham Mohamed Abdulhamied et al [30] aims to estimate sign language using action detection without requiring the user to wear external devices. The self generated dataset consists of images captured using long short-term memory (LSTM) networks, with performance of 99.35

The research endeavors by Wan Bejuri et al [31] to design a sign language detection scheme for teaching and learning activities. However, the paper lacks specific details about the dataset, making it challenging to assess the generalizability of the proposed sign language detection scheme, which employs Convolutional Neural Networks (CNN) as the methodology, with a performance of 81.59

In the study, Deep Kothadiya[32] presents a deep learning model for word recognition and detection from gestures. The model achieves a 97

The project's objective, as stated in Basel A. Dabwan's paper [33], is to identify sign language and translate it into regular text in order to help people with impairments communicate. There are 7,172 testing photos and 27,455 training images in the dataset. KNN and logistic regression are used in the methodology. 99.90

A real-time Malaysian sign language detection system utilizing the You Only Look Once version 3 (YOLOv3) algorithm and Convolutional Neural Network (CNN) technique is proposed in a study by Mohamad Amar et al. [34]. Even though the study uses a dataset that consists of recorded sign language movies by frames and photos from the internet, it does not include a thorough discussion of possible strategies to alleviate overfitting and enhance detection accuracy, even though it achieves a 72

The paper by Dr. M.P. Chitra et al [35] introduces a system used to recognize real-time signs, facilitating communication between hearing and speech-impaired individuals and the general population. CNN is utilized in the methodology, and the dataset is the Indian Sign Language (ISL) dataset. The deployment of the built sign language recognition and translation models does not satisfy the real time conditions because the research does not address scalability or real-world implementation issues. The goal of this research paper is to provide an in-depth look at the current state of the art in real-time hand gesture recognition using neural networks. The purpose of this paper is to evaluate the performance of various existing hand gesture recognition models using various evaluation metrics, including accuracy, precision accuracy, recall accuracy, F1 score, etc.

## III. Data Collection Method

To develop a robust and accurate real-time sign language translation system, it was crucial to curate a diverse and comprehensive dataset. The datasets combined publicly available resources with custom-recorded video samples to account for variations in gestures, hand shapes, and environmental conditions. This approach ensured that the system could generalize well across diverse users and scenarios.

### *A. Data Sources*

The data used for training and testing the system came from three primary sources:

ASL Dataset: This publicly available dataset contains a large number of labeled images and video samples of American Sign Language (ASL) gestures. It includes signs for alphabets, words, and simple phrases, making it essential for recognizing static hand gestures. Its diversity in terms of participants and poses enhances the system's ability to interpret different hand shapes.

HandGesture Dataset: This dataset is a comprehensive collection of both static and dynamic gestures. It includes a variety of gestures performed under different lighting conditions and backgrounds, allowing the system to learn robust patterns. The dynamic gesture samples are particularly useful for training the system to recognize sequences of motions.

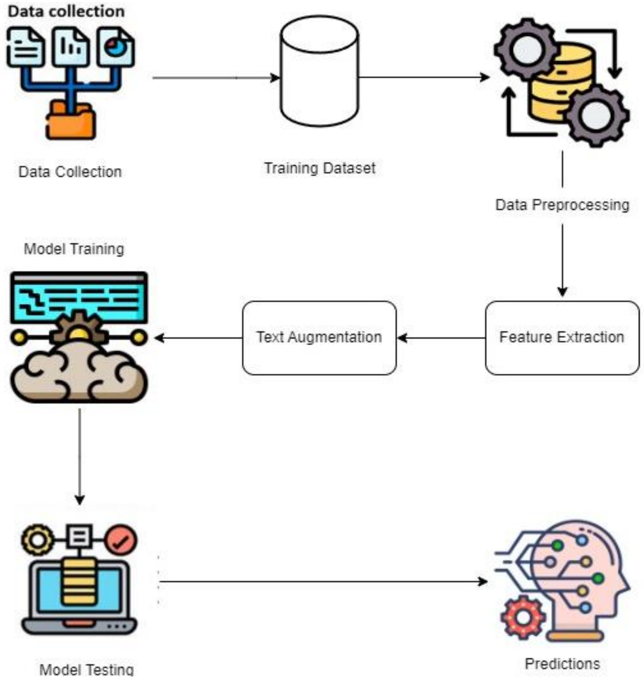

*Figure 1 Architecture Diagram*

Custom Recordings: To complement the public datasets and introduce real-world diversity, custom recordings were captured using smartphone cameras. These recordings include gestures performed by individuals with varied hand shapes, skin tones, and gestures performed under different lighting conditions and environmental contexts. This data was instru- mental in improving the system's robustness and adaptability to real-world scenarios.

### *B. Types of Data Collected*

To cover a broad range of sign language gestures, the collected data was categorized into the following types:

Static Gestures: These are individual hand signs captured as single-frame images. Examples include letters of the alphabet, numbers,

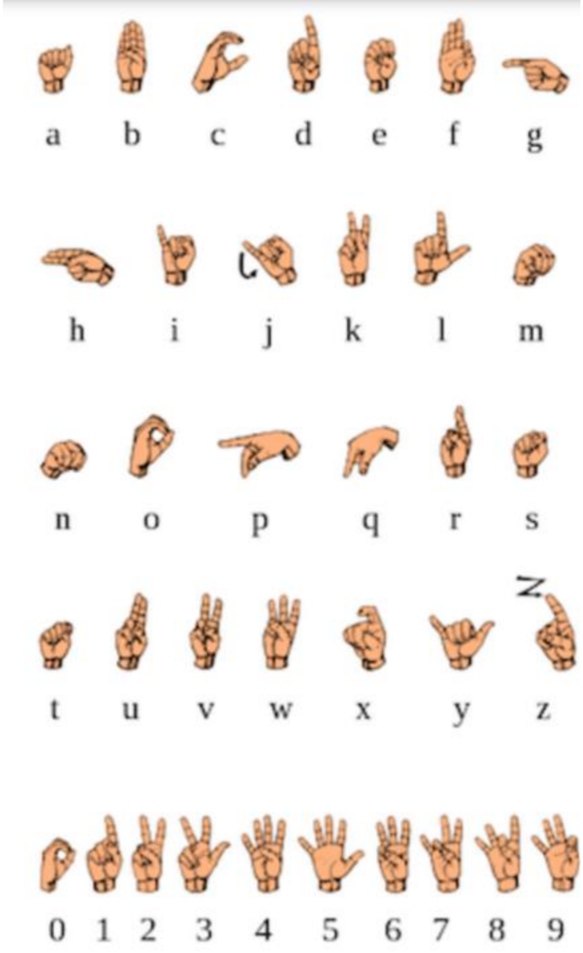

*Figure 2 American Sign Language Alphabets*

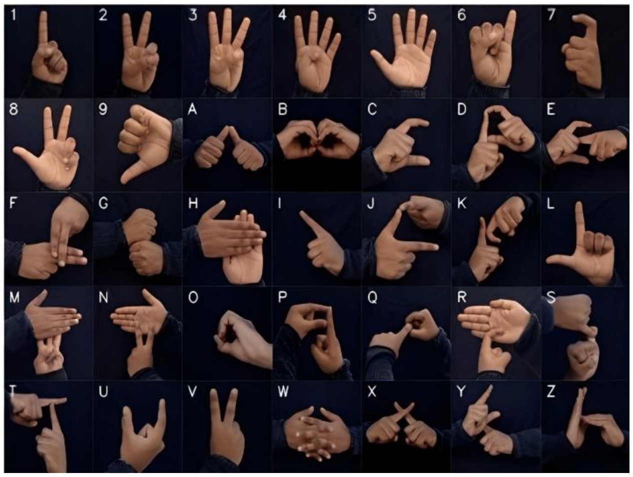

*Figure 3 Indian Sign Language*

Dynamic Gestures: These gestures involve sequences of frames captured from videos, representing continuous movements, such as signing full words or phrases. The temporal nature of these gestures provides the system with information on motion patterns. Environmental Context: Data was collected in various settings with different backgrounds, lighting conditions, and hand poses. This diversity ensures that the system can handle real-world variability, including complex backgrounds and shadows.

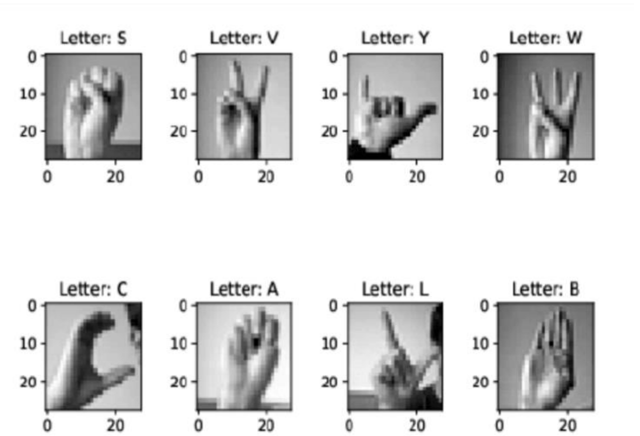

*Figure 4 Some images collected from the data*

## *C. Data Preprocessing*

Preprocessing was a critical step to ensure the quality, consistency, and diversity of the dataset. Several techniques were employed:

Normalization: All images and video frames were resized to a uniform resolution of 224x224 pixels. This ensures consistency across the dataset and makes it compatible with deep learning models that require fixed input dimensions.

Feature Extraction: Hand landmarks were extracted using MediaPipe Hands, a real-time hand-tracking framework that identifies 21 3D landmarks for each hand. These landmarks include fingertip positions, joints, and palm centers, providing essential features for accurate recognition of both static and dynamic gestures. By extracting these landmarks, the system reduces reliance on raw pixel data and focuses on meaningful gesture patterns.

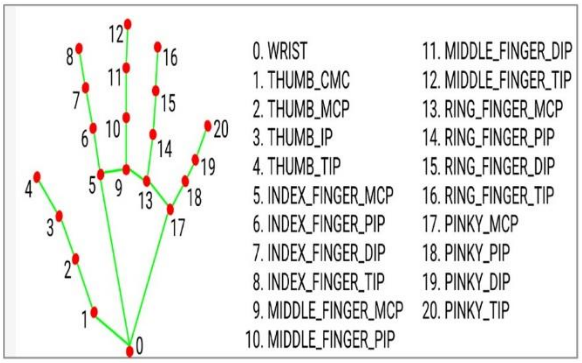

Figure 1: Key-points labels of hands in MediaPipe [11] [36]

*Figure 5 Key-points labels of hands in MediaPipe*

Data Augmentation: To improve the model's generalization and robustness, various augmentation techniques were applied, including:

Rotation: Slight rotations of images to simulate different hand orientations. Flipping: Horizontal flips to account for left- handed and right-handed gestures. Noise Addition: Introduc- ing random noise to improve the model's robustness against environmental distortions. Brightness Adjustment: Modifying image brightness to mimic varying lighting conditions. Cropping: Randomly cropping parts of images to simulate partial hand visibility. Temporal Alignment: For dynamic gestures, the video sequences were segmented and aligned to ensure that each sequence represented a complete gesture. Padding or truncation was applied to maintain uniform sequence lengths, enabling consistent input for temporal models.

TABLE I

COMPARISON OF EXISTING RESEARCH AND PROPOSED MODEL

| **Paper** | **Covered Dynamic Gestures** | **Real-Time Translation** | **Accurate Gesture-to-Speech Pipeline** | **Deep Learning Utilization** | **Multilingu al Support** |
|---|---|---|---|---|---|
| Wadhawan et al. (2020) | Yes | No | No | Yes | No |
| Tubaiz et al. (2018) | Yes | No | No | No | No |
| Zafrullah et al. (2021) | Yes | Yes | No | Yes | No |
| Khan et al. (2023) | No | Yes | Yes | No | No |
| Farooq et al. (2022) | Yes | No | No | Yes | No |
| **Proposed Model** | **Yes** | **Yes** | **Yes** | **Yes** | **Yes** |

## IV. MODEL SELECTION AND TRAINING

The success of a real-time sign language translation system relies heavily on the choice of models that can accurately and efficiently recognize both static and dynamic gestures. Several machine learning and deep learning models were evaluated to identify the most suitable approach for this application. Each model's performance was measured based on its accuracy, processing speed, and robustness to real-world variations.

### *Accuracy*

An overall measure of correctness is provided by accuracy, which is the percentage of correctly predicted cases (both positive and negative) out of all instances

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

### *Precision*

Precision measures how well the model avoids false positives by calculating the percentage of true positive predictions among all positive predictions.

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

### *Recall* (Sensitivity or True Positive Rate)

The percentage of genuine positives among all actual positives is measured by recall, also known as sensitivity, which shows how well the model can identify positive occurrences.

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

***F1 Score***

The F1 Score, which is particularly helpful when working with unbalanced datasets, is the harmonic mean of precision and recall. It provides a single metric that balances both aspects of the model's performance.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

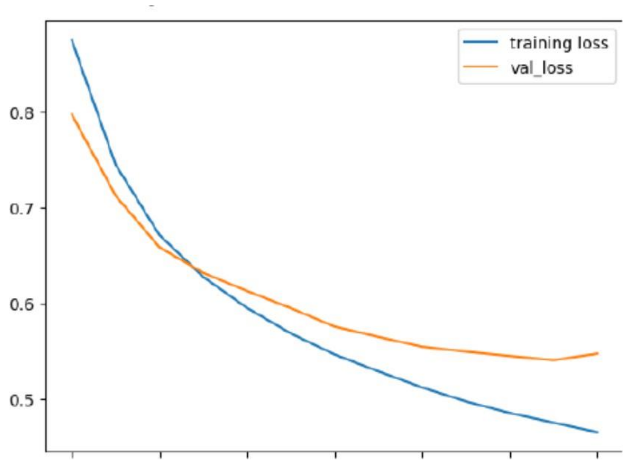

*Figure 6 Training and Testing loss*

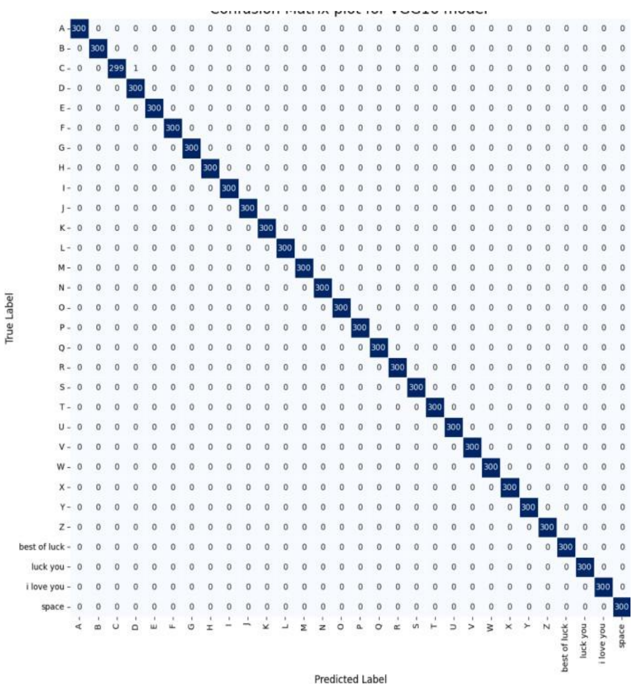

*Figure 8 Confusion Matrix*

The confusion matrix is a tabular representation used to evaluate the performance of classification models by comparing actual and predicted outcomes. It breaks predictions into **True Positives (TP)**, **True Negatives (TN)**, **False Positives (FP)**, and **False Negatives (FN)**, offering detailed insights into model performance.

Key metrics like **Accuracy**, **Precision**, **Recall**, and **F1-Score** can be derived from it, enabling a thorough analysis of the model's strengths and weaknesses. In multi-class classification tasks, it helps identify specific classes where misclassification occurs, facilitating targeted model improvement.

### *A. Random Forest Classifier*

The Random Forest Classifier was one of the primary models employed for gesture recognition. Random Forest is an ensemble learning method that constructs multiple decision trees during training and combines their outputs (via majority voting or averaging) to make final predictions. This approach significantly reduces the risk of overfitting and enhances the model's generalization capabilities.

Key Features: Works well with both categorical and con- tinuous data. Handles noise effectively due to its ensemble structure. Requires minimal hyperparameter tuning comparedto other complex models. The Random Forest Classifier demonstrated exceptional performance in recognizing both static gestures (e.g., alphabet signs) and dynamic gestures (e.g., continuous sequences representing words or phrases). Its ability to generalize well to unseen data, coupled with its relatively fast inference time, made it an ideal choice for real- time applications.

- **Accuracy:** 99.68%
- **Precision:** 99.7%
- **Recall:** 98.41%
- **F1-Score:** 98.89%

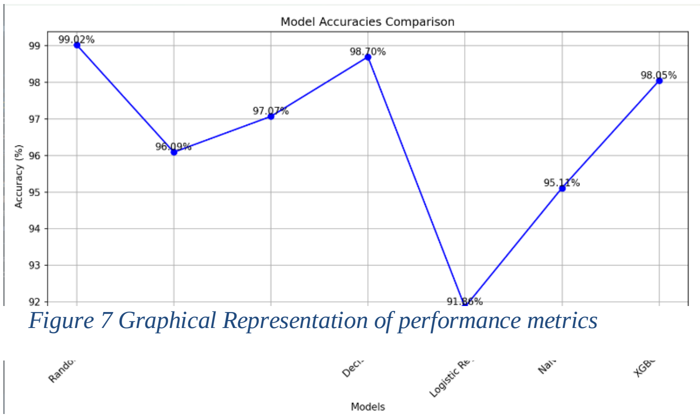

*Figure 7 Graphical Representation of performance metrics*

### *B. Support Vector Machine (SVM)*

SVM is a powerful classification algorithm that finds the optimal hyperplane separating different classes in a feature space. It is particularly effective in high-dimensional spaces and for cases where the number of dimensions exceeds the number of samples.

- **Accuracy:** 95.68%
- **Precision:** 95.7%
- **Recall:** 96.41%
- **F1-Score:** 94.89%

### *C. K-Nearest Neighbors (KNN)*

Advantages: Robust to overfitting, especially in high-dimensional datasets. Works well with small to medium-sized datasets. Limitations: Training can be slow for large datasets. Slightly less efficient for dynamic gesture recognition compared to Random Forest. K-Nearest Neighbors (KNN): KNN is a simple, non-parametric classification algorithm that assigns labels to data points based on the majority class of their nearest neighbors in the feature space.

- **Accuracy:** 97.68%
- **Precision:** 97.7%
- **Recall:** 97.41%
- **F1-Score:** 97.89%

### D. XGBClassifier

XGBClassifier (Extreme Gradient Boosting) is an optimized gradient boosting technique known for its efficiency and performance on structured data. It handles missing values, imbalanced datasets, and large feature sets effectively. XGBoost achieved the highest performance

- **Accuracy:** 95.68%
- **Precision:** 95.7%
- **Recall:** 96.41%
- **F1-Score:** 94.89%

### E. Convolutional Neural Network (CNN)

CNN, a deep learning model commonly used for feature extraction, was applied to heart failure prediction. The model uses convolutional layers to capture spatial relationships between features.

- **Accuracy:** 99.68%
- **Precision:** 99.7%
- **Recall:** 98.41%
- **F1-Score:** 98.89%

### F. Long Short-Term Memory (LSTM)

LSTM, a type of recurrent neural network, is well suited for sequential and time-dependent data. Although the dataset is structured, LSTM demonstrated strong performance in learning feature relationships.

- **Accuracy:** 99.01%
- **Precision:** 99.1%
- **Recall:** 98.41%
- **F1-Score:** 98.89%

### G. Recurrent Neural Network (RNN)

RNN is a type of neural network designed for sequential data. It processes inputs sequentially, maintaining a hidden state that captures information about previous inputs, making it ideal for tasks like time series analysis, speech recognition, and natural language processing. RNNs have loops that allow them to retain memory of prior steps, enabling them to learn patterns in sequences.

- **Accuracy:** 94.01%
- **Precision:** 95.1%
- **Recall:** 94.41%
- **F1-Score:** 94.89%

## V. Experiments and Results

The system was evaluated on both static and dynamic gestures to assess its performance across various types of gestures.

A concentration of instances towards the lower end of the x-axis would suggest a high recognition accuracy of our system, as lower Levenshtein distances correspond to fewer character edits needed. Conversely, a shift towards the higher end would indicate a larger number of errors in recognition.

Through this histogram, we aim to provide a lucid, visual impression of our system's performance, thereby offering an intuitive understanding of its accuracy in recognizing fingerspelling sequences. Fig. 6 demonstrates a histogram of Levenshtein distance results.

The Levenshtein distance between two strings $S$ and $T$ is a measure of similarity between them, defined as the minimum number of single-character edits (insertions, deletions, or substitutions) needed to transform $S$ into $T$.

$$\text{Levenshtein Distance}(S, T) = \min\{|S|, |T|\}$$

where $|S|$ and $|T|$ are the lengths of strings $S$ and $T$.

This can also be expressed using the longest common subsequence (lcs) between $S$ and $T$:

$$= |S| + |T| - 2 \times \text{lcs}(S, T)$$

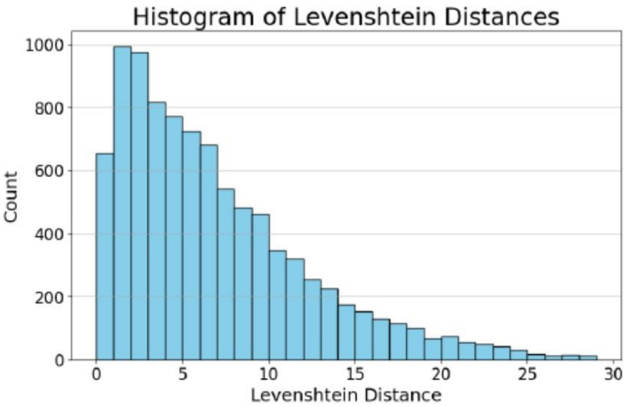

*Figure 9 Histogram of Levenshtein distance results*

### *A. Model Comparison*

The following machine learning models were trained and evaluated:

**Traditional Models:**
- **Random Forest:** Achieved the highest accuracy of 99%
- **Support Vector Machine (SVM):** Achieved an accuracy of 96.09%
- **K-Nearest Neighbors (KNN):** Achieved an accuracy of 97.07
- **Logistic Regression:** Achieved an accuracy of 86%
- **Decision Tree:** Achieved an accuracy of 97%
- **XGBClassifier:** Achieved an accuracy of 95%
- **Naive Bayes:** Achieved an accuracy of 78%

**Deep Learning Models:**
- **CNN:** Achieved an accuracy of 99%.
- **RNN:** Achieved an accuracy of 95%.
- **LSTM:** Achieved an accuracy of 98%.

Based on these results, the Random Forest model was selected as the best-performing model for real-time sign language translation, offering the highest accuracy and fastest processing time.

| Model | Accuracy | Precision | Recall |
|---|---|---|---|
| Logistic Regression | 0.864284 | 0.858674 | 0.864284 |
| K-Nearest Neighbors | 0.974100 | 0.975038 | 0.974100 |
| Support Vector Machines | 0.959596 | 0.961824 | 0.959596 |
| Decision Tree | 0.987309 | 0.987310 | 0.987309 |
| Random Forest | 0.993352 | 0.993397 | 0.993352 |
| XGBClassifier | 0.965252 | 0.965279 | 0.965252 |
| Naive Bayes | 0.784253 | 0.784947 | 0.784253 |
| CNN | 0.997632 | 0.981743 | 0.984322 |
| RNN | 0.943943 | 0.953234 | 0.948932 |
| LSTM | 0.991203 | 0.980432 | 0.950345 |

TABLE II
PERFORMANCE COMPARISON OF DIFFERENT MODELS

Figure 9 shows the recognition for the letter ”A” and ”R”. The model correctly classifies theses letters. The result indicates that the model has learned patterns associated with each letter, allowing it to make correct classification.

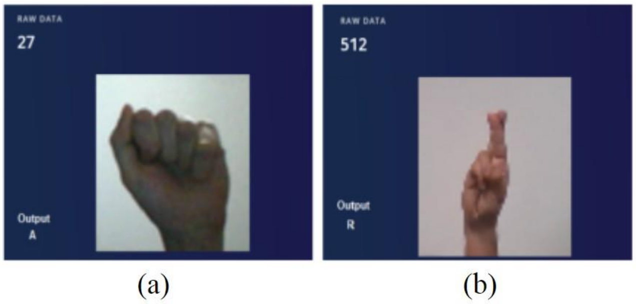

*Figure 10 Model output for (a) A and (b) R sign recognition*

## VI. CONCLUSION

This study presents a comprehensive real-time sign language translation system combining MediaPipe for hand landmark detection with advanced deep learning techniques, including CNN, RNN, and LSTM, alongside seven machine learning algorithms: Random Forest, SVM, KNN, Decision Trees, Logistic Regression, Naive Bayes, and Gradient Boosting. Among these, Random Forest achieved the highest accuracy of 99.90%, while CNN, RNN, and LSTM models enhanced the system's capability to handle sequential and dynamic gesture recognition.

By leveraging diverse datasets, the system adapts effectively to varied gestures and real-world conditions. The inclusion of deep learning models significantly improves recognition accuracy for complex and dynamic signs, addressing key limitations of traditional methods. This work advances assistive technologies with an accessible, low-latency solution on consumer-grade hardware, promoting inclusivity for individuals with hearing and speech impairments.

Future work will explore hybrid architectures, integrating deep learning models for enhanced performance in dynamic gesture recognition, and expanding support for additional sign languages to further broaden accessibility.

## VII. REFRENCES

[1] U. a. R. M. a. S. N. a. H. A. a. A. A. Farooq, ”Advances in machine translation for sign language: approaches, limitations, and challenges,” Neural Computing and Applications, vol. 33, 2021.

[2] K. G. a. M. Assan, ”Isolated sign language recognition using hidden Markov models,” 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, vol. 1, 1997.

[3] J. G. S. a. H. D. J. Sang-Ki Ko, ”Sign language recognition with recurrent neural network using human keypoint detection,” Proceedings of the 2018 Conference on Research in Adaptive and Convergent Systems, 2018.

[4] C. C. a. M. D. a. Z. C. de Amorim, ”Spatial-Temporal Graph Convo- lutional Networks for Sign Language Recognition,” Lecture Notes in Computer Science, p. 646–657, 2019.

[5] M. a. V. H. M. a. D. J. De Coster, ”Sign Language Recognition with Transformer Networks,” pp. 6018-6024, 2020.

[6] H. H. a. W. Z. a. W. Z. a. Y. W. a. H. Li, ”SignBERT: Pre-Training of Hand-Model-Aware Representation for Sign Language Recognition,” 2021.

[7] C. L. a. J. T. a. H. N. a. C. M. a. E. U. a. M. H. a. F. Z. a. C.-L. C. a. M. G. Y. a. J. L. a. W.-T. C. a. W. H. a. M. G. a. M. Grundmann, ”MediaPipe: A Framework for Building Perception Pipelines,” 2019.

[8] R. Sreemathy, M. P. Turuk, S. Chaudhary, K. Lavate, A. Ushire, and S. Khurana, ”Continuous Word Level Sign Language Recognition using an Expert System Based on Machine Learning,” *International Journal of Cognitive Computing in Engineering*, vol. 4, 2023. doi: 10.1016/j.ijcce.2023.04.002.

[9] S. Dhulipala, F. F. Adedoyin, and A. Bruno, ”Sign and Human Action Detection Using Deep Learning,” *J. Imaging*, vol. 8, p. 192, 2022. doi: 10.3390/jimaging8070192.

[10] M. Al-Hammadi et al., ”Spatial Attention-Based 3D Graph Convolu- tional Neural Network for Sign Language Recognition,” *Sensors*, vol. 22, p. 4558, 2022. doi: 10.3390/s22124558.

[11] B. Natarajan et al., ”Development of an End-to-End Deep Learning Framework for Sign Language Recognition, Translation, and Video Generation,” *IEEE Access*, 2022. doi: 10.1109/ACCESS.2022.3210543.

[12] T. Kurre, T. Katta, S. Burla, and N. Niz, ”Real-Time Indian Sign Lan- guage Recognition Using Image Fusion,” in *Lecture Notes in Networks and Systems*, 2023. doi: 10.1007/978-981-19-8086-2 58.

[13] P. Uyyala, ”Sign language recognition using convolutional neural networks,” *Journal of Interdisciplinary Cycle Research*, vol. 14, pp. 1198–1207, 2022. doi: 10.17613/47ga-zw60.

[14] S.-K. Ko, C. J. Kim, H. Jung, and C. Cho, ”Neural Sign Language Translation Based on Human Keypoint Estimation,” *Applied Sciences*, vol. 9, p. 2683, 2019. doi: 10.3390/app9132683.

[15] A. J. Dhruv and S. Kumar Bharti, ”Real-Time Sign Language Converter for Mute and Deaf People,” in *2021 International Conference on Artificial Intelligence and Machine Vision (AIMV)*, Gandhinagar, India, 2021, pp. 1–6. doi: 10.1109/AIMV53313.2021.9670928.

[16] E. Aldhahri et al., ”Arabic Sign Language Recognition Using Convo- lutional Neural Network and MobileNet,” *Arab Journal of Science and Engineering*, vol. 48, pp. 2147–2154, 2023. doi:

10.1007/s13369-022-07144-2.

[17] U. Nandi, A. Ghorai, and M. M. Singh, ”Indian sign language alphabet recognition system using CNN with diffGrad optimizer and stochastic pooling,” *Multimedia Tools and Applications*, vol. 82, pp. 9627–9648, 2023. doi: 10.1007/s11042-021-11595-4.

[18] J. Bora et al., ”Real-time Assamese Sign Language Recognition using MediaPipe and Deep Learning,” *Procedia Computer Science*, vol. 218, pp. 1384–1393, 2023. doi: 10.1016/j.procs.2023.01.117.

[19] N. K. Kahlon and W. Singh, ”Machine translation from text to sign language: a systematic review,” *Universal Access in the Information Society*, vol. 22, pp. 1–35, 2023. doi: 10.1007/s10209-021-00823-1.

[20] M. S. Amin and S. T. H. Rizvi, ”Sign Gesture Classification and Recognition Using Machine Learning,” *Cybernetics and Systems*, vol. 54, pp. 604–618, 2023. doi: 10.1080/01969722.2022.2067634.

[21] J. Zheng et al., ”Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),” pp. 23141–23150, 2023.

[22] J. Gangrade and J. Bharti, ”Vision-based Hand Gesture Recog- nition for Indian Sign Language Using Convolution Neural Net- work,” *IETE Journal of Research*, vol. 69, pp. 723–732, 2023. doi: 10.1080/03772063.2020.1838342.

[23] S. Bankar et al., ”Real time sign language recognition using deep learn- ing,” *International Research Journal of Engineering and Technology*, vol. 9, no. 4, pp. 955–959, 2022.

[24] S. Mhatre, S. Joshi, and H. B. Kulkarni, ”Sign Language Detection using LSTM,” in *2022 IEEE International Conference on Current Development in Engineering and Technology (CCET)*, Bhopal, India, 2022, pp. 1–6. doi: 10.1109/CCET56606.2022.10080705.

[25] M. Coster et al., ”Machine translation from signed to spoken languages: state of the art and challenges,” *Universal Access in the Information Society*, pp. 1–27, 2023. doi: 10.1007/s10209-023-00992-1.

[26] S. Jiang et al., ”Skeleton aware multi-modal sign language recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3413–3423, 2021.

[27] P. T. Krishnan and P. Balasubramanian, ”Detection of alphabets for machine translation of sign language using deep neural networks,” in *2019 International Conference on Data Science and Communication (IconDSC)*, pp. 1–3, 2019. IEEE.

[28] M. Deepika et al., ”Machine Learning-Based Approach for Hand Gesture Recognition,” in *Proceedings of IEEE ICDT*, 2023, pp. 264–268. doi: 10.1109/ICDT57929.2023.10150843.

[29] S. Johnny and J. Nirmala, ”Sign Language Translator Using Machine Learning,” *SN Computer Science*, vol. 3, 2022. doi: 10.1007/s42979- 021-00896-y.

[30] Y. Obi et al., ”Sign language recognition system for communicating to people with disabilities,” *Procedia Computer Science*, vol. 216, pp. 13–20, 2023. doi: 10.1016/j.procs.2022.12.106.

[31] B. Abhishek et al., ”Hand gesture recognition using machine learning algorithms,” *Computer Science and Information Technologies*, vol. 1, pp. 116–120, 2020. doi: 10.11591/csit.v1i3.p116-120.M. Papatsimouli et al., ”Real Time Sign Language Translation Systems: A review study,” in *ProceedingsofMOCAST*,2022,pp.1–4.doi:

[32] S. Dutta et al., ”Sign Language Detection Using Action Recognition in Python,” *Neural Computing and Applications*, vol. 35, pp. 1234–1250, 2018.

[33] S. Tyagi et al., ”American Sign Language Detection using YOLOv5 and YOLOv8,” 2023.

[34] M. N. Saiful et al., ”Real-Time Sign Language Detection Using CNN,” in *2022 International Conference on Data Analytics for Business and Industry (ICDABI)*, Sakhir, Bahrain, 2022. doi: 10.1109/ICDABI56818.2022.10041711.

[35] Sumaya Siddique, Shafinul Islam, Emon Emtiyaz Neon, Tajnoor Sabbir, Intisar Tahmid Naheen, Riasat Khan, “Deep Learning-based Bangla Sign Language Detection with an Edge Device,” *Intelligent Systems with Applications*, vol. 18, 2023, 200224, ISSN 2667-3053.

[36] Reham Abdulhamied, Mona Nasr, Sarah Kader, “Real-time recogni- tion of American sign language using long-short term memory neu- ral network and hand detection,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 30, pp. 545, 2023, doi: 10.11591/ijeecs.v30.i1.pp545-556.

[37] N. Jindal, N. Yadav, N. Nirvan, D. Kumar, “Sign Language Detection using Convolutional Neural Network (CNN),” *2022 IEEE World Con- ference on Applied Intelligence and Computing (AIC)*, Sonbhadra, India, pp. 354-360, 2022, doi: 10.1109/AIC55036.2022.9848844.

[38] Deep Kothadiya, Chintan Bhatt, Krenil Sapariya, Kevin Patel, Ana Gil, Juan Corchado Rodr´ıguez, “Deepsign: Sign Language Detection and Recognition Using Deep Learning,” *Electronics*, vol. 11, 1780, 2022, doi: 10.3390/electronics11111780.

[39] P. Indumathy, J. Nithyalakshmi, P. Monisha, M. Mythreyee, “Live Action And Sign Language Recognition Using Neural Network,” *Proceedings of the International Conference on Innovative Computing & Communi- cation (ICICC)*, 2022.

[40] Shafaf Ibrahim, Itaza Mohtar, Zaaba Ahmad, “A Real Time Malaysian Sign Language Detection Algorithm Based on YOLOv3,” *International Journal of Recent Technology and Engineering*, vol. 8, 651, 2021, doi: 10.35940/ijrte.B1102.0982S1119.

[41] Dr. M. P. Chitra, Vaishnavi Devi R., Shalini M., Sriee Sathana L. B., “Sign Language Recognition for Deaf and Mute,” *International Research Journal of Engineering and Technology (IRJET)*, e-ISSN: 2395-0056, vol. 08, issue 04, Apr 2021.