



# Machine Learning Framework for Intelligent Hand Gesture Recognition: An Application to Indian Sign Language and Hand Talk

Shashi Malviya<sup>1\*</sup>, Anita Mahajan<sup>2</sup> and Kamal Kumar Sethi<sup>3</sup>

<sup>1,2,3</sup> Department of Artificial Intelligence and Data Science  
Acropolis Institute of Technology and Research, Indore, India  
\*shashimalviyamtech21, anitamahajan@acropolis.in, kamalsethi@acropolis.in

**Abstract.** Communicating with individuals with hearing disabilities poses significant challenges. The research presented in this paper represents an effort to further explore the complexities associated with character classification in Indian Sign Language (ISL). It's important to note that sign language alone may not suffice for effective communication, especially for individuals with hearing or speech impairments. The gestures made by individuals with disabilities can appear jumbled or confusing to those who are unfamiliar with the language. Sign language recognition has long been recognized as a crucial tool to aid individuals with hearing impairments. Over the years, researchers have dedicated significant efforts to advancing this field of study. Recently, there has been a growing focus on developing solutions that can be universally applied in India, where the need for such technology is particularly pronounced. The primary objective of this paper is to develop an accurate and reliable sign language recognition system. By critically evaluating different methodologies, the aim is to identify the most effective method for accurately recognizing and interpreting sign language gestures, ultimately contributing to the advancement of assistive technologies for the deaf community.

**Keywords:** Indian sign language, Dataset, Media pipe, Classification, Neural Network, KNN, SVM, Random Forest

## 1. Introduction

Sign language serves as the primary mode of communication for individuals who are hearing impaired, relying on gestures instead of spoken words to convey meaning. It encompasses hand shapes, movements, orientation, facial expressions, and lip patterns to express messages. However, regional variations and the limited availability of sign language interpreters create communication barriers between the deaf community and the

hearing majority. Over 2 million people in India are deaf, and they have difficulty interacting with the hearing community because the latter is not very familiar with sign language. As such, the necessity for sign language interpreters who can help the deaf and the hearing is critical to communication. Regretfully, translators are costly and difficult finding [2, 3]. In recent years, there has been an incredible burst of studies on ISL gesture recognition. These varied initiatives, carefully done by researchers worldwide, have created a wealth of information and creativity in the field of recognition of gestures. These studies have set out on an excellent objective of creating trustworthy and precise systems that recognize the complex web of actions that makes up Indian Sign Language, driven by the rapidly developing fields of machine learning, computer vision, and signal processing. [2, 3, 4]

The range of strategies used in ISL gesture recognition studies shows how difficult the issue is and how interdisciplinary collaboration is necessary. From handwritten feature extraction methods to neural network architectures, researchers have explored a range of approaches in an attempt to record and interpret the specifics of sign language.

The development of carefully selected datasets specifically designed for ISL makes it easier for researchers to develop and assess gesture recognition models and benchmark their methods, which has advanced the field. [1, 4, 6] By creating a real-time hand gesture identification system with a focus on Indian Sign Language (ISL) recognition, this research study offers a novel approach for speech impairments. The methodology targets 26 English letter gesture sets and using machine learning techniques to recognize ISL gestures in real-time. Although obstacles, the system performs almost perfectly in real-world situations, indicating its potential to greatly improve communication accessibility for those with difficulty hearing.

## 2. Literature Survey

In the Study, a deep learning architecture based on computer vision is presented for a signer-independent system that recognizes Indian Sign Language [1]. The author proposes the Signet model based on supervised learning, which is capable of identifying all 24 alphabets. The proposed CNN architecture consists of six hidden layers and one dropout layer. In Study the author employs an approach involving counting fingertips and calculating the distance between fingertip and palm centroid using PCA algorithm for hand gesture recognition. [2], Data is acquired through a 3megapixel camera. The recognition process involves segmentation to obtain red, blue, and green masks, fingertip identification algorithm, and PCA (Principal Component Analysis).

Tripathi et al [3] employed orientation histogram in conjunction with PCA (Principal Component Analysis) for continuous gesture recognition. They collected data using a Canon EOS camera with an 18–55 mm lens, capturing 10 sentences and evaluating them at different distances. Although their approach proved effective, it was conducted in a controlled environment. The work proposed by Rajam et al [4] employed image processing for South Indian sign language recognition, achieving 98.125% accuracy. Data captured via LG Smart Cam USB web camera comprised 32 signs. Suggested enhancement with neural network integration. In the work [5] utilized neural networks

and SVM for alphabet recognition, achieving 94.37% accuracy. Dataset sourced from the internet covered 17 alphabets, but lacked comprehensive coverage.

Dixit et al. [6] utilized MSVM (Multi-Class Support Vector Machine) for hand gesture recognition, achieving an impressive accuracy of 96%. Their data creation process involved 60 classes; however, the approach exhibited signer dependency. In their study, Sharma et al. [7] employed the VGG16 model for alphabet recognition, attaining an accuracy of 97%. Their dataset, comprising 26 classes, was developed internally. However, the research underscored the importance of a more diverse dataset to improve performance.

The study in [8] employed K-nearest correlated neighbor classification for alphabet recognition, achieving 90% accuracy. Their manually constructed dataset encompassed single-handed and double-handed gestures. However, the study encountered challenges with incorrect classification in double-handed gestures. In the approach proposed by Bhagat et al. [9] utilized CNNs for alphabet and number recognition, achieving 98.81% accuracy. Data collected via Microsoft Kinect RGB-D camera comprised 36 classes. However, the study lacked automation. Alaria et al. [10] utilized CNNs (Convolutional Neural Networks) for alphabet recognition, achieving an accuracy of 85%. Their dataset, consisting of 26 classes, was captured using a Raspberry Pi 4 camera. The study suggests enhancing performance through transfer learning techniques.

### 3. Proposed Methodology:

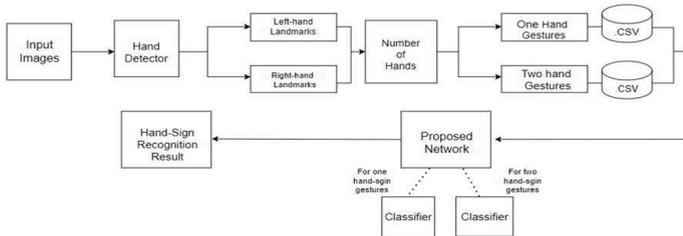


Fig. 1 Schematic diagram of Proposed Methodology

In our course on hand sign recognition, we have proposed a methodology that can determine hand gestures with high reliability in real-time. The proposed methodology is shown in Fig.1. Throughout the development of this methodology, we underwent multiple phases, which are defined in the subsequent sections. In this course the hand landmarks are detected in real-time via frameworks like Media Pipe. Spatial distribution of these landmarks determines the number of hands present. A dataset is curated, comprising labeled hand gesture images, for model training. Trained models predict hand sign gestures based on capture data, by employing well know classifiers. Real-time recognition is achieved by feeding webcam images into the trained models, which analyze hand

landmarks to classify gestures as intended.

#### 4. Image Acquisition

In the course of hand sign recognition, the process of image acquisition serves as a fundamental step for understanding and interpreting gestures accurately. Leveraging the media pipeline that is offered by the computer vision pre-processing CV2 library gives a structured framework designed for real-time detection of hand images through web cameras. This pipeline offers a streamlined approach to acquiring visual data, capturing static images upon which subsequent feature extraction processes can be executed with precision and efficiency. The CV2, also known as Open CV [11], stands as a cornerstone in the field of computer vision, offering a rich suite of functionalities tailored for image and video analysis.

Various aspects of image record and pre-processing is seamlessly handled by its flexible capabilities and large toolkit. We ensure a seamless flow of visual data towards the recognition system by using CV2. One of CV2's key benefits is its high the preliminary processing capabilities. Equipped with a diverse array of functions, CV2 enhance image quality, adjust color spaces, reduce noise, and perform other essential pre-processing tasks.

##### 4.1 Acquiring Hand Landmarks

After setting up the media pipeline by employing Computer Vision techniques to visual data from the video streams. Leveraging Open CV's capabilities, captured frames in real-time for subsequent analysis. For hand tracking and landmark detection, we integrated Media Pipe, a powerful framework designed for various machine learning-based tasks, including hand detection and tracking. Media Pipe facilitates the detection of hands within images by employing a comprise 16 pre-trained Tensor Flow and Tensor Flow Lite models on top of the Media Pipe framework [12] via which it traces the hand landmarks, i.e. key points that are used to track the hands show in Fig 2. Through Media Pipe, we acquired intricate hand landmarks, encompassing critical points such as



**Fig. 2** Showing hand landmarks on hand given by media pipe framework fingertips, knuckles, and palm regions. These landmarks, detected with reliable accuracy, provide essential spatial information crucial for gesture recognition. Also by leveraging media pipe functionalities we distinguished left-hand landmarks and right-hand landmarks.

##### 4.2 Hand Counting Using Landmark Analysis

In hand counting using landmark analysis, the landmarks are acquired using the Media Pipe framework, which offers a reliable hand tracking and landmark detection

capabilities. The landmarks are represented as a set of key points detected on the hand within an image or video frame. Each landmark is associated with a specific coordinate in a three-dimensional space, typically denoted by  $(x, y, z)$  coordinates. These coordinates represent the position of the landmark relative to the camera frame. The  $(x, y)$  coordinates denote the position of the landmark within the 2D image plane, while the  $z$ -coordinate represents the depth or distance of the landmark from the camera. For a hand there are 21 landmarks which mark the overall skeleton of a hand. For two hands, the count of landmarks get increased to 42 i.e. 21-21 points for each hand. Observing this deviation, we can efficiently distinguish gestures requiring two hands and one hand.

## 5. Dataset Preparation for Training

In preparing the dataset for training, we make use of Media Pipe to discern between gestures requiring one hand and those requiring two. Leveraging this capability, we categorized and stored hand landmarks separately based on the number of hands involved. Furthermore, to gain independence of the relative distance from the video capturing device, we opted to use only the  $x$  and  $y$  coordinates, discarding the  $z$  coordinate. This decision mitigates the influence of hand distance from the camera on gesture interpretation.

We computed relative coordinates and applied normalization, essential for consistent model training and mitigate outlier effects with varying hand positions relative to the device. Additionally, we organized the data as vectors, as with one hand gesture comprising 21 points, each described by  $(x, y)$  coordinates. This arrangement results in a 42-length feature vector for one hand. For gestures involving two hands, the dataset consists of 42 points, yielding 84 features in total. These vectors were systematically stored in CSV files labeled with corresponding gesture labels. Notably, we prioritized dataset balance by generating a significant number of instances for each gesture, mitigating potential data imbalance issues during model training. The procedure of dataset generation is well-elaborated in the forthcoming section.

### 5.1 Building Classifier for hand sign recognition

In our hand sign prediction framework, we implement a two-modal approach for classification tasks, distinguishing between one-hand and two-hand gestures. Using visual information obtained with Media Pipe, the number of hands identified dictates the model that is used. We thoroughly examine several classifiers in order to maximize results, choosing them based on their proven efficacy in earlier research. Our choice of classifiers includes random forests, support vector machines (SVM),  $k$ -nearest neighbors (KNN), and deep neural networks. We evaluate classifier performance through cross-validation to select the best model for both one- and two-hand gestures. This thorough process guarantees reliable and precise hand sign prediction. Our approach promises efficient gesture recognition in a range of scenarios and applications by utilizing well-established classifiers and the unique characteristics of one-hand and two-hand gestures. This will enhance accessibility in a variety of settings and improve human-computer interaction.

## 5.2 Recognizing Hand Sign Gestures

This phase is the most important one, in which we use the suggested framework to immediately predict and identify gestures. Our gesture recognition system's accuracy and efficiency are enhanced by the concept of gesture separation, which also makes it possible to understand hand sign instructions in dynamic contexts with easily.

We ensure trustworthy and immediate recognition of a variety of hand motions by utilizing the capabilities of the suggested network. During the hand sign recognition stage, we first receive an image and then use a combined media pipeline that consists of Media Pipe and Open CV (cv2). In this pipeline, we use Media Pipe to extract hand landmarks from the frame that CV2 has retrieved. Subsequently, we ascertain the number of hands present in the frame following this detection, a trained classifier, enriched with acquired knowledge from its training phase, is invoked. The classifier analyzes the extracted hand landmarks, predicting and classifying them into their corresponding hand gestures classes. This final phase represents a culmination of efforts, showcasing the culmination of our system's capabilities in real time hand gesture recognition and interpretation.

## 6. Dataset

In response to the inadequacy of existing datasets for Indian Sign Language (ISL), we undertook the creation of our own comprehensive dataset. Despite the presence of open datasets, they lacked the depth and variety necessary for training accurate ISL interpretation models. ISL, utilized primarily by the deaf and hard of hearing community in India, encompasses a diverse array of gestures and signs, including representations of English alphabets and numerical digits, but in this study the focus is only on English alphabets. Our dataset includes representations of all 26 English alphabets, with 19 depicted using two-hand signs and the remainder using one-hand signs.

### 6.1 Environment Stimuli

During our dataset generation process, we maintain precise control over the environment, ensuring a balanced background with optimal brightness levels. We use a standard webcam for data capture to ensure consistency and accuracy, crucial for precise analysis and modeling.

### 6.2 Acquiring Hand Landmarks

Media Pipe framework, introduced by Google, is utilized for constructing machine learning pipelines to analyze time-series data like video and audio. Initially developed for real-time video and audio analysis on YouTube, Media Pipe's subsequent public release enabled researchers and developers to seamlessly integrate and utilize this framework in their projects. Comprising 16 pre-trained Tensor Flow and Tensor Flow Lite models, Media Pipe solutions encompass a variety of tasks such as hand detection, face detection, text analysis, and more [13]. Leveraging these capabilities, we have employed Media Pipe to infer hand landmarks from video frames depicting hand sign gestures. The hand-tracking model outputs 21 3D landmark points, accurately delineating key points on the hand. These landmarks are returned in sequences of spatial coordinates, denoted by

their x, y, and z coordinates, accurately tracking the hand's precise location within the field of view of the capturing device. An illustration of the landmarks depicted by using a static image is shown in Fig 3 (a) and Fig 3 (b)



(a)

**Fig 3. (a) Showing ISL hand sign for English alphabet –A**



(b)

**Fig-3(b) Corresponding Landmarks obtained by Media pipe represented in 3-D view.**

We record only x, y coordinates and dropping z -coordinates of the landmarks representing the key points on the hand to remove depth-wise relation of hand gesture with the capturing device. Preceding with determining the relative positions of these landmarks against a base point, i.e. first landmark point. This step allows us to get the relative points which reduces disparities with left and right landmarks. Followed by, we normalize these coordinates based on the length of the list of landmarks. This normalization process ensures that the hand gestures are represented consistently across different hand sizes and positions within the frame. Subsequently, we also separate gesture requiring one hand and two hand by tracking the number of key points.

#### 6.4 Characteristics of the Dataset

A dataset of landmarks for all 26 English alphabets of the Indian Sign Language (ISL) was collected, transformed by the pre-processing procedure for each hand sign to identify key landmarks, show in Fig 4 The combined dataset consists of 43,359 records, with 34,440 records for 19 gestures requiring both hands and 8,919 for 7 gestures with one hand. The dataset was organized to ensure balance and uniform spread across all classes. Providing a solid foundation for training and evaluating machine learning models for sign language recognition.

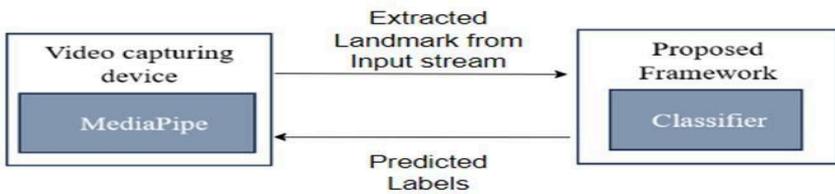


**Fig 4** Preview of Landmarks after pre-processing: relative position and normalization

## 7. Implementation and Result

### 7.1 Proposed system and its components

The system is a streamlined pipeline that efficiently processes input data to produce real-time hand gesture classifications. It consists of two sub-systems: a video capturing device that captures live video streams, and a framework that handles various gestures. The pipeline end is connected to the framework, which performs preprocessing and hand detection. The media pipe extracts hand landmarks from the frame and passes onto other end of the pipeline. A related classifier that has been previously trained for the detected hand configuration is then invoked by the framework based on the number of hands involved. It analyzes the landmarks and characteristics that have been retrieved to produce smooth and effective gesture predictions. Real-time answers to dynamic motions are made possible by this optimized procedure, which guarantees quick output generation.



**Fig. 5** System block diagram showing: video-capturing device support with cv2 library for transferring extract stream to our proposed framework for gesture recognition

### 7.2 Building Classifier

We chose Random Forest, k Nearest Neighbors (KNN), Support Vector Machines (SVM), and Deep Neural Networks (DNN) as our classifiers. Previous experimental results and the algorithms' track record of successfully managing a range of data properties have served as our guidelines. These classifiers are well-known for handling high-dimensional data and a lot of inputs, which makes them appropriate for our task. Additionally, they function well in situations where there are complex interactions between classes since they are excellent at handling classes that are closely related to one another. Our goal is to create a comprehensive and flexible model that can effectively predict outcomes in our data domain by utilizing the capabilities of each classifier.

### 7.3 Deep Neural network

Artificial neural networks with dense a connection which connect every node in one layer to every node in the next is known as deep neural networks(DNNs). In multi-class classification problems, they are used to divide inputs into various classes. The concept is

that every hidden layer receives input to each neuron, generating a dense link. Even with big data sets, Deep Neural Networks are able to identify complex patterns. With this skill, we were able to identify and anticipate hand gestures who involved one or two hands [14,15]. In multi-class classification, the final layer choose class with the highest probability by a soft max activation function. The output of the soft max activation function is the class with the highest probability.

#### **7.4 Random Forest**

Random Forest is a machine learning method that builds several decision trees using collective learning principles, which improves prediction performance and decreases overfitting. It provides consistent results even with high dimensions via voting or averaging the results from all trees. It works well for problems involving regression and classification, especially when handling big datasets and high dimensionality [16]. We were able to identify the hand gestures using this ability.

#### **7.5 K-Nearest Neighbors**

A flexible supervised learning method for classification and regression problems is K-Nearest Neighbors (KNN). Using a similarity principle, it identifies a new data point's label or value by comparing it to the average or majority vote of its K-nearest neighbors in the training dataset. By allocating data points to the class with the majority vote among its neighbors, KNN works well for multiclass classification [17, 18]. It is a well-liked option for many kinds of datasets due to its simplicity, convenience of use, and capacity to manage both numerical and categorical data. In addition, compared to other algorithms, KNN is less susceptible to outliers.

#### **7.6 Support Vector Machines**

A supervised learning method for classification and regression problems is Support Vector Machines (SVM). In a high-dimensional space, it determines a hyperplane that maximizes the margin between classes. While SVM inherently supports binary classification, it can be extended to handle multiclass classification by breaking down the problem into multiple binary classification tasks. SVMs are versatile and efficient, capable of handling high-dimensional and nonlinear data relationships [19,20]. They are robust to outliers, perform well with small datasets, and ensure optimality in results due to their convex optimization nature.

### **8. Performance Metrics**

To develop an effective hand sign recognition system, the proposed network needs to be evaluated with performance measures that will guide and direct the network to achieve the goal objective. These parameters provide insights on how well a classifier can discern the gesture from the input stream. They provide a numerical value through which the capabilities and limitations of a classifier can be understood. In our study, we evaluate the classifier performance by considering the following metrics. The accuracy score measures how often a machine learning model correctly predicts the outcome [21]. It quantifies the percentage of correct classifications made by the model against all the prediction cases

that the model had made during prediction of the entire dataset. Precision describes the proportion of correctly predicted examples among all the true cases predicted by the model. It represents the model's capability of being truly correct [22].

Recall measures the model's ability to correctly identify positive instances from the entire pool of actual positive instances, calculated as ratio of number of all true cases, against all true cases that exist for an instance [22]. F1 score gives a common predictive measure based on test precision and recall, by combining both metrics for quick comparisons and is used to generalize precision and recall as one metric and it is typically used to compare different models which have discrepancy in their loss and recall [22]. It is determined by taking the harmonic mean of precision and recall. Confusion matrix is a table that is used to describe the performance of a classification model on a set of test data for which the true values are known [23]. It allows visualization of the performance of an algorithm and it is particularly useful for understanding the performance of a multi-class classification problem. The mathematical formulation of these metrics are given in Equation (1) (2) (3) (4) as:

$$\text{Accuracy} = (\text{True Positive} + \text{True Negative}) / \text{True Positive} + \text{True Negative} + \text{False Positive} + \text{False Negative} \quad (1)$$

$$\text{Precision} = \text{True Positive (TP)} / (\text{True Positive (TP)} + \text{False Positive (FP)})$$

$$\text{Precision} = \text{True Positive (TP)} / (\text{True Positive (TP)} + \text{False Positive (FP)})$$

...(2)

$$\text{F1 Score} = 2 \times (\text{recall} \times \text{precision}) / (\text{recall} + \text{precision})$$

$$\text{F1 Score} = 2 \times (\text{recall} \times \text{precision}) / (\text{recall} + \text{precision})$$

...(3)

$$\text{Recall} = \text{True Positive (TP)} / (\text{True Positive (TP)} + \text{False Negative (FN)})$$

$$\text{Recall} = \text{True Positive (TP)} / (\text{True Positive (TP)} + \text{False Negative (FN)})$$

...(4)

## 9. Results

The results of our work, as shown in Table 1, underscore the proficiency of our approach in classifying the 26 English alphabets of Indian Sign Language gestures using two distinct datasets - one for single-hand gestures and another for two-hand gestures. To ensure optimality in the results, we employed K-Fold Cross-Validation, enabling thorough exploration of the parameter space and ensuring the selection of optimal configurations. According to our research, all models demonstrated very high accuracies upon

deployment, highlighting the efficacy of our approach. Specifically, the model SVM for two-handed gestures and KNN for one-handed gestures emerged as the top performers in terms of accuracy and robustness. Detailed performance analysis for these models is outlined in the classification reports and confusion matrix in Fig 4 and Fig 5 and confusion matrix in Fig 6.

**Table 1: Comparison of Classifier Accuracies (in percentage) for One-handed and Two-handed gestures.**

| Classifier    | Accuracy (One Hand) | Accuracy (Two Hands) |
|---------------|---------------------|----------------------|
| DNN           | 99.55               | 99.94                |
| K – NN        | 99.94               | 99.91                |
| SVM           | 99.89               | 99.99                |
| Random Forest | 99.89               | 99.98                |

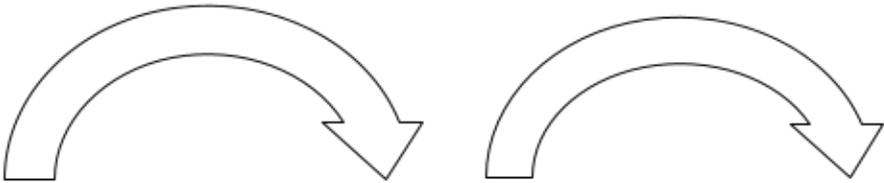
**Table 2 Classification report for SVM as a two-hand classifier**

| Labels                | Labels No. | Precision | Recall | F-1 Score | Support |
|-----------------------|------------|-----------|--------|-----------|---------|
| A                     | 0          | 1.00      | 1.00   | 1.00      | 462     |
| B                     | 1          | 1.00      | 1.00   | 1.00      | 250     |
| D                     | 2          | 1.00      | 1.00   | 1.00      | 321     |
| E                     | 3          | 1.00      | 1.00   | 1.00      | 358     |
| F                     | 4          | 1.00      | 1.00   | 1.00      | 355     |
| G                     | 5          | 1.00      | 1.00   | 1.00      | 356     |
| H                     | 6          | 1.00      | 1.00   | 1.00      | 350     |
| K                     | 7          | 1.00      | 1.00   | 1.00      | 346     |
| M                     | 8          | 1.00      | 1.00   | 1.00      | 342     |
| N                     | 9          | 1.00      | 1.00   | 1.00      | 357     |
| P                     | 10         | 1.00      | 1.00   | 1.00      | 342     |
| Q                     | 11         | 1.00      | 1.00   | 1.00      | 334     |
| R                     | 12         | 1.00      | 1.00   | 1.00      | 369     |
| S                     | 13         | 1.00      | 1.00   | 1.00      | 362     |
| T                     | 14         | 1.00      | 1.00   | 1.00      | 320     |
| W                     | 15         | 1.00      | 1.00   | 1.00      | 321     |
| X                     | 16         | 1.00      | 1.00   | 1.00      | 426     |
| Y                     | 17         | 1.00      | 1.00   | 1.00      | 414     |
| Z                     | 18         | 1.00      | 1.00   | 1.00      | 414     |
| <b>Accuracy</b>       |            |           |        | 1.00      | 1784    |
| <b>Macro Accuracy</b> |            | 1.00      | 1.00   | 1.00      | 1784    |

|                          |      |      |      |      |
|--------------------------|------|------|------|------|
| <b>Weighted Accuracy</b> | 1.00 | 1.00 | 1.00 | 1784 |
|--------------------------|------|------|------|------|

**Table 3 Classification report for KNN as a one-hand classifier**

| Labels                   | Labels No. | Precision | Recall | F-1 Score | Support |
|--------------------------|------------|-----------|--------|-----------|---------|
| C                        | 0          | 1         | 1      | 1         | 251     |
| I                        | 1          | 1         | 1      | 1         | 215     |
| J                        | 2          | 1         | 1      | 1         | 256     |
| L                        | 3          | 1         | 1      | 1         | 266     |
| O                        | 4          | 1         | 1      | 1         | 308     |
| U                        | 5          | 1         | 1      | 1         | 236     |
| V                        | 6          | 1         | 1      | 1         | 252     |
| <b>Accuracy</b>          |            |           |        | 1.00      | 1784    |
| <b>Macro Accuracy</b>    |            | 1.00      | 1.00   | 1.00      | 1784    |
| <b>Weighted Accuracy</b> |            | 1.00      | 1.00   | 1.00      | 1784    |

**Fig 6. Confusion matrix for SVM (above) and KNN (below).**

Further, we have also analyzed the real-time performance of our model, and the comprehensive findings are meticulously documented in the Table 4 provided. This table encapsulates the intricate nuances of our model's efficacy, showcasing its prowess in accurately interpreting gestures. Notably, it delineates whether the gesture is executed with one hand or two. Moreover, if our machine learning model demonstrates the capability to discern these gestures accurately, it is indicated with a resounding "yes" in the corresponding column.

**Table 4 Hand gesture predictions for alphabets using different machine learning methods.**

| Alphabets | ISL Hand Sign      |                    | Prediction Remarks |     |                 |               |
|-----------|--------------------|--------------------|--------------------|-----|-----------------|---------------|
|           | One Handed Gesture | Two Handed Gesture | DNN                | SVM | KNN             | Random Forest |
| A         |                    | Yes                | Yes                | Yes | Yes             | Yes           |
| B         | -                  | Yes                | Yes                | Yes | Yes             | Yes           |
| C         | Yes                | -                  | Yes                | Yes | Yes             | Yes           |
| D         | -                  | Yes                | Conflict with P    | Yes | Conflict with P | Yes           |
| E         | -                  | Yes                | Yes                | Yes | Yes             | Yes           |
| F         | -                  | Yes                | Yes                | Yes | Yes             | Yes           |
| G         | -                  | Yes                | Yes                | Yes | Yes             | Yes           |
| H         | -                  | Yes                | Yes                | Yes | Yes             | Yes           |

|   |     |     |                 |                 |                 |                 |
|---|-----|-----|-----------------|-----------------|-----------------|-----------------|
| I | Yes | -   | Yes             | Yes             | Yes             | Yes             |
| J | Yes | -   | Yes             | Yes             | Yes             | Yes             |
| K | Yes | -   | Yes             | Yes             | Yes             | Yes             |
| L | Yes | -   | Yes             | Yes             | Yes             | Yes             |
| M | -   | Yes | Conflict with N | Conflict with N | Conflict with N | Yes             |
| N | -   | Yes | Yes             | Conflict with Z | Yes             | Yes             |
| O | -   | Yes | Yes             | Yes             | Yes             | Yes             |
| P | -   | Yes | Conflict with D | Yes             | Conflict with D | Yes             |
| Q | -   | Yes | Conflict with S | Yes             | Yes             | Conflict with S |
| R | -   | Yes | Yes             | Yes             | Yes             | Yes             |
| S | -   | Yes | Yes             | Yes             | Yes             | Yes             |
| T | -   | Yes | Yes             | Yes             | Yes             | Yes             |
| U | -   | Yes | Yes             | Yes             | Yes             | Yes             |
| V | -   | Yes | Yes             | Yes             | Yes             | Yes             |
| W | -   | Yes | Yes             | Yes             | Yes             | Yes             |
| X | Yes | -   | Yes             | Yes             | Yes             | Yes             |
| Y | -   | Yes | Yes             | Yes             | Yes             | Yes             |
| Z | -   | Yes | Yes             | Yes             | Yes             | Yes             |

---

## 10. Application View

We present our application, showcasing our developed one-hand and two-hand classifier models, effectively predicting the correct gestures. In the Fig. 7, we demonstrated hand gestures representing the letters 'A', 'D', 'E', 'G', 'J', and 'L', and our application accurately identified each of them. Similarly, our application successfully recognized several other words through their respective hand gestures.

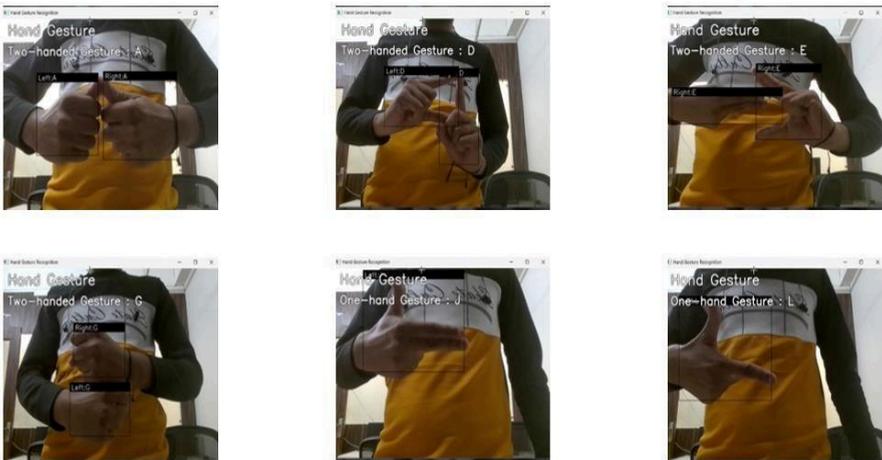


Fig 7. View of application showing gesture prediction by our proposed classifier

### 11. Conclusion

This paper introduces a real-time ISL (Indian Sign Language) hand gesture recognition system designed to aid communication for individuals with speech impairments. Focusing on identifying 26 English alphabet gesture sets, the system achieves high accuracy scores of 99.99% for two-handed gestures and 99.94% for one-handed gestures using SVM and KNN classifiers respectively. In practical scenarios, the system demonstrated near perfect performance with occasional conflicts between similar signs: sign – P sign -D and sign –N sign -M, highlighting its potential for practical sign recognition applications.

### 12. Future Work

In addition to recognizing English alphabet gestures, there is significant potential for expanding the capabilities of our real hand gesture recognition system. By incorporating recognition for digits and words in addition to alphabets, we can enhance the versatility and utility of the system for individuals with speech impairments thus broadening the scope of communication possibilities. Exploration of deep learning architectures could lead to improved accuracy and robustness in gesture recognition, even in challenging real-world environments. Moreover, considering the diverse linguistic and cultural backgrounds of users, adapting the system to recognize gestures from different sign languages could significantly benefit a broader range of individuals.

## References

- [1] Kocakanat, K., & Serif, T., Turkish traffic sign recognition: Comparison of training step numbers and lighting conditions. *European Journal of Science and Technology*. doi:10.31590/ejosat.1015972(2021)
- [2] Sruthi, C. J. International conference on communication and signal processing (ICCSP). IEEE.(2019)
- [3] Deora, D., & Bajaj, N. Indian sign language recognition. In 1st IEEE international conference on emerging technology trends in electronics, communication & networking..(2012)
- [4] Tripathi, K., Baranwal, N., & Nandi, G. C. Continuous Indian sign language gesture recognition and sentence formation. *Procedia Computer Science*, 54, 523–531.(2015)
- [5] Rajam, P., & Subha, G.. Real time Indian sign language recognition system to aid deaf dumb people. In IEEE 13th international conference on communication technology(2011).
- [6] Dixit, K., & Singh Jalal, A.. Automatic Indian sign language recognition system. IEEE.(2013)
- [7] Sharma, A. Benchmarking deep neural network approaches for Indian Sign Language recognition. *Neural Computing and Applications*, 33, 6685–6696.(2021)
- [8] Gupta, B., Shukla, P., & Mittal, A.K-nearest correlated neighbor classification for Indian sign language gesture recognition using feature fusion. In IEEE International conference on computer communication and informatics (ICCCI). (2016)
- [9] Bhagat, N., Kamal, Y., & Vishnusai, G. N. . Indian sign language gesture recognition using image processing and deep learning. In IEEE Digital Image Computing: Techniques and Applications (DICTA).(2019)
- [10] Alaria, S. . Simulation and analysis of hand gesture recognition for indian sign language using CNN. *International Journal on Recent and Innovation Trends in Computing and Communication*, 10, 10–14.(2022)
- [11] Python OpenCV Tutorial. (n.d.). Retrieved, from <https://pythonexamples.org/python-opencv/>
- [12] Media. (n.d.). Retrieved, from Google for Developers website: <https://developers.google.com/media>
- [13] Bora, J., Dehingia, S., & Boruah, A.. Anuraag Anuj Chetia, Dikhit Gogoi Real-Time Assamese Sign Language Recognition Using Media Pipe and Deep Learning *Procedia Computer Science*, 218, 1384–1393.(2023)
- [14] Melanie. Dense neural networks: Understanding their structure and function. Retrieved, from Data Science Courses | Data Scientist website: [https://datascientest.com/en/dense-neural-networks-understanding-their-structure-and-function\(2024\)](https://datascientest.com/en/dense-neural-networks-understanding-their-structure-and-function(2024))
- [15] Rithani, M., Kumar, R. P., & Doss, S. A review on big data based on deep neural

- network approaches. *Artificial Intelligence Review*, 56(12), 14765–14801. doi:10.1007/s10462-023-10512-5(2023)
- [16] What is random forest? Retrieved, from [https://www.ibm.com/topics/random-forest\(2024\)](https://www.ibm.com/topics/random-forest(2024))
- [17] (N.d.-a). Retrieved, from <https://www.ibm.com/docs/en/knn-usage>
- [18] A Complete Guide to K-Nearest Neighbors (Updated 2024). (n.d.). *Analytics Vidhya*.(2024)
- [19] Srivastava, T. Guide to K-Nearest Neighbors algorithm in machine learning. Retrieved, from *Analytics Vidhya* website: [https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/\(2024\)](https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/(2024))
- [20] Daniel Gutierrez, O. Build a multi-class support vector machine in R. Retrieved, from *Open Data Science - Your News Source for AI, Machine Learning & more* website: [https://opendatascience.com/multi-class-support-vector-machine-r/\(2018\)](https://opendatascience.com/multi-class-support-vector-machine-r/(2018))
- [21] Goyal, C. Understanding multiclass classification using SVM. Retrieved, from *Analytics Vidhya* website: [https://www.analyticsvidhya.com/blog/2021/05/multiclass-classification-using-svm/\(2021\)](https://www.analyticsvidhya.com/blog/2021/05/multiclass-classification-using-svm/(2021))
- [22] Loss for Data-imbalanced NLP Tasks. 58th Annual Meeting of the Association for Computational Linguistics. (n.d.). 465–476.
- [23] (N.d.-b). Retrieved from <https://www.sciencedirect.Com/topics/engineering/confusion-matrix>.
- [24] Nair, V., & V, A. (2013). A Review on Indian Sign Language Recognition. *International Journal of Computer Applications*, 73(22), 33–38.(2013)

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

