American Journal of Electronics & Communication

Indian Sign Language Classification (ISL) using Machine Learning

Subhalaxmi Chakraborty
University of Engineering and Management
Kolkata, India
subhalaxmi2008@gmail.com

Nanak Bandyopadhyay
University of Engineering and Management, Kolkata
Kolkata, India
nanakbandyopadhyay@hotmail.com

Piyal Chakraverty
University of Engineering and Management, Kolkata
Kolkata, India

Swatilekha Banerjee
University of Engineering and Management, Kolkata
Kolkata, India
swatilekha123@hotmail.com

Zinnia Sarkar
University of Engineering and Management, Kolkata
Kolkata, India
zinnia.rnj21@gmail.com

Sweta Ghosh
University of Engineering and Management, Kolkata
Kolkata, India
ghosh.swetag@gmail.com
chakravertypiyal@gmail.com

***Abstract*— Communication is a crucial for humans, it is most vital. People with hearing or speaking disabilities need a way to communicate with other people of the society and vice versa. This paper presents a novel methodology in classifying the English Alphabets shown via various hand gestures in The Indian Sign Language (ISL) using Mediapipe Hands API, launched by Google. The objective of using this API is to detect 21 landmarks in each hand along with their x, y and z coordinates in 3D space. Due to the scarcity of proper dataset available on the internet for ISL, at the very beginning, we have created a dataset having a size of 15000, per English character, each consisting of the coordinates of 21 landmarks recognized by Mediapipe Hands API. From the literature, we found that prediction has been done for The American Sign Language and other foreign sign languages using Mediapipe API effectively. The novelty of our proposed work lies in using the same API for the Indian Sign Language. In this paper, we have discussed a comparative analysis of different classification algorithms like Support Vector Machine (SVM), Random Forest, K-nearest neighbors (KNN) , Decision Tree and other algorithms in terms of accuracy with the highest accuracy among all being 99%. It is relevant to mention in this connection that the classification of the Indian Sign Language (ISL) using Mediapipe API is faster than the other conventional methods and outperforms in computational capability. This model can be used in web applications, mobile applications, desktop applications and in many more places.**

***Keywords*—sign language, machine learning, computer vision, mediapipe, Indian sign language**

I. INTRODUCTION

Communication is exceptionally pivotal to humans because it empowers us to express. The modes of communication include but are not limited to speech, body language, gestures, reading, writing etc. Unfortunately, there comes a communication gap for the speaking and hearing-impaired minority. Visual aids or an interpreter can be used in some cases to communicate with them [1]. Here the Sign Languages play an important role. These are virtual-gesture languages that are extensively used by deaf and hard-hearing people to communicate with others.

It has its own vocabulary that is distinct from that of spoken and written languages. Every word or alphabet is attributed to a certain gesture or body language (which entails movement of hands, limbs, or body to convey the speaker's thoughts) in sign language. [2]. Sign languages vary by area, such as American Sign Language (ASL), Indian Sign Language (ISL), and so on. This research focuses on Indian Sign Language. Indian Sign Language is mainly spoken in South Asian countries. ISL has a variety of qualities that set it apart from other sign languages. Number signs, family relationships, and the use of space are also essential elements of ISL. In addition, ISL has no temporal inflection [3]. The objective of the proposed work is to create a model that can correctly, easily, and consistently interpret alphabets in Indian Sign language based on their corresponding gestures in all lighting conditions. Recognizing sign language signs can be done in a variety of ways. Sanil et. Al used one of these methods: they trained a machine learning model to segment the skin portion of the image, extracted relevant features from the skin segmented images, used the extracted features as input into various supervised learning models for testing, and then used the trained models for classification. Sanil Jain and K.V.Sameer Raja used a number of approaches to train multiple machine learning models, with the maximum accuracy of 54.63 percent [4]. Rathna et. al used a different method of detecting hands to train the machine learning algorithm. They used the Microsoft XBOX360 Kinect Camera to obtain a dataset of depth-based segmented RGB images for classifying 36 distinct gestures. They used a Deep Convolutional Neural Network to identify the characters and got an accuracy of 89.30 percent [5]. This paper explores how one of Google's frameworks, Mediapipe Hands, can be used to detect hands with both speed and precision.

II. BACKGROUND

A. *Sign Language & Gestures*

Sign language is a form of communicating for people who are deaf or hard of hearing. It's a series of hand signals, facial expressions, and body language that helps them to communicate with the rest of society. There isn't a single sign language. Rather, depending on the country, there are a number of sign languages around the world. Some countries, such as America, India, China, Mexico, Japan, Australia, and the United Kingdom, have their own sign language. Some gestures are performed with one hand, and others are performed with both hands [6]. According Hearing deficiency impacts 466 million people worldwide, with 34 million of them being teenagers, according to the World Health Organisation. WHO

predicts that by 2050, over 900 million people will have hearing damage that is incapacitating[7].

B. *Indian Sign Language*

India is a country rich in ethnic diversity as well as linguistic distinctions. Also for people with normal hearing and voice, communicating in a country like India is difficult. As a result, communication is much more complex for disabled persons. Also and, there are only a few schools that cater to them. Nonetheless, it falls short of meeting the needs of disabled people in this densely populated region. Furthermore, most rural areas have not yet been developed enough to provide opportunities for the deaf and dumb.

C. *Mediapipe Hands API*

Hands by MediaPipe (by Google) is a high-resolution hand and finger tracking solution. Machine learning (ML) is used to infer 21 3D landmarks of a hand from a single frame. MediaPipe's system achieves real-time efficiency on a mobile phone, and also scales to several hands, while existing state-of-the-art methods focus largely on strong desktop environments for inference.[8]

MediaPipe Hands makes use of a machine learning pipeline that consists of several models that work together:

- A palm detection model that uses the entire image to generate an aligned hand bounding box. [8]
- A hand landmark model that returns high-fidelity 3D hand key points from the cropped image area identified by the palm detector.[8]

With proper inputs to the Mediapipe solutions API (discussed later), it provides the following output.

- *MULTI_HAND_LANDMARKS*

  A series of detected/tracked hands, each of which is represented by a list of 21 hand landmarks, each of which is made up of the letters x, y, and z. The picture width and height were used to normalise x and y to [0.0, 1.0]. The landmark depth is represented by z, with the origin being the depth at the thumb (landmark number 0), and the smaller the value, the closer the landmark is to the camera. The magnitude of z is measured on a scale that is similar to that of x. Fig 1 gives a visual representation of the hand landmarks.[8]
- *MULTI_HANDEDNESS*

  Collection of the detected/tracked hands' handedness (i.e., is it a left or right hand). Each hand is made up of two parts: a mark and a score. Label is a string with the value "Left" or "Right" in it. Score is the expected handedness's approximate likelihood, which is always greater than or equal to 0.5. (and the opposite handedness has an estimated probability of 1 - score). [8]

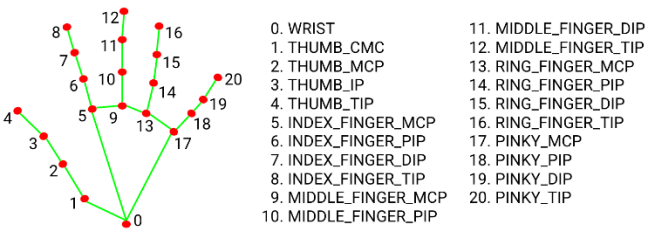

Fig. 1 Hand landmarks

D. *Euclidian Distance and Slope of a line*

The length of a line segment between two points in Euclidean space is known as the **Euclidean distance** in mathematics.[9]

Let point x and point y have Cartesian coordinates $(x1, x2)$ and $(y1, y2)$ in the Euclidean plane respectively. The difference between p and q is then calculated as:

$$d(x, y) = \sqrt{(y_1 - x_1)^2 + (y_2 - x_2)^2} \quad (1)$$

The slope of a line in the plane containing the x and y axes is defined as the change in the y coordinate divided by the corresponding change in the x coordinate between two distinct points on the line, and is commonly expressed by the letter m [10]. The following equation describes this:

$$m = \frac{\Delta y}{\Delta x} = \frac{vertical\ change}{horizonal\ change} \quad (2)$$

The difference in x between two points $(x1, y1)$ and $(x2, y2)$ is $x2 - x1$, while the change in y is $y2 - y1$. The following equation is obtained by substituting all quantities into the above equation:

$$m = \frac{y_2 - y_1}{x_2 - x_1} \quad (3)$$

Euclidean Distance and the slope will be later used while creating the dataset.

III. Literature Survey

There has been a lot of research into sign language recognition for American Sign Language (ASL), and there are many standard databases online. Indian Sign Language (ISL) research is still in its early stages. This may be attributed to ISL's low prevalence, rudimentary nature, and lack of recognition in society. Many variations remain between sign languages used in different areas due to a lack of standardization [10]. As a consequence, there is currently no standard dataset for ISL, so research is focused on existing datasets for other sign languages, or on generating or accumulating datasets from a variety of sources.

Hand gesture recognition can be performed using either vision-based or sensor-based methods [11].

- **Vision Based:** Vision-based techniques necessitate the use of a video camera to capture images or video of hand gestures. Three key stages comprise the high-level pipeline for hand gesture recognition using a vision-based approach:

a) To distinguish the hand gesture from the image, this step includes clean-up and filter operations such as blurring, thresholding, image morphing, skin masking, and so on. This step is also know as the **Image Pre-processing** stage.

b) **Feature extraction**: This stage extracts features from the cleaned images. The image dataset is given a numerical representation. Since raw pixel data may be discordant and subject to noise, inversion, rotation, or illumination, feature extraction is necessary, making the learning process difficult. Feature extraction can be done

manually or automatically, using feature detection algorithms.

c) **Classification**: The image dataset's attributes serve as the learning model's training data. This knowledge can be used to train classification models such as SVM, Random Forest, K-nearest neighbours KNN, Decision Tree, and Neural Networks. The aplhabets can be predicted using the trained model.

- **Sensor Based:** This method necessitates the use of sensors and instruments to capture the hand's motion, direction, and velocity.
  This method is brilliantly portrayed by T Raghuveera, R Deepthi, R Mangalashri and R Akshaya [12]. They used Microsoft Kinect to collect images of the hand gestures. This gave them the advantage of being able to obtain depth information from the Kinect's infrared sensor in addition to RGB color information.
  The use of sensors in Sign Language is very well explained by Edon Mustafa [14].

IV. Proposed Work

The aim of this research is to build a machine learning model that can rapidly and reliably identify Indian Sign Language movements. This machine learning model can then be used in a variety of ways to assist people with speech difficulties in communicating effectively. The objectives are described in more detail below.

- To develop or implement a system for recognising hand gestures that can be captured and used to train a machine learning algorithm.
- To make the dataset as wide and varied as practicable, since there is no readily accessible dataset for Indian Sign Language that can satisfy all of the criterias of the research.
- To prepare the dataset for the classification of different alphabets using various machine learning methods.
- To build a program that can detect hand movements and accurately identify the alphabet.

V. Workflow

The diagram below illustrates the steps involved in detecting the hands and developing a fully trained and working machine learning model for gesture detection at a high level.

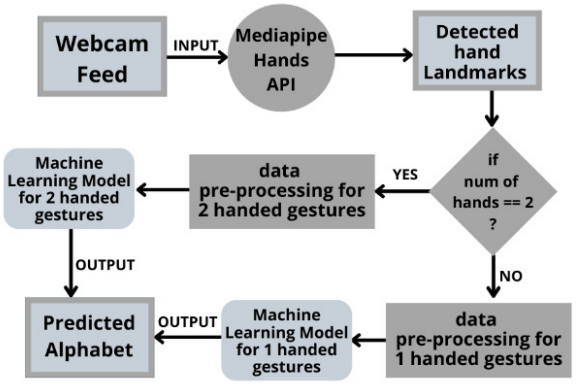

Fig. 2 Workflow diagram

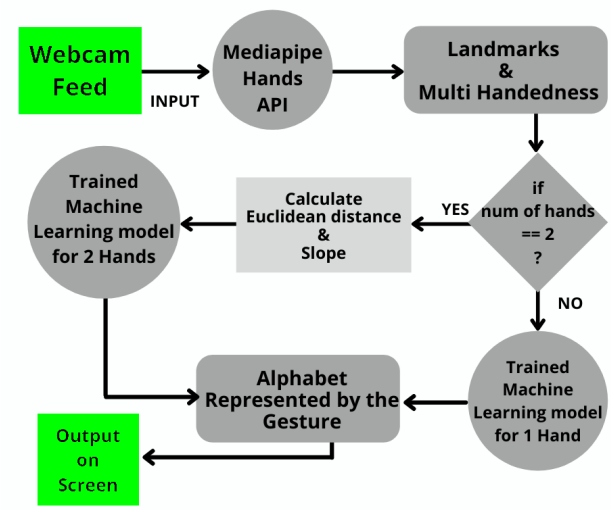

Fig. 3 Character prediction flowchart

A. *Hand Detection*

The palm detection model from Mediapipe handles the whole hand detection process. The webcam feed is fed into the

Mediapipe API, which returns a list of detected/tracked hands, each of which is defined by a list of 21 hand landmarks, each of which is made up of x, y, and z. The landmark depth is represented by z, with the origin being the depth at the thumb, and the smaller the value, the closer the landmark is to the camera. It also returns a list of the detected/tracked hands' handedness, as stated in section II (C).

B. *Dataset*

The dataset used for this work was made by the team themselves since there was no readily available dataset for ISL that fulfilled all the requirements of the work. As the Indian Sign Language consists of hand gestures that require the use of one as well as two hands, two separate datasets were made instead of one. One dataset contained all the data regarding the hand gestures that can be shown using only one hand and the other one had all the data

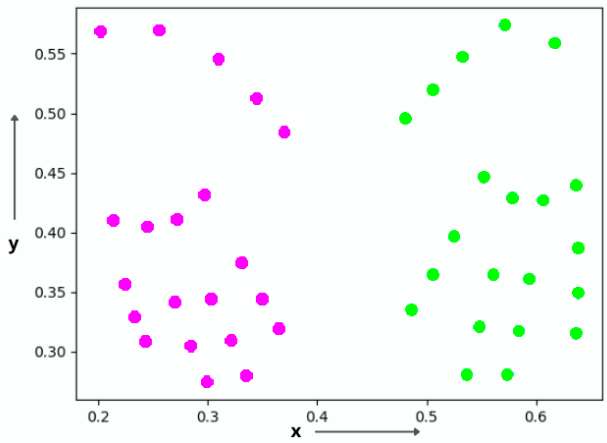

Fig. 4 Hand Landmark representation in 2D plane

regarding the hand gestures requiring two hands. A python script was written for collecting the data from all the team members. The script was capable of detecting and extracting the landmarks of one or two hands using libraries like openCV and mediapipe. These landmarks consist of the x, y and z coordinates as mentioned in section 3.1. The landmarks' coordinates were used accordingly for single-handed and double-handed gestures.

1) Dataset for single-handed gestures

***Note** - For this work, single-handed gestures are considered to be shown using only the right hand.*

For each time a gesture was shown in the camera, the Mediapipe API returned the collection of handedness of the detected/tracked hands (i.e., is it a left or right hand) and collection of detected/tracked hands, where each hand is represented as a list of 21 hand landmarks and each landmark is composed of x, y and z. Using the count of detected hands, the python script written for data collection sends the data to the function that handles one handed gestures data pre-processing. Only the x and y coordinates are used from all the 21 landmarks detected by Mediapipe. These coordinates were inserted in a list in proper order preceded by the alphabet that is represented by the gesture. 15000 such records were dumped in the CSV file for each character.

Below is an example of how the single-handed gestures data looks.

A list for one instance of the Alphabet C

[C,0.94911146,0.62622416,0.85656536,0.6220784,0.770489
93,0.5701999,0.7059876,0.536494,0.63958037,0.5170507,0.825
8685,0.3889605,0.78857744,0.27900457,0.74550724,0.2527436
3,0.7029729,0.2544515,0.86787266,0.36955082,0.8263026,0.25
575498,0.776415,0.24780819,0.72802174,0.26539296,0.908185
66,0.37579635,0.8749907,0.2651433,0.8263938,0.24938047,0.7
7778522,0.2553631,0.9495631,0.3972244,0.9301878,0.30835307
,0.8907019,0.27846164,0.84412473,0.2635035]

2) Dataset for double-handed gestures

Gestures that utilized two hands were in majority. Detecting these gestures and extracting the landmarks were done the same way as for the one-handed gestures. Only here the python script for data collection used the data in the function which handles pre-processing for two handed gestures. The z coordinates were omitted here as well. The first approach for data collection was to create a list containing the coordinates of the landmarks for both the hands. Figure 3 represents the landmarks represented in 2D plane. This resulted in a huge dataset size. For each similar landmark of two separate hands, four values were inserted in the list. Each list for a single character had 85 elements in it including the character which the gesture represents. 15000 such lists had to be dumped in a CSV file which would result in a gigantic file size.

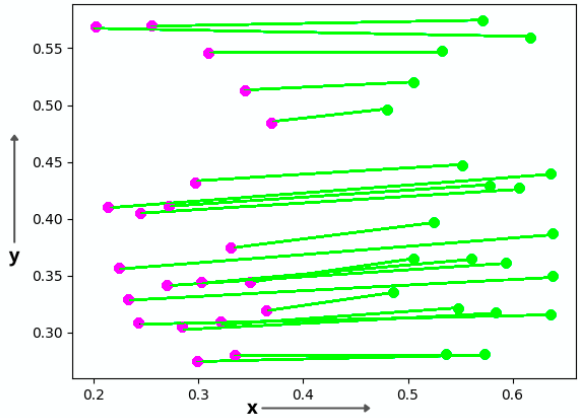

Fig. 5 Euclidean distance between similar landmarks

Better options were to be thought of. The figure below displays how Mediapipe detects the x and y coordinates and how they can be scattered in a 2D plane. The Euclidean distance between each similar landmarks of different hands were calculated as shown in figure 4.

The slope of those lines was calculated as well with respect to the x-axis.

Thus now, for each similar landmark of two separate hands, only two values were calculated instead of four. This resulted for each list of a single gesture to consist of only 43 elements including the character that the gesture represents. Below is an example of two-handed gesture lists

A list for one instance of the Alphabet A

[A,0.43240565, -0.118828712, 0.28146366, -0.076334043,
0.179769501, -0.024106368, 0.11335197, -0.001396183,
0.037270946, 0.165703438, 0.254766403, -0.011468012,
0.217078681, -0.077803008, 0.223184551, -0.018855066,
0.239964854, 0.001133083, 0.337602944, -0.058893716,
0.296307455, -0.057513888, 0.30030761, -0.042237095
,0.321719255, -0.015095411, 0.420603901, -0.082614887,
0.385677697, -0.094802589, 0.377734235, -0.061607914,
0.394410228, -0.030871268, 0.508491004, -0.105871774,
0.478781214, -0.121624107, 0.449774288, -0.102564637,
0.453709334, -0.072778826]

C. Training Machie Learning Models

Both the datasets were trained separately using various machine learning models. The following description is given considering the dataset containing the all data regarding two handed gestures.

The dataset was divided into two parts, i.e. the training data set and the testing dataset in a ratio of a:b. Then the training data was used to train a Kernel SVM model, an SVM model, Random Forest classifier model, KNN classifier, Decision Tree, Naive Bayes and Logistic Regression. Exempt the Naive Bayes model and the Logistic Regression Model, all the other models worked exceedingly well.

VI. Results and Discussion

In this proposed work, using the mediapipe, prediction of the alphabets is done. Here, Fig. 6 displays a comparative analysis of the acquired results. Among Kernel SVM, SVM, Random Forest, KNN and Decision Tree, could have been used as all of these algorithms gave good results in their training phase. As the accuracy value is best in case Kernel SVM , this has been used. The webcam feed was given as input to the Mediapipe API and using it's returned value; it was decided whether the user was showing gestures using one hand or two hands. As mentioned earlier, most of the gestures required two hands and a few used only one hand. If the user was showing gestures using two hands, the Euclidean distance between

the similar landmarks of different hands were found along with the slope that each of those lines made with the x axis. Then those distances and slopes were given to the trained Kernel SVM model and the predicted Alphabet was received as a returned value. On the other hand, if the user displayed a hand gesture using only one hand, the landmarks of that hand gesture would be sent to the trained Kernel SVM model for one hand gestures. The returned values were the predicted alphabets which were displayed on the screen. Fig. 7 shows the predicted output of hand gestures.

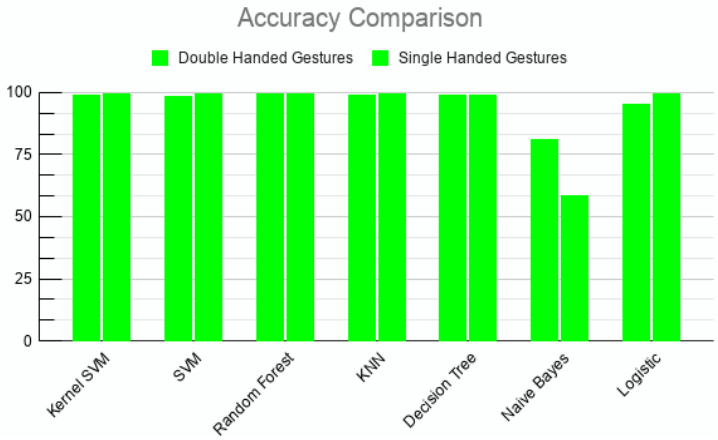

Fig. 6 Accuracy Comparison between Single and Double handed Signature

Fig. 7 Predicted alphabets

VII. Future Scope

This initiative seeks to make connectivity simple for all and will benefit from the following:

- The trained machine learning model can be integrated in a web or smartphone framework to provide an end-to-end solution that is simple to use for all.
- Other sign languages, in addition to Indian Sign Language, may be added. A functionality that encourages the user to incorporate sign languages on their own will greatly improve the accessibility and user experience.

REFERENCES

[1] Muskan Dhiman and Dr G.N. Rathna on SIGN LANGUAGE RECOGNITION, Department of Electrical Engineering, DSP Lab, Indian Institute of Science, Bangalore

[2] Ashok Kumar Sahoo on Indian sign language recognition using neural networks and kNN classifiers, August 2014, Journal of Engineering and Applied Sciences 9(8):1255-1259

[3] Dewang Sultania, "Indian Sign Language Recognition", unpiblished.

[4] Sanil Jain and K.V.Sameer Raja, "Indian Sign Language Character Recognition", Indian Institute of Technology, Kanpur.

[5] Paranjoy Paul and Dr. G N Rathna, "Real time Indian Sign language recognition", in Summer Research Fellowship Programme of India's Science Academies.

[6] Sign Language Alphabets From Around The World, www.ai-media.tv

[7] Deafness and hearing loss, www.who.int

[8] Mediapipe Hands, google.github.io/mediapipe/solutions

[9] Tripathi, K., & Nandi, N. B. G. C. Continuous Indian Sign Language Gesture Recognition and Sentence Formation. Procedia Computer Science, 54, 523–531. doi:10.1016/j.procs.2015.06.060 ,2015

[10] R. Johnson and J. Johnson, "Distinction between West Bengal Sign Language and Indian Sign Language Based on Statistical Assessment", Sign Language Studies, vol. 16, no. 4, pp. 473-499, 2016. Available: https://www.jstor.org/stable/26191231.

[11] M. Cheok, Z. Omar and M. Jaward, "A review of hand gesture and sign language recognition techniques", International Journal of Machine Learning and Cybernetics, vol. 10, no. 1, pp. 131-153, 2017. Available: 10.1007/s13042-017-0705-5

[12] Raghuveera, T., Deepthi, R., Mangalashri, R. et al. "A depth-based Indian Sign Language recognition using Microsoft Kinect". Sādhanā 45, 34 (2020). https://doi.org/10.1007/s12046-019-1250-6

[13] R. Johnson and J. Johnson, "Distinction between West Bengal Sign Language and Indian Sign Language Based on Statistical Assessment", Sign Language Studies, vol. 16, no. 4, pp. 473-499, 2016. Available: https://www.jstor.org/stable/26191231.

[14] Mustafa, Edon & Dimopoulos, Konstantinos. (2014). "Sign Language Recognition using Kinect". Avaiable: https://www.researchgate.net/publication/266144236_Sign_Language_Recognition_using_Kinect