Sign Language Detection For Deaf And Dumb People

Dr. Nirmala J Saunshimath
Assistant Professor, Dept. CSE, NMIT Bangalore, India
nirmala.saunshimath@nmit.ac.in

Shreya
Dept. CSE, NMIT
Bangalore, India
1nt20cs167.shreya@nmit.ac.in

Swathi K T
Dept. CSE, NMIT
Bengaluru, India
1nt20cs190.swathi@nmit.ac.in

Uday Kumar G
Dept. CSE, NMIT
Bengaluru, India
1nt20cs200.uday@nmti.ac.in

Parinitha P
Dept. CSE, NMIT,
Bengaluru, India
1nt20cs124.parinitha@nmit.ac.in

Abstract - People with speech and hearing impairments rely heavily on sign language in their daily lives. This group comprises approximately 1% of the Indian population. For these reasons alone, incorporating a framework that is capable of understanding Indian Sign Language would be extremely advantageous to these people. Our method in this study recognizes Indian sign language alphabets (A-Z) and numerals (0-9) in a live video stream using the Bag of Visual Words model (BOVW). It then outputs the expected labels as both text and audio. Segmentation is done based on skin colour as well as background subtraction. SURF (Speeded Up Robust Features) features have been extracted from the images and histograms are generated to map the signs with corresponding labels. The Support Vector Machine (SVM) and Convolution Neural Networks (CNN) are used for classification. An interactive Graphical User Interface (GUI) is also developed for easy access.

I. Introduction

Human life has always depended heavily on communication. Being able to communicate and engage with others is a fundamental human need. However, our perspective and the manner we communicate with people can differ greatly from those around us due to a variety of factors such as our upbringing, education, social background, and so forth. Furthermore, it is crucial to make sure that we are understood in the manner that we want.

Despite this, regular people have little trouble relating to one another and are able to express themselves clearly through speaking, writing, gestures, body language, reading, and other common forms of communication. Those with speech impairments, however, are limited to using sign language, which makes it more challenging for them to interact with the majority. This suggests the need for devices that can translate sign language into spoken or written language and vice versa, known as sign language recognizers.However, these IDs are expensive, difficult to utilise, and have limitations. The primary motivation for the development of automatic sign language recognition systems is the fact that researchers from several nations are currently working on these recognizers.

Indian Sign Language (ISL) uses both static and dynamic signals, single- and double-handed gestures, and many signs for the same alphabet in various parts of the country. It makes the introduction of such a programme exceedingly challenging. Furthermore, there isn't a common dataset available. The intricacy of Indian sign language is demonstrated by all these items.

In sign language recognition, there are primarily two methods that are employed: the sensor-based method and the vision-based method [5]. While webcameras are used in a vision-based approach to capture images or videos, gloves or other devices that recognise finger gestures and convert them into corresponding electrical signals are used in the sensor-based approach for sign determination. Signers prefer vision-based gesture detection because it doesn't require specialised technology and has the advantage of spontaneity [6]. But in a complicated environment, hand segmentation is crucial for identification. Therefore, a frameworkthat can solve this issue is proposed.

In this paper we provide an approach for developing a big, diverse, and reliable real-time system for Indian Sign Language that can recognise letters (A– Z) and numerals (0–9). Rather than relying on expensive equipment like gloves or the Kinect, we have identified indicators in these pictures byemploying camera photographs. This report also discusses the accuracy attained in the outcome. In order to facilitate communication between individuals with hearing or speech impairments and the able-bodied, real-time, accurate, and efficient judgement on ISL sign recognition is necessary.

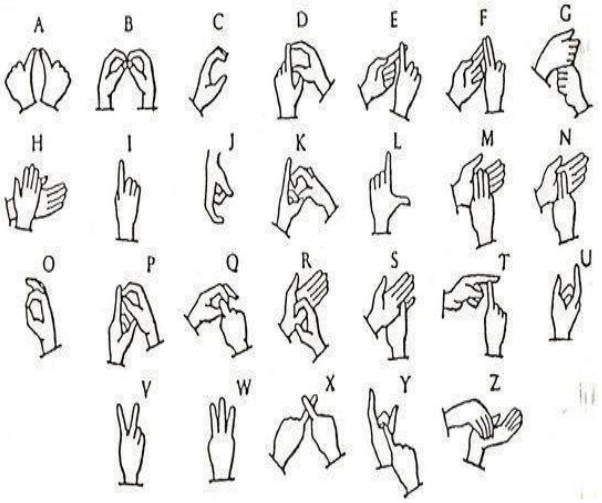

Fig 1 : Hand sign of Indian Sign Language(ISL)

II. Literature Survey

In the Paper "A Deep Learning based Indian Sign Language Recognition System" authors proposed a signer independent deep learning based methodology for building an Indian Sign Language (ISL) static alphabet recognition system. Here, they review various existing methods in sign language recognition and implement a Convolutional Neural Network (CNN) architecture for ISL static alphabet recognition from the binary silhouette of signer hand region.

In the Paper "American sign language finger spelling recognition with phonological feature- based tandem models". Authors have studied the recognition of fingerspelling sequences in AmericanSign Language from video using tandem-stylemodels, in which the outputs of multilayer perceptron (MLP) classifiers are used asobservations in a hidden Markov model (HMM)- based recognizer. They focus on recognition of one constrained but important part of the language: fingerspelling, in which signers spell out a word as a sequence of handshapes or hand trajectories corresponding to individual letters.

These works mainly relied on feature extraction, pattern recognition, and other similar techniques. But the majority of the time, a system with just one functionality is insufficient. As a result, to address this issue, hybrid approaches were presented.

After reading through these works, the authors were inspired to develop a unique dataset and an algorithm that would function flawlessly on it without compromising the video detection's accuracy. Because SURF characteristics would shorten the measurement time and make the system rotation-invariant, we chose to use them. In order to make the system usable outside of controlled situations, the paper's authors have also tackled the issue of background reliance.

III. Methodology

In order to build a highly accurate system that would be beneficial for real-time users, sign language recognition requires effective and reliable data. Here, the authors have addressed the sign detection and classification issue by utilising the specially created dataset. Dataset, Image Acquisition, Data Pre-processing, Feature Extraction, and Sign Classification are the several steps at which the data flows for sign language recognition.

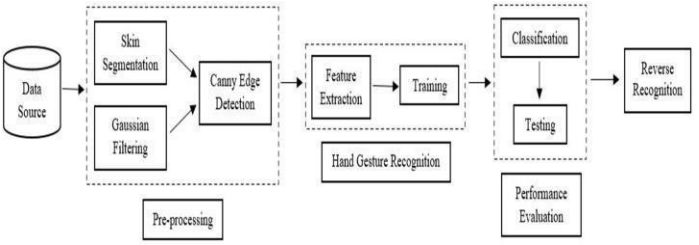

Fig 2 : Flow diagram of the application

1. Data Collection

Data collection is an essential component of research work in every field since it promotes the growth of deep learning and machine learning models. The largest obstacle we encountered when gathering data was the lack of standardized databases for Indian sign language. Consequently, we tried to manually create a dataset as part of this project in order to help us solve this issue.

Signs from the live video were transformed into frames, a pixel value threshold was used to further extract the frames. Because the generated frames had a resolution of 250*250, pre-processing requires less computing resources. There were overa thousand pictures of each sign in each folder. As a result, 36,000 photos in total from both image collecting techniques were included in the dataset. Both hands and one hand were required to make the signs. The pictures were taken with varying rotations and saved in grayscale with a .jpg extension.the dataset images can be seen in fig 3.

Fig 3 : Images captured from video

2. Pre - Processing

In this step, the image is prepared for feature extraction and detection. All of the photographs have the same dimensions in order to maintain scale homogeneity. For the photos taken against a simple background, the captured video frame is transformed into the HSV colour space. Because the skin's tint differs from the background, it may be removed with ease. The frame is then subjected to an experimental threshold that determines hue and removes pixels with skin tones from the picture. In addition, the picture is binarized, noise is removed by blurring, and the maximum contour is extracted from the output, with the assumption that the contour with the largest area signifies the hands. By using the median filter and morphological techniques, errors are further eliminated. Converted images can be seen in fig 4.

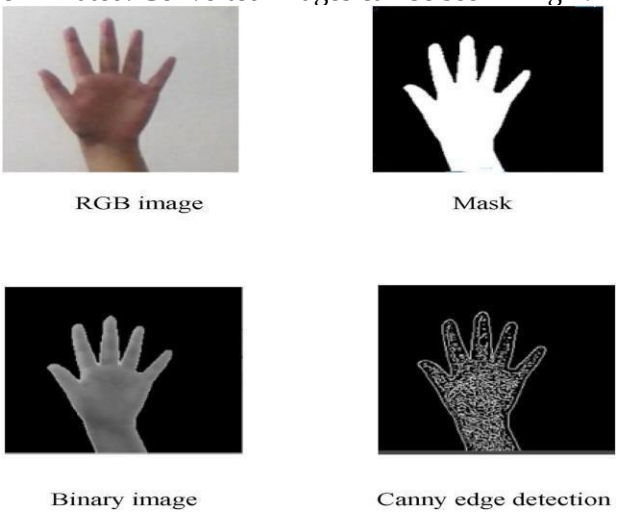

Fig 4 : image at each processing step

3. Feature Extraction

In this step, a Bag of Visual Words (BOVW) is constructed, comprising feature extraction, feature clustering, model codebook development, and histogram generation.

The definition of the popular image classification model known as the Bag of Visual Words (BOVW) was taken from data retrieval and NLP's (Natural Language Processing) Bag of Words (BOW) [26]. This involves counting the instances of each word in a text, utilising the frequency of each word to extract the keywords, and creating a frequency histogram based on that information. This concept is modified such that the features of the image are used as words in place of words. The image descriptors and key points are utilised to create a language in which each image is represented as a frequency histogram of obtained features. Later on, this frequency histogram can be used to forecast the category of another similar image. For this SURF (Speeded Up Roburst Features) is used which is feature descriptor and detector.

Image is represented as the set of characters given by SURF.

IMage = {d1,d2,d3,….,dn}

Where di are the characters like colors,shape etc. Obtained from SURF as shown in Fig 5.

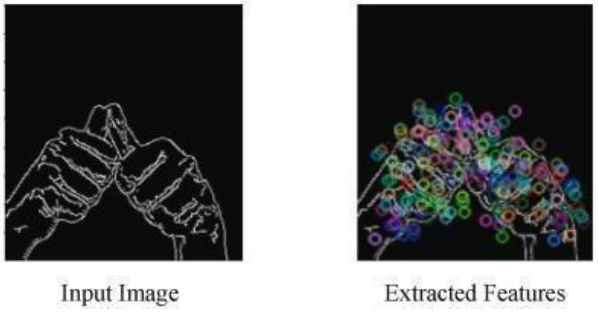

Fig 5 : Output obtained from SURF

Clustering all of the features that are produced after the SURF is applied is the next step in the feature extraction process. In order to employ the core and cluster them as the dictionary's visual keyword, related features are grouped together. Although the K-means technique can be used for clustering, we have chosen to employ micro batch K-means due to the size of the data. While it uses less memory and takes less processing time, it is similar to K-means. It eliminates the need for all of the data to be in the memory at once by using tiny random batches of fixed-size data at a time. Every iteration updates the clusters using a fresh random sample taken from the

dataset, and this process is continued until convergence.

4. Classification

After the feature detection stage we enter classification stage ,which is done using SVM(Support Vector Machine) and CNN(Convolutional Neural Network).

i. SVM(Support Vector Machine)

For classification and regression issues, the Support Vector Machine (SVM) is a supervised model that can resolve both linear and non-linear issues. It functions using the concept of decision planes, which define decision boundaries.

SVM with a linear kernel has been utilised for this classification. In order to classify and recognise ISL signs, we have fed the SVM the visual word histograms as feature vectors. 28,800 photos are used in total for the training. Following training, the classifier's performance is assessed using the testing set, which consists of 7236 images in total. The classifier's performance is measured using a number of criteria, including accuracy, precision, recall, and so on.

ii. CNN(Convolutional Neural Network).

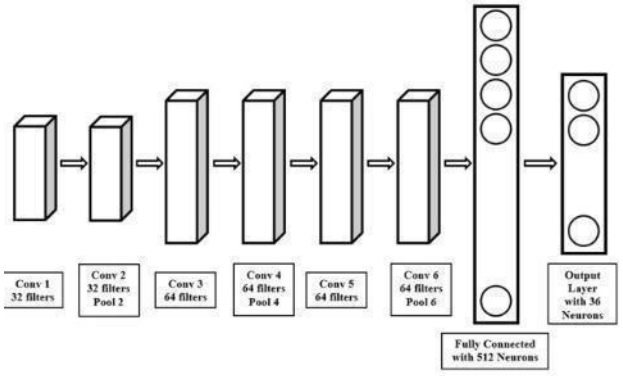

Fig 6 : CNN Architecture

CNNs are models of functional extraction inspired by the visual cortex of the human brain. When a filter map is applied to specific areas of a picture, CNNs analyse the images piece by piece. Thesesegments are referred to as features, and they are used to compare two images by identifying nearly identical features at nearly same positions. When it comes to picture recognition and classification, CNNs outperform other neural networks.

Our overall architecture, which consists of numerous convolutional and dense layers, is a fairly typical CNN architecture. Every CNN has three layers. The max-pool layer and dropout layer are placed after a set of two convolutional layers with a total of 32 filters and a 3 × 3 window size. A second set of two convolutional layers with 64 filters—a max pooling layer and a dropout layer—follows it. Lastly, there is a fully connected hidden layer with 512 neurons of the ReLU activation function and an output layer of the softmax activation function. There are also two more convolutional layers with64 filters and a max pooling layer. The Architecture design is shown in Fig 6.

5. Output Sign

The algorithm automatically converts predicted class labels which are returned as numerical vectors into text and speech. Better communication and user convenience are the goals of this. Following the classifier's identification of the label, the label is sent as a key to a dictionary, which returns the associated sign as value. The user is then shown this. Pyttsx3, a Python text to speech package, is used to convert text to speech. Threading is done because it slows down the live video stream by causing theframes to process very slowly. This makes it possible to simultaneously accomplish the translation of text to speech and the prediction of signs. This ensures that sound is continuous without disturbance.

6. Reverse recognition

To enable a dual way of communication between the hearing majority and the speech disabled, the reversal procedure is crucial in a sign language recognition system [28]. This form of communication has been incorporated into oursystem. The user provides text (English alphabets) as input in this case, which is mapped onto labels and accompanying signs (pictures from a database) that are presented to the user sequentially. The Google Speech API is used for voice recognition.

IV. Results

The Dataset collected is divided 2 parts which training dataset and testing dataset. 80% of the dataset is training dataset and 20% is testing dataset.

SVM and CNN, both produced good accuracy results on the images; however, CNN outperformed SVM with fewer features. 26 alphabets and 10

digits make up the 36 signs that the system is trained to recognise. Even though there is room for improvement, the current results are encouraging.

a) Performance of SVM and CNN

SVM reported a 99.14% accuracy rate on the test set. An overall accuracy of 99% can be seen in the computed precision and recall values of the alphabets and digits identified using SVM.
On the training set during the most recent epoch, we have seen an overall accuracy of 94% using CNN, while a testing accuracy of more than 99%. Accuracy of CNN is calculated for 50 epochs for which a ghraph is drawn as shown in figure 7.

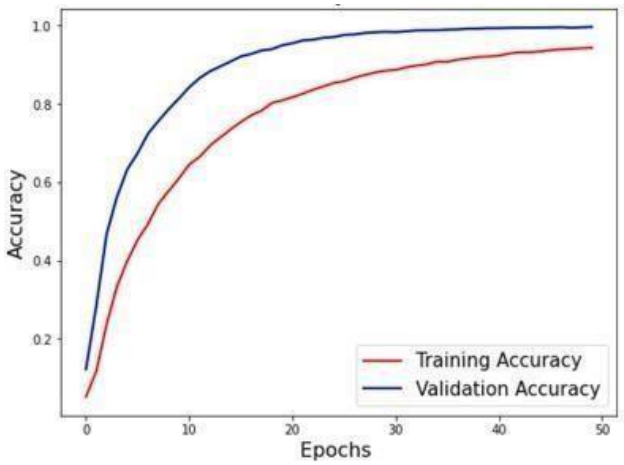

Fig 7 : Accuracy graph

Algorithm	Accuracy(in %)
SVM	99.17
CNN	99.64

Table 1 : Accuracy measure of SVM and CNN

b) Real Time Testing

An interactive GUI is created for the system's users (Fig. 8) that includes a fully functional sign-in and sign-up mechanism. By using the "predict sign" button, users can make predictions about signs using our dataset-trained model, or they can use the "create signs" button to construct their own database. Additionally, there is a voice to sign conversion tool available.

The proposed recognition approach based on the SURF features is resistant against rotation, orientation, and other effects, and has the advantage of rapid computations. Because it is a user-independent model, it can also resolve background dependence issues as long as the camera remains motionless. On the other hand, it can be used freely on plain backdrops. The backgrounds in the earlier paintings were either simple or complicated and were employed in regulated settings. For most models, the recognition accuracy is around to 0.94. But the majority of the signs that have been observed only employ one hand or basic hand motions. Our application can translate visual information into text or speech with a high accuracy of 99% and recognise double hand signs. By assisting researchers in using this approach with a standard dataset, it is a fundamental step forward in reducing some of the limitations.

Fig 8 : GUI of the Application

V. Conclusion and future work

In this paper, we presents a novel method for classifying and recognising Indian sign language signs (0–9) and A–Z using SVM and CNN. Our primary objective is to develop a more real-time recognition utility, enabling the system to be utilised in any location. It is accomplished by creating a unique data set, solving the background dependence issue, and making the system rotationally invariant. With 99% accuracy, thesystem has been successfully trained on all 36 ISL static alphabets and digits. In the future, more indicators in different languages from different nations can be added to the collection, creating a more functional framework for real-time applications. This can be further extended to simple words and expressions recognition tasks.

VI. References

[1] K. Bantupalli and Y. Xie, "American Sign Language Recognition using Deep Learning and Computer Vision," 2018 IEEE International Conference on Big Data (Big Data), 2018, pp. 4896-4899, doi: 10.1109/BigData.2018.8622141.

[2]U. von Agris, M. Knorr and K. -F. Kraiss, "The significance of facial features for automatic sign language recognition," 2008 8th IEEE International Conference on Automatic Face & Gesture Recognition, 2008, pp. 1-6, doi: 10.1109/AFGR.2008.4813472.

[3] T. Kim, K. Livescu and G. Shakhnarovich, "American sign language fingerspelling recognition with phonological feature-based tandem models," 2012 IEEE Spoken Language Technology Workshop (SLT), 2012, pp. 119-124, doi: 10.1109/SLT.2012.6424208.

[4] D. Li, C. R. Opazo, X. Yu and H. Li, "Word-level Deep Sign Language Recognition from Video: A New Large-scale Dataset and Methods Comparison," 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), 2020, pp. 1448-1458, doi: 10.1109/WACV45572.2020.9093512.

[5] S. C.J. and L. A., "Signet: A Deep Learning based Indian Sign Language Recognition System," 2019 International Conference on Communication and Signal Processing (ICCSP), 2019, pp. 0596-0600, doi: 10.1109/ICCSP.2019.8698006.

[6] Shagun Katoch, Varsha Singh, Uma Shanker Tiwary, Indian Sign Language recognition system using SURF with SVM and CNN, Array, Volume 14, 2022, 100141, ISSN 2590-0056

[7] Y. Liao, P. Xiong, W. Min, W. Min and J. Lu, "Dynamic Sign Language Recognition Based on Video Sequence With BLSTM-3D Residual Networks," in IEEE Access, vol. 7, pp. 38044-38054, 2019, doi: 10.1109/ACCESS.2019.2904749.

[8] J. Huang, W. Zhou, H. Li and W. Li, "Attention-Based 3D-CNNs for Large-Vocabulary Sign Language Recognition," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 29, no. 9, pp. 2822-2832, Sept. 2019, doi: 10.1109/TCSVT.2018.2870740.

[9]V. Slavuj , B. Kovacic and I. Jugo Department of Informatics of Rijeka, Croatia –"Intelligent Tutoring System for Language Learning"-May 2015

[10]G. Ananth Rao, P.V.V. Kishore Department of Electronics and communication Engineering-"Selfie video based continuous Indian sign language recognition system" – February 2017

[11] Satheesh Kumar Raju, Anil Kumar G S, Sunny Arokia Swamy Department of electrical and electronics Engineering-"Double Handed Indian Sign Language to Speech and Text"-2015

[12] Oscar Koller, Simon Hadfield, and Richard Bowden-"Multi channel Transformers for Multi-articulatory Sign Language translation."-2020

[13] V.V. Kishore, M.V.D. Prasad, Ch. Raghava prasad, R Rahul-" 4-Camera Model for sign language recognition using Elliptical Fourier descriptors and ANN"-2015

[14] Tripathi, Neha, G. C. Nandi-"Continuous Dynamic Indian Sign language Gesture Recognition with Invariant Backgrounds"-2015

[15] Daleesha M. Viswanathan, Sumam Mary Idicula "Recent developments in Indian sign language recognition: an analysis"

[16] Anuja V. Nair, V. Bindu "A review on Indian sign language recognition"

[17] K.K. Dutta, S.A.S. Bellary "Machine learning techniques for Indian sign language recognition"

[18] Garima Joshi, Vig Renu, Sukhwinder Singh "DCA-based unimodal feature-level fusion of orthogonal moments for Indian sign language dataset"

[19] K. Manikandan, Ayush Patidar, Pallav Walia, Aneek Barman Roy "Hand gesture detection and conversion to speech and text"

[20] J. Sivic, A. Zisserman "Video Google: a text retrieval approach to object matching in videos"