

Hindi Sign Language Detection Using Cnn

**Prof. Pragya Sinha¹, Ifham Khwaja², Kaushik Rathod³,
Naman Sanklecha⁴**

¹Assistant professor and Guide, Vishwakarma University, Pune, Bachelor in Technology in Computer Engineering

^{2,3,4}Student, Vishwakarma University, Pune, Bachelor in Technology in Computer Engineering

Abstract:

Significant challenges arise for individuals who are deaf and mute, as effective communication is crucial in today's world. Bridging the communication gap is of utmost importance, and advancements in machine learning offer a solution. This research focuses on developing a Hindi sign language detection system to address the needs of Hindi speakers in India, where Hindi is the most spoken language. By enabling communication in Hindi through sign language, this system ensures that individuals who are not proficient in English can fully participate in society and access education effortlessly. Existing systems primarily cater to American Sign Language (ASL) and Indian Sign Language (ISL), leaving a gap for Hindi sign language. Through the proposed methodology and leveraging machine learning techniques, the system aims to revolutionize the lives of the deaf and mute, empowering them to express themselves, interact seamlessly, and be understood by a wider audience. By addressing this crucial problem, the research contributes to inclusivity, accessibility, and equal opportunities for Hindi speakers.

Keywords: CNN, OpenCV, pooling layer, VGG, Epochs

INTRODUCTION

In a world where effective communication is crucial, imagine the challenges faced by individuals who are both deaf and mute. Everyday interactions, accessing education, and expressing themselves become arduous tasks. However, advancements in technology, particularly in the field of machine learning, present an incredible opportunity to bridge the communication gap. Sign language detection using machine learning can revolutionize the lives of the deaf and dumb, enabling them to communicate effortlessly and participate fully in society. Previously people have worked on this technology, but they have mostly worked on ASL (American Standard Language) which a world-wide accepted language, it also has an Indian alternative (Indian Standard Language) which is widely used in India but Hindi is the most spoken language in India so there is a need for Hindi Sign language. This language is not very popular like ASL but many Indians cannot understand the English language. So it is important to address this problem.

MOTIVATION

Hindi is the most spoken language in India but the sign language system are so far developed for only ASL and ISL, we want to develop a Hindi sign language detection system so that people can

communicate in Hindi through sign language as many people in India don't understand English. Few systems have been developed for HSL but they are not quite efficient.

LITERATURE REVIEW

Previously many people have worked on this project and published papers on it. We studied the papers and studied the methodologies and the drawbacks associated with them. The sign detection system [1] The paper is about using machine learning for Indian Sign Language recognition and explaining the process behind it. Captures motion information. The UCI skin segmentation dataset (with roughly 2,00,000 points) was used for training, and they used a webcam to collect the data based on draft photographs. Pre-processing includes image adjustments such as cropping, filtering, brightness, and contrast. Such a process employs picture augmentation, image cropping, and image segmentation. In this plan, 7Hu movement is used and 7 movement is found. The gesture database was created using this. The data is then fed into a supervised machine learning algorithm. The algorithms used are support vector machines, random forests. The main disadvantage is that it only works with numbers, not words and letters of the ISL language. [2] In this study, they study learning Vietnamese using Google's Mediapipe platform. Simple recurrent neural networks (RNNs) can recognize wrist, hand, and finger movements. By using OpenCV, various movements in the video are recorded and divided into frames. To process data, we employ Long Short Term Memory (LSTM). The Mediapipe framework is largely responsible for the research's accuracy.

The outcomes are better the more accurate Mediapipe's practical experience is. The precision of Mediapipe's hand detection is also impacted by changes in position and data size.

The downside is that it has very low accuracy (0.75), we will work to improve it. [3] The system has five flexible sensors (one for each finger) and each has a voltage divider to measure them (only resistors are used). Arduino Lilypad for reading sensors (one worn with microcontroller). It is also designed as a lithium battery charger with all its accessories: 5 flexible sensors, Arduino Lilypad, five 10 ohm resistors, bluetooth hc-05 and 3.7V to 700mAh Li-polymer battery. It uses the PS algorithm CHC and DROP algorithm to train the data and sensors in the glove to detect hand movement and capture data. One of the downsides is that it requires gloves to use and is difficult to carry around. [4] In this paper, These images were obtained with a real-time camera feed and used in this study using OpenCV. They create a region of interest (ROI), which ought to be where our hands are seen. The ROI box is 200 * 200 pixels as well. By calculating the background's average weight and deducting it from the frame, we can begin manually inspecting the image. Then, they converted the black-and-white images that we had originally captured with the BGR using cv2, which they did by calling `cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)`. The next step involves using the `cv2.findContours()` function to locate the largest contour. The function accepts three parameters: image position, contour take mode, and contour approach method. Their output is a modified image with contours and hierarchy. They then created a CNN model from these images. One drawback is that it's only trained with a limited number of letters and not all are included. Accuracy is also lower than other models.

METHODOLOGY

1. Data collection:

We can take videos of hand gestures or Search for existing Hindi sign datasets online We have to ensure that the dataset contains a sufficient number of diverse sign images with different backgrounds, lighting conditions, and perspectives.

a. Implement sign detection with OpenCV:

- a. Load the pre-trained CNN model that you saved after training.
- b. Capture or load an input image containing Hindi signs.
- c. Preprocess the input image by resizing it to the same size used during training and normalizing the pixel values.
- d. Use the pre-trained CNN model to classify the regions of interest (ROIs) within the image.
- e. We will use Sliding window technique to identify ROIs:
- f. For each ROI, pass it through the CNN model and obtain the predicted class probabilities.
- g. Set a threshold probability value, above which you consider the detection as positive.
- h. Mark the ROI on the original image with a bounding box if the predicted class probability above the threshold, indicating a successful detection.
- i. Repeat this process for all ROIs in the image.
- j. Finally, display the original image with the bounding boxes around the detected Hindi signs.

b. Preprocess the dataset by performing the following steps:

- Resize the images to a consistent size. For example, you can resize them to 224x224 pixels.
- Normalizes image pixel values between 0 and 1. It divides each pixel value by 255 to complete the normalization.

2. Data Preprocessing:

- a. Convert the video dataset into individual frames.
- b. Resize the frames to a consistent resolution suitable for processing.
- c. Normalize the frames to enhance contrast and reduce variations in lighting conditions.
- d. Annotate the videos with corresponding labels indicating the sign language gesture being performed.

3. Feature Extraction:

- a. To find and localize hand gestures in video frames, use hand detection algorithms. You can utilize models that have already been trained to recognize hands, such as Single Shot MultiBox Detector (SSD) or You Only Look Once (YOLO).
- b. Employ hand tracking algorithms, such as the Kanade-Lucas-Tomasi (KLT) tracker or OpenCV's built-in tracking algorithms, to track hand movements across consecutive frames.
- c. Extract hand shape information by separating the hand region from the tracked bounding box and describing it with scale-invariant feature transform (SIFT) or histograms of oriented gradients (HOG) descriptors.
- d. Capture hand motion features by computing optical flow, which tracks the movement of hand pixels across frames. Techniques like Lucas-Kanade or Dense Optical Flow can be used for this purpose.

4. Training the model:

- a. The data is broken down into three categories: training, validation, and testing.
- b. The CNN model will be trained using the training process, the hyperparameters will be tuned using the validation process, and performance will be tracked during the training process. The CNN model's performance will then be evaluated using the testing process.
- c. About 70% of the data are used for training, 15% are used for validation, and 15% are used for testing.
- d. Design the architecture of the CNN model. We can start with a relatively simple architecture and increase its complexity if needed.
- e. A typical CNN design comprises of a pooling layer for spatial down sampling, numerous convolutional layers for classification, and fully linked layers.
- f. We will use deep learning frameworks Keras to define and train the model.
- g. Compile the model by specifying the loss function as `sparse_categorical_crossentropy`, optimizer as `sgd`, and evaluation metrics as accuracy.
- h. Apply the training dataset to the model. To enable the model to learn the features and patterns in the photos, iterate through the dataset numerous times in multiple epochs (passes through the complete dataset).
- i. Monitor the model's performance using the validation dataset during training to identify overfitting or underfitting.
- j. Adjust hyperparameters like batch size, number of layers) based on the validation performance to improve the model's accuracy.

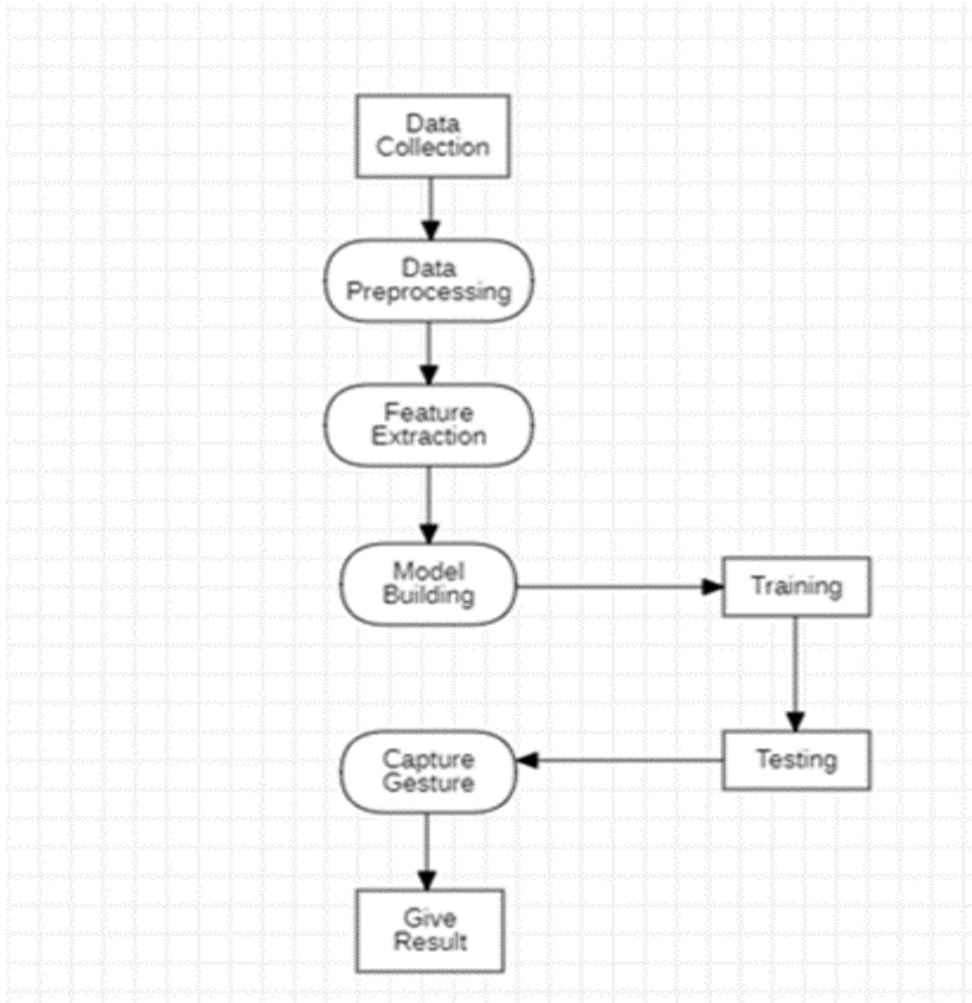
5. Evaluate the model:

- c. After the model has been trained, test it on a dataset with unused data to determine how well it performed.
- d. To measure the model's effectiveness on the testing dataset, we will utilize metrics like accuracy, precision, recall, and F1-score.
- e. These metrics will help us understand the model's overall performance and identify any areas for improvement.

6. Fine-tuning and optimization:

- a. If the initial performance of the sign detection system is not satisfactory, you can consider various techniques to improve it.
- b. Increase the size of the training dataset by collecting more images or using data augmentation techniques such as random rotations, translations, or flips.
- c. Modify hyperparameters to enhance the performance of the model, such as learning rate, batch size, or number of layers.
- d. Modify the CNN architecture by adding more layers, changing the number of filters, or using different types of layers (e.g., dropout, batch normalization) to improve accuracy.
- e. Use a pre-trained model (like ResNet, VGG, or MobileNet) that has been trained on a large dataset and fine-tune it using your dataset of Hindi signs to inculcate transfer learning.

- f. Experiment with different optimization algorithms and techniques to improve training efficiency and convergence.



Block Diagram of proposed System

Conclusion

In conclusion, it can be said that this technology can help the blind and dumb people communicate with people. This technology to be aligned for hindi language as it is already available for English language. Through our research we read several research papers and studied their approach and the drawbacks they had. We developed an approach that can help us to rectify our system and solve the problems.

References

1. Sign Language Recognition Based on Intelligent Glove Using Machine Learning Techniques Paul D. Rosero-Montalvo^{1,2} , Pamela Godoy-Trujillo¹ , Edison Flores-Bosmediano¹ , Jorge Carrascal-García³ , Santiago Otero-Potosi³ , Henry Benitez-Pereira³ and Diego H. Peluffo-Ordoñez ~
2. Search Vietnamese using Mediapipe Khat Duy Bach ICT Department, FPT University, Hanoi, Vietnam Phung Thai Duong ICT Department, FPT University, Hanoi, Vietnam Bui Ngoc Anh ICT Department, FPT University, Hanoi, Vietnam. Ngo Tung Son Department of ICT, FPT University, Hanoi, Vietnam.

3. Sign Language Recognition Based on Intelligent Glove Using Machine Learning Techniques Prof. Radha S. Shirbhate¹, Mr. Vedant D. Shinde², Ms. Sanam A. Metkari³, Ms. Pooja U. Borkar⁴, Ms. Mayuri A. Khandge⁵
4. SIGN LANGUAGE RECOGNITION FOR HINDI VARNAMALA USING CNN Pratibha Gupta¹, Priya Rajput², Priyanka Katiyar³, Srishti Sharma⁴, Shaivya Shukla⁵, Dr. Umesh Dwivedi⁶,^{1,2,3,4,5}Student, Computer Science And Engineering , Babu Banarasi Das Northern India Institute Of Technology, Lucknow-226028, India. ⁶Associate Professor, Computer Science And Engineering, Babu Banarasi Das Northern India Institute Of Technology, Lucknow-226028, India.