SIGN LANGUAGE DETECTION SYSTEM

Gayatri Dube, Venkata Shreya Appalla

Guide by Jaishankar natarajan

School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India

**Abstract**

This paper presents a system to aid people having vocal or hearing disabilities to communicate. It discusses a method of developing a sign language recognition system which detects, recognises, and translates hand gesture into text. The algorithm can extract and recognise signs from static and dynamic backgrounds using OpenCV libraries. The dataset thus collected is trained using CNN- a machine learning algorithm- using Tensorflow and Keras.

Keywords – Sign Language, machine learning, CNN, OpenCV, Tensorflow, Keras

**Introduction**

Communication is a vital part of one’s life without which, going about the daily life is tedious. People who have trouble speaking or have difficulty in hearing express themselves using sign language. Usage of hand gestures to communicate is called sign language. Sign Language involves fingerspelling which spells out letters, numbers, and words character by character. It also involves combining movement of arms, body, and hand shapes. Sign language has its own vocabulary and grammar rules like any other language and share a few similarities with other languages.

Since a vast majority of people lack the knowledge of Sign Language and hence it is an inconvenient method of communication. It is a rather cumbersome method of communication and cannot be used or understood by most people especially in emergencies.

Hence, an evident solution to this problem is present in the world of Machine Learning and Image Detection. Development of a Sign Language Recognition System detects hand gestures and can solve this issue. Various machine algorithms can be used to build such a system for effective communication without any barriers. By providing a low-cost solution, this paper aims to bridge the gap between hearing impaired people and us by enabling to interpret the signs without a professional interpreter.

**Literature** **Review**

Number of authors have used various techniques, depending on the nature of sign language and the signs.

Shashank Salian et. al [3] proposed a real time system using machine learning algorithms such as CNN to recognise sign language. OpenCV along with Tensorflow and Keras were used for image processing and training the model. The dataset used for image processing in the project was from Massey University. This dataset contains exhaustive amount if data and contains png images in RGB mode which were pre-processed the images were resized to an aspect ratio of 1:1 using imagemagik tool for training the CNN. The authors of this paper proposed using SGD (Stochastic Gradient Descent) to reduce the cost of training the dataset using CNN. The final step was to process the real time images from the webcam using OpenCV to get the image of the hand by employing various techniques such as Convex Hull and Background Subtraction and applying similar image pre-processing techniques as done on the dataset for obtaining the result i.e., the translated meaning of the hand gesture.

Another approach proposed by Priyanka C Pankajakshan and Thilagavathi B [4], uses ANN as a supervised learning algorithm for pattern recognition. The feed forward type of ANN architecture was used for gesture recognition. The project was implemented in the 2014 version of Matlab.

The study done by Ahmed Kasapbasi et. al [2] differs from other research work as a custom data set was consisting of gestures from the American Sign Language. The dataset tried to include different illumination conditions and hence the dataset was collected at different times of the day, with varied distance from the camera as well. The study proposed by the authors uses CNN. OpenCV libraries were used for capturing real time images from the webcam and for displaying the translated output. The network is trained using the Backpropagation algorithm and Tensorflow and Keras were used to implement the system. The study achieved an accuracy of 99.38%.

Anup Kumar et. al.[5] have carried out the project based on two modules: the first module was ‘gesture to speech module’ and there second module was ‘speech to gesture module’. The Gesture to Speech Module they have taken takes gesture as an input in the form the live dataset and then they produce the output as speech according to the sign given in the dataset for this, they have SVM classifier. Whereas the second module works the other way around, using the user's speech as the input and producing the equivalent image or video as the output by using whereas the second module works the other way around, using the user's speech as the input and producing the equivalent image or video as the output.

Joyeeta Singh and Karen Das [6] research suggests a new method for deciphering Indian Sign Language, which considers live dataset of the signs. The three stages of the proposed system are pre-processing, feature extraction, and classification. Histogram matching and skin filtering are included in the pre-processing stage. Eigen values and Eigen Vectors were taken into consideration during the feature extraction stage, and lastly Eigen value weighted Euclidean distance is used to detect the sign.

Anup Nandy et. al.[7] proposes the same but uses K-nearest neighbour metrics and Euclidean distance for testing pattern on the features vector as Very accurate classification results are provided by K nearest neighbours.

**Materials and Methods**

Sign language can be detected using computer vision-based techniques or sensor-based systems. Sensor based systems require the users to wear gloves which transforms the user’s hand movements into useful data. In the proposed work, the system simply uses a webcam for the acquisition of images of the hand gestures, allowing human-computer to contact without the need for any additional hardware.[1]

The webcam is used to create the database required for training the model as well as for capturing real time images of the hand gestures and comparing them to the dataset so that the system can predict the output.

Data acquisition and pre-processing

Feature extraction

Gesture Classification

Sign Recognised in the form of text

The proposed system is to develop sign language recognition system. For recognition of sign language we have used the OpenCV, Tenserflow.

For data acquisition, live dataset are captured form webcam using OpenCV and Python. Then the labelled map is created which contains the each sign (number) along with their respective id. This label map will have 11 different signs which are label from 0 to 10.

In Feature Extraction the data is collected in the from the webcam, then it will subtract the background and will extract the sign given in the live dataset after feature extraction gesture classification is done

In gesture classification the tranining and testing of the data is carried out. Then the traning and the testing of the data are then converted into Tenserflow records using generate tfrecords, which is subsequently used to train the tenserflow object Detection API Tenserflow Object Detection is the an oen source frameworkmake it simple to create, train and deploy an object identification model. After training the dataset the model will be ready for testing. Then the detection is done by OpenCV and webcam. This system will detect the gesture and convert the gesture into text form

**Dataset**

Sign language can be detected using computer vision-based techniques or sensor-based systems. Sensor based systems require the users to wear gloves which transforms the user’s hand movements into useful data. In the proposed work, the system simply uses a webcam for the acquisition of images of the hand gestures using the OpenCV libraries, allowing human-computer to contact without the need for any additional hardware.[1]

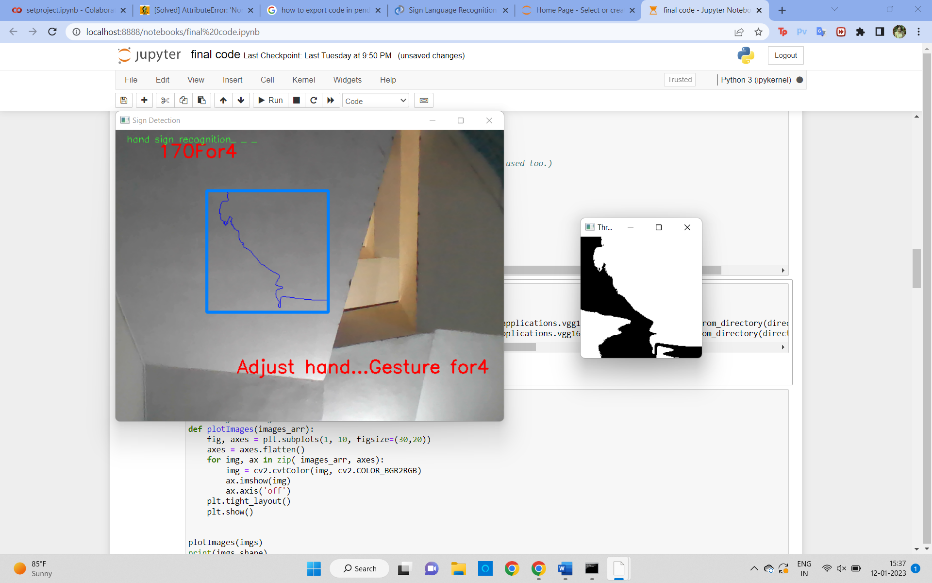


Figure 1

The blue box on the screen in the Region of Interest which detects the hand gesture.

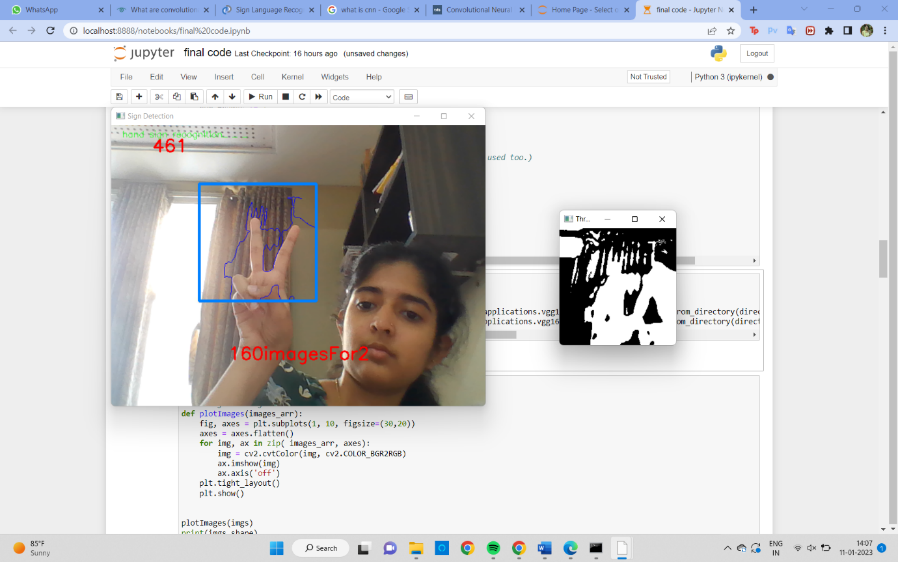


Figure 2

In the example above, the dataset for the number 2 is being created which will be saved in the train folder.

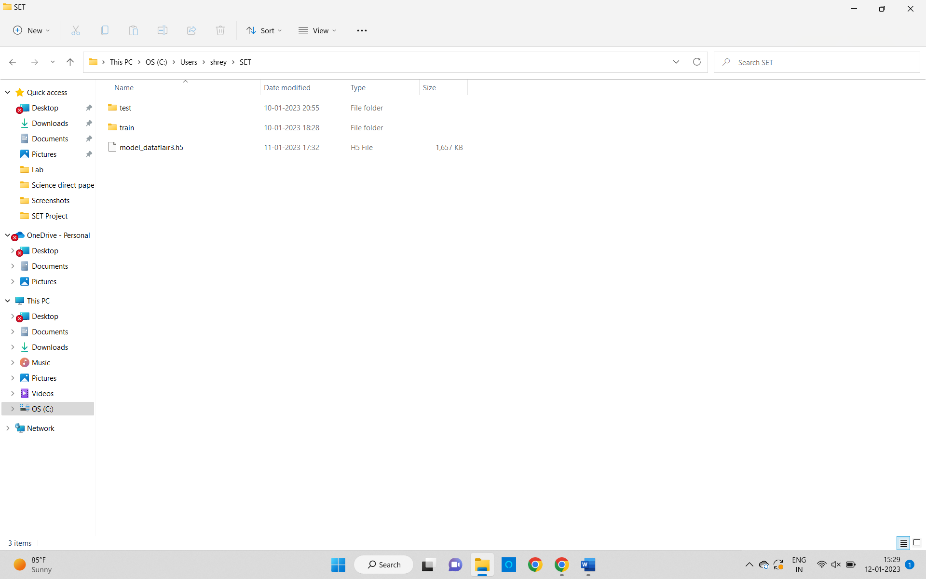


Figure 3

Once the images are captured, they will be saved in a directory containing two folders namely test d train.

Graphical user interface, text, application

Description automatically generated

Figure 4

Each of the folder will further contain 10 folders consisting of the images of the numbers from 1 to 10.

Table

Description automatically generated

Figure 5

The webcam is used to create the database required for training the model as well as for capturing real time images of the hand gestures and comparing them to the dataset so that the system can predict the output.

For training the dataset, we have captured and saved 300 images of each number. As for the test dataset, we save 40 images of each number.

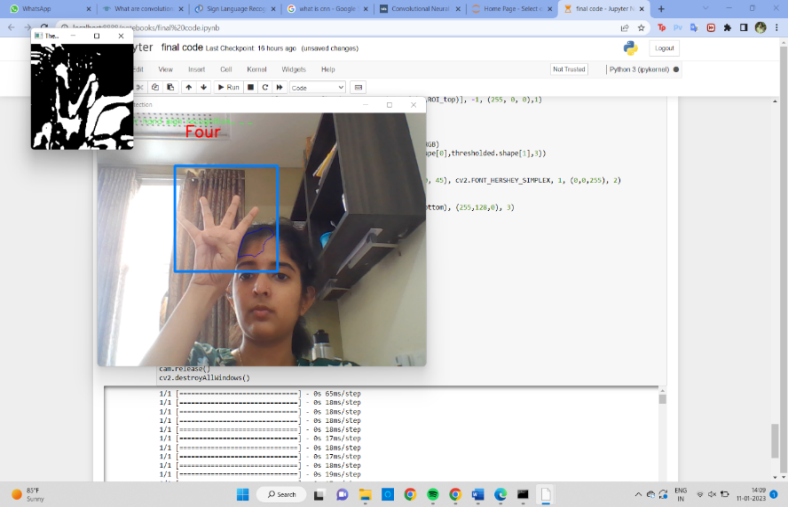
**Experimental** **Procedure**

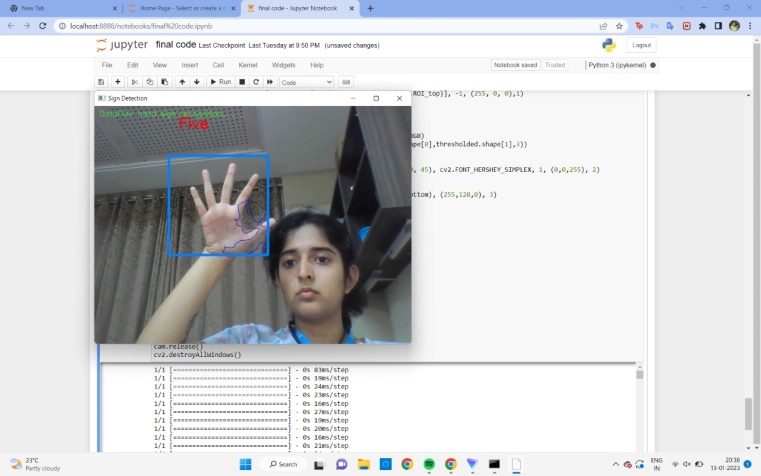
After creating the dataset, now we will train the CNN model

We are now training a CNN on the newly produced data set. First, we load the data using Keras' ImageDataGenerator, which allows us to utilise the flow from directory function to load the train and test set data, and each of the number folder names will be the class names for the images loaded. Reduce LR on plateau and early stopping are used in training call-backs, and both are reliant on validation dataset loss. The accuracy and loss are calculated using the validation dataset after each epoch, and if the validation loss is not decreasing, the LR of the model is reduced using the Reduce LR to prevent the model from overshooting the loss minima, and we also use the early stopping algorithm so that if the validation accuracy continues to decrease for some epochs, the training is stopped. he example includes the call-backs as well as the two optimization techniques used - SGD (stochastic gradient descent, which means the weights are modified at each training instance) and Adam (a combination of Adagrad and RMSProp). The example includes the call-backs as well as the two optimization techniques used - SGD (stochastic gradient descent, which means the weights are modified at each training instance) and Adam (a combination of Adagrad and RMSProp). We discovered that the model SGD seemed to provide higher accuracies. As we can see, throughout training, we achieved 100% training accuracy and approximately 81% validation accuracy. After creating the model, we fit it to the train batches for ten epochs using the call-backs outlined above.

**Results** **and** **Discussion**

Like creating the live dataset, a blue box is created which is the Region of Interest (ROI) using OpenCV libraries which will detect the hand gesture. Next, the max contour is calculated. The detection of the contour indicates the presence of a hand in the ROI. In figure 6, we can see that the model is successfully able to detect the number and in red colour it is also presenting the number in text format





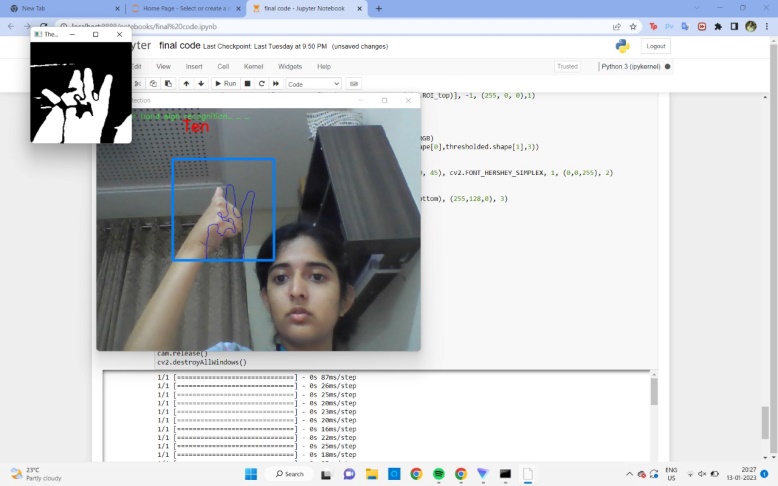


Figure 6 Predicting the hand gesture

**Conclusion**

A system to recognize sign language is developed using Convolutional Neural Network. Neural Network Systems are used to extract features from the image gestures that are given to the system as the input followed by the application of segmentation to extract the shape of the hand. A ROI is created with varied backgrounds while collecting images of the hand gesture. A database is created for ten numbers of with each number consisting of 300 images for the training dataset and 40 images each for the testing dataset. These collected images will be compared to the real time images collected from the webcam to get the translated (sign language to text) output.

**References**

1. Shagun Katoch et.al., “Indian Sign Language recognition system using SURF with SVM and CNN”, 2022, Elsevier Inc.
2. Ahmed Kasapbasi et.al. , “DeepASLR: A CNN based human computer interface for American Sign Language recognition for hearing-impaired individuals”,2021, Elsevier Inc.
3. Shashank Slain et.al., “Proposed System for Sign Language Recognition”, 2017,IEEE
4. Priyanka C Pankajakshan, Thilagavathi B, “Sign Language Recognition System”, 2015, IEEE
5. Anup Kumar et.al., “Sign Language Recognition”, 2016, IEEE
6. Joyeeta Singha, Karen Das, “Recognition of Indian Sign Language in Live Video”, Volume 70-No.19, 2013, International Journal of Computer Applications
7. Anup NAndy et.al., “Recognition of Isolated Sign Language Gesture in Real Time”, 2010, Springer-Verlag Berlin Heidelberg