Sign Language Conversion For Hearing And Speech Impaired

Diksha Bendale, Kunal Goel, Rutuja Chaudhari, Supriya Thorat,
Under Guidance of-
Mr Keshav Tambre, Asst. Prof. at IsquareIT
Department of Information Technology,
International Institute of Information Technology
Pune, India

***Abstract*:: Inability to talk is taken into account to be a real disability. People with this disability use different modes to speak with others, there are several methods available for his or her communication, one such common method of communication is dactylology. Sign language allows people to communicate with human body language; each alphabet has a set of human hand signs representing a particular expression. The purpose of the paper is to convert sign language to text and sentences with human gesture classification. Sign Language Recognition is one among the foremost growing fields of research. Many new techniques have been developed recently within this area. Sign Language is especially used for the communication of deaf-mute people. This paper shows the sign language recognizing 26 hand gestures in Sign language using CONVOLUTIONAL NEURAL NETWORK. The proposed system contains three modules such as Image capture module which contains image capturing using webcam and database formation, the feature extraction module having colour detection and shape detection, the pattern generation module having trained model and sign recognition. The final output will be predicted text from the sign expressed by the user. KEYWORDS: Sign Language, Hand Gesture Recognition, Image Processing.**

I. INTRODUCTION

As computer innovation keeps on developing, the need for characteristic correspondence amongst people and machines additionally increments. Even though the mouse is exceptionally valuable for gadget control, it can be badly arranged to use for physically disabled individuals and individuals who aren't acquainted with utilizing the mouse for connection. Sign Language is the most natural and expressive way for hearing impaired people. People, who are not deaf, never try to learn sign language to interact with deaf people. This leads to isolation of the deaf people. The difference between normal people and the deaf community can be minimized if the computer can be programmed in such a way that it can translate sign language to text format. The strategy proposed in this paper makes utilization of a webcam through which hand gestures given by the user are captured and identified accordingly.

There are various categories in sign language like ISL (Indian Sign Language), ASL (American Sign Language), BSL (British Sign Language), etc. But none of the sign languages is universal or international. Here this proposed system can recognize the various alphabets of Sign Language; this will reduce the noise and give accurate results. The research problem in computer recognition is sign language for enabling communication with hearing-impaired people. This system introduces efficient and fast techniques for the identification of the hand gesture representing an alphabet of the Sign Language.

Human gestures are a vital sign of human communication and an attribute of human actions informally called body language. A lot of methods are being used to track human gestures. To get maximum accuracy and to bring out the system unique plenty of methods are attempted and the best case is user-defined actions (gestures) to manage the system. For example, consider a person who cannot speak and wants to say “Hello” to a group of people who don't know sign language. The user stands in front of the system and waves the hands and the system throws out the speech “HELLO”. These signs are processed for feature extraction using some color model. The extracted features are compared by using a pattern-matching algorithm. The features are compared with a testing database to calculate sign recognition. Finally, the recognized gesture is converted into text. This system provides an opportunity for deaf-dumb people to communicate with non-signing people without the need for an interpreter

II. PREVIOUS WORK

In the papers [1], [2] several types of Artificial Intelligence-based models have been proposed with several types of their implementations. The implementations involve a lot of technologies needed to be installed

A. Glove Approach

In this approach, the person who is giving the signs wears a glove. In this, the hand gesture signal is in the form of analogue and is converted into digital form. The sensor in this glove is used to detect hand gestures and is converted into code. The drawback of this approach is the signer has to wear sensor hardware along with the glove and hence this approach is not a cost-effective approach.

B. Leap Motion Controller

It is a sensor that detects hand movement and converts that signal into computer commands. It consists of two IR cameras and three infrared LED's. LED generates IR light signal and the camera generates 300 frames per second of reflected data. These signals are sent to the computer through a USB cable for further processing.

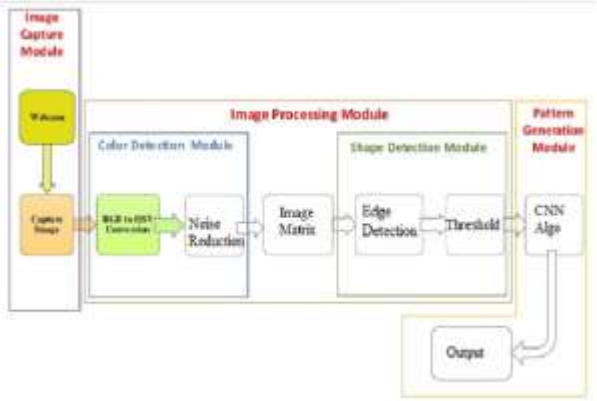
III. SIGN LANGUAGE CONVERSION FOR HEARING AND SPEECH IMPAIRED

A. Proposed Model

The proposed model is based on a vision-based approach as other approaches are not cost-effective as they need external hardware.

The paper presents a model which captures images from the web camera, the image frame through which the desired input has to be taken can be manipulated from the software. The image is then processed and further converted into grayscale using OpenCV. An image in which the only colors are shades of grey is a Grayscale image. Less information needs to be provided for each pixel to make the feature extraction from the images easier, thus there is a need for differentiating such images from colored images. The image is then processed to recognize the gestures through image processing algorithms like CNN used in this model. CNN's are used for image classification and recognition due to their high accuracy. The gestures are then recognized according to the sign given by the user and the text is then displayed on the screen.

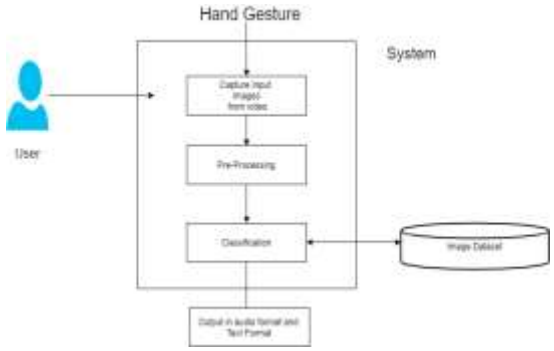
B. Architecture



- **Image Capture Module**
  > Collecting hand gestures using a webcam by capturing images.
- **Image Processing Module**
  > Colour detection
  > Shape detection
- **Pattern Generation Module**
  > Training, Testing and output.

We apply a 2D CNN model using a Tensorflow library. The images are scanned with a filter of size 3 by 3 using the convolution layer. To extract important features from the input image, the dot product between the frame pixel and weights of the filter is calculated. After each convolution layer, we apply the pooling layers. Each pooling layer decreases the amount of activation map present in the previous layer by merging all the features that were learned in the previous layers' activation maps. Thus, it generalises the features represented by the network and reduces overfitting of the training data. In our case, the activation function is a Rectified Linear Unit and the input layer of the convolutional neural network has feature maps of size 3 by 3. The dropout is set to 50 per cent and the layer is flattened. The last layer of the network contains an entirely connected output layer with ten units, and the activation function is Softmax. Then lastly to compile the model we use Adam as the optimiser and category cross-entropy as the loss function.

C. Working



Data Processing: The load_images.py script contains functions that will load the Raw Image Data and at the same time save the images as NumPy arrays into the file storage of the system. By applying filters and ZCA whitening to reinforce features, the data of the loaded image from load_images.py will be preprocessed through resizing. The processed image data will be split into training, validation, and testing data and written to storage during the training phase. The fun_util.py script loads the relevant data split into a dataset class in the training phase. For use of the trained model in classifying gestures, a private image is loaded and processed from the filesystem.

Training: The model is trained using hyperparameters obtained from a config file that lists the training rate, batch size, image filtering, and several epochs. The configuration used to train the model is saved alongside the model architecture for future evaluation and tweaking for improved results. Using Adam Optimizer with Cross-Entropy Loss the model is trained so that the training and validation datasets are loaded as Data loaders. The model with the best validation accuracy is saved to storage for further evaluation and use by evaluating every epoch on the validation set. Upon finishing training, the training and validation error and loss is saved to the disk, alongside a plot of error and loss overtraining.

Classify Gesture: After a model has been trained, it's often used to classify a replacement ASL gesture that's available as a file on the filesystem. The user inputs the file path of the gesture image and therefore the test data.py script will pass the file path to process data.py to load and equivalently preprocess the file because the model has been trained.

D. Methodology

The system proposed is a vision-based approach in which all the signs are represented with bare hands then it eliminates the matter of using any artificial devices for interaction.

❑ CNN:

CNN is a category of Deep Neural Networks which will recognize and classify particular features from images and are widely used for analyzing visual images. Some of its applications are image and video recognition, image classification, medical image analysis, computer vision and language processing. The preprocessing required for CNN is far lower as compared to other classification algorithms.

The term ‘Convolution” in CNN denotes the function of convolution which happens to be a special kind of linear operation wherein two functions are multiplied to process a third function which expresses how the form of one function is modified by another. In simple terms, two images that may be represented as matrices are multiplied to provide an output that's used to extract features from the image.

There are two major parts of CNN architecture:

1. A convolution tool that separates and identifies the varied features of the image for analysis during a process called Feature Extraction
2. A fully connected layer that uses the output from the convolution process and predicts the class of the image-based upon the features extracted in previous stages.

There are three layers that structure the CNN. To form a CNN architecture these layers are stacked.

1. Convolutional Layer

This is the 1st layer and it is used to extract the various features from the input images. In this layer, the mathematical process of convolution is performed between the input image and a filter of a specific size MxM. By sliding the filter over the input image, the scalar product is taken between the filter and parts of the input image with regard to the dimensions of the filter (MxM).

The output is termed as the Feature map which provides us information about the image like the corners and edges. Later, this feature map is fed to other layers to find out several other features of the input image.

2. Pooling Layer

Many times a Convolutional Layer is followed by a Pooling Layer. The primary aim of this layer is to decrease the dimensions of the convolved feature map to scale back the computational costs. This is performed by reducing the connections between layers and independently operating on each feature map. Depending upon the method used, there are

several kinds of Pooling operations.

In Max Pooling, the biggest element is taken from the feature map. Average Pooling calculates the average of the elements during a predefined sized Image section. To compute Sum Pooling we take the total sum of the elements present within the predefined section. The Pooling Layer usually is a bridge between the Convolutional Layer and FC Layer

3. Fully Connected Layer (FC)

The FC layer consists of the weights and biases with the neurons. It is then used to join the neurons between two different layers. These layers are usually placed before the output layer and form a previous couple of layers of a CNN Architecture.

The input image obtained from the previous layer are flattened and sent to the FC Layer. Then the flattened vector undergoes a few more Fully Connected layers where the mathematical function's operations usually occur. The classification process begins to take place in this stage.

There are two more important parameters in addition to the above three layers which are the dropout layer and also the activation function.

4. Dropout

Mostly, overfitting is caused within the training dataset when all the features are connected to the FC layer. Overfitting occurs when a specific model works so well on the training data causing a negative impact on the model's performance when used on brand new data.

To overcome this problem, a dropout layer is utilised wherein a couple of neurons are dropped from the neural network during the training process leading to a reduced size of the model. When we pass a dropout of 0.3, 30% of the nodes are selected randomly from the neural network and dropped out.

5. Activation Functions

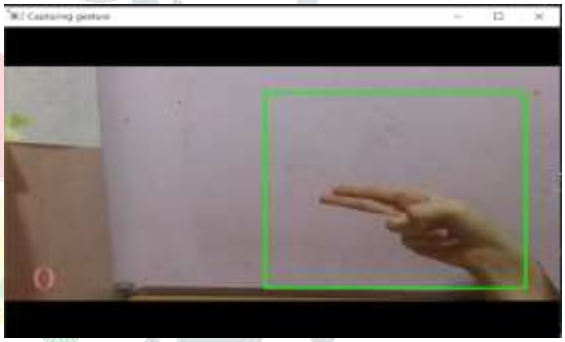
Finally, the Activation function is one of the most important parameters of the CNN model. They are used to learn and approximate any sort of continuous and complicated relationship between variables of the network. In simple words, it decides which information of the model should be fired in the forward direction and which of them should not be fired at the end of the network.

It adds non-linearity to the network. There are several commonly used activation functions like the ReLU, Softmax, tanH and also the Sigmoid functions. Each of these functions has a specific usage. Sigmoid and Softmax functions are preferred for a multi-class classification for a binary classification CNN model, generally, softmax is used.

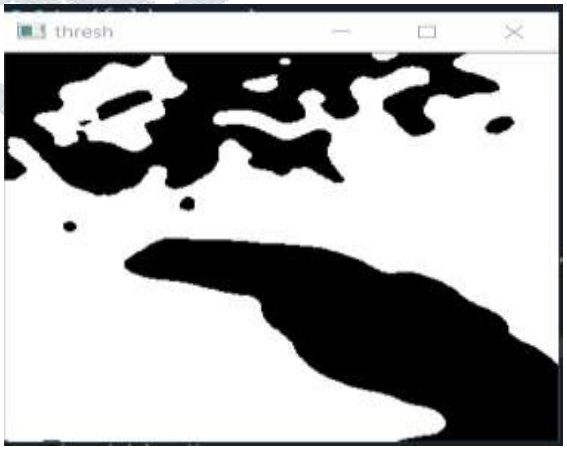
❑ DATA SET GENERATION:

For the project, we tried to seek out already made datasets but we couldn't find a dataset within the sort of raw images that matched our requirements. All we could find were the datasets within the sort of RGB values. Hence we decided to make our own data set. The steps we followed to make our data set are as follows.

We used the Open computer vision(OpenCV) library to produce our dataset. Firstly we captured around 1000 images of each of the symbols in ASL for training purposes and around 200 images per symbol for testing purposes. The first step is that we capture each frame shown by the webcam of our machine. In each frame, we define a region of interest (ROI) which is denoted by a green bounded square as shown in the image below.

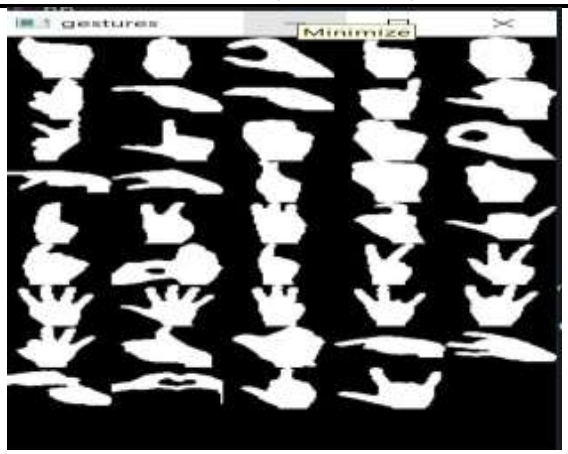


From this entire image, we extract our ROI which is RGB and convert it into a grayscale image.



Finally, we apply our gaussian blur filter to our image which helps us extract various features of our image.

Given below are all the processed gestures of the alphabets.



❑ GESTURE CLASSIFICATION

The approach which we used for this project is: Our approach uses two layers of algorithms to predict the final symbol of the user.

Algorithm Layer 1:

1. Apply gaussian blur filter and threshold to the frame crazy OpenCV to urge the processed image after feature extraction.

2. This processed image is passed to the CNN model for prediction and if a letter is detected for quite 50 frames then the letter is printed and taken into consideration for forming the word.

3. Spaces between the words are considered using the blank symbol.

Algorithm Layer 2:

1. On getting detected various sets of symbols show similar results.

2.Then classification is done by using classifiers made for those sets only.

❑ TRAINING AND TESTING :

We convert our input images(RGB) into grayscale and apply gaussian blur to remove unnecessary noise. We apply an adaptive threshold to extract our hand from the background and resize our images. We feed the input images after preprocessing to our model for training and testing after applying all the operations mentioned above. The prediction layer estimates how likely the image will fall under one of the classes. So the output is normalized between 0 and 1 and such that the sum of each value in each class sums to 1. We have achieved this using the softmax function. At first, the output of the prediction layer will be somewhat far from the actual value. To make it better we have trained the networks using labelled data. Cross-entropy is a performance measurement used in the classification. It is a continuous function that is positive at values that are not the same as labelled values and is zero exactly when it is equal to the labelled value. Therefore we optimized the cross-entropy by minimizing it as close to zero. To do this in our network layer we adjust the weights of our neural networks. TensorFlow has an inbuilt function to calculate the cross-entropy. As we have found out about the cross-entropy function, we have optimized it using Gradient Descent. In fact, the best gradient descent optimizer is called Adam Optimizer.

IV. CONCLUSION

The project is a simple demonstration of how CNN can be used to solve computer vision problems with good accuracy. We have developed software using artificial intelligence technologies such as Machine Learning, CNN, deep learning etc. to capture real-time sign language gestures and display the letters of individual gestures helping to form sentences using concepts of machine learning and various technologies.

This project can be upgraded in a few ways in the future:

It could be built as a mobile or a web application for the users to conveniently access the project.

The main area where this will be used is publically like ticket issuing counters, hospitals etc.

This can be even used to teach sign language to normal people.

Further, this could be used to take words and display the gesture for the same.

Thus the software will help to bridge the gap between non-sign language and sign language speaking people.

ACKNOWLEDGEMENT

We would like to give special thanks to Prof. Keshav Tambre for his mentoring and insights about the subject.

REFERENCES

[1] He, Siming. (2019). Research of a Sign Language Translation System Based on Deep Learning. 392-396. 10.1109/AIAM48774.2019.00083.

[2] International Conference on Trends in Information Sciences and Computing (TISC). : 30-35, 2012.

[3] Herath, H.C.M. & W.A.L.V.Kumari, & Senevirathne, W.A.P.B & Dissanayake, Maheshi. (2013). IMAGE BASED SIGN LANGUAGE RECOGNITION
SYSTEM FOR SINHALA SIGN LANGUAGE

[4] M. Geetha and U. C. Manjusha, , "A Vision Based Recognition of Indian Sign Language Alphabets and Numerals Using B-Spline Approximation", International Journal on Computer Science and Engineering (IJCSE), vol. 4, no. 3, pp. 406-415. 2012.

[5] Huang, J., Zhou, W., & Li, H. (2015). Sign Language

Recognition using 3D convolutional neural networks. IEEE International Conference on Multimedia
and Expo (ICME) (pp. 1-6). Turin: IEEE.
[6] Jaoa Carriera, A. Z. (2018). Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. Computer Vision and Pattern Recognition (CVPR),
2017 IEEE Conference on (pp. 4724-4733). IEEE. Honolulu.
[7] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. Computer Vision and Pattern
Recognition, 2009. CVPR 2009. IEEE Conference on (pp. 248-255). IEEE. Miami, FL, USA .
[8] Soomro, K., Zamir , A. R., & Shah, M. (2012). UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. Computer Vision and Pattern
Recognition, arXiv:1212.0402v1, 1-7.
[9] Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., & Serre, T. (2011). HMDB: a large video database for human motion recognition. Computer Vision (ICCV), 2011 IEEE International Conference on (pp. 2556-2563). IEEE