Indian Sign Language Database Creation and Detection

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology

in

CSE Core

by

Debangsu Sarkar - 19BCE0902

Swayam Atul Mehta – 20BCE0255

Under the guidance of
Prof. / Dr. Gopalkrishan T

SCOPE

VIT, Vellore.

April, 2022

Executive Summary

Sign language is one of the oldest and most natural form of language for communication, but since most people do not know sign language and interpreters are very difficult to come by, we have come up with a real time method using neural networks for finger spelling based Indian sign language. In our method, the hand is first passed through a filter and after the filter is applied the hand is passed through a classifier, a Convoluted Neural Network which predicts the class of the hand gestures. We also recognized the importance of developing a Software which makes it easier to develop Datasets for the various regional interpretations of Indian Sign language.

INDEX

1. INTRODUCTION

1.1 OBJECTIVE

To exchange information and communicate among their community, the deaf and hard of hearing people use a commonly agreed upon sign language. Computer recognition of sign language deals from sign gesture acquisition and continues till text/speech generation. These signs can be static, in the form of single letters which can be captured as an image, or dynamic, in the form of continuous string of characters forming a word and then a sentence which can be captured as a video. Static gesture recognition is simpler than dynamic gesture recognition, and we will attempt to recognise that first, but both recognition systems are important to the human community. We aim to develop an Indian sign language recognition dataset and use it in the deep learning model which depends on neural networks to interpret gestures of sign language and hand poses to natural language. We are going to describe the Sign language recognition steps, the data acquisition, data pre-processing and transformation, the feature extraction, classification and then describe how we got the results. We will also describe some future directions for research in this regard.

Keywords: Sign Language Recognition, Hand Tracking, Hand Gesture Recognition, Gesture Analysis, Face Recognition, Sign Language, ISL, Hearing Disability, Convolutional Neural Network (CNN), Artificial Intelligence, Computer Vision, Machine Learning, Image Processing, Dataset creation, Neural Network.

1.2 MOTIVATION

For interaction between normal people and D&M people a language barrier is created as sign language structure which is different from normal text. So, they depend on vision-based communication for interaction. If there is a common interface that converts the sign language to text the gestures can be easily understood by the other people. So, research has been made for a vision-based interface system where D&M people can enjoy communication without really knowing each other's language. The aim is to develop a user-friendly human computer interface (HCI) where the computer understands the human sign language. There are various sign languages all over the world, namely Indian Sign Language (ISL), French Sign Language, British Sign Language (BSL), Indian Sign language, Japanese Sign Language and work has been done on other languages all around the world.

1.3 BACKGROUND

Sign language is a visual gestural mode of communication used predominantly by people who are deaf or hard of hearing as well as people who cannot speak. It makes use of three dimensional space through hand movements, facial expressions and body language to convey meaning. It has its own vocabulary and syntax which is purely different from other spoken/written languages. A spoken language makes use of vocal tracts along with linguistic elements like vowels, consonants, tone, etc. to convey a message. On the other hand, a sign language makes use of above mentioned visual elements altogether eliminating the use of oratory and auditory systems of the human body. Both spoken and sign languages involve complex grammar which plays a key role in connecting words into phrases and sentences.

The Indian Sign Language (ISL), often referred to as the Indo-Pakistani Sign Language (IPSL), is the predominant sign language in South Asia. Number signs, family relationship and spatial use are some crucial features of ISL which distinguish it from other sign languages. Unlike the Indian Sign Language (ISL), ISL is devoid of temporal inflection in its fingerspelling chart.

A sign language recognition system serves as an easy, efficient and accurate mechanism to transform sign language into text/speech. Computerized digital image processing and classification methods are used to recognize the alphabet flow and interpret the words and phrases of sign language. The four essential components of a gesture recognition system are – modeling, analysis, recognition and application systems.

1.4 Literature Review

Authors and Year (Reference)	*Title (Study)*	*Concept / Theoretical model/ Framework*	*Methodology used/ Implementation*	*Dataset details/ Analysis*	*Relevant Finding*	*Limitations/ Future Research/ Gaps identified*
Ashok K Sahoo, Gouri Sankar Mishra and Kiran Kumar Ravulakollu (2014)	SIGN LANGUAGE RECOGNITION : STATE OF THE ART	A survey paper: The sign language recognition steps are described. Data acquisition, data preprocessing, transformation, feature extraction, classification and results obtained are examined. Some future directions for research in this area also suggested.	It's mostly a survey paper, so implementation details are not contained but it dedicates to find various implementations done by the scientists, like data acquisition devices are used by some of the researchers in order to acquire input signs. These are the list of input devices: • CyberGlove® • Sensor Glove • Polhemus FASTRAK	It discussed on the various datasets already used by the researchers, like Lifeprint Fingurespell Library for ISL, CAS-PEAL for CSL and many others and discussed on the various usages and advantages of each.	While significant progress has already been made in computer recognition of sign languages of other countries but a very limited work has been done in ISL Computerization. ISL is majorly missing in various literature. Most of the researchers create their own database for sign language recognition. This database can be also classified into digits, alphabets and phrases. Current systems are mainly focused on static signs/ manual signs/ alphabets/ numerals. Systems should be able to distinguish face, hand and other parts of body simultaneously.	The paper fails to discuss about the various implementation mechanisms used by various researchers and is mostly an exercise into the various datasets available and when to use one or create our own dataset.

Ahmed KASAPBASI, Ahmed Eltayeb AHMED ELBUSHRA, Omar AL-HARDANEE Arif YILMAZ (2022)	Machine learning methods for sign language recognition: A critical review and analysis	The authors developed a dataset and a Convolutional Neural Network-based sign language interface system to interpret gestures of sign language and hand poses to natural language. The dataset created in this study is a new addition in the field of sign language recognition (SLR). This dataset may be used to develop SLR systems. Furthermore, the research compares the results of the dataset with two different datasets from other studies.	The dataset contains images varying 0.5 m, 0.75 m and 1 m hand distance to illustrate variance in illumination and depth conditions. The neural network developed in this study is a Convolutional Neural Network (CNN) which enhances the predictability of the Indian Sign Language alphabet (ISLA).	The dataset contains images and corresponding letters of ISLA. The creation of the dataset was dependent on many factors such as illumination and the distance between the camera and hand which we adjusted to improve the performance of the CNN model. The dataset was created under variable conditions which increases the number of contributions, comparisons, results and conclusions in the field of SLR and may enhance such Systems.	The framework has achieved a 99.38% accuracy with excellent prediction and small loss (0.0250) with using the homemade dataset in contrast to the first public dataset: 99.41% with a 0.0204 loss and the second dataset: obtained accuracy was 99.48% and the loss was 0.0210, although the dataset was the largest one among all the other datasets and contains 104,000 images which ultimately led to the superior prediction and has higher validation accuracy and lower OOB error.	The variation in size, position, shape and background of the hand, lighting, and the distance of the hand from the camera is still not taken much into account although is stressed much throughout the paper. This study can be improved by adding more images for more letters and words into the dataset. More images can be added to improve accuracy and reduce loss. By adding new words and terms, the proposed system may be improved to predict a complete word. Predicted words can be turned into speech by utilizing a text-to-speech engine.

G. Ananth Rao, P.V.V. Kishore (2018)	Selfie video based continuous Indian sign language recognition system	This paper introduces a novel method to bring sign language closer to real time application on mobile platforms. Selfie captured sign language video is processed by constraining its computing power to that of a smart phone. Pre-filtering, segmentation and feature extraction on video frames creates a sign language feature space. Minimum Distance and Artificial Neural Network classifiers on the sign feature space is trained and tested iteratively. Sobel edge operator's power is enhanced with morphology and adaptive thresholding giving a near perfect segmentation of hand and head portions compensating for the small vibrations of the selfie stick.	Their implementation is a demonstration of Image Processing until we have the result of the picture's features being extracted. They started with a video capturing sensor: A smart phone camera that records the video and is processed per frame and is segmented by frame and object and then the monotone image is taken and the features are extracted after passing through Sobel's masks. Then 2D DCT is calculated of the head contour. For faster classification as it should be real time, Minimum distance classifier is used (MDC).	The paper doesn't go into the details of the dataset they made or used except for a few examples of sentences that they have used but they had this info that: A formal database of 18 signs in continuous sign language was recorded with 10 different signers. Pre-filtering, segmentation and contour detection are performed with Gaussian filtering, sobel with adaptive block thresholding and morphological subtraction respectively. Hand and head contour energies are features for classification computed from discrete cosine transforms.	Thus, they tested and simulated the idea of sign language recognition into smartphones. Thus, to create a database we don't need much specialised equipment but just a phone and its camera. They have proved that an ANN works better in this case than a WMS.	The ANN model isn't deeply explored and most of the discussion is around the video capture, the image processing, feature extraction and the formation of the database. A mobile app isn't developed to show how the Android API might react to it and the image processing is done independently.

Junfu Pu, Wengang Zhou, Houqiang Li (2019)	Iterative Alignment Network for Continuous Sign Language Recognition	The framework used for this research was a 3D residual network (3D-ResNet) for feature extraction and an encoder-decoder network for sequence modelling.	Their method integrated the encoder-decoder network and connectionist temporal classification (CTC) into a unified deep architecture. To explore the correspondence between the input sequence and target translation, soft dynamic time warping (soft-DTW) was used to align the CTC-decoder and LSTM-decoder. Their system consisted of 4 tiers of neural network: (1) Feature Extractor (2) Sequence Encoder (3) Target Decoders (4) Alignment Constraint	Two public datasets:(1) RWTH-PHOENIX-Weather multi-signer (7K sign videos with a total of 77K words) for German SLR. (2) CSL (5k videos made by 50 signers and 100 sentences each, each sentence consists an average of 5 words) for Chinese SLR.	Their approach had the strong capability to deal with the unseen sentence recognition problem. It was more effective and superior with better performance compared to existing methods.	Substitution error was ignored in the Word Error Rate (WER) for RWTH-PHOENIX-Weather-2014. Training the network in an end-to-end way did not provide good results. This Align-end2end method can be explored further.

Oscar Koller (2020)	Quantitative Survey of the State of the Art in Sign Language Recognition	This paper provides an overview of the field of sign language recognition following a quantitative meta-study approach.	The author has covered 300 published studies and manually labelled them based on their basic recognition characteristics which include modelled vocabulary size, number of contributing signers, the features and modalities of the employed sign language, its dataset quality and available input data type. A detailed structured overview comparing over 25 research studies that have evaluated their approaches on the RWTH-PHOENIX-Weather corpus.	There is no dataset as such since it is a survey paper but the author has covered studies which have been done mainly using datasets for Indian, Chinese and German sign languages.	It is observed that large vocabulary (> 1000 signs) and 50-200 vocabulary tasks have experienced a large gain in the number of published results since 2015. RGB data started attracting a lot of interest after 2005 and depth as input modality gained popularity after 2010. Hand shape has been tackled by a much larger fraction of results published after 2015.	In many studies, data augmentation is not carefully described and also an ablation study that details the effect of various augmentation methods is left for coming research. More efforts are needed to create real-life large vocabulary continuous sign language tasks that should be made publicly accessible with well-defined train, development and test partitions.

Ming Jin Cheok, Zaid Omar, Mohamed Hisham Jaward (2017)	A review of hand gesture and sign language recognition techniques	This paper reviews the state-of-the-art techniques used in recent hand gesture and sign language recognition research.	The vision-based recognition techniques reviewed have been categorized into: data acquisition, pre-processing, segmentation, feature extraction and classification. At each stage, the used algorithms have been elaborated along with comparison of their merits. The sensor-based approaches included data glow, electromyography, WiFi and Radar.	There is no dataset as such since it is a survey paper but the author has reviewed the benchmark databases like Purdue RVL-SLLL, RWTH-PHOENIX-Weather, ATIS Sign Language Corpus, SIGNUM Corpus, RWTH-BOSTON-50, RWTH-BOSTON-104 and RWTH-BOSTON-400	Most literature reviewed in this paper focused on recognition of only one hand. Most vision-based researches which were reviewed used a standard camera or a webcam. The most commonly applied pre-processing techniques included Median and Gaussian filter to remove noises. Tracking of hand movement was often carried out using Particle filtering, CAMShift method, and Adaboost tracking algorithm. The most commonly used color spaces were HSV, YCbCr and CIE Lab. The research also showed that skin color segmentation with other features such as edge detection and threshold improved the segmentation result.	There are significant gaps to be filled for gesture recognition to be able to be put into actual use. The numbers of research using benchmark database are far less compared to those collected their own database. Future works using benchmarked databases are advised to allow for direct comparison between algorithms used.

1.5 Hardware and Software Requirements

We have developed this project using OpenCV and Keras modules of python.

The prerequisites software & libraries for the sign language project are:

- Python (3.7.4)
- IDE (Jupyter)
- Numpy (version 1.16.5)
- cv2 (openCV) (version 3.4.2)
- Keras (version 2.3.1)
- Tensorflow (as keras uses tensorflow in backend and for image preprocessing) (version 2.0.0)

2. PROJECT DESCRIPTION AND GOALS

2.1 Data Acquisition

The different approaches to acquire data about the hand gesture can be donein the following ways:

I. Use of sensory devices

It uses electromechanical devices to provide exact hand configuration, and position. Different glove-based approaches can be used to extract information. But it is expensive and not user friendly.

II. Vision based approach

In vision-based methods computer camera is the input device for observing the information of hands or fingers. The Vision Basedmethods require only a camera, thus realizing a natural interaction between humans and computers without the use of any extra devices. These systems tend to complement biological vision by describing artificial vision systems that are implemented in software and/or hardware.

The main challenge of vision-based hand detection is to cope with the large variability of human hand's appearance due to a huge number ofhand movements, to different skin-color possibilities as well as to thevariations in viewpoints, scales, and speed of the camera capturingthe scene.

2.2 Data Preprocessing and Feature Extraction for Vision-Based Approach

- In [1] the approach for hand detection combines threshold-based colordetection with background subtraction. We can use Adaboost face detector to differentiate between faces and hands as both involve similar skin-color.
- We can also extract necessary image which is to be trained by applying a filter called Gaussian blur. The filter can be easily applied using open computer vision also known as OpenCV and is describedin [3].
- For extracting necessary image which is to be trained we can use instrumented gloves as mentioned in [4]. This helps reduce computation time for preprocessing and can give us more concise and accurate data compared to applying filters on data received from video extraction.
- We tried doing the hand segmentation of an image using color segmentation techniques but as mentioned in the research paper skin color and tone is highly dependent on the lighting conditions due to which output, we got for the segmentation we tried to do were no so great. Moreover, we have a huge number of symbols to be trained for our project many of which look similar to each other like the gesture for symbol 'V' and digit '2', hence we decided that in order to produce better accuracies for our large number of symbols, rather than segmenting the hand out of a random background we keep background of hand a stable single color so that we don't need to segment it on the basis of skin color. This would help us to get better results.

2.3 Gesture Classification

- In [1] Hidden Markov Models (HMM) is used for the classification of the gestures. This model deals with dynamic aspects of gestures. Gestures are extracted from a sequence of video images by tracking the skin-color blobs corresponding to the hand into a body– face space centered on the face of the user. The goal is to recognize two classes of gestures: deictic and symbolic. The image is filtered using a fast look–up indexing table. After filtering, skin color pixels are gathered into blobs. Blobs are statistical objects based on the location (x, y) and the colorimetry (Y, U, V) of the skin color pixelsin order to determine homogeneous areas.

- In [2] Naïve Bayes Classifier is used which is an effective and fast method for static hand gesture recognition. It is based on classifying the different gestures according to geometric based invariants which are obtained from image data after segmentation. Thus, unlike many other recognition methods, this method is not dependent on skin color. The gestures are extracted from each frame of the video, with astatic background. The first step is to segment and label the objects of interest and to extract geometric invariants from them. Next step is the classification of gestures by using a K nearest neighbor algorithm aided with distance weighting algorithm (KNNDW) to provide suitable data for a locally weighted Naïve Bayes" classifier.
- According to paper on "Human Hand Gesture Recognition Using a Convolution Neural Network" by Hsien-I Lin, Ming-Hsiang Hsu, andWei-Kai Chen graduates of Institute of Automation Technology National Taipei University of Technology Taipei, Taiwan, they construct a skin model to extract the hand out of an image and then apply binary threshold to the whole image. After obtaining the threshold image they calibrate it about the principal axis in order to center the image about it. They input this image to a convolutional neural network model in order to train and predict the outputs. They have trained their model over 7 hand gestures and using their model they produce an accuracy of around 95% for those 7 gestures.

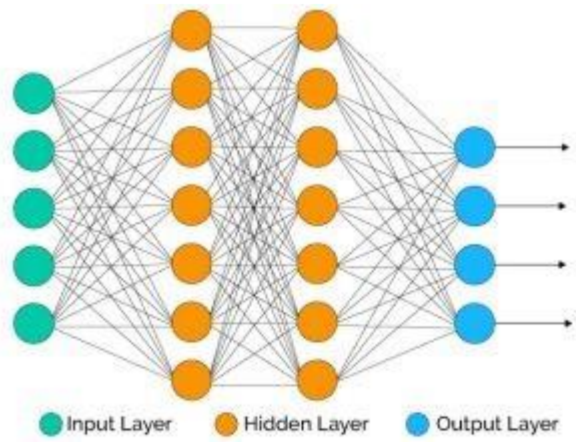
2.4 Key Words and Definitions

I. Feature Extraction and Representation:

The representation of an image as a 3D matrix having dimension as ofheight and width of the image and the value of each pixel as depth (1 in case of Grayscale and 3 in case of RGB). Further, these pixel values are used for extracting useful features using CNN.

II. Artificial Neural Networks:

Artificial Neural Network is a connection of neurons, replicating the structure of human brain. Each connection of neuron transfers information to another neuron. Inputs are fed into first layer of neurons which processes it and transfers to another layer of neurons called as hidden layers. After processing of information through multiple layers of hidden layers, information is passed to final output layer.

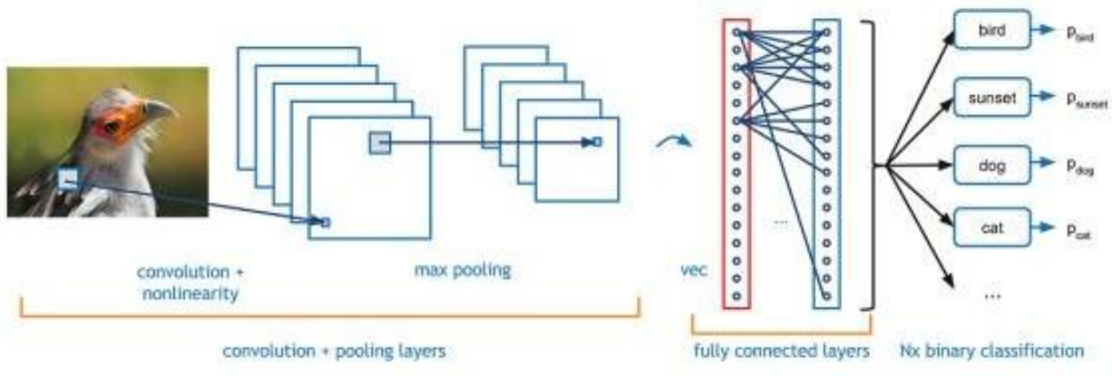


There are capable of learning and they have to be trained. There are differentlearning strategies:

1. Unsupervised Learning
2. Supervised Learning
3. Reinforcement Learning

III. Convolution Neural Network:

Unlike regular Neural Networks, in the layers of CNN, the neuronsare arranged in 3 dimensions: width, height, depth. The neurons in a layer will only be connected to a small region of the layer (window size) before it, instead of all of the neurons in a fully-connected manner. Moreover, the final output layer would have dimensions (number of classes), because by the end of the CNN architecture we will reduce the full image into a single vector of class scores.

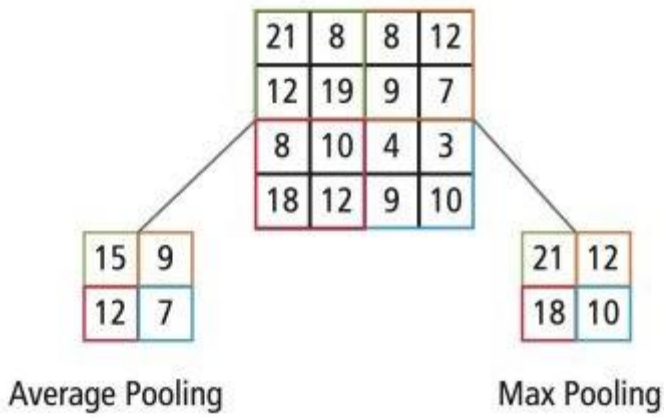


a) Convolution Layer: In convolution layer we take a small window size [typically of length 5*5] that extends to the depth of the input matrix. The layer consists of learnable filters of window size. During every iteration we slid the window by stride size [typically 1], and compute the dot product of filter entries and input values at a given position. As we continue this process well create a 2-Dimensional activation matrix that gives the response of that matrix at every spatial position. That is, the network will learn filters that activate when they see some type of visual feature such asan edge of some orientation or a blotch of some color

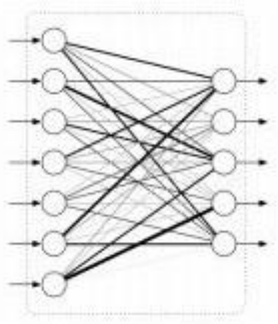
b) Pooling Layer: We use pooling layer to decrease the size of activation matrix and ultimately reduce the learnable parameters. There are two types of pooling:

i. **Max Pooling:** In max pooling we take a window size [for examplewindow of size 2*2], and only take the maximum of 4 values. Well lidthis window and continue this process, so well finally get a activation matrix half of its original Size.

ii. **Average Pooling:** In average pooling we take average of all values in a window.



c) Fully Connected Layer: In convolution layer neurons are connected only to a local region, while in a fully connected region, well connect the all the inputs to neurons.



d) Final Output Layer: After getting values from fully connected layer, well connect them to final layer of neurons [having count equal to total number of classes], that will predict the probability of each image to be in different classes.

IV. TensorFlow:

Tensorflow is an open-source software library for numerical computation.First we define the nodes of the computation graph, then inside a session, the actual computation takes place. TensorFlow is widely used in MachineLearning.

V. Keras:

Keras is a high-level neural networks library written in python that works as a wrapper to TensorFlow. It is used in cases where we want to quickly build and test the neural network with minimal lines of code. It contains implementations of commonly used neural network elements like layers, objective, activation functions, optimizers, and tools to make working with images and text data easier.

VI. OpenCV:

OpenCV (Open-Source Computer Vision) is an open source library of programming functions used for real-time computer-vision. It is mainly usedfor image processing, video capture and analysis for features like face and object recognition. It is written in C++ which is its primary interface, however bindings are available for Python, Java, MATLAB/OCTAVE.

2.5 Methodology

The system is a vision-based approach. All the signs are represented with bare hands and so it eliminates the problem of using any artificial devices forinteraction.

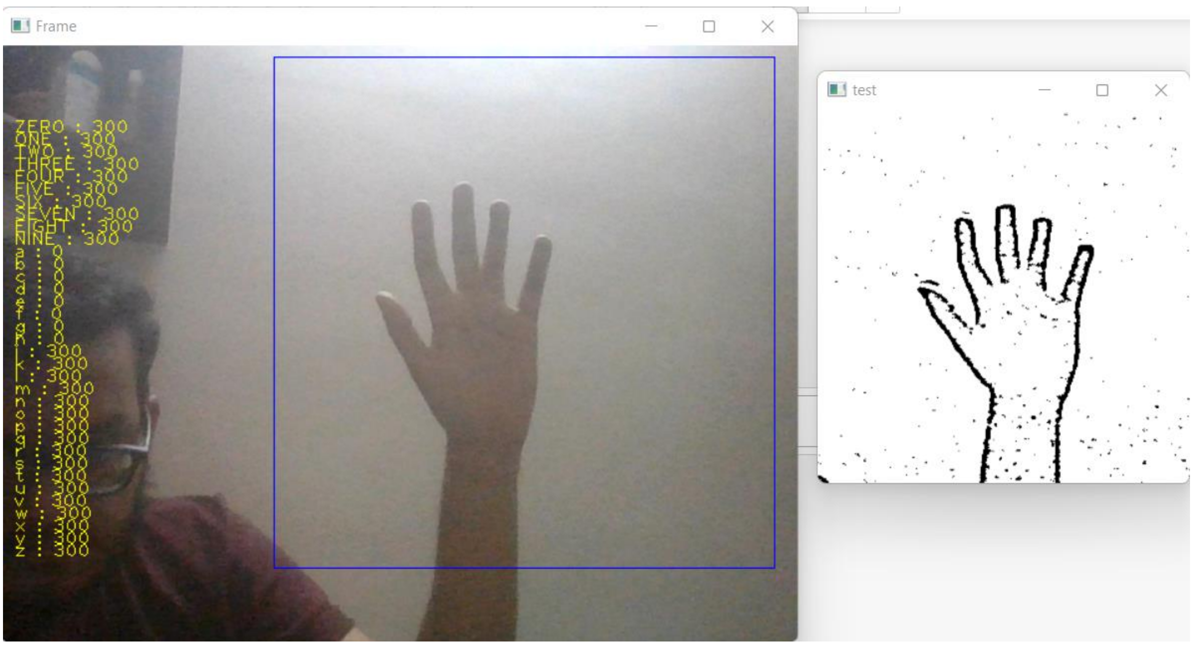
2.6 Data Set Generation

For the project we tried to find already made datasets but we couldn’t find dataset in the form of raw images that matched our requirements. All we could find were the datasets in the form of RGB values. Hence, we decidedto create our own data set. Steps we followed to create our data set are as follows.

We used Open computer vision (OpenCV) library in order to produce our dataset. Firstly, we captured around 800 images of each of the symbol in ISLfor training purposes and around 200 images per symbol for testing purpose.

First, 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 blue bounded square as shown in the image below.

Finally, we apply our gaussian blur filter to our image which helps us extracting various features of our image. The image after applying gaussian blur looks like below.



3. GESTURE CLASSIFICATION

The approach which we used for this project is:

Our approach uses two layers of algorithm to predict the final symbol of the user.

Algorithm Layer 1:

1. Apply gaussian blur filter and threshold to the frame taken with OpenCV toget the processed image after feature extraction.
2. This processed image is passed to the CNN model for prediction and if aletter is detected for more than 50 frames then the letter is printed and taken into consideration for forming the word.
3. Space between the words is considered using the blank symbol.

Algorithm Layer 2:

1. We detect various sets of symbols which show similar results on gettingdetected.
2. We then classify between those sets using classifiers made for those setsonly.

Layer 1: CNN

Model:

1. **1st Convolution Layer:** The input picture has resolution of 128x128 pixels.It is first processed in the first convolutional layer using 32 filter weights(3x3 pixels each). This will result in a 126X126 pixel image, one for each Filter-weights.
2. **1st Pooling Layer:** The pictures are down sampled using max pooling of 2x2 i.e we keep the highest value in the 2x2 square of array. Therefore, ourpicture is down sampled to 63x63 pixels.
3. **2nd Convolution Layer:** Now, these 63 x 63 from the output of the first pooling layer is served as an input to the second convolutional layer. It is processed in the secondconvolutional layer using 32 filter weights (3x3 pixels each).This will result in a 60 x 60 pixel image.
4. **2nd Pooling Layer:** The resulting images are down sampled again using max pool of 2x2 and isreduced to 30 x 30 resolution of images.
5. **1st Densely Connected Layer:** Now these images are used as an input to a fully connected layer with 128 neurons and the output from the second convolutional layer is reshaped to anarray of 30x30x32 =28800 values. The input to this layer is

an array of 28800 values. The output of these layer is fed to the 2nd Densely Connected Layer. We are using a dropout layer of value 0.5 to avoid overfitting.

6. **2nd Densely Connected Layer:** Now the output from the 1st Densely Connected Layer is used as an input to a fully connected layer with 96 neurons.
7. **Final layer:** The output of the 2nd Densely Connected Layer serves as an input for the final layer which will have the number of neurons as the number of classes we are classifying (alphabets + blank symbol).

Activation Function:

We have used ReLu (Rectified Linear Unit) in each of the layers (convolutional as well as fully connected neurons). ReLu calculates max(x,0) for each input pixel. This adds nonlinearity to the formula and helps to learn more complicated features. It helps in removing the vanishing gradient problem andspeeding up the training by reducing the computation time.

Pooling Layer:

We apply **Max** pooling to the input image with a pool size of (2, 2) with reluactivation function. This reduces the amount of parameters thus lessening thecomputation cost and reduces overfitting.

Dropout Layers:

The problem of overfitting, where after training, the weights of the network are so tuned to the training examples they are given that the network doesn't perform well when given new examples. This layer "drops out" a random set of activations in that layer by setting them to zero. The network should be able to provide the right classification or output for a specific example even if some of the activations are dropped out[5].

Optimizer:

We have used Adam optimizer for updating the model in response to the output of the loss function. Adam combines the advantages of two extensions of two stochastic gradient descent algorithms namely adaptive gradient algorithm (ADA GRAD) and root mean square propagation (RMSProp).

Training and Testing:

We convert our input images (RGB) into grayscale and apply gaussian blur to remove unnecessary noise. We apply adaptive threshold to extract ourhand from the background and resize our images to 128 x 128.

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 values in each class sums to 1. We have achieved this using 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 labeled data. The cross-entropy is a performance measurement used in the classification. It is a continuous function which is positive at values which isnot same as labeled value and is zero exactly when it is equal to the labeled 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 the cross-entropy function, we have optimized it usingGradient Descent in fact with the best gradient descent optimizer is called Adam Optimizer.

4. CHALLENGES FACED

There were many challenges faced by us during the project. The very first issue we faced was of dataset. We wanted to deal with raw images and that too square images as CNN in Keras as it was a lot more convenient working with only square images. We couldn’t find any existing dataset for that hence we decided to make our own dataset. Second issue was to select afilter which we could apply on our images so that proper features of the images could be obtained and hence then we could provide that image as input for CNN model. We tried various filter including binary threshold, canny edge detection, gaussian blur etc. but finally we settled with gaussian blur filter. More issues were faced relating to the accuracy of the model we trained in earlier phases which we eventually improved by increasing the input image size and also by improving the dataset.

5. RESULTS

We have achieved an accuracy of 95.8% in our model using only layer 1 of our algorithm, and using the combination of layer 1 and layer 2 we achieve an accuracy of 98.0%, which is a better accuracy then most of the current research papers on Indian sign language. Most of the research papersfocus on using devices like kinect for hand detection. In [7] they build a recognition system for flemish sign language using convolutional neural networks and kinect and achieve an error rate of 2.5%. In [8] a recognition model is built using hidden markov model classifier and a vocabulary of 30 words and they achieve an error rate of 10.90%. In [9] they achieve an average accuracy of 86% for 41 static gestures in Japanese sign language. Using depth sensors map [10] achieved an accuracy of 99.99% for observed signers and 83.58% and 85.49% for new signers. They also used CNN for their recognition system. One thing should be noted that our model doesn't uses any background subtraction algorithm whiles some of the models present above do that. So, once we try to implement background subtraction in our project the accuracies may vary. On the other hand, most of the above projects use kinect devices but our main aim was to create a project which can be used with readily available resources. A sensor like kinect not only isn't readily available but also is expensive for most of audience to buy and our model uses a normal webcam of the laptop hence it is great plus point. Below are the confusion matrices for our results.

6. CONCLUSION

In this report, a functional real time vision based Indian sign language recognition for D&M people have been developed for ISL alphabets. We achieved final accuracy of 98.0% on our dataset. We are able to improve ourprediction after implementing two layers of algorithms in which we verify and predict symbols which are more similar to each other. This way we are able to detect almost all the symbols provided that they are shown properly, there is no noise in the background and lighting is adequate.

7. FUTURE SCOPE

We are planning to achieve higher accuracy even in case of complex backgrounds by trying out various background subtraction algorithms. We are also thinking of improving the preprocessing to predict gestures in low light conditions with a higher accuracy.

8. REFERENCES

[1] Sahoo, Ashok & Mishra, Gouri & Ravulakollu, Kiran. (2014). Sign language recognition: State of the art. ARPN Journal of Engineering and Applied Sciences. 9. 116-134.

[2] Ahmed KASAPBAŞI, Ahmed Eltayeb AHMED ELBUSHRA, Omar AL-HARDANEE, Arif YILMAZ, DeepISLR: A CNN based human computer interface for Indian Sign Language recognition for hearing-impaired individuals,
Computer Methods and Programs in Biomedicine Update,
(https://www.sciencedirect.com/science/article/pii/S2666990021000471).

[3] G. Ananth Rao, P.V.V. Kishore,Selfie video based continuous Indian sign language recognition system,Ain Shams Engineering Journal,Volume 9, Issue 4,2018

LINKS:

[1] (PDF) Sign language recognition: State of the art (researchgate.net)

[2] DeepISLR: A CNN based human computer interface for Indian Sign Language recognition for hearing-impaired individuals | Elsevier Enhanced Reader

[3] Selfie video based continuous Indian sign language recognition system - ScienceDirect

9. APPENDIX A

OpenCV

OpenCV (Open-Source Computer Vision Library) is released under a BSDlicense and hence it's free for both academic and commercial use. It has C++, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. OpenCV was designed for computational efficiency andwith a strong focus on real-time applications. Written in optimized C/C++,the library can take advantage of multi-core processing. Enabled with OpenCL, it can take advantage of the hardware acceleration of the underlying heterogeneous compute platform.

Adopted all around the world, OpenCV has more than 47 thousand people ofuser community and estimated number of downloads exceeding 14 million. Usage ranges from interactive art, to mines inspection, stitching maps on theweb or through advanced robotics.

Convolution Neural network

CNNs use a variation of multilayer perceptrons designed to require minimalpreprocessing. They are also known as shift invariant or space invariant artificial neural networks

(SIANN), based on their shared-weights architecture and translation invariance characteristics.
Convolutional networks were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in arestricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field. CNNs use relatively little pre-processing compared to other image classification algorithms. This means that the network learns the filters thatin traditional algorithms were hand-engineered. This independence from prior knowledge and human effort in feature design is a major advantage. They have applications in image and video recognition, recommender systems, image classification, medical image analysis, and natural languageprocessing.

Tensorflow

TensorFlow is an open-source software library for dataflow programmingacross a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for bothresearch and production at Google.

TensorFlow was developed by the Google brain team for internal Google use. It was released under the Apache 2.0 open-source library on November9, 2015.

TensorFlow is Google Brain's second-generation system. Version 1.0.0 was released on February 11, 2017. While the reference implementation runs on single devices, TensorFlow can run on multiple CPUs and GPUs (with optional CUDA and SYCL extensions for general-purpose computing on graphics processing units). TensorFlow is available on 64-bit Linux, macOS,Windows, and mobile computing platforms including Android and iOS.
Its flexible architecture allows for the easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops toclusters of servers to mobile and edge devices.