HAND SIGN (ISL) RECOGNITION MODEL FOR ALPHABETS USING CNN MODEL

Mr Vijay Kumar[1], Rachna Narula[2] , Amrita Ticku[3]

Assistant professor in Department of Computer Science & Engineering,
Bharati Vidyapeeth's College of EngineeringGGSIPU, New Delhi, India

A B S T R A C T

Sign language is a means of communication when we use gestures to communicate with people, especially the deaf and dumb. In these sets of functions, the features included and the linguistic diversity of the areas have been major obstacles that have led to small ISL research. This paper proposes a hand gestures detection sensor system using the Deep Learning Algorithm, Convolution Neural Network (CNN) to process image and predict gestures. Most sign language tools are available using expensive external sensors.Our project aims to extend a step forward in this field by making our data set take pictures using a webcam and contains modules such as pre-processing features and features for rendering, training and model testing and sign-to-text conversion. then use different methods to remove features to extract useful information. There are 26 signs in Indian Sign Language corresponding characters where the proposed algorithm provided accurate results of up to 94% of all letters. translate it into text so that ordinary people can understand it.

K E Y W O R D S: I n d i a n S i g n L a n g u a g e (I S L) , C o n v o l u t i o n N e u r a l N e t w o r k (C N N) , M u l t i l a y e r P e r c e p t r o n (M L P) , D e e p L e a r n i n g , D i l a t i o n , M o r p h o l o g i c a l O p e r a t o r ,

1. Introduction

Communication is the process of sharing ideas or messages through gestures or text. Communication involves sharing thoughts, information, messages, and any other type of information. Communication is the only tool for communicating orally, in writing, viewing, or behaviour. People who are totally disabled or who have a speech impediment or part of it are often referred to as deaf and dumb. Sign language is widely used by deaf people and is used as a means of communication. Communicating with people who have good hearing aids through the lips or sign language is a language that uses gestures and other gestures, including facial expressions and body language, used primarily by deaf people. Hand gestures is a powerful way to convey our thoughts to others. Hand gestures may stop or change. Strong gestures involve the use of the hand and the movement of the hands as you speak. Sign language is a basic and natural way of communicating with the deaf. The community ignores these disabled people and isolates them. To bridge the gap between the common people and the deaf, each must learn sign language. We aim to develop an effective program to help overcome this barrier to communication.The purpose of this activity will be to visualize Indian Sign Language using the CNN Model. The model was established using 26 signs in Indian Sign Language. The dataset contained approximately 500-700 images per letter. Gesture recognition and sign language recognition has been a well-researched American Sign Language (ASL) topic, but few research studies in Indian Sign Language (ISL) have been published. But instead of using advanced technology like gloves or Kinect. We aim to solve this problem using modern computer technology and machine learning methods.

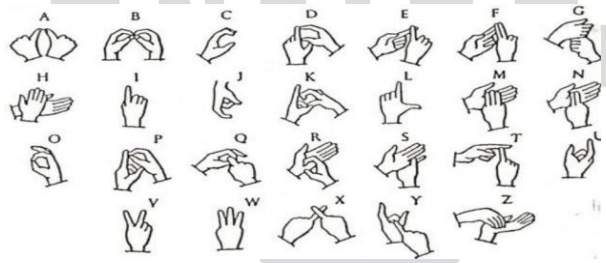


Figure 1 : Indian Sign Language

2. Motivation

Communication is one of the keys to living in a community. Deaf and mute people speak to each other using sign language but ordinary people find it difficult to understand their own language. Much work has been done in recognizing American Sign Language but Indian Sign Language is very different from American Sign Language. ISL uses two hands to communicate and ASL uses one hand to communicate. Using both hands often leads to the protection of the limbs. In addition to this, the lack of data sets and the variation in sign language and location have resulted in concerted efforts in obtaining ISL gesture. Our project aims to take a basic step in closing the communication gap between ordinary and deaf people through Indian Sign Language. The effective expansion of this project into common words and expressions will not only enable deaf and dumb people to communicate more easily and efficiently with the outside world, but also provide the

development of independent systems of understanding and assistance.

3. Objective

- Communication has always had a profound effect on all domains. The purpose of this project is to identify the metaphorical imagery in the image so that the communication gap between the average person and the person with a hearing impairment is easily closed.
- We aim to be recognized from the image database and use computer vision and machine learning or deep learning techniques to produce relevant features andsubsequent classifications.
- Help in communicating to the deaf anddumb.
- Build a recognition system that does not require much hardware.

4. Literature Review

The method was developed by Teena Varma, Ricketa Baptista, Daksha Chithirai Pandi, Ryland Coutinh [1]. The main task was to preview the images obtained for this project. In this paper, the raw images obtained are pre-processed using the following algorithms- 1. Gray scaling, 2. Skin masking 3. Threshold 4. Canny Edge detection. Postpre-processing images were included in the Convolution Neural Network model for training andtesting purposes. Various CNN structures were developed and tested in our database to identify the best architecture for hand-recognition . During the live scanning test, a 98% accurate Canny edge algorithm showed better results than othertechniques.Another approach was developed by Prof. Radha S.Shirbhate, Mr. Vedant D. Shinde, Ms. Sanam A. Metkari, Ms. Pooja U. Borkar, Ms. Mayuri A. Khandge [2], In this program, you have gone through an automated system to recognize sign language gestures in real time, using a variety of tools. The first step in this system is taking picturesthrough a webcam. After that the Captured Photos are in RGB mode. So the first step is to convert RGB images into binary images and then pre-processing will be done featuring cropping, filtering, brightness & brightness adjustment and much more. After thatthe extracting of the feature will be done on pre- processed data. SVM is a supervised machine learning model with integrated learning algorithm that analyzes segmentation and regression analysis data. The results of this test have a 100% accuracy level.
Chava Sri Varshini, Guguloth Sreekanth, Gurram Hruday, Mysakshi Chandu, Shaik Khadar Sharif [3].They then created a CNN model and performed successful training and validation of that created model. Here they used two methods to extract features are thresholding and RGB. Among these themodel trained with the converted Threshold data obtained 98.69% accuracy and, we obtained an accurate prediction of the given effect on ROI. But in the case of RGB converted data sets we did not get accurate results for the given input action. A different approach was developed by Sakshi Goyal, Ishita Sharma, Shanu Sharma [4], This system suggests that the method is built in relation to a single user. Real-time images will be scanned first and then stored in a directory and in a newly scanned image and a feature removal will take placeto determine what character the user is using. Using the SIFT (scale invariance Fourier transform) algorithm. The proposed algorithm provides 95% accurate results of 9 alphabets and their images are taken at all angles and distances. Mehreen Hurroo, Mohammad Elham Walizad [5], They use a webcam to shoot hand gestures. The images are made into a series of processing functions where the background is detected and removed using the HSV coloring algorithm (Hue, Saturation, Value). Segmention and then done to getthe skin tone region. Binary pixels are issued in eachframe, and the Convolutional Neural Network is used for training and classification. They had achieved 90% more accuracy. Another method was developed by Pavitra Kadiyala[6], who had developed the first install and import ofthe required libraries. Upload and split databases fortesting and training. Added Data and Define CNN Model. Then Train the training data on the model and evaluating in the test. After that the maintenanceof the model and integration with the website Downloading input from the user on the website andpredicting the sign. Ghali Upendra, Kokkiligadda Bhuvanendra, D Gayathri [7] In this program they made the basic steps in detecting hand gestures which are: Data acquisition-> Pre-Data Pre-processing-> Training-> Symbolization. They achieved 95.8% accuracy.

Raimundo F. Pinto Jr., Carlos D. B. Borges, Antonio
M. A. Almeida, and la'lis C. Paula Jr. [8] In their paper Static Hand Gesture Recognition Based on Convolutional Neural Networks they introduced a system of recognizing the static sign of the Italian sign language of the alphabet. Their modeling approach involves image acquisition, image processing followed by segmentation. In the case of image processing they used color segmentation to extract hands from frames / images using MLP and then applied image processing to those images.
Jayashree R. Pansare et al. [9] proposed a system of
26 handwriting ASL characters from a complexbackground using the Euclidean distance measurement.
Paranjoy Paul and Dr. G N Rathna [10] proposed a real-time sign language recognition system in which they separate hands using the XBOX360 camera's depth sensor and train the CNN model. To create anIndian Sign Language Recognition System they use2 types of images 1) RGB and 2) Images from XBOX360 and then combined them to segment outhand in images.
Lionel Pigou, Sander Dieleman, Peter-Jan Kindermans and Benjamin Schrauwen [11] proposed a model in which they first reduced noise in depth maps of CLAP14 dataset using a thresholding, followed by background removal and median filtering. 91.7% accuracy in cross validation containing different users with different domains from the training set and 95.68% test accuracy.
Kartik Shenoy, Tejas Dastane, Varun Rao, Devendra Vyavaharkar [12] in their paper "Real Time Indian Sign Language (ISL) Recognition" used a grid-based feature extraction followed by KNN machine learning algorithm for classifying images.
The Tensorflow transfer learning API is used by Sharvani Srivastava, Amisha Gangwar [13] and Richa Mishra to create an ISL

recognition system.Nachamai. M [14] in his paper "ALPHABET RECOGNITION OF AMERICAN SIGN LANGUAGE: A HAND GESTURE RECOGNITION APPROACH USING SIFT ALGORITH" introduced a consistent ASL recognition method using the SIFT (Scale Invariant Feature Transform) algorithm. SIFT is a flexible algorithm and because of that feature its resultspromise real-time and formatted images. Flexibilityof the scale is the main purpose of selecting this algorithm.
Hoshang kolivand, Saba Joudaki, Mohd Shahrizal Sunar and David Tally [15] in their paper "A new framework for gn language alphabet hand posture recognition using geometrical features through an artificial neural network" have proposed a system that uses Depth-based geometrical sign language recognition via SVM. Mayuresh Keni, Shireen Meher, Aniket Marathe[16] Proposed Sign Language System. In this projectit uses image processing system to identify, especially English sign language used by deaf people to communicate and translate them into text so that ordinary people can understand. The system does not need the background to be completelyblack. It works on any domain.

5. Proposed Methodology:

Using RGB colouring photos, a Deep learning model that can distinguish Indian Sign Language Alphabets. There were a total of 26 alphabets in it. This system is divided into three main phases, and the first phase of this system is data collecting. The second phase includes the application of various image processing techniques to extract the necessarydetails from the image and convert the image into binary colour. The third and last phase to choose an appropriate deep learning algorithm to work as classification model.

The primary steps of the proposed system are as follows:

1. Data Collection
2. Image Pre-processing
3. Image Augmentation
4. CNN Classifier5.

Figure 2: Steps Followed

5.1 Data Collection:

The proposed recognition model requires a large dataset for training, so all the data used for building this system is all collected by us using webcam withappropriate lighting conditions so the system can easily work in similar lighting. A total of 18,200 images are collected for training and validating the CNN classifier.

5.2 Image Pre-processing:

With the help of image pre-processing, we are able to convert the RGB images into single channel binary images highlighting the important features in each image such as edges. The image processing techniques helped in increasing model accuracy as well as reducing the time required to train the model.

```
def process_img(img):
    #Coversion to gray
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    #Bilateral filtering
    blur = cv2.bilateralFilter(gray,9,75,75)

    #Adative Thresholding
    th3 = cv2.adaptiveThreshold(blur,255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C,cv2.THRESH_BINARY_INV,11,2)

    #Noise Removal
    closing = cv2.morphologyEx(th3,cv2.MORPH_CLOSE,None,iterations = 2)

    #Dilation
    dilation = cv2.dilate(closing,None,iterations = 1)


    #Threshold
    ret, res = cv2.threshold(dilation, 0, 255,  cv2.THRESH_BINARY_INV)


    return res
```

Figure 3 : Function used to pre-process images

5.3 Image Augmentation:

An effective technique employed to expand image dataset by creating transforming the existing imagespresent in dataset when the amount of dataset is enough and dataset doesn't have enough variations in it. The transformations used in augmentations are flipping images, rotating images, cropping the sides of image, scaling up or down an image, shearing images, shifting images etc.

The image augmentation can be easily applied on training set images using ImageDataGenerator that comes with keras.

```
from keras.preprocessing.image import ImageDataGenerator

train_datagen = ImageDataGenerator(
        rotation_range=10, # rotate the image 30 degrees
        width_shift_range=0.1, # Shift the pic width by a max of 10%
        height_shift_range=0.1, # Shift the pic height by a max of 10%
        rescale=1/255, # Rescale the image by normalizing it.
        shear_range=0.2, # Shear means cutting away part of the image (max 20%)
        zoom_range=0.2, # Zoom in by 20% max
        horizontal_flip=True, # Allo horizontal flipping
        fill_mode='nearest' # Fill in missing pixels with the nearest filled value
        )
```

Figure 4: Augmentation used

5.4 Convolution Neural Network (CNN)Classifier

One of the most popular Deep learning algorithm that is widely put to use for image categorization is Convolution Neural Network. A Convolution Neural Network takes an input image, assign weight to many features/details inthe image, and understand difference between them. CNN requires far less pre-processing than other classification algorithms. The Convolution Operation extracts high-level characteristics from the input picture, like edgesas an example. The first Convolutional Layer isoften in charge of gathering minor or lower- grade information like colour, gradient direction, and so on. The architecture adapts to the High-Level characteristics as well, by adding additional layers, giving us a smart network of can understands almost all of the images present in dataset in the same way that humans do.

```
# Initializing the CNN
classifier = Sequential()

# First convolution layer and pooling
classifier.add(Convolution2D(32, (3, 3), input_shape=(sz, sz, 1), activation='relu'))
classifier.add(MaxPooling2D(pool_size=(2, 2)))

# Second convolution layer and pooling
classifier.add(Convolution2D(32, (3, 3), activation='relu'))
# input_shape is going to be the pooled feature maps from the previous convolution layer
classifier.add(MaxPooling2D(pool_size=(2, 2)))

# third convolution layer and pooling
classifier.add(Convolution2D(32, (3, 3), activation='relu'))
# input_shape is going to be the pooled feature maps from the previous convolution layer
classifier.add(MaxPooling2D(pool_size=(2, 2)))
# Flattening the layers
classifier.add(Flatten())

# Adding a fully connected layer
#units changed 128 to 400 - 500 - 200
classifier.add(Dense(units=200, activation='relu'))
classifier.add(Dropout(0.40))
#units changed 96 to 800 - 300 - 150
classifier.add(Dense(units=150, activation='relu'))
classifier.add(Dropout(0.40))
classifier.add(Dense(units=64, activation='relu'))
classifier.add(Dense(units=26, activation='softmax')) # softmax for more than 2
```

Figure 5: CNN Model used

6. Implementation

The implementation approach consists of 4 phases. It includes image data acquisition, image pre- processing, augmentation and then classification.

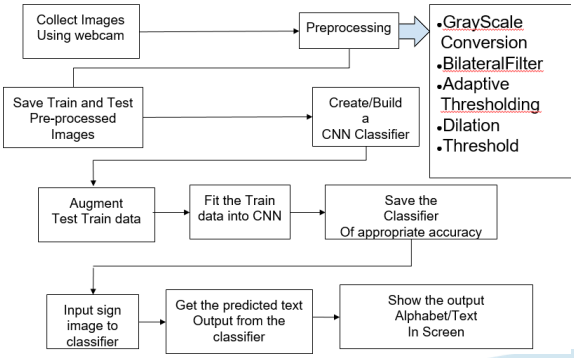


Figure 6: *Block diagram of methodology adopted inour project.*

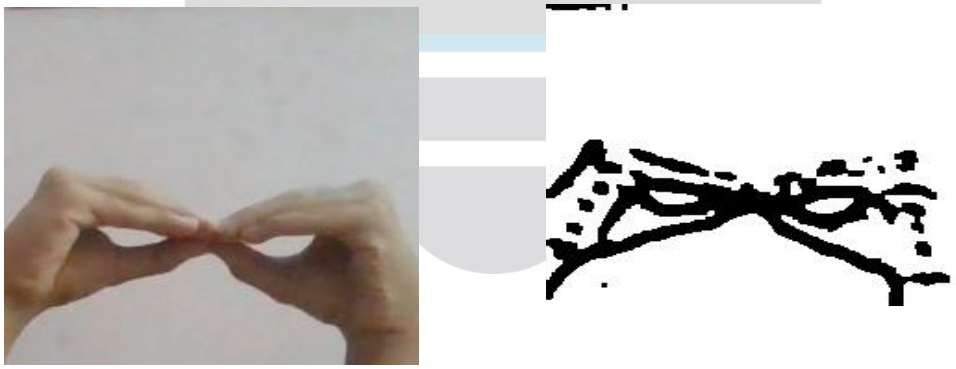
6.1 Image Data Acquisition

Hand gestures for Indian Sign Language alphabets were employed for recognition in this work. All of the steady photographs representing hand gestures were acquired in real time with the resolution of 330 X 300 pixels using HP Pavilion ec2004ax webcam and the data set was prepared. The acquired images were stored and saved in jpg format. Because the suggested approach does not include any wrist cropping mechanism, the user does not need to wear any type of clothing to prevent armprojection.

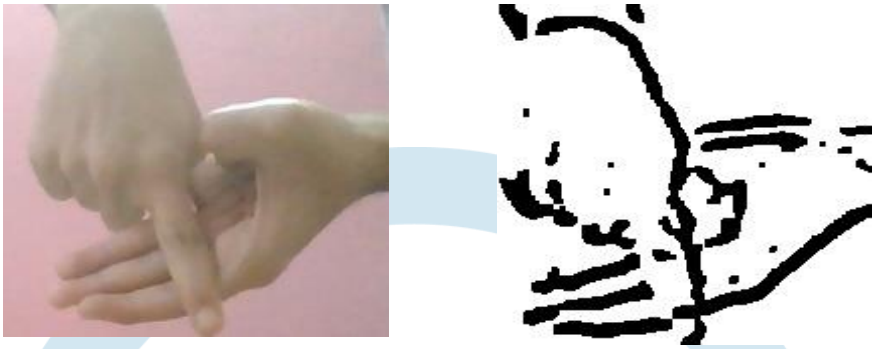
There are 700 images in total for each alphabet of Indian Sign Language alphabets in the image database. 700 Images acquired for each symbol making total 26*700 number of images. Out of 700,500 images were used in training and 200 for testing.

Because this work does not require a sophisticated background, the experimental setup requires a simple plain background. The signer must keep a shorter distance (around one meter) from the web camera. The camera is set in a predetermined position to capture the sign displayed by the signer, preventing the camera from vibrating. To obtain a clear hand image, proper illumination must be provided during image acquisition.

ISL for A



ISL for B



ISL for E

Figure 7: Sample dataset Images

6.2 Image Pre-processing

The acquired dataset imagers are in RGB colour space. The input image is first downsized to 200 × 200 pixels during the pre-processing step. Resizing of each and every image is carried out to lower the image processing time as well as feature extraction time and increase the system's accuracy.
Approach followed in Image Pre-processing:

- Gray-Scale Conversion
- Adaptive Thresholding
- Dilation
- Threshold

6.2.1 Gray-Scale Conversion

This process the RGB images were converted to black and white or grayscale images. This helps in reducing any effect causes by illumination changes and scaling. Grayscalerepresentations are frequently employed for extracting descriptors rather than operating directly on colour images since they simplify the technique and reduce computational requirements.

Figure 8: Image before and after Grayscaleconversion.

6.2.2 Adaptive Thresholding

The threshold value in basic thresholding is global, meaning it is the same for all pixels in the image. Adaptive thresholding is a method for calculating threshold values for smaller regions/areas, resulting in varying threshold values for different regions.
The adaptive thresholding can be performed easily by using cv2.adaptiveThreshold() method provided in OpenCV.

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_3 (Conv2D)           (None, 198, 198, 32)      320
 max_pooling2d_3 (MaxPooling  (None, 99, 99, 32)       0
 2D)
 conv2d_4 (Conv2D)           (None, 97, 97, 32)        9248
 max_pooling2d_4 (MaxPooling  (None, 48, 48, 32)       0
 2D)
 conv2d_5 (Conv2D)           (None, 46, 46, 32)        9248
 max_pooling2d_5 (MaxPooling  (None, 23, 23, 32)       0
 2D)
 flatten_1 (Flatten)         (None, 16928)             0
 dense_4 (Dense)             (None, 200)               3385800
 dropout_2 (Dropout)         (None, 200)               0
 dense_5 (Dense)             (None, 150)               30150
 dropout_3 (Dropout)         (None, 150)               0
 dense_6 (Dense)             (None, 64)                9664
 dense_7 (Dense)             (None, 26)                1690
_________________________________________________________________
```

Figure 9: Image before and after Adaptivethresholding

6.2.3 Dilation

Because the segmented images still contains noise, it is subjected to image morphological process like dilation.In Dilation Operation, The higher value among all pixels in the vicinity is the value of the output pixel.In case of binary image a pixel is set to if any of its neighbour pixels has a value. Morphological Dilatation enhances visibility and fills up small gaps. Fill forms appear larger and linesappear thicker.In the field of mathematical morphology, dilation isone of two basic operators, the other being erosion. It's most commonly used on binary images, but thereare grayscale versions as well. The primary effect ona binary picture is to gradually expand the bounds offoreground pixels regions. As a result, foreground pixel areas increases in size, while gaps within suchregions shrink significantly.

Figure 10: Image before and after Dilation

6.2.4 Thresholding

Thresholding is a widely usedsegmentation technique for distinguishing between foreground and background objects. In thresholding,value of each pixel is compared to a fixed threshold
value. If the pixel value is less than or equal to the threshold, it is set to minimum (generally), otherwise it is set to the maximum value (generally 255).

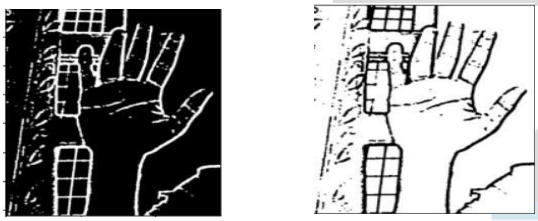


Figure 11: Image before and after Thresholding

6.2.5 CNN Classifier Architecture

- 3 Convolutional Layer
- 3 Dense Layer as follows:
  - 1st layer has 200 neurons with **"relu"** as an activation function
  - 2nd layer has 150 neurons with **"relu"** as an activation function
  - 3rd layer has 64 neurons with **"relu"** as an activation function
- Output Dense layer has 26 outputs equal tothe number of symbols with **softmax** as anactivation function

Figure 12: Model Summary

7. Results & Analysis

The proposed method was implemented using AMDRyzen™ 5 5600H processor running at 3.30 GHZ clock speed and the code was written in Python using dependencies like OpenCV, NumPy, and TensorFlow. Convolutional Neural Network is used for the classification and 26 ISL alphabets were classified in the proposed work The standard Indian Sign Language dataset was used. The dataset comprised all 26 alphabets. 72% of the data sample was used for training and 28% for testing. Out of 700 images of each class, 500 images from each alphabetclass are used for training the convolutional neural network and 200 images are used during testing. The implementation gave upto 94% accuracy in identifying the test sample for this dataset. Some sample images used for building recognition model are shown in the figure below.

```
Epoch 1/10
449/449 [==============================] - 104s 232ms/step - loss: 2.4120 - accuracy: 0.2708 - val_loss: 1.0723 - val_accuracy:
0.6779
Epoch 2/10
449/449 [==============================] - 104s 232ms/step - loss: 0.9897 - accuracy: 0.6728 - val_loss: 0.5101 - val_accuracy:
0.8389
Epoch 3/10
449/449 [==============================] - 105s 233ms/step - loss: 0.6347 - accuracy: 0.7837 - val_loss: 0.3672 - val_accuracy:
0.8900
Epoch 4/10
449/449 [==============================] - 104s 232ms/step - loss: 0.4692 - accuracy: 0.8403 - val_loss: 0.2700 - val_accuracy:
0.9116
Epoch 5/10
449/449 [==============================] - 107s 237ms/step - loss: 0.3833 - accuracy: 0.8717 - val_loss: 0.2697 - val_accuracy:
0.9216
Epoch 6/10
449/449 [==============================] - 120s 268ms/step - loss: 0.3162 - accuracy: 0.8904 - val_loss: 0.2517 - val_accuracy:
0.9358
Epoch 7/10
449/449 [==============================] - 108s 241ms/step - loss: 0.2656 - accuracy: 0.9143 - val_loss: 0.2471 - val_accuracy:
0.9316
Epoch 8/10
449/449 [==============================] - 107s 238ms/step - loss: 0.2512 - accuracy: 0.9187 - val_loss: 0.1997 - val_accuracy:
0.9400
Epoch 9/10
449/449 [==============================] - 105s 235ms/step - loss: 0.1959 - accuracy: 0.9352 - val_loss: 0.1602 - val_accuracy:
0.9553
Epoch 10/10
449/449 [==============================] - 113s 251ms/step - loss: 0.2022 - accuracy: 0.9359 - val_loss: 0.1557 - val_accuracy:
0.9616
```

Figure 13: Train accuracy and Validation accuracyafter each epoch

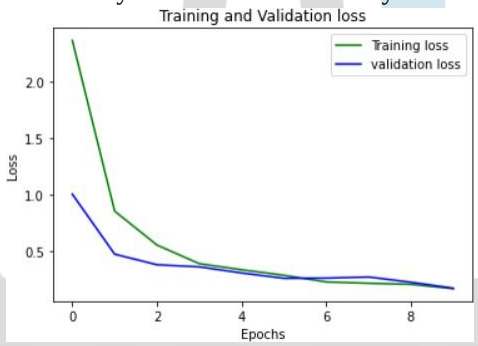


Figure 14: Training Loss vs Validation Loss

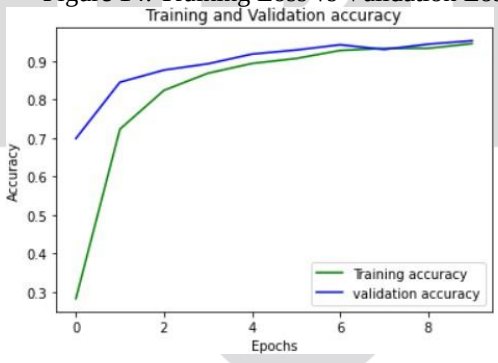


Figure 15: Training Accuracy vs Validation Accuracy

Table 1 : Comparison between our approach and other approach

Approach Used	Accuracy	Difficulties Encountered

Color Segmentation and Convolutional Neural Network[1]	93%	Complex Skin color leads to poor detection.
Hand Segmentation using depth sensor followed by CNN classification.[2]	89%	Along with a normal camera a depth-sensing camera is also required. The images from both camera needs to be merged accurately with is time-consuming.
Transfer Learning[5]	85.45%	It's hardware costly to build such a model as algorithm-like transfer leaning requires high CPU and GPU power.
Contour based	91%	Use of hand gloves
Neural Network with Back propagation	80.28%	Input feature vectors of Neural Network should have integer values..

Our Work	Accuracy	Advantages
Image Processing + CNN Based	96% validation accuracy	Recognizes 26 hand gestures with appropriate accuracy. Bare hands used for classification, removing any necessity any data gloves.

Conclusion And Future Scope

In our work, we have presented a hand sign recognition system that identifies basic human sign language which is A to Z alphabets from images. Todo so we have used the concept of movement of hands and fingers in various directions. Our project uses our own dataset which contains 500 images per alphabet, the photos are scaled to a 200x200 size setup. Our Dataset has 17,200 training photos, each class has 500 training photos, and 200 validation. A functional hand sign language recognition model is proposed. We achieved an accuracy of 93.59% on our dataset and 96% on the validation dataset. Because of the above suggested pre-processing the model has quite less chances of confusing between the similar (Not identical) set of sign.
To detect the hand gesture, the proposed method uses CNN algorithm. We have extracted these regions using image pre-processing and image segmentation.The system can be better by further reducing the calculation time. The future scope involves making the program robust in the sense that it should recognise signs through a video and convert it into corresponding sentence. Another goal is to improve the accuracy of the system to detect hand gestures. The techniques used can also be applied to real-time systems. Therefore, the system should be able to perform hand gesture without background restrictions This paper represents a hand sign recognition method using OpenCV. It was also learned during the test that transfer learning alone can take a long time to train the system. But using a standard classifier like Hybrid can greatly reduce that time. This system is much simpler than standard methods of extracting features. Features can also be extracted from RGB images directly. Tests were performed using our data set which achieved 96% accuracy. In the future, we would like to increase the accuracy and try to find the most distinct database,so that it works better.

REFERENCES

1. Teena Varma, Ricketa Baptista, Daksha Chithirai Pandi, Ryland Coutinho,"Sign Language Detection using Image Processing and Deep Learning", International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 07 Issue: 11 | Nov 2020 www.irjet.net p-ISSN: 2395-0072
2. Prof. Radha S. Shirbhate, Mr. Vedant D. Shinde, Ms. Sanam A. Metkari, Ms. Pooja U. Borkar, Ms. Mayuri A. Khandge, "Sign language Recognition Using Machine Learning Algorithm", International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 07 Issue: 03 |Mar 2020 www.irjet.net p-ISSN: 2395-0072Pattern Recogn 29(6):1007–1017

3. Chava Sri Varshini , Guguloth Sreekanth , Gurram Hruday , Mysakshi Chandu , Shaik Khadar Sharif, "Sign Language Recognition" International Journal of Engineering Research & Technology (IJERT) http://www.ijert.org ISSN: 2278-0181 IJERTV9IS050781 Published by : www.ijert.org Vol. 9 Issue 05, May-2020
4. Sakshi Goyal , Ishita Sharma , Shanu Sharma, "Sign Language Recognition System For Deaf And Dumb People", International Journal of EngineeringResearch & Technology (IJERT) Vol. 2 Issue 4, April - 2013 ISSN: 2278-0181
5. Pavitra Kadiyala," SIGN LANGUAGE DETECTION", International Research Journal of Engineering and Technology (IRJET) e-ISSN:2395-0056 Volume: 08 Issue: 07 | July 2021 www.irjet.net p-ISSN: 2395-0072
6. Mehreen Hurroo, Mohammad Elham Walizad, "Sign Language Recognition System using Convolutional Neural Network and Computer
7. Vision", International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 IJERTV9IS120029 Published by : www.ijert.org Vol. 9 Issue 12, December-2020
8. Ghali Upendra, Kokkiligadda Bhuvanendra, D Gayathri," Sign Language Interpreter using Convolution Neural Network", International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 08 Issue: 04 |Apr 2021 www.irjet.net p-ISSN: 2395-0072
9. Raimundo F. Pinto Jr., Carlos D. B. Borges, Antonio M. A. Almeida, and la'lis C. Paula Jr. , "Static Hand Gesture Recognition Based on Convolutional Neural Networks." , Hindawi Journal of Electrical and Computer Engineering Volume2019, Article ID 4167890, 12 pages
10. Jayashree R.Pansare, Sharavan H.Gawande, Maya Ingle, "Real-Time Static Hand Gesture Recognition for American Sign Language (ASL) in Complex Background", Journal of Signal and Information Processing, 2012, Vol.3, pp 364-367.
11. Paranjoy Paul and Dr. G N Rathna , " Real TimeIndian Sign Language Recognition" published by authorcafe,
12. http://reports.ias.ac.in/report/19049/real-time- indian-sign-language-recognition
13. Lionel Pigou, Sander Dieleman, Peter-Jan Kindermans, Benjamin Schrauwen, "Sign Language Recognition using Convolution Neural Network." Lecture Notes in Computer Science, Springer, 2015,pp. 572–78, doi:10.1007/978-3-319-16178-5_40.Kartik Shenoy, Tejas Dastane, Varun Rao, Devendra Vyavaharkar, "Real Time Indian Sign Language (ISL) Recognition", IEEE – 43488, 9th ICCCNT 2018 July 10-12, 2018, IISC, Bengaluru Bengaluru, India
14. Sharvani Srivastava, Amisha Gangwar, Richa Mishra, "Sign Language Recognition System using Tensorflow Object Detection API", International Conference on Advanced Network Technologies and Intelligent Computings. ANTIC 2021: Advanced Network Technologies and Intelligent Computing pp 634–646
15. Nachamai. M., "ALPHABET RECOGNITION OF AMERICAN SIGN LANGUAGE: A HAND GESTURE RECOGNITION APPROACH USING SIFT ALGORITHM", International Journal of Artificial Intelligence & Applications (IJAIA),Vol.4, No.1, January 2013
16. Hoshang kolivand, Saba Joudaki,Mohd Shahrizal Sunar, David Tally, "A new framework for sign language alphabet hand posture recognition using geometrical features through artificial neural network", Neural Computing and Applications(2021) 33:4945–4963
17. https://doi.org/10.1007/s00521-020-05279-7
18. Mayuresh Keni, Shireen Meher, Aniket Marathe., "Sign Language Recognition System", International Journal of Scientific & Engineering Research, Volume 4, Issue 12, December-2013 ISSN 2229-5518