ISSN 2278-3091
Volume 10, No.3, May - June 2021
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse131032021.pdf
https://doi.org/10.30534/ijatcse/2021/131032021

Indian Sign Language Recognition Using Canny Edge Detection

Vipul Brahmankar[1], Nitesh Sharma[1], Saurabh Agrawal[1], Saleem Ansari[1], Priyanka Borse[1], Khalid Alfatmi[2]

[1] Department of Computer Engineering, Shri Vile Parle Kelavani Mandal's Institute of Technology, Dhule, India, {vipul.brahmankar, niteshsharma2702, saurabha861, saleemansari907585, priyankaborse29}@gmail.com

[2] Assistant Professor, Department of Computer Engineering, Shri Vile Parle Kelavani Mandal's Institute of Technology, Dhule, India, Khalid.Alfatmi@svkm.ac.in

ABSTRACT

In our society, it is very difficult for hearing impaired and speech impaired people to communicate with ordinary people. They use sign languages to communicate, which use visually transmitted sign patterns, generally includes hand gestures. Sign languages being difficult to learn and non-universal, there is a barrier of communication between the hearing impaired and ordinary people. To break this barrier a system is required that can convert sign language to voice and vice versa in real-time. Here, we propose a real-time two-way system, for communication between hearing-impaired and normal people, which converts the Indian Sign Language (ISL) letters into equivalent alphabet letters and vice versa.

In the proposed system, using a camera, images of ISL hand gestures are captured. Then Image pre-processing is done so that these images are ready for feature extraction. Here, a novel approach of using the Canny Edge Detection Algorithm. Once the necessary details are extracted from the image, it is matched with the data set, which is classified using Convolutional Neural Network, and the corresponding text is generated. This text is converted into a voice. Similarly, using a microphone, the voice input of an ordinary person is captured and converted into text. This text is then matched with the data set and a corresponding sign is generated. This system reduces the gap in communication between hearing-impaired and ordinary people. Our method provides 98 % accuracy for the 35 alphanumeric gestures of ISL.

Key words: Sign Language, Hand Gesture, Edge Detection, Convolutional neural network, Sign to voice, Voice to sign

1. INTRODUCTION

The sign language is an important technique of communication for hearing impaired and speech impaired people. It is a full-fledged natural language having its own grammar and lexicons. It makes use of gestures instead of sound to convey meanings. Sign language has nicely structured code gestures of which every gesture has a meaning assigned to it. It can be used to express complex meaning by combining basic elements. Sign language is not universal just like spoken language, it has variations and local dialects according to countries and regions. Some of the common sign languages are Indian Sign Language (ISL), British Sign Language (BSL), American Sign Language (ASL), etc. This paper is based on ISL. Figure 1. shows Indian Sign Language alphabet gestures. The majority of people cannot understand sign language.

Thus, it is difficult for hearing impaired and speech impaired people to communicate with normal people. So, is true for normal people to communicate with them. This led to the requirement of sign language interpreters, a person who can translate sign language to spoken language and vice versa. However, such interpreters are limited. This resulted in the development of an automatic sign language recognition system, which could automatically translate the sign into the corresponding text or vice versa without the help of sign language interpreters.

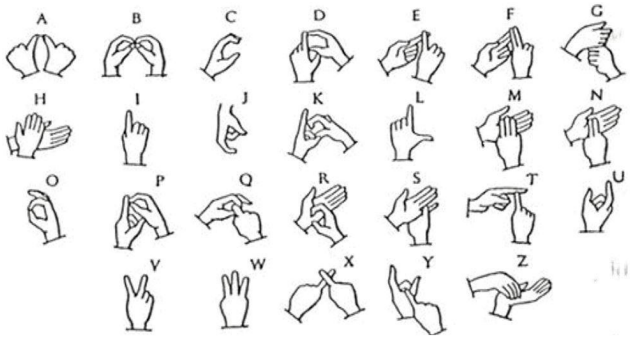

Figure 1: Indian sign language alphabets

Over the past decade, significant research has been made in the sign language recognition system. A few of these systems based on gesture acquiring methods are given below. Hand gloves-based system: Sensors attached to glove are used for detecting hand gesture signals that are in the analog form. These analog signals are converted to a digital signal using ADC. The gloves have accelerometer and flex sensors used to

detect bend signals [1]. Leap motion-based system: Leap Motion is a sensor consisting of two IR cameras and three IR LED's which detects the hand movement and converts that signal into computer commands. IR light signals are generated by LED and the camera generates 300 frames per second of data reflected by that light. These data are sent to the computer for further processing [1]. Kinect sensor-based system: Microsoft Kinect motion sensor consists of an RGB camera, depth sensor, and multi-array microphone that can be used to recognize hand movement and speech [1]. Computer Vision-based system: A web camera is used to capture images. These images are processed using different image processing techniques. Then compared with the dataset for recognition of alphabets [1]. This is the most cost-effective method.

The proposed methodology focuses on a computer vision-based two-way communication system between sign language users and non-users. The system has two modules, Sign-to-Voice (STV) and Voice-to-Sign (VTS). The process in brief for STV consists of acquiring an image using a webcam and preforming image processing on it. Features extracted from the image are matched with the dataset, giving associated text, which is further converted to audio given as output. Similarly, for VTS, voice input captured using a microphone is converted to text. Sign corresponding to text is displayed on screen for sign language users. Such a system can help in the improvement of hearing and speech impaired community via human-computer interaction.

2. SURVEY OF LITERATURE

2.1 Framework

Casam Njagi NYAGA et al. makes use of the grounded theory methodology in order to propose a framework for hand gesture sign language identification [2]. The proposed structure is introduced in a blueprint that shows the steps to be followed when designing hand motion gesture-based communication acknowledgment frameworks [2].

2.2 Computer Vision based systems

Nishi Intwala et al. proposed a system that uses the Convolutional Neural Network algorithm for the identification and categorization of the 26 Indian sign language letters into their identical alphabet letters by capturing a real-time image of that sign and converting it to its text equivalent [3]. Here, the GrabCut algorithm for segmentation and MobileNet as image classification is used. The outcomes indicated a 96% precision for the testing pictures and a precision of 87.69% for runtime image [3].

Pranali Loke et al. proposed an inexpensive approach working from the android application, uses the HSV model for segmentation. Processing is done at the server side where the image is mapped to the corresponding gesture using neural network using MATLAB [4].

Neel Kamal Bhagat et al. proposed a new model (RGB-D) that accomplishes mapping between the depth and the RGB pixels and various models were utilized for preparing where the depth segmented static model accomplishes accuracy of 98.81 % [5].

2.3 Other Existing Systems

Nikhita Praveen et al. presented a smart glove technique for the detection of sign language gestures. It is a two-part framework, one consisting of sensors attached to the smart glove that recognize sign giving corresponding text and other translating text to corresponding audio output [6]. The disadvantage of this technique is that it has higher equipment costs because of sensors on the hand gloves.

Harsh Vardhan Verma et al. presented a methodology for sign language gesture recognition using Microsoft Kinetic and OpenNI framework [7]. The Kinect camera makes use of a structured light technique for the generation of real-time depth maps containing discrete range measurements of the physical position [7]. The proposed framework makes use of a kinetic depth sensor to record the user's motion. The system considers the gestures as a sequence of frames and performs feature extraction on every frame and after performing other calculations the system returns the most significant results as an output [7].

2.4 Image classifier

Comparison between Support Vector Machine (SVM) (Traditional Machine Learning approach) and Convolutional Neural Network (CNN) (Deep Learning approach) is given in Table 1.

Table 1: Comparison between Image Classifier

SVM	CNN
Higher accuracy for small-scale dataset [8].	Higher accuracy for large-scale dataset [8]
Accuracy decreases with picture size [8].	Accuracy increases with picture size [8].
Widely preferred for binary classification.	Mostly used for image classification.

Here, Convolutional Neural Network (CNN) is used for image classification. CNN is a special type of neural network, which is generally widely used in the field of image recognition [8]. The biggest difference between CNN and traditional neural network is that partial connection network is used, and the concept of local receptive field is proposed [8]. It has a strong ability to extract local features of image using convolutional kernel [8]. The CNN network structure is

composed of an input layer, 2 convolutional layers, 2 pooling layers, 2 fully connected layers and output layers, a total of 8 layers [8].

2.5 Edge detectors

Comparison between different edge operators is given in Table 2.

Table 2: Comparison of different edge detectors [10]

Edge Detector	Advantages	Disadvantages
Classical Sobel, prewitt, Kirsch	Simple. Detects edges and their orientations.	Sensitive to noise. Inaccurate.
Zero Crossing Laplacian, Second direction Derivative	Detects edges and their orientations. Have fixed characteristics in all directions.	Responds to some of the existing edges. Sensitive to noise.
Laplacian of Gaussian (LoG) Marr-Hildreth	Finds correct places of edges. Tests wider area around the pixel [10].	Malfunctions at the corners, curves and where the gray level intensity function varies. Does not detects edges and their orientations because of using the Laplacian filter [10].
Gaussian Canny, Shen-Castan	Uses probability for finding error rate, Localization and response. Improves signal to noise ratio, Better detection specially in noise conditions [10].	Complex. False zero crossing. Time consuming.

Compared to Sobel, Prewitt, and Robert's operators, the Canny edge detection algorithm is computationally more expensive. However, the performance of Canny edge detection algorithm is better than all other operators under all scenarios [10]

Mamta Juneja et al. studied and compared different Image Edge Detection algorithms and examined the performance of those algorithms in different environments [9]. The performance was evaluated by examining the edge maps in comparison to each other through statistical assessment [9]. Among the various methods investigated, it was found that the Canny edge detection algorithm has the capability of detecting both strong and weak edges, and seems to be more appropriate than other methods like Laplacian of Gaussian [9].

3. PROPOSED METHODOLOGY

The proposed system incudes real-time two-way communication system consisting of two modules: Sign-to-Voice (STV) and Voice-to-Sign (VTS).

3.1 Sign-to-Voice module (STV)

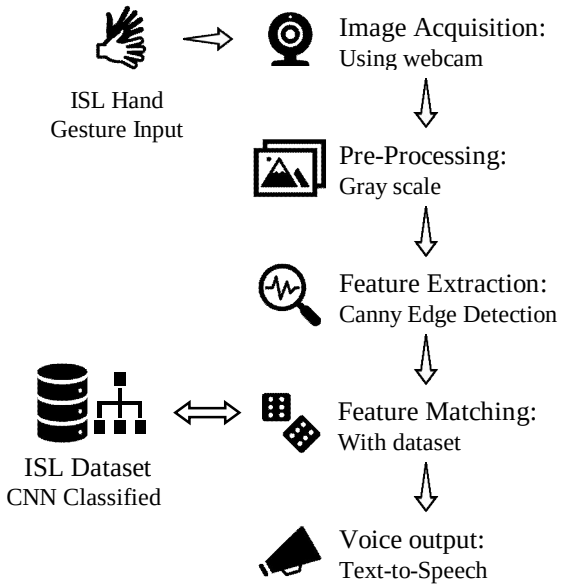

Figure 2: Flowchart of Sign-to-Voice Module

Figure 2. shows flowchart of STV module which includes:

A) Image Acquisition:

Input ISL hand gesture image can be captured using any camera. This image is in RGB format. Fig 3 shows input sign gesture image in RGB format.

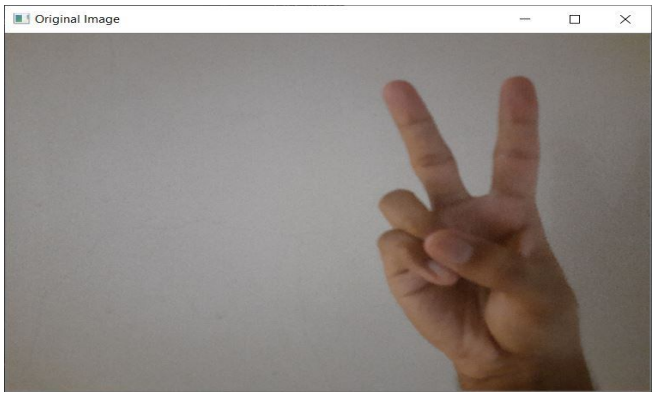

Figure 3: Original RGB image

B) Image Preprocessing:

The RGB format (3D) image captured using camera is converted to a Grey-Scale (1D) image. That means the input color image will be converted to black and white image.

Figure 4: Gray-scale image

Figure 4. shows image after converting it to gray-scale. This is a one-dimensional image.

C) *Feature Extraction:*

Here, Canny Edge detection algorithm (optimal edge detector) is used.

Canny Edge Detection algorithm is a multi-stage algorithm detecting a wide range of edges in images. It is composed of 5 steps [11]:

i) Noise Reduction:

Edge detection results are highly sensitive to image noise. Hence, any noise in image is filtered out by convolving image with Gaussian smoothing filter.

ii) Gradient computation:

Compute gradient magnitude and gradient angle. It calculates edge strength(intensity) and edge direction. The Sobel operator uses a pair of 3x3 convolution masks giving G_x and G_y i.e. gradient in x and y directions respectively. Then, the approximate absolute gradient magnitude at each point can be calculated using (1). Once, gradient in x and y directions are known, gradient angle can be easily calculated using (2).

$$G = \sqrt{G_x^2 + G_y^2} \quad (1)$$

$$\theta = tan^{-1}\frac{G_x}{G_y} \quad (2)$$

Now, as shown in the fig 5, gradient angle is rounded to one of four angles which are 0, 45, 90, 135 representing vertical, horizontal and two diagonals.

For example, θ in [0°, 22.5°] or [157.5°, 180°] is given specific value as 0°.

Similarly, θ in [22.5°, 67.5°] is given specific value as 45°, θ in [67.5°, 112.5°] is given specific value as 90° and θ in [112.5°, 157.5 °] is given specific value as 135°.

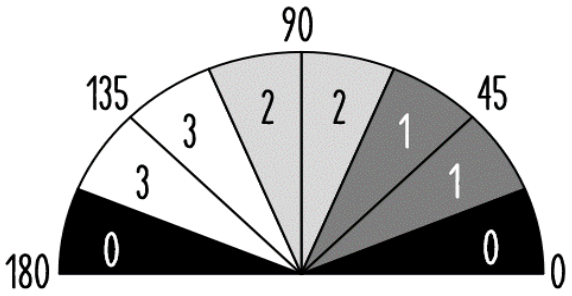

Figure 5: Partition of edge angles into sectors

iii) Non-maxima suppression:

After applying Sober filter, we get edge image having non-uniform thickness and intensity. Non-maxima suppression is performed to thin out the edges and reduce blurring effect. Here, it compares gradient magnitude at every pixel with two neighbouring pixels depending on its rounded gradient angle. If magnitude of pixel in consideration is greater than two neighbours, it will be considered to be on edge, else it will be set to 0. This results into thinned and sharp edges

Figure 6 shows two neighbours to be considered for comparison according to rounded gradient angle of pixel in consideration.

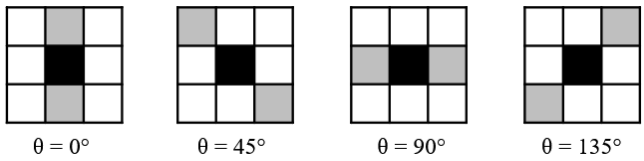

Figure 6: Neighbours to consider for Non-maxima suppression.

For example, if value of rounded gradient angle of pixel is 90, it will set to zero if its gradient magnitude is less than gradient magnitude of pixels to its west and south.

iv) Double threshold:

As only high intensity pixels are required, pixels with weak gradient magnitude are required to be filtered out. In double threshold, high threshold and low threshold values are defined. Intensity higher than high threshold are strong pixels, lower than low threshold are reduced to 0 and intensity between both are weak pixels.

v) Hysteresis:

Finally, hysteresis includes converting weak pixels to

strong pixels. For each weak pixel, if any one of the 8-connected neighbourhood pixels is strong, that weak pixel is converted into strong pixel. Intensity of all the remaining weak pixels are reduced to 0.

D) Feature Matching:

The resulting image after processing, is compared with the images in the dataset to recognize the corresponding meaning of gesture in text. A dataset consisting of images of ISL gestures is stored in the database. The images in dataset are also feature extracted using Canny edge detection. This dataset is fed to Convolutional Neural Network for image classification.

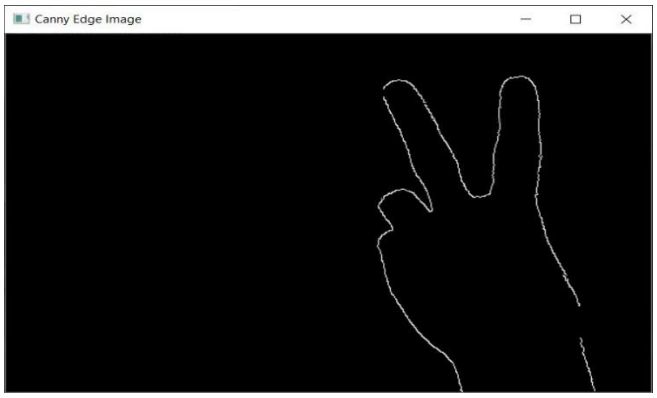

Figure 7: Image after applying Canny Edge detection algorithm

Figure 7 shows final image obtained after applying Canny edge detection algorithm.

E) Voice output:

Voice of corresponding sign gesture is given as output by using Text-to-Speech engine.

3.2 Voice-to-Sign (VTS):

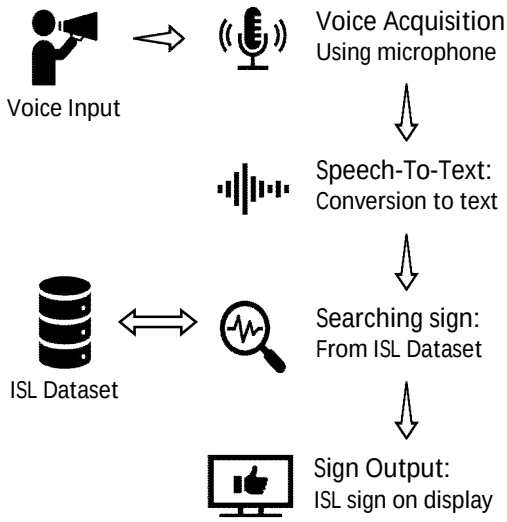

Figure 8: Flowchart of Voice-to-Sign Module

Figure. 8. shows flowchart of VTS module which includes:

A) Voice acquisition:

Input voice of ordinary person can be captured using microphone.

B) Text conversion:

Voice input is converted to text by using Speech-to-Text engine

C) Searching sign:

Sign corresponding to the text obtained is searched in the dataset. A dataset having digitally created ISL hand gestures and their corresponding text meaning is stored in database.

D) Sign output:

The sign obtained from dataset is displayed on the screen.

4. IMPLEMENTATION

The system is implemented using Python 3.8 and GUI is developed using PyQt5.

4.1 Dataset:

A) Data creation and augmentation:

As no standard ISL dataset is available, we created our own dataset for this experiment. The dataset consists of 35 (viz. A-Z and 1-9) Alphanumeric gestures of ISL. The dataset was captured from five subjects, comprising of male and female genders. For each gesture, we captured 1400 images of size 128 x 128. The total numbers of images were 49000. These images were captured using OpenCV and webcam. The dataset was then split into 70% and 30% for the training and testing set, respectively.

The images in the training set were augmented by shearing up to 0.2 scales, zooming up to 0.2 scales, and flipping the images horizontally. This prevents the classifier from overfitting and also improves its efficiency.

B) Dataset Preprocessing:

Images in dataset are transformed from BRG to HSV. Pixels in given HSV range are retained to isolate skin color (hand gesture) from background. After segmentation the images are converted to gray scale. Further, we apply canny edge detection algorithm to get edges. This is done using python OpenCV. Figure 9 shows dataset preprocessing flow.

The dataset is ready to train the image classifier.

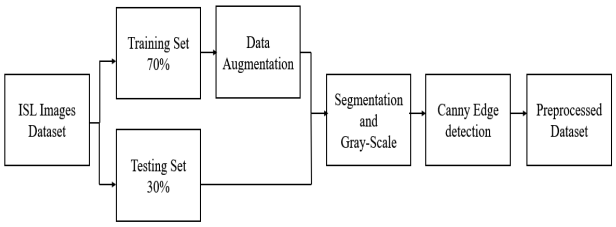

Figure 9: ISL image dataset preprocessing flow

4.2 Image classification:

We have used keras and tensorflow for implementing CNN model.

A) **CNN architecture:**

There are 15 layers in our CNN model. Each 128x128 image in the training set is fed to a series of convolutional, ReLU, and max-pooling layers, which reduces the images to 128 feature maps of size 6x6. Then they are fed to three fully-connected layers with 4608, 128, and 96 neurons respectively and each accompanied by activation function ReLu. For the first two fully connected layers dropout of 40% is used to avoid overfitting. Finally, the last layer uses softmax activation to classify the gestures into 35 classes. Figure 12. shows the architecture of the CNN model.

5. RESULTS AND DISCUSSION

Here we discuss the training of the model discussed in the previous section and its results. The model was trained on 'ADAM' optimizer and 'categorical_crossentropy' loss function with 100 epochs and batch size as 32.

The CNN architecture achieves training accuracy of 95.9% and validation accuracy of 98.5%.

The plot of epochs vs Accuracy and the plot of epochs vs Loss for the CNN model are shown in Figure 10 and Figure 11 respectively.

For the evaluation of real-time performance of the model, we asked 10 people to perform the gestures in front of the webcam in real time. This resulted into accurate prediction of all the gestures by all the people.

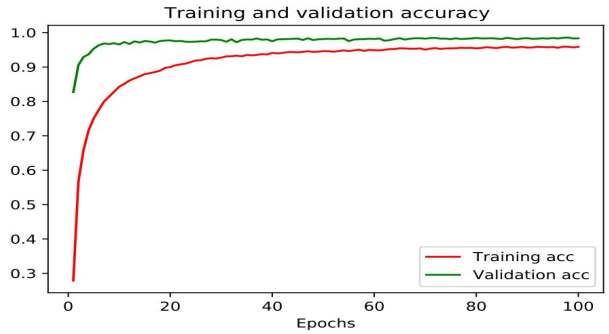

Figure 10: Training and testing accuracy of model

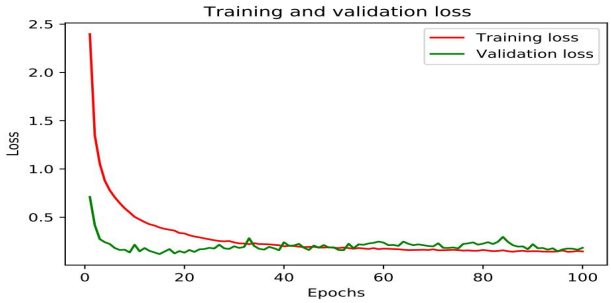

Figure 11: Training and testing loss of model

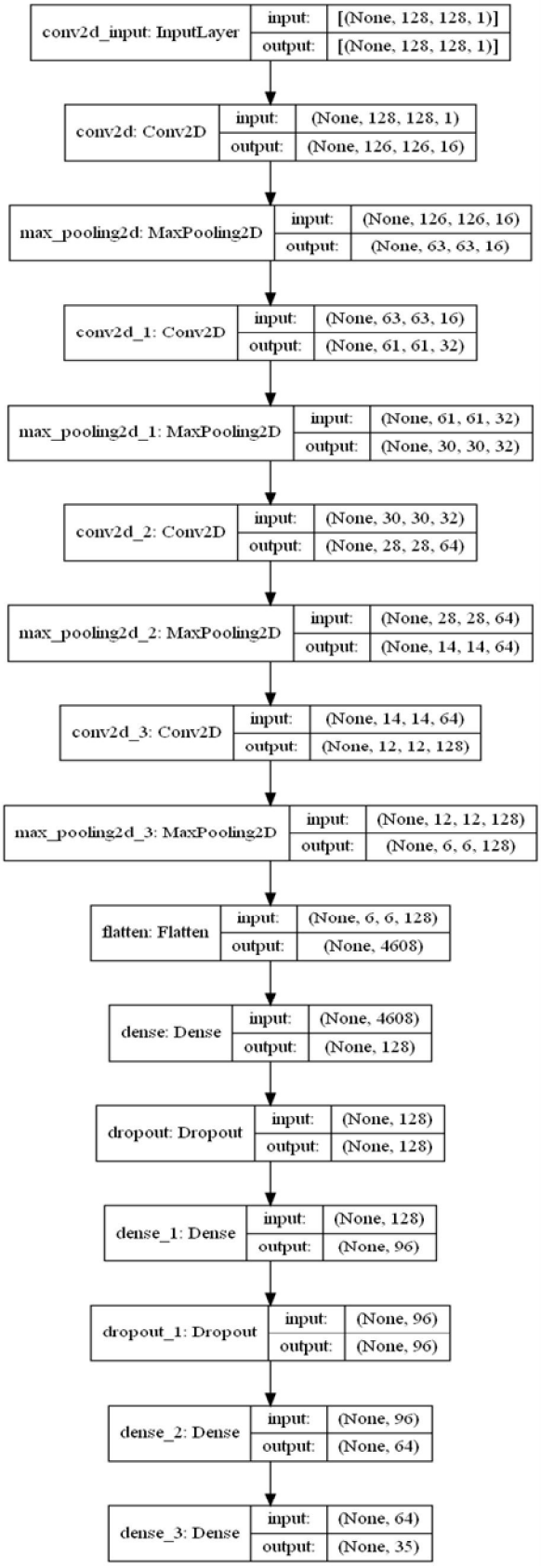

Figure 12: CNN Architecture

Classification Report

	precision	recall	f1-score	support
1	0.96	0.96	0.96	420
2	0.99	0.99	0.99	420
3	0.98	0.99	0.99	420
4	1.00	0.95	0.97	420
5	0.99	0.99	0.99	420
6	0.99	0.94	0.97	420
7	1.00	1.00	1.00	420
8	0.98	0.99	0.98	420
9	0.99	1.00	1.00	420
A	0.97	1.00	0.98	420
B	0.93	1.00	0.96	420
C	0.99	1.00	0.99	420
D	0.99	0.99	0.99	420
E	0.97	0.98	0.97	420
F	1.00	0.99	0.99	420
G	1.00	1.00	1.00	420
H	1.00	1.00	1.00	420
I	1.00	1.00	1.00	420
J	0.90	1.00	0.95	420
K	1.00	0.89	0.94	420
L	0.98	0.94	0.96	420
M	1.00	1.00	1.00	420
N	1.00	1.00	1.00	420
O	0.97	1.00	0.99	420
P	0.99	0.86	0.92	420
Q	1.00	0.99	0.99	420
R	1.00	1.00	1.00	420
S	1.00	1.00	1.00	420
T	1.00	1.00	1.00	420
U	1.00	0.99	1.00	420
V	0.99	1.00	1.00	420
W	1.00	1.00	1.00	420
X	0.98	1.00	0.99	420
Y	0.95	0.99	0.97	420
Z	0.98	1.00	0.99	420
accuracy			0.98	14700
macro avg	0.98	0.98	0.98	14700
weighted avg	0.98	0.98	0.98	14700

Figure 13: Model Evaluation Report

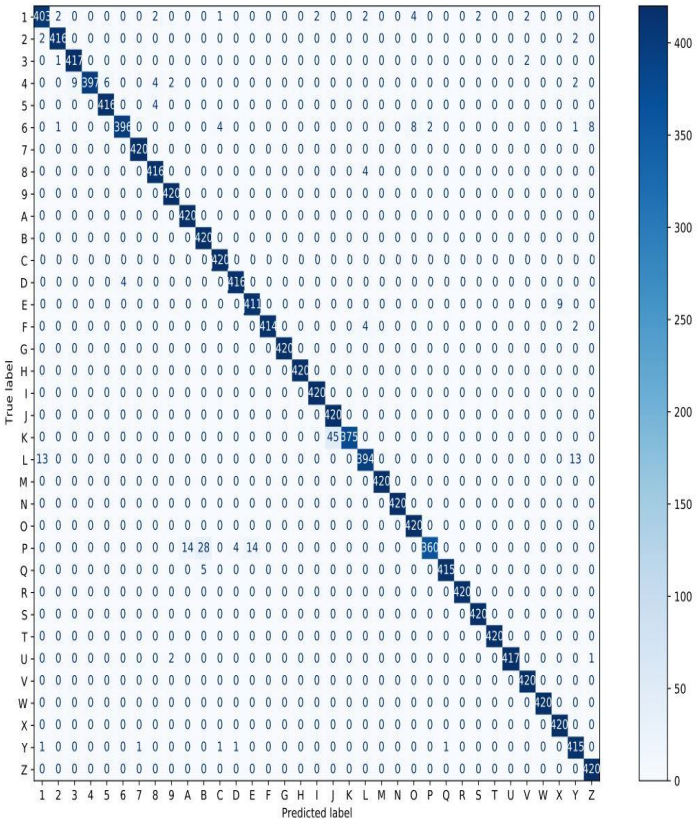

Figure 14: Confusion Matrix

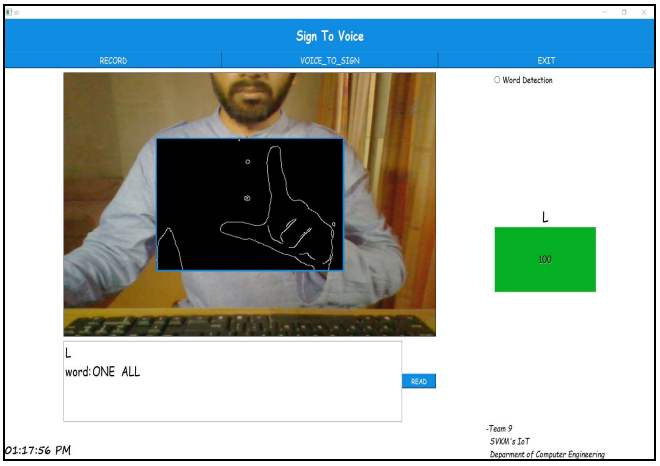

Figure 15: Sign-To-Voice module example

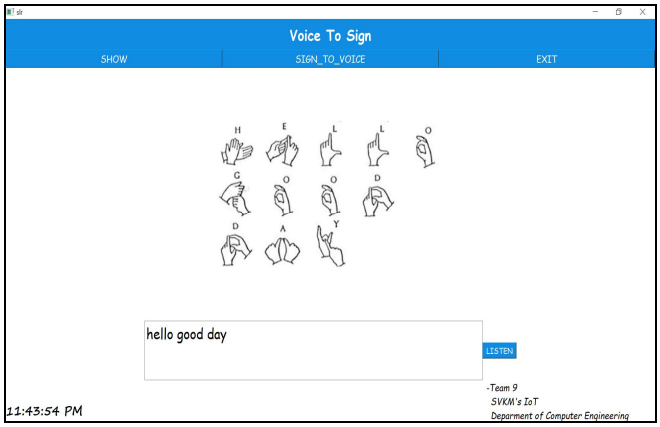

Figure 16: Voice-To-Sign module example

Figure 13 shows CNN model evaluation report. Figure 14 shows a 35 x 35 confusion matrix for 35 ISL gestures considered in project. Figure 15 and figure 16 shows GUI for STV and VTS module respectively.

6. CONCLUSION

The main purpose of this paper is, helping and serving the hearing and speech impaired community for communicating with ordinary people. In this paper, we propose a real-time two-way communication system that is capable of convert Sign-to-Voice and Voice-to-Sign. Using the CNN model and image processing, a robust sign language hand gesture recognition system can be developed. The proposed method provides a different approach for ISL hand gesture recognition using canny edge detection with an accuracy of 98%. The multiple stages of the Canny edge detection algorithm ensure the accuracy of the system.

In the future, further research can be focused on the recognition of dynamic ISL gestures and sentence formation. Portability of the system in order to extend it to mobile phones and smart home devices.

REFERENCES

1. H. B. D. J. T. **Review on Classification Methods used in Image based Sign Language Recognition System**, *IJRITCC*, vol. 5, no. 5, pp. 949 - 959, May 2017.
2. C. N. Nyaga and R. D. Wario. **Towards a Sign Language Hand Gesture Recognition Design Framework**, *in 2020 IST-Africa Conference (IST-Africa)* , Kampala, Uganda, 2020, pp. 1-8.
3. N. Intwala, A. Banerjee, Meenakshi and N. Gala. "**Indian Sign Language converter using Convolutional Neural Networks**, *in 2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*, Bombay, India, 2019, pp. 1-5.
4. P. Loke, J. Paranjpe, S. Bhabal and K. Kanere. **Indian sign language converter system using an android app**, *in 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)*, Coimbatore, 2017, pp. 436-439.
5. N. K. Bhagat, Y. Vishnusai and G. N. Rathna. **Indian Sign Language Gesture Recognition using Image Processing and Deep Learning**, *2019 Digital Image Computing: Techniques and Applications (DICTA)*, Perth, Australia, 2019, pp. 1-8.
6. N. Praveen, N. Karanth and M. S. Megha. **Sign language interpreter using a smart glove**, *in 2014 International Conference on Advances in Electronics Computers and Communications*, Bangalore, 2014, pp.1-5.
7. H. V. Verma, E. Aggarwal and S. Chandra. **Gesture recognition using kinect for sign language translation**, *in 2013 IEEE Second International Conference on Image Information Processing (ICIIP-2013)*, Shimla, 2013, pp. 96-100.
8. Pin Wang, En Fan, Peng Wang. **Comparative Analysis of Image Classification Algorithms Based on Traditional Machine Learning and Deep Learning**, *Pattern Recognition Letters (2020)*, vol. 141, pp. 61-67, Jan. 2021.
9. Mamta Juneja, Parvinder Singh Sandhu. **Performance Evaluation of Edge Detection Techniques for Images in Spatial Domain**, *International Journal of Computer Theory and Engineering*, vol. 1, no. 5, pp. 614-621, 2009.
10. R. Maini and H. Aggarwal. **Study and comparison of various edge detection techniques**, *IJIP*, vol. 3, 2006.
11. Ramesh Jain, Rangachar Kasturi, Brian G. Schunk. **Machine Vision**, *in MCGraw-Hill*, 1995.