International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.8, Issue.9, September 2020 E-ISSN: 2347-2693

Indian Sign Language Recognition for Static and Dynamic Hand Gestures

Manav Prajapati[1*], Mitesh Makawana[2], Sahil Hada[3]

[1,2,3]Dept. of Computer Engineering, Birla Vishvakarma Mahavidyalaya, V.V. Nagar, Anand, India

[*]*Corresponding Author: manavprajapati1398@gmail.com, Tel.: +91-7984361284*

DOI: https://doi.org/10.26438/ijcse/v8i9.5458 | Available online at: www.ijcseonline.org

Received: 08/Sept/2020, Accepted: 15/Sept/2020, Published: 30/Sept/2020

Abstract — Humans are called as social animals and because of that communication becomes a very integral part of a human being. Humans use verbal and non-verbal forms of speech for communication purposes, but not all humans are capable of verbal speech, for e.g. Deaf and Mute people. Hence, Sign Languages are developed for them, but still there is a hindrance in the communication for them. So, using the hand gestures, this paper presents a system where CNN network is used to for the classification of Alphabets and Numbers. CNN is used because alphabets and number gestures are static gestures in Indian Sign Language and CNNs give very good results for image classification. This uses hand-masked (skin-segmentation) images for training the model. For the dynamic hand gestures, the system uses LSTM network for the classification task. LSTM are well known for accurately predicting the data which is distributed in time-frame. This paper presents two models, CNN and LSTM for predicting different type of hand gestures i.e. static as well as dynamic.

Keywords **—** Indian Sign Language, CNN, Skin-segmentation, LSTM

NOMENCLATURE

ISL: Indian Sign Language
ROI: Region of Interest
CNN: Convolutional Neural Network
RMSProp: Root Mean Square Propagation
LSTM: Long Short Term Memory

I. INTRODUCTION

Everyone uses language to communicate with others whether it is English, Spanish, Sign language or Language of touch or smell. Sign language is language used by deaf and mute people for conversation. It varies from country to country with its own vocabulary. Indian sign language (ISL) is a collection of the gestures used by Indian deaf and mute communities. These gestures also vary in different regions of India.

It is always a challenge for the normal person to communicate with deaf-mute people and vice versa. Sign language interpreter is a solution for this problem. It provides a bridge between normal and deaf-mute community for communication. There are mainly two approaches for sign language recognition, glove based, and computer vision based[1]. In this paper, computer-vision based approach is discussed for interpreting the ISL in two different ways. Recognition of alphabet of ISL consists of frame extraction from the camera, hand masking, feature extraction, and classification and recognition. This is the identification of the alphabet from a single frame. The second way is to recognize gestures for words. The sequence of frames from the camera is used to identify the gesture. It consists of the same modules as alphabet recognition however, it took a sequence of frames instead of only one frame. This paper focused on ISL recognition using deep learning and computer vision.

The rest of the paper is organized as follows; Section II presents related work done in gesture recognition. Section III contains a methodology about two approaches of Indian sign language recognition. The first approach for static hand gestures and the second for dynamic hand gestures. Discussion on results and conclusion is explained in Section IV and V respectively.

II. RELATED WORK

Many techniques are developed to recognize sign language. They are mainly divided into two approaches, using various motion tracking sensors or computer vision. Lots of study work is done on a sensor-based approach by using gloves and wires [1, 2, 3]. The wearing of these devices continuously is inconvenient therefore; further work is majorly focus on computer-vision based approach. Lots of work has been carried out using computer-vision based approach. Authors have suggested different ways to recognize sign language using CNN (Convolution Neural Network), HMM (Hidden Markov Model) and contour [4, 5, 6, 7]. Various methods are used for image segmentation like, HSV and color difference images [4, 5]. Authors have proposed the Support Vector Machine (SVM) method for classification [6, 8]. Archana and Gajanan have also compare various methods for segmentation and feature extraction [9]. All the previous papers have recognized the

ISL alphabets successfully. However, in real life, the deaf and mute people use word's gestures to convey the message. We can identify words using these previous techniques if the word has static gesture.

Many words in ISL required motion of hand. Video classification methods are good to identify these dynamic gestures instead of simple image classification technique. Video-based action recognition has already drawn attention from various researchers [10, 11, 12]. Instead of taking color image data for each frame of video, some researchers did difference between consecutive frames and then randomly gives these segments to TSN (Temporal Segment Network) [11]. Sun, Wang and Yeh discuss categorization and captioning for video using LSTM (Long Short Term Memory) [13]. Juilee, Ankita, Kaustubh and Ruhina have proposed methods for Indian sign language recognition using videos [14].

After some searching for sign language recognition system, we found that most of the research is carried out only on static sign language gestures or classification of video for various actions that leads us to study the dynamic gestures identification using video recognition techniques.

III. METHODOLOGY

III. 1. Static Gesture Classification

We have carried out experiments on the dataset provided by [15]. The dataset contains 36 folders representing 0-9 and A-Z, with each folder consisting of the skin-color segmented hand images for its corresponding alphabet or number. For every alphabet and number, there are 220 images each with a size of 110 x 110 pixels. Figure 1 shows the image for each label from the dataset.

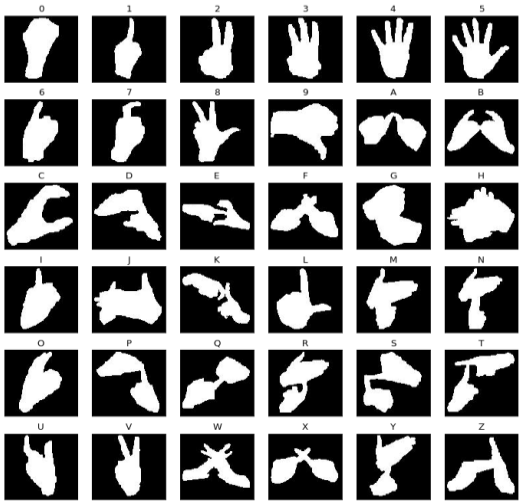

Figure 1. Hand-masked image dataset

All these images are split into training and testing dataset with 80:20 ratio. So, the training dataset contains 6336 images and the testing dataset contains 1584 images corresponding to 36 classes. Also, as the number of images per class is less, we did data augmentation to feed more data to the CNN model. The data augmentation includes tasks like rotation, width_shift, height_shift, rescaling, etc. The CNN model specification is mentioned in the following Table 1.

Table 1 CNN Specification

Property	Value
Convolution Layer	3 Layers (32, 64, 128 nodes)
Convolution Layer (Kernel Size)	3, 3, 2
Max Pooling Layer	3 Layers - (2, 2)
Fully Connected Layer	128 nodes
Output Layer	36 nodes
Activation Used	Softmax
Optimizer	RMSProp
Hyperparameters	
Learning rate	0.01
No. of epochs	10

After training the model, the following steps are performed to predict the output.

1. *Frame Extraction*: OpenCV library was used to capture the video from the webcam for live-predictions. After capturing the video, a single frame is taken and we define a Region of Interest (ROI) in that frame. Region of Interest is the area where a person performs their hand-gestures.
2. *Skin-segmentation*: The ROI from the frame is to be converted to hand-masked images for giving it to the model for predicting purpose. Firstly, we need to blur the image in order to reduce the noise. This task is carried out by applying Gaussian Blur. After blurring, the ROI is converted from RGB to HSV color scale. Converting image to HSV color scale helps for better skin detection rather than in RGB. Then the lower and higher limits are set for extracting the skin. Here in our case, we used (108, 23, 82) as lower range and (179, 255, 255) as a higher range. This range gave us the best results. After selecting the range, we compare each pixel values and if the pixel value is not in range, then it is converted to black color otherwise it is converted to a white pixel. This gives us hand-masked images. Still, the hand-masked image has noise in it and an uneven edge. To fix this problem, we use Dilate and Erode functions available in OpenCV to smooth out the edges.
3. *Prediction*: The ROI from the frame is converted to a hand-masked image. This hand-masked image is then given as input to the CNN model and prediction is performed. This predicted value which 0-9 or A-Z is given as output on the original frame. But this leads to another problem, the output on the frame continuously flickers. To solve this problem, we used the prediction of 25 frames and used the maximum predicted class as the output.

Figure 2 shows the hand-masked image and final output for alphabet L.

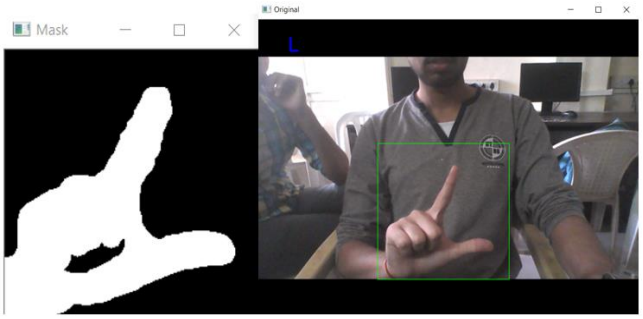

Figure 2. Hand-mask and Predicted Output

III. 2. Dynamic Gesture Classification

Neural Networks can help us predict values from complex data and majority of the time, the inputs are not related to time or they are not needed in chronological order. This was the case with static gestures in ISL and hence multi-layer CNN architecture would suffice. But for dynamic gestures, the previous state must be preserved and CNNs are not able to do that. So in this case, LSTM networks become useful. LSTMs are a type of Recurrent Neural Network (RNN) having a chain-like structure of repeating modules that helps to learn long-term dependencies in the sequential data.

Model: "sequential_1"

Layer (type)	Output Shape	Param #
time_distributed_1 (TimeDist	(None, 8, 7, 7, 1280)	2257984
time_distributed_2 (TimeDist	(None, 8, 1280)	0
lstm_1 (LSTM)	(None, 8, 64)	344320
dropout_1 (Dropout)	(None, 8, 64)	0
lstm_2 (LSTM)	(None, 64)	33024
dense_1 (Dense)	(None, 64)	4160
dropout_2 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 24)	1560
dropout_3 (Dropout)	(None, 24)	0
dense_3 (Dense)	(None, 5)	125

Total params: 2,641,173
Trainable params: 2,607,061
Non-trainable params: 34,112

Figure 3 Model Architecture

Firstly, in order to train the model, we need data. We created a dataset which contains videos of 12 classes. These classes are Today, Tomorrow, Yesterday, Bye, Mom, Dad, Time, Eat, What, Me(I), Thank-you, and Namaste. Each class has approx. 20-25 hand-gesture videos of 5-7 seconds. The videos are captured using the back camera of mobile-phones having an average of 28-30 frames per second. The dataset is created among different persons with different backgrounds to ensure various circumstances are covered for training. Now, this dataset is split with 20% data for validation and rest for training. The model architecture is shown in Figure 3.

In the input, we continuously pass a sequence of 8 frames/images extracted from the videos of our training dataset. Before giving these 8 frames as input, we apply RGB Difference filter. In RGB Difference, we subtract a current frame from their previous frame. So only the changing pixels remain in the frame and the rest of the static image gets removed. In this way, it helps in catching visual features that are changing with time. Here in our case, it helps us in capturing a pattern of gesture and it also removes the background, hence becoming independent of various background scenarios.

These frames are then resized to size 224 x 224 pixels, the reason being the next layer is the MobileNetV2 layer which only accepts image size of a maximum 224 x 224 pixels. We have used MobileNetV2 with 'Imagenet' weights as the pre-trained model. MobileNetV2 is used for image segmentation and using it as a pre-trained model helps us to eliminate the task of building a CNN model for image segmentation. TimeDistributed layer is used which results in passing these 8 frames individually to separate MobileNetV2 layers. Now, in order to insert a sequence of frames into LSTM, we need to flatten it, hence we use the TimeDistributed GlobalAveragePooling layer. Finally, we have a multi-layer LSTM structure with some Dropout and Fully Connected layers to reduce overfitting. LSTM will help us in recognition of pattern forming in dynamic/moving hand gestures. Lastly, SGD is used as an optimizer because it gives better results when there is less dataset available. Adam also gives good results when the dataset is large.

IV. RESULTS AND DISCUSSION

IV. 1. Static Gesture Classification

During the testing of the model for static hand gestures and deciding the best architecture, we used various optimizers like RMSProp, SGD, Adam and trained our model for 10 epochs each. Well, RMSProp gave us the best results with 73.6% accuracy. The graph of accuracy vs. epoch is shown in Figure 4.

Skin segmentation is an integral part of our system for predicting static hand gestures. We concluded that (108, 23, 82) as a lower range and (179, 255, 255) as a higher range gives the best results. Figure 5 and 6 shows the skin-segmentation and predicted gesture.

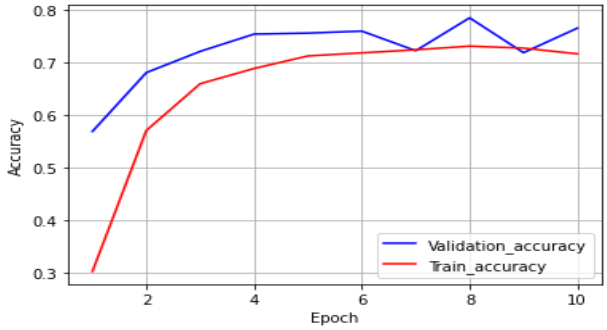

Figure 4 Accuracy vs. Epoch graph for the CNN model

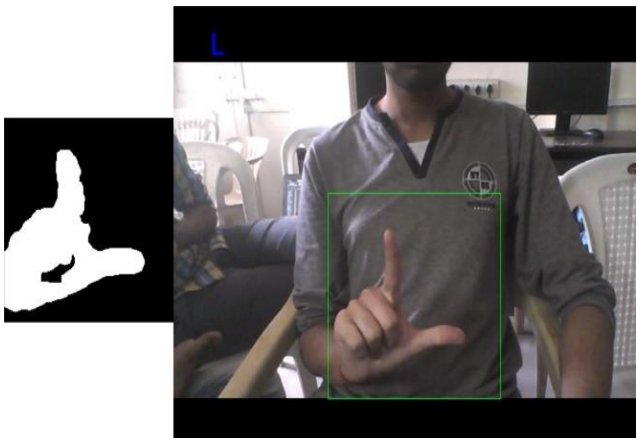

Figure 5 Prediction of Alphabet Letter 'L'

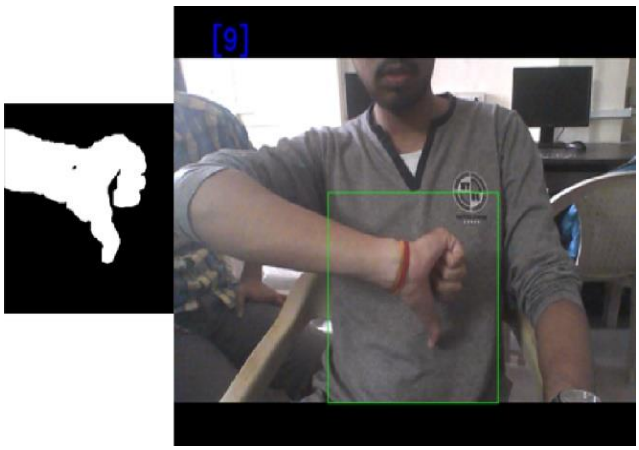

Figure 6 Prediction of Number '9'

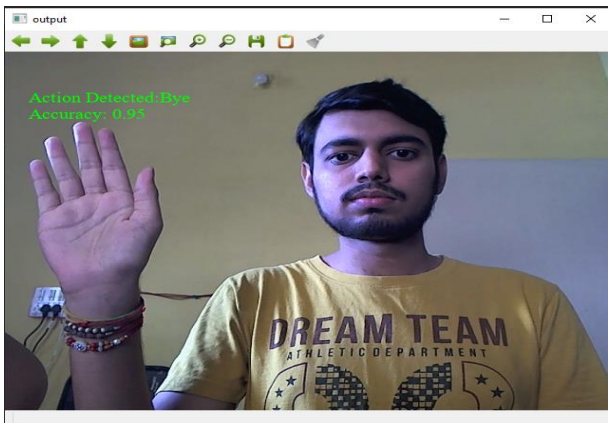

Figure 7 Prediction of Word "Bye"

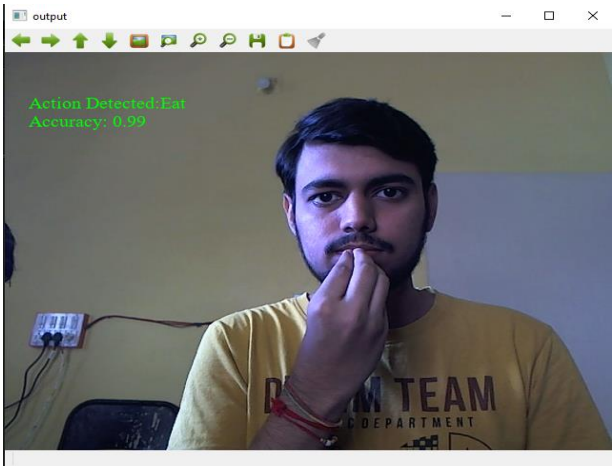

Figure 8 Prediction of Word "Eat"

There are few limitations in recognizing the static hand gesture using skin-segmentation. The foremost one is that it requires a non-skin colored background. If the background contains color which is in the range of skin color then it is hard to mask skin, hence the result will be wrong prediction. For example, if the background is of yellowish shade which comes under our range then this issue occurs. The second issue is with similar-looking hand-gestures. There is an overlap of the same gestures used in alphabet and number. For example, the alphabet 'V' and number '2' have the same gestures and hence our system is not able to differentiate properly. There is also be a problem of similar-looking hand-gestures which results in lower accuracy. For example, alphabet 'M' and 'N' are very similar. Another similar looking pair is 'F-X' and '1-I'.

IV. 2. Dynamic Gesture Classification

In dynamic hand gesture recognition, we have used multi-layer LSTM for capturing the pattern formed by moving gestures. We have trained our model to recognize 12 frequently used words. The model gives approx. 85% accuracy. Figure 7 and 8 show the predicted gestures and accuracy. If there is no action performed, the result will be "No action performed".

To overcome the issue of background color, we have used RGB Difference to eliminate the static portion of the sequence of frames. This also helps in detecting a pattern in hand gesture as only the moving hand remains in the frame. The only issue with this approach is if a person is in a moving background, then the sequence of frames will also have a background in it and hence it will affect the prediction accuracy. Furthermore, to increase accuracy one can add more videos to the dataset with diverse backgrounds and people.

V. CONCLUSION AND FUTURE SCOPE

The Deaf-mute community faces the problem in communication every day. This paper presents two approaches to recognizing hand-gestures: Static and Dynamic gestures. For static gesture classification, a CNN model is implemented which classifies the gestures into alphabets (A-Z) and numbers (0-9) with 73% accuracy. Along with the model, it uses skin-segmentation for hand-masking. For dynamic gestures, we trained our model having multi-layer LSTM with MobileNetV2 for 12 words and gave very satisfying results with 85% accuracy. For the future work in static gestures, one can build another approach for skin-segmentation which is independent of skin-color. For dynamic gestures, one can also increase the size of the dataset with various backgrounds.

REFERENCES

[1] M. Mohandes, M. Deriche, J. Liu, "Image-Based and Sensor-Based Approaches to Arabic Sign Language Recognition", IEEE Transactions on Human-Machine Systems, Vol.**44**, Issue.**4**, pp. **551-557**, **2014**.

[2] C. Zhu, W. Sheng, "Wearable Sensor-Based Hand Gesture and Daily Activity Recognition for Robot-Assisted Living", IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, Vol.**41**, Issue.**3**, pp.**569-573**, **2011**.

[3] T. Jaya, and V. Rajendran, "Hand-Talk Assistive Technology for the Dumb", International Journal of Scientific Research in Network Security and Communication (IJSRNSC), Vol.**6**, Issue.**5**, pp.**27-31**, **2018**.

[4] L. K. Ramkumar, S. Premchand, G. K. Vijayakumar, "Sign Language Recognition using Depth Data and CNN", SSRG International Journal of Computer Sciences and Engineering (SSRG - IJCSE), Vol.**6**, Issue.**1**, pp.**9-14**, **2019**.

[5] P. Gupta, A. K. Agrawal, S. Fatima, "Sign Language Problem and Solutions for Deaf and Dumb People", In the Proceedings of the International Conference on System Modeling & Advancement in Research Trends (SMART), Moradabad, India, **2014**.

[6] A. S. Nikam, A. G. Ambekar, "Sign Language Recognition Using Image Based Hand Gesture Recognition Techniques", In the Proceedings of the Online International Conference on Green Engineering and Technologies (IC-GET), Coimbatore, India, pp.**1-5**, **2016**.

[7] N. S. Lele, "Image Classification Using Convolutional Neural Network", International Journal of Scientific Research in Computer Science and Engineering (IJSRCSE), Vol.**6**, Issue.**3**, pp.**22-26**, **2018**.

[8] J. L. Raheja, A. Mishra, and A. Chaudhary, "Indian Sign Language Recognition Using SVM", Pattern Recognition and Image Analysis, Vol.**26**, Issue.**2**, pp.**434-441**, **2016**.

[9] A. S. Ghotkar, G. K. Kharate, "Study of Vision Based Hand Gesture Recognition Using Indian Sign Language", International Journal on Smart Sensing and Intelligent Systems, Vol.**7**, Issue.**1**, **2014**.

[10] K. Simonyan, A. Zisserman, "Two-stream Convolutional Networks for Action Recognition in Videos", Advances in Neural Information Processing Systems, pp.**568-576**, **2014**.

[11] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D Lin, X. Tang, and L. Van Gool, "Temporal Segment Networks: Towards Good Practices for Deep Action Recognition", In the Proceedings of the European conference on Computer Vision, pp.**20-36**. Springer, Cham, **2016**.

[12] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar and L. Fei-Fei, "Large-scale Video Classification with Convolutional Neural Networks", In the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.**1725-1732**, **2014**.

[13] J. Sun, J. Wang, T. C. Yeh, "Video understanding: from video classification to captioning". In the Proceedings of the Computer Vision and Pattern Recognition, pp.**1-9**, Stanford University, **2017**.

[14] J. Rege, A. Naikdalal, K. Nagar, R. Karani, "Interpretation of Indian Sign Language through Video Streaming", International Journal of Computer Science and Engineering (IJCSE), Vol.**3**, Issue.**11**, pp.**58-62**, **2015**.

[15] Pradip Patel, Narendra Patel, "Vision Based Real-time Recognition of Hand Gestures for Indian Sign Language using Histogram of Oriented Gradients Features", in International Journal of Next-Generation Computing, Vol. **10**, No. **2**, July **2019**.

AUTHORS PROFILE

Mr. Manav Prajapati completed his Bachelor of Technology in Computer Engineering from Birla Vishvakarma Mahavidyalaya (BVM), V.V. Nagar, Anand in 2020. He is currently working at Tata Consultancy Services Ltd. as Assistant System Engineer Trainee. The major interests are towards Machine Learning, Computer Vision, Web Development, and Software Engineering.

Mr. Mitesh Makwana completed his undergraduate study in Computer Engineering from Birla Vishvakarma Mahavidyalaya, V.V. Nagar, Anand in 2020. The major interests are towards Machine Learning, Deep Learning, and Android Development.

Mr Sahil Hada completed his Bachelor of Technology in Computer Engineering from Birla Vishvakarma Mahavidyalaya, V. V. Nagar, Anand in 2020. His major interests are Machine Learning, Game Development, and Software Engineering. He is planning to pursue a Masters of Information Technology in Information Systems and Data Science.

