Conversion of Sign Language into Devanagari Text Using CNN

Vimal Gaur, Rinky Dwivedi, Pankhuri, and Shivam Dhar

Abstract Sign language is an increasingly valuable interactive medium for both deaf and hearing disabled individuals. These languages are considered as a group of predefined languages that use the visual–manual mode to communicate knowledge. These languages are considered to be unitary for deaf people. This paper proposes a useful hand motion picture recognition design method to provide a simple way to interact with the visually impaired and the average person. This is accomplished by considering Devanagari Sign Language finger-spelling comprehension in real time. In this paper, authors have created dataset of Devanagari Sign Language by capturing the gestures through the Web camera. From these captured images, we found the region of interest (ROI). Convolutional neural network has been used for classification as it eases the process of feature creation. For training this network, a hand signal has been used as input and the resultant output produces character when a suitable match is found with an accuracy of 98% and testing accuracy of 99.59%.

Keywords Deep learning · Convolutional neural network (CNN) · Devanagari sign language · Region of interest (ROI)

1 Introduction

India is the second-most populous nation in the world with over a population of billion people. More than a million people are now suffering from hearing problems and vision loss. According to a survey, it has been calculated that one percent of people are deaf and twelve percent have hearing problems. With the introduction of artificially intelligent computers combined with the availability of big data

V. Gaur (✉)
Maharaja Surajmal Institute of Technology, JanakPuri, Delhi 110058, India
e-mail: vimalgaur@msit.in

R. Dwivedi · Pankhuri · S. Dhar
Maharaja Surajmal Institute of Technology, C-4 Janakpuri, New Delhi 110058, India
e-mail: rinkydwivedi@msit.in

X.-Z. Gao et al. (eds.), *Advances in Computational Intelligence and Communication Technology*, Lecture Notes in Networks and Systems 399,
https://doi.org/10.1007/978-981-16-9756-2_8

and vast computing tools, there has been a massive rise in healthcare, robots, automated self-driving cars, and human–computer interface (HCI) [1]. HCI is used in augmented reality systems, face recognition systems, and hand gesture recognition systems. This paper aims to identify different alphabets of the Devanagari family which is surely a part of the human–computer interface. The main aim is to deal with the difficulties faced by deaf and dumb people in communicating with people in their day-to-day life. For effective communication, some signs are required. In this paper, these signs have been gathered together to construct a sign language that differs from one region to another [2, 3]. Nearly, 466 million individuals around the globe experience the problem of deafness (upwards of 5% of the global population). Of those 34 million include children, as per the World Health Organization (WHO) [4]. These estimates are expected to rise by more than 500 million by the year 2050 [9]. Moreover, in several cases, severe deafness disease is a global clumped in lower and middle-income nations. There has been a lot of work in this area in the past using sensors (like a glove sensor) and other image processing techniques (like edge detection, Hough transform,), but these have not yielded satisfactory results. However, with new deep learning techniques such as CNN, productivity in this area has skyrocketed, and opening up many new possibilities in the future [5]. People with hearing disabilities remain back at Internet conversations, workplaces, information sessions, and classes. They mostly use text messages to communicate with others, which is not the ideal tool. Only with the growing acceptance of telemedicine services, deaf individuals can interact spontaneously via their healthcare systems, friends and coworkers, irrespective of how much the other person is familiar with the sign language. Hand gesture recognition systems can be generally grouped into two categories: the first one is the hardware-based system and the second one is the technical vision system. In hardware-based systems, individual needs certain instruments to extract the characteristics that describe the signal of the hand gesture. The cyber glove is an extraction device that is used for extracting the characteristics like orientation, gestures, and color of the hands. It is commonly seen in the recognition of sign language. Vision-based devices use optical imaging systems that extract the lines and interpret the signal. The model demonstrated in this paper is a vision-based solution where no special device has been worn by the consumer. This paper has been structured as: Sect. 2 presents literature review. Problem statement has been presented in Sect. 3. Based upon this problem statement, objectives of research have been formulated and described in Sect. 4. Section 5 presents methodology. Section 6 presents results and discussions, and Sect. 7 concludes and presents future scope.

2 Literature Review

As per the World Health Organization (WHO), the estimates of having to hear people have finally hit 400 million. It is due to this fact that current findings have been intensified to make it possible for people with disabilities to interact. In recent years, several researches have been carried out. Few of them have been listed as below:

In [1], the authors have created smartphone SLR software for the identification of Indian Sign Language (ISL) gestures. In this research work, they have used the selfie mode continuous sign language capture method and CNN model for image recognition. In [2], the researchers have proposed a sign language recognition system that converts sign language in real time, thus allowing people who are not familiar with sign language to interact effectively with having to hear people. American Sign Language has been taken as input, and convolutional neural network has been trained on the dataset by Massey University, Institute of Information and Mathematical Sciences (2011). After the network has been trained, the network model and network weights have been stored for the real-time recognition of sign languages. Also, they have used certain frames for the determination of skin and for the hand gesture determination they used a convex hull algorithm. In [3], the researchers have focused on creating vision-based software that includes the text-based interception of hand gestures, thereby promoting contact between signatory and non-signatory. The dataset used by them is American Sign Language (ASL). The proposed model accepts the video frames as input and removes spatial and temporal information from them. In this research, they have used CNN to identify the spatial features and RNN to train on the temporal features. In [4], the authors have used the strategies for segmentation of images and identification of features. They have employed FAST and SURF algorithms to establish a relationship between image segmentation and object detection. The system created by them goes through multiple phases, such as capturing the data using the KINECT sensor, performing the segmentation of images, feature detection and extraction of ROI. The K-nearest neighbors (KNN) algorithm has been used to distinguish images. Text-to-Speech (TTS) conversion has been performed for the audio output. In [5], an Indian Sign Language translator has been created using a convolutional neural network algorithm by the researchers to identify the 26 letters of the Indian Sign Language into their corresponding alphabetic characters by taking a real-time representation of the gesture and translating that to its identical. In the implementation, a database has been generated using different image preprocessing methods to make the database available for the extraction of a feature. After the features were extracted, the images were sent to the CNN model using the Python program. In [6], a sign language finger-spelling alphabet detection system has been proposed using different image processing methods, supervised machine learning, and deep learning. The histogram of oriented gradients (HOG) and local binary pattern (LBP) characteristics within each symbol has been derived from the input image, and afterward, the multiclass support vector machines (SVMs) were used to train such collected features, and the end-to-end convolutional neural network (CNN) architecture was employed in the training set for the comparison. A Bangla Sign Language to text translator device has been made [7] by the researchers through customized region of interest (ROI) segmentation and convolutional neural network (CNN). They used customized ROI segmentation to allow the system person to change the pre-loaded boundary box on the display screen to the deaf person's hand area and thus only the area inside the video frame would be forwarded to the CNN prediction model. The model has also been incorporated with the ARM cortexA53 embedded Raspberry Pi, which provided durability and portability to the

system. In [8], the authors of the paper have designed an American Sign Language translator app Using OpenCV. They have used a ResNet-34 CNN Classifier for the classification of American Sign Language (ASL) hand gestures. "Firstly this application extracts the signs from the input video, this video then converted into frames, later ResNet-34 CNN classifier has been used for classifying frames and finally the text corresponding to the sign is displayed as an output on the screen". In [9], the researchers carried out a consistent study of the numerous methods used to translate sign language to text/speech form. Later, they have employed the best possible method to create an android app which can convert the ASL signs to text or speech in real-time.

After doing a thorough analysis on these studies, it has been analyzed that there is a requirement of real-time software for detecting hand gestures so that they can be converted into Devanagari script. Table 1 shows the methodologies that have been adopted by different researchers.

Table 1 Methodologies by different researchers

S. no	Author	Dataset	Accuracy (%)	Algorithm
1	Rao et al.	Real-time images	92.88	CNN
2	Taskiran et al.	Massey University, Institute of information and mathematical sciences	98.05	CNN
3	Bantupalli et al.	American sign language dataset	Created a vision-based application for sign language to text conversion	CNN(spatial features) RNN (temporal features)
4	Andewale et al.	Images are collected using Kinect sensor	78	Unsupervised machine learning algorithms
5	Intwala et al.	Real-time images	87.69	CNN
6	Kania et al.	Exeter University Dataset and Massey University web pages	93.3	Snapshot learning model
7	Khan et al.	Custom image dataset	94	Implemented in Raspberry Pi
8	Kurhekar et al.	Real-time images	78.5	A ResNet-34 CNN classifier
9	Tiku et al.	Real-time images	98.82	SVM

3 Problem Statement

In this paper, a real-time software has been developed for detecting hand gestures and converting them into a Devanagari script, using a deep learning approach (CNN). The results are required to be compiled using 25 alphabets of the Devanagari script.

4 Objectives of Research

- To generate a dataset that is skin tone independent using a webcam.
- To apply appropriate image processing and preprocessing techniques for achieving better accuracy and to obtain region of interest (ROI).
- To design the model and building CNN to train raw images and achieve optimal precision.

5 Methodology

5.1 Dataset Generation

Due to non-availability of datasets on Devanagari script, we instigated to create the datasets with 200 images from different angles. For the creation of the dataset, we used the webcam of personal computer. The webcam used for data gathering purposes is of 2.1 Megapixel and relative illumination is greater than 40%.

Steps to create dataset: OpenCV module has been used to produce dataset. Nearly 600 images of each alphabet of Devanagari script have been captured for training purposes and 200 images for testing purposes. When the frame of the camera opens, it displays everything in RGB values, but a specific region of interest (ROI) has been created inside that frame only, in the form of a small rectangle, which has videos displayed in the form of adaptive Gaussian filter. The small rectangle in our frame converts RGB into grayscale and finally applies an adaptive Gaussian filter in real-time [10, 11].

5.2 Gesture Classification

We have used one algorithm for predicting the final symbol displayed by the user.

- Applying an adaptive Gaussian filter and threshold to the frame, take using OpenCV, to extract features and then result in the processed image.
- Image after processing is passed through CNN model for prediction, and if a letter is recognized by the system, it is printed.

- In case more than one letter shows a similar result, we tried to increment the number of distinct positions

A. Layer 1: CNN model:

- First Convolutional Layer: The input picture is first passed through the convolutional layer. This layer uses 32 filter weights.
- First Pooling Layer: The pictures are down sampled using max pooling. The resultant output is a 2 * 2 array.
- Second convolutional layer: This array is then processed using 32 filter weights. The output of this layer is reshaped to an array of 30*30*32 = 28,800.
- Second Pooling Layer: This processed image is then down sampled again and reduced into images with less resolution.
- First Densely Connected Layer: These less-resolution images along with an input array of 28,800 are send to first densely connected layer having 128 neurons. The output of this will be used in the second densely connected layer. A dropout layer with a value of 0.5 has been added to avoid the overfitting of the model.

B. Layer 2: Activation Function:

- Rectifies layer unit has been used in every layer (convolutional as well as fully connected neurons). For each input pixel, max(x, 0) is calculated by ReLu. This adds non-linearity to the formula and trains the model on more complicated features. Due to this, vanishing gradient problems is also eradicated and the training process also fastens.

Layer 3: Pooling Layer: Max pooling layer has been applied to input image with a pool size of (2, 2) with ReLu activation function. This lessens the number of parameters and ultimately leading to reducing computation cost and overfitting of the model.

Layer 4: Dropout Layer: The problem of overfitting has been resolved by tuning the weights of the network. So, in dropout layer a random set of activations sets these weights to zero. Thereby network helps in providing accurate classification even if some activations are dropped out.

Layer 5: Optimizer: For updating the model in accordance with the output of loss functions, an Adam optimizer has been used. It adds benefits to the system by combining extensions of two stochastic gradient descent algorithms, namely adaptive algorithm (ADA GRAD) and root mean square propagation (RMSProp).

5.3 Training and Testing

Initially, the input images in the colored form have been converted into grayscale images and then passed under an adaptive Gaussian filter for removing unwanted noise. The complete dataset is divided into 70% training and 30% testing. The input images after preprocessing have been fed to the model for testing and training. The prediction layer estimates how likely the image will fall in which category. So the output is normalized between 0 and 1 such that the sum of values in every class sums to one. We have achieved this using the softmax function. The result of the prediction layer varies as compared with the actual result. That is why the model has been trained under labeled data. Classification uses cross-entropy as performance measurement, which is a continuous function with assigns positive values to those which are not the same as the labeled value and zero value to those which are the same as the label. We adjusted the weights of neural networks such that the cross-entropy is minimized to zero. An in-built function to calculate cross-entropy is already present in TensorFlow. After identifying the cross-entropy function, we have optimized it with Adam optimizer.

6 Results and Discussions

While training the dataset without augmentation, the CNN model achieved high training accuracy up to 98% and 99.59% testing accuracy, demonstrating real-time environment results. As discussed, region of interest has been demonstrated in Fig. 1.

Fig. 1 Demonstration of region of interest (ROI)

Figures 2 and 3 shows the snippets of the model. As shown, accuracy varies from 95.13 to 99.59% when the number of epochs increase from 1 to 5.

The difference in accuracy is because hand gestures in real-time are not in the same position. Also, the pixel position and shapes of the images are different. Additionally, the background is scrambled. The comparison of accuracy of the CNN model with the current state-of-the-art have been shown in Table 2.

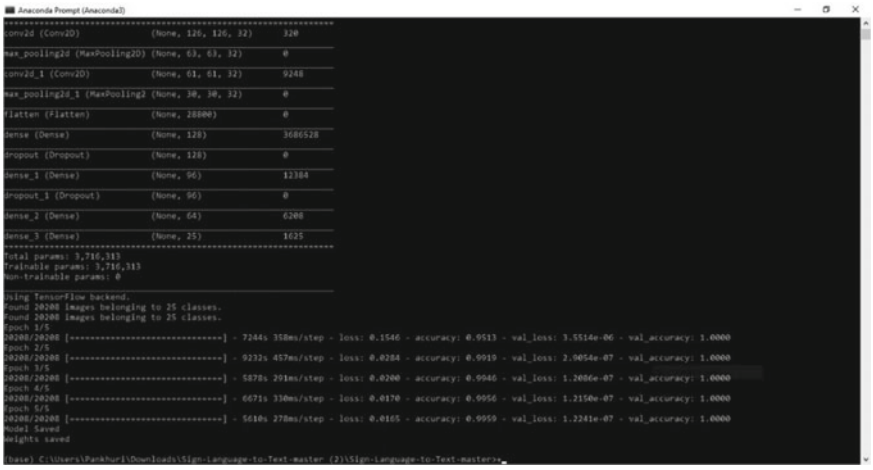

Fig. 2 Snippet 1 of the model

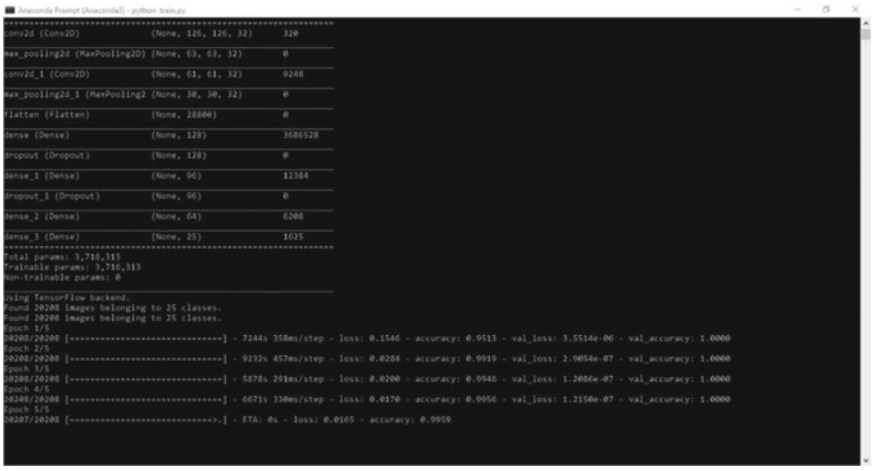

Fig. 3 Snippet 2 of the model

Table 2 The accuracy comparison of the CNN model with the current state-of-the-art methods

Method	Accuracy (%)
CNN with Gaussian filter	99.59
CNN [1]	92.88
CNN [2]	98.05

7 Conclusion and Future Scope

The study shows that Devanagari script used in India, Nepal, and Tibet includes 25 different letters and is written from left to right. Hence, 600 different images were collected as training set and 200 for testing test. From this collection of images, a database of twenty-five different signs were used. These set of images are subjected to batch segmentation, detection of each alphabet is done, and region of interest is extracted from specific bounding box. The combination of adaptive Gaussian filter and CNN showed that the gesture classification could determine the best matched feature. This is achieved by comparing it with existing database. The accuracy percentage while training the model comes out to be 98%, and testing accuracy has been calculated as 99.59%. We have achieved higher accuracy even in cluttered backgrounds and build a model which is comparatively less dependent on light. In future work, we will try to improve accuracy by removing background and performing heavy preprocessing on the images.

References

1. Rao GA, Syamala K, Kishore PVV, Sastry ASCS (2018) Deep convolutional neural networks for sign language recognition. In: 2018 conference on signal processing and communication engineering systems (SPACES), Vijayawada, pp 194–197. https://doi.org/10.1109/SPACES.2018.8316344
2. Taskiran M, Killioglu M, Kahraman N (2018) A real-time system for recognition of american sign language by using deep learning. In: 2018 41st international conference on telecommunications and signal processing (TSP), pp 1–5. https://doi.org/10.1109/TSP.2018.8441304
3. Bantupalli K, Xie Y (2018) American sign language recognition using deep learning and computer vision. In: 2018 IEEE international conference on big data (Big Data), Seattle, WA, USA, pp 4896–4899. https://doi.org/10.1109/BigData.2018.8622141
4. Victoria A, Adejoke O (2018) Conversion of sign language to text and speech using machine learning techniques. 5:58–65
5. Intwala N, Banerjee A, Meenakshi, Gala N (2019) Indian sign language converter using convolutional neural networks. In: 2019 IEEE 5th international conference for convergence in technology (I2CT), Bombay, India, pp 1–5. https://doi.org/10.1109/I2CT45611.2019.9033667
6. Nguyen HBD, Do HN (2019) Deep learning for american sign language finger-spelling recognition system. In: 2019 26th international conference on telecommunications (ICT), Hanoi, Vietnam, pp 314–318. https://doi.org/10.1109/ICT.2019.8798856
7. Khan SA, Joy AD, Asaduzzaman SM, Hossain M (2019) An efficient sign language translator device using convolutional neural network and customized ROI segmentation. In: 2019 2nd

international conference on communication engineering and technology (ICCET), Nagoya, Japan, pp 152–156. https://doi.org/10.1109/ICCET.2019.8726895
8. Kurhekar P, Phadtare J, Sinha S, Shirsat KP (2019) Real time sign language estimation system. In: 2019 3rd international conference on trends in electronics and informatics (ICOEI), Tirunelveli, India, pp 654–658. https://doi.org/10.1109/ICOEI.2019.8862701
9. Tiku K, Maloo J, Ramesh A, Indra R (2020) Real-time conversion of sign language to text and speech. In: 2020 second international conference on inventive research in computing applications (ICIRCA), Coimbatore, India, pp 346–351. https://doi.org/10.1109/ICIRCA48905.2020.9182877
10. Fernandes L, Dalvi P, Junnarkar A, Bansode M (2020) Convolutional neural network based bidirectional sign language translation system. In: 2020 third international conference on smart systems and inventive technology (ICSSIT), Tirunelveli, India, pp 769–775. https://doi.org/10.1109/ICSSIT48917.2020.9214272
11. Beasley WC (1940) Partial deafness and hearing-aid design: I. characteristics of hearing loss in various types of deafness. J Soc Motion Picture Eng 35(7):59–85. https://doi.org/10.5594/J10069.s