Advances and Applications in Mathematical Sciences
Volume 22, Issue 11, September 2023, Pages 2199-2211

REAL-TIME SIGN LANGUAGE RECOGNITION USING CNNS

JATIN AGGARWAL[1], LAKSHAY GUPTA[1], MOHIT GUPTA[1], NITIN[1] and PALAK GIRDHAR[2]

[1]Student
Department of Computer Science
and Engineering
Bhagwan Parshuram Institute of Technology
PS-4, Sec.-17, Rohini, Delhi-89, India
E-mail: jatin22200@gmail.com
lakshaygupta.bpit@gmail.com
mohitgupta0178@gmail.com
nitkash888@gmail.com

[2]Assistant Professor
Department of Computer Science
and Engineering
Bhagwan Parshuram Institute of Technology
PS-4, Sec.-17, Rohini, Delhi-89, India
E-mail: palakgirdhar@bpitindia.com

Abstract

People with hearing or listening impairments communicate among themselves using a system of signals known as sign language rather than speaking. Building a programme that can identify sign language movements or activities is vital and required in order to facilitate communication between the hearing and dumb community. An essential step toward improving communication between the disabled (deaf/dumb) community and the general public is the development of a real-time sign language recognition system. As a result, we present the creation and application of a convolutional neural network-based Sign Language Recognizer. In order to recognise sign language, a model called ASLR-Net (American Sign Language Recognition) is put out in this paper. Region of Interest (ROI) brings in novelty and lowers the need for storage memory.

2020 Mathematics Subject Classification: 68T07.
Keywords: Deep learning, CNN, Assistive Technology.
Received June 1, 2023; Accepted June 22, 2023

1. Introduction

Sign Language Recognition (SLR) is the evolving area of research in the domain of computer vision. Sign language is the best way of communicating with the people with low or no hearing ability. It is a natural language used by people for communicating letters, words and sentences by using different signs or movements of hands. Sign language helps people with disability to express their views and helps to bridge the gap between hearing impaired people and with other persons. Sign language gives an easy medium for communication and way of expressing their feelings in front of normal person. Most of the research has been done on American Sign Language (ASL) recognition, whereas Indian Sign language (ISL) is still a challenging task due to the involvement of both the hands. With the development of artificial intelligent algorithms and availability of big data and presence of large computational resources has led to a major growth in various fields of robotics, healthcare, automated self-driving cars and Human-Computer Interaction (HCI). Sign language recognition or hand-gesture recognition is the challenging area of research in the application of HCI.

People with disability of speech and hearing-impaired use hand gesture to communicate with other people. However, apart from few, less number of people are able to understand this sign language. Most of the people may require an interpreter for better communication, which may be a costly affair. The purpose of this work is to narrow down the communication gap by developing a software for automatic recognition of real-time sign language. In literature, various machine learning approaches have been used for the recognition of sign language. With the advent of deep learning techniques, there is a significant improvement in the recognition rate and in feature extraction phase. Traditional approaches depend on manual feature extraction, whereas with the advancement of technology and evolution of deep learning, it is able to understand the complex pattern with improved recognition rate.

To bridge this communication gap, we proposed a basic Convolutional Neural Network (CNN) called ASLR-Net to develop the real-time sign language recognizer. We have developed a CNN model with 3 Convolution, 3 Maxpooling, 4 Dropout, a Flatten, and 2 fully connected layers. The model is

trained with the MNIST American Sign language dataset having 27,455 training and 7,172 testing samples with 785 features and 24 classes.

The paper is organized as follows. Section 2 describes the related work. The proposed system design and architecture is demonstrated in Section 3. Section 4 describes the experimental results and analysis. Finally, the research has been concluded in Section 5.

2. Related Work

Researchers are paying close attention to sign language recognition because of the many applications that can be used in many areas such as deaf communication systems, human-machine interaction, machine control, etc. Research on SL recognition can be split into two categories. categories on the basis of the type of symbols: (i) recognition based on stationary signs and (ii) recognition based on strong symptoms. Most studies are performed on the detection of static symptoms. SL recognition research began in the late 1990s.

American Sign Language Character Recognition with Capsule Networks is research that uses two methods i.e. LeNet and CapsNet for solving this problem. LeNet is accepted as the foundation of deep learning and built for optical character recognition [4]. CapsNet which claims to revolutionize the problem of object recognition just like deep learning and was tested with an optical character recognition problem like LeNet. CNNs work by detecting features in images. The features like edge and color changes are detected. Then, by combining them more complex features are learned and classification made. Along with these operations, while transferring the features that come from previous layers to the next layers, they are calculated as weighted sums. No information about poses and orientations is kept in these processes. Also, subsampling layers in CNNs cause some features to get lost. This model after augmenting the data provides an accuracy of about 95% but fails to make an accurate classification in a few cases.

Sign Language Recognition Using Convolutional Neural Networks [14] the author uses data set from the ChaLearn Looking at People 2014 That dataset consists of 20 different Italian gestures, performed by 27 users with variations in surroundings, clothing, lighting, and gesture movement. The

videos are recorded with a Microsoft Kinect. In the preprocessing stage, the author crops the highest hand and the upper body using the given joint information. Furthermore, the noise in the depth maps is reduced with thresholding, background removal using the user index [14], and median filtering CNNs [2] are inspired by the visual cortex of the human brain. The artificial neurons in a CNN will connect to a local region of the visual field, called a receptive field. This is accomplished by performing discrete convolutions on the image with filter values as trainable weights. The architecture of the model consists of two CNNs, one for extracting hand features and one for extracting upper body features. Each CNN is three layers deep. A classical ANN with one hidden layer provides classification after concatenating the outcomes of both CNNs. The authors observed a validation accuracy of 91.70%. The accuracy on the test set is 95.68% and they observed a 4.13% false-positive rate, caused by the noise movements.

Sign Language Recognition Using Deep Learning and Computer Vision uses a custom CNN model to recognize gestures in sign language. A convolutional neural network of 11 layers is constructed, four Convolution layers, three Max-Pooling Layers, two dense layers, one flattening layer, and one dropout layer. The American Sign Language Dataset from MNIST is used to train the model to identify the gesture [16]. The dataset contains the features of different augmented gestures. Introduced a custom CNN (Convolutional Neural Network) model to identify the sign from a video frame using Open-CV. Initially, a feature extracted dataset is used to train the custom model that has 11 layers with a default image size. The major issue faced is due to the background of the image. As the model is trained with a segmented grayscale gesture image, it does not support background subtraction from the image when the frames are dropped from a video. The current model yields better accuracy with a segmented hand gesture which is done by the Open-CV with a Region of Interest (ROI) box implemented in the driver program. The model lacks accuracy with noisy images when it is dropped from the video frame. The model performance was not as expected if a person wears ornaments like rings as the dataset used to train the model was clean without the inclusion of any ornaments.

3. Methodology

3.1. Dataset

The MNIST American Sign Language Dataset is employed in this study to evaluate the performance of the suggested model. Every alphabet from A to Z, with the exception of J and Z, has a one-to-one mapping in the training and test scenario. Each alphabet is assigned a label with a value between 0 and 25.

27,455 cases made up the training data, while 7172 cases in the test data had labels in the header row. The grayscale values for the single 28x28 pixel image at positions pixel1, pixel2, ..., pixel784 range from 0-255. A few data samples from the dataset are shown in Figure 1.

Figure 1. Samples of the MNIST Sign Language dataset.

3.2. Pre-Processing

In this work, the data we use from the MNIST data set is in a .csv format (which represents a single 28x28 pixel image with grayscale values between 0-255) which we then convert into 2-D images.

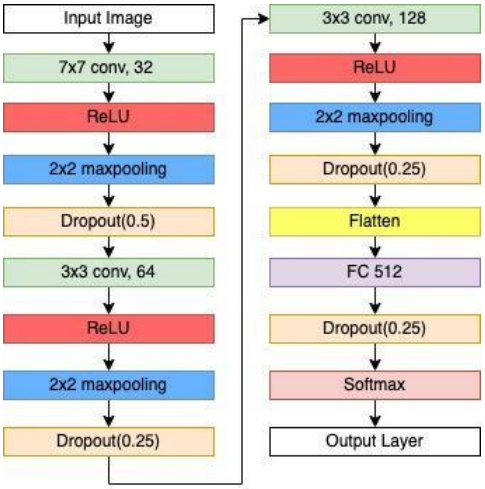

Figure 2. Layers in the proposed ASLR-Net. Conv refers to the convolutional layer.

3.3. Proposed Architecture

The proposed methodology for the sign recognition is shown in the figure 3.

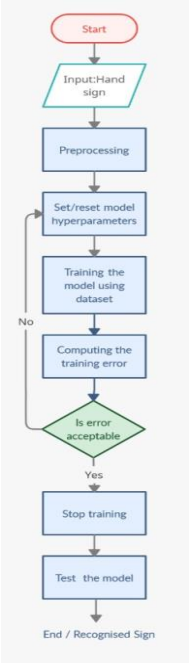

Figure 3. Proposed Sign Language Recognition System.

(a) Convolutional Neural Network (CNN)

Convolutional neural networks are a subset of deep neural networks that are frequently employed for feature extraction from input images. The foundation of CNN is the Convolution Layer. This layer's objective is to extract features from the input. With a given input and filter size, the convolutional layer executes the mathematical convolution operation. Different feature maps are computed using different size filters/kernels. The application of an elementwise non-linear activation function to the convolved result and the convolving operation on the input with kernels are both necessary for the production of feature maps. By carefully choosing the number of layers and neurons, the CNN design aids in boosting performance. Such guidelines that standardise the acceptance of the number of neurons and layers do not exist. The proposed model maximizes the recognition accuracy.

(i) Input Layer

The input layer is the first layer of the network. Pre-processed images are passes to this layer for execution. Here, input layer has 784 features where features resemble image pixels at resolution 28×28.

(ii) Convolution Layer

The foundation of a CNN is this layer, which also houses the majority of the computation. Filter, input data, and a feature map are the main elements employed at this tier.

Similar to a conventional neural network, convolution is a linear procedure that includes multiplying a set of weights with the input.

(iii) Activation Function

An Activation Function checks whether a neuron needs to be activated or not. This function decides whether the neuron's input which it provides to the network is needed to be taken into account for predicting using mathematical operations. Here, Rectified Linear Unit (ReLU) activation function have been used. It handles the problem of vanishing gradient, which is commonly observed with other activation functions. ReLU generates the output if the

given input is positive, otherwise it will give zero as output. The ReLU function is written as follows:

$$\mathrm{Re}\,LU(y) = \max(0,\ y) \tag{1}$$

where y is the given input.

(iv) Pooling Layer

A two-dimensional filter is slid over each channel of the feature map during the pooling operation, summarising the features held within the filter's region. It decreases the size of the feature maps, which also aids in cutting down on a CNN's training time and memory requirements. The pooling operations max-pooling, min-pooling, and average pooling can all be carried out. Here, we employ the max-pooling procedure in our task. The feature map's maximum value from the chosen area is taken into account while it is being downscaled (pooled). The following equation can be used to get the resultant dimension of the max-pooling operation:

$$N_{out} = floor\left(\frac{N_{in} - F}{S}\right) + 1 \tag{2}$$

where the terms N_{in}, F, and S, respectively, stand for the input image, kernel, and stride sizes.

(v) Fully Connected Layer

The features map generated by the previous layer is fed into the fully connected layer (FC) layer. In an FC layer, the neurons in one layer are connected to the neurons in another layer. The FC layer also behaves like a convolution layer with a filter of size 1×1.

(vi) Dropout Layer

A regularization technique called dropout randomly sets input elements to zero with a given probability. When a model's testing accuracy is too low compared to its training accuracy, over-fitting problems arise. By randomly setting activation to zero during the training process, a dropout layer followed by the FC layer in CNN models enables the prevention of the over-fitting issue and improves performance. The study's assumed dropout probability was 0.5.

(vii) Output Layer

The output layer is the last layer of neurons in an artificial neural network that produces given outputs for the program. Though they are made much like other artificial neurons in the neural network, output layer neurons may be built or observed in a different way, given that they are the last "actor" nodes on the network. Softmax activation function is used for multiclass classification. It distributes the output probabilities in the given classes. The class with the highest probability is chosen as the final classification. The mathematical expression for the Softmax function is given by

$$\sigma(X)_i = \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x_k}} \tag{3}$$

$$\text{for } i = 1, 2, 3, ..., K.$$

where X_i is the inputs from the previous FC layer used to each Softmax layer node and K is the number of classes.

4. Experimental Results

4.1. Performance Measures

(a) Accuracy

A metric for rating classification models is accuracy. It provides the percentage of correctly predicted events. It can be calculated as follows for binary classification:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{4}$$

where,

True Positives (TP)

False Positives (FP)

True Negatives, or TN

False Negatives (FN)

(b) Precision

A measure of precision counts how many correctly positive forecasts were made. It is determined by dividing the total number of accurately anticipated positive samples by the number of positive samples overall. It is spelled as follows:

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

(c) Recall

The capacity of a model to locate each and every pertinent case in a data collection. Recall is calculated mathematically by dividing the total number of true positives by the total number of true positives plus false negatives. You can write it as follows:

$$Recall = \frac{TP}{TP + FP} \tag{6}$$

(d) F-1 Score

The F1 score is defined as the harmonic mean of precision and recall. F-Measure provides a way to combine both precision and recall into a single measure that captures both properties. The F1 score formula is shown here:

$$F1Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{7}$$

4.2. Result Analysis

The well-known MNIST ASL dataset was used to train the suggested customized CNN model. There are 27455 training samples in the dataset. The suggested model is trained using its 784 characteristics. The goal of the model's training is to minimize loss. To calculate the loss, a sparse categorical cross-entropy function is used. The training samples are divided into training and validation sets at various points during the model's development. The model is trained using a 512-batch size across 50 epochs. The test set has 7172 samples, and the model's test accuracy is said to be between 96 and 97 percent. Table 1 displays the suggested method's performance metrics, such as accuracy score, precision, recall, and F1 score. Further, to establish the results, it is compared with state-of-the art methods shown in table 2. For

better visualization, training accuracy and loss plots are shown in Figure 5 and Figure 6.

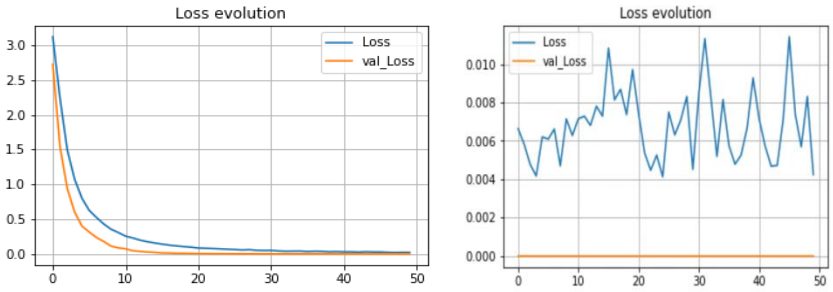

Figure 5. Loss Evolution Graph (first iteration and last iteration).

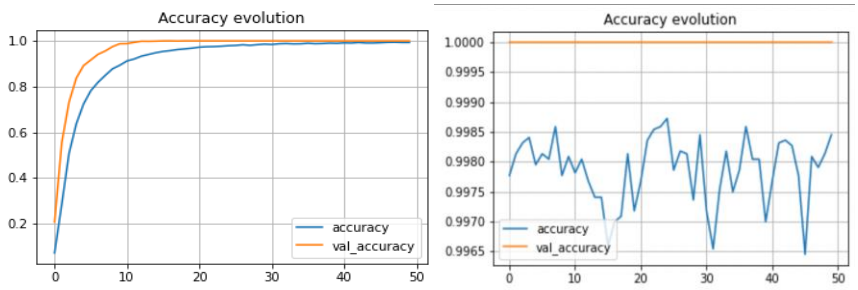

Figure 6. Accuracy Evolution Graph (first iteration and last iteration).

Table 1. Comparison with state-of-the-art methods.

Models Used	Accuracy Score	Precision	Recall	F1 score
LeNet	82.19	81.24	81.82	80.95
CapsNet	88.93	84.48	89.04	86.41
CapsNet augmented	95.08	91.11	95.63	93.22
Proposed Method	**0.969**	**0.962**	**0.965**	**0.962**

5. Conclusion

Proposed model, reached 96.9% accuracy on the test set, while the accuracy of capsnet augmented reached only 95.08% and of custom CNN (11 layer Model) [16] reached only 93%. Hence proposed model outperforms the

state-of-the-art methods in testing accuracy over the same dataset, also with lesser training time.

Dataset	Model	Epochs	Batch size	Optimizer	No. of Layers	Accuracy	No. of output classes
MNIST ASL Dataset	CapsNet augmented	30	128	Adam	7	95.08%	24
	Custom CNN (11 Layer) Model	10	128	Adam	11	93%	24
	Proposed Model	50	512	Adam	13	96.9%	24

6. Future Scope

The proposed model could be trained to recognize different sign languages such as Indian Sign Language. For now, it is limited to ASL i.e. the American Sign Language. Classes could be increased, and words or phrases could be introduced. The model could be improved and advanced to recognize gestures that require motions and both hands.

References

[1] S. Sinhal, S. Gupta, P. Purohit and R. Shah, Advanced gesture recognition system using deep learning, IRJMETS 3(4) (2021), 223-228.

[2] M. M. Rahman, M. S. Islam, M. H. Rahman, R. Sassi, M. W. Rivolta and M. Aktaruzzaman, (2019, December), A new benchmark on American sign language recognition using convolutional neural network. In 2019, International Conference on Sustainable Technologies for Industry 4.0 (STI) (pp. 1-6). IEEE.

[3] S. S. Rautaray and A. Agrawal, (2010, December), A novel human-computer interface based on hand gesture recognition using computer vision techniques. In Proceedings of the First International Conference on Intelligent Interactive Technologies and Multimedia.

[4] M. Bilgin and K. Mutludoğan, (2019, October), American sign language character recognition with capsule networks. In 2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT) (pp. 1-6). IEEE.

[5] F. Khan, B. Halim and A. Rahman, Computer vision-based mouse control using object detection and marker motion tracking, International Journal of Computer Science and Mobile Computing 9(5) (2020), 35-45.

[6] M. Salvi, S. Kegade, A. Shinde and B. Tekwani, Cursor manipulation with hand recognition using computer vision, Information Technology in Industry 9(1) (2021), 1455-1456.

[7] N. Rajendra and B. Rao, Cursor movement by object detection based on image processing, Int. J. Res. Publ. Rev. 2(5) (2021), 18-22.

[8] G. A. Rao, K. Syamala, P. V. V. Kishore and A. S. C. S. Sastry, (2018, January), Deep convolutional neural networks for sign language recognition. In 2018 Conference on Signal Processing and Communication Engineering Systems (SPACES) (pp. 194-197). IEEE

[9] K. S. Varun, I. Puneeth and T. P. Jacob, (2019, April), Hand gesture recognition and implementation for disables using CNN'S, In 2019, International Conference on Communication and Signal Processing (ICCSP) (pp. 0592-0595). IEEE.

[10] O. Yadav, S. Makhwana and P. Yadav, Cursor movement by hand gesture, International Journal of Engineering Sciences and Research Technology 3 (2017), 243-247.

[11] S. Masood, A. Srivastava, H. C. Thuwal and M. Ahmad, Real-time sign language gesture (word) recognition from video sequences using CNN and RNN, In Intelligent Engineering Informatics (pp. 623-632). Springer, Singapore (2018).

[12] R. Rastgoo, K. Kiani and S. Escalera, Real-time isolated hand sign language recognition using deep networks and SVD, Journal of Ambient Intelligence and Humanized Computing 13(1) (2022), 591-611.

[13] R. Elakkiya, Machine learning based sign language recognition: A review and its research frontier, Journal of Ambient Intelligence and Humanized Computing 12(7) (2021), 7205-7224.

[14] L. Pigou, S. Dieleman, P. J. Kindermans and B. Schrauwen, (2014, September), Sign language recognition using convolutional neural networks, In European Conference on Computer Vision (pp. 572-578). Springer, Cham.

[15] N. Aloysius and M. Geetha, Understanding vision-based continuous sign language recognition, Multimedia Tools and Applications 79(31) (2020), 22177-22209.

[16] R. S. Sabeenian, S. S. Bharathwaj and M. M. Aadhil, Sign language recognition using deep learning and computer vision, J. Adv. Res. Dyn. Contr. Syst. 12 (2020), 964-968.

[17] S. Sharma and K. Kumar, ASL-3DCNN: American sign language recognition technique using 3-D convolutional neural networks, Multimedia Tools and Applications 80(17) (2021), 26319-26331.

[18] Y. C. Bilge, R. G. Cinbis and N. Ikizler-Cinbis, Towards zero-shot sign language recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence (2022).