2021 International Conference on Industrial Electronics Research and Applications (ICIERA) | 978-1-6654-3542-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICIERA53202.2021.9726736

Recognition of American Sign Language using Deep Learning

Aeshita Mathur*
Dept. of CSE
Delhi Technological University
New Delhi, India
aeshitamathur@gmail.com

Deepanshu Singh
Dept. of Applied Physics
Delhi Technological University
New Delhi, India
deepanshusinghhudda21@gmail.com

Rita Chhikara
Dept. of CSE
The NorthCap University
Gurugram, India
ritachhikara@ncuindia.edu

***Abstract*—Deaf and mute communities have always faced a communication barrier, but advances in the field of Deep Learning are reducing this barrier. As a form of communication, sign language is one of the most ancient and natural, but since few people speak it and interpreters are extremely rare, this paper proposes to use neural networks to handle fingerspelling based on American Sign Language. A comparative study for Sign Language Recognition (SLR) is presented by implementing a variety of Deep Learning models. The paper proposes a CNN architecture that outperforms (by around 4%) various pre-trained models for SLR.**

***Index Terms*—Sign Language, Deep Learning, Convolutional neural Network, Resnet50, VGG16, Inception v3**

I. Introduction

Communication between individuals and groups relies heavily on spoken language. It is very important for humans to communicate in order to understand each other's needs and, in turn, build relationships. A large part of the population relies upon spoken language for expressing thoughts and emotions, and without it, misunderstandings may arise. Even though spoken language has existed for a long time, a section of the population still has difficulty communicating. Those with speech and hearing impairments cannot communicate using this medium and therefore need other means to communicate. The need for sign language is evident here. Sign language involves visual means of communication such as hand gestures, facial expressions, and body language. The same as a spoken language, sign languages are composed of following elements: Manual features which consists of shape of hand shape, the orientation, movement and position of the fingers or palm, and Non-manual features which are gaze of the eye gaze, shakes, different types of facial expressions like mouth related gestures, orientation of shoulders. Combining manual and non-manual features provides a gloss, which is a fundamental building block of sign language and represents the closest meaning of a sign [1]. Akin to spoken languages, there are several types of sign languages which have come into being based on geography and dialect. There is no universal sign language. Various sign languages exist for example Greek, American, Indian etc. In this paper sign language recognition has been performed using the latest technology of deep learning. The glosses are inferred in the process of sign language recognition/prediction. The glosses are captured from the video of a signer. The contribution of this paper is in bridging the gap between the deaf communities and hearing majority. Various pre-trained CNN Architectures and proposed deep learning methodologies have been applied and compared to recognize sign language and develop an efficient learning model with increased recognition rate. For experimental purposes, American Sign Language (ASL) has been chosen. ASL has shown exceptional results as it is considered central with respect to Deaf culture [2][3]. The benchmark datasets not being available for American Sign Language (ASL) led to creation of dataset through Open Computer Vision (OpenCV) Library. The dataset has 26 English alphabets, 600 images each for training and 150 images each for testing.

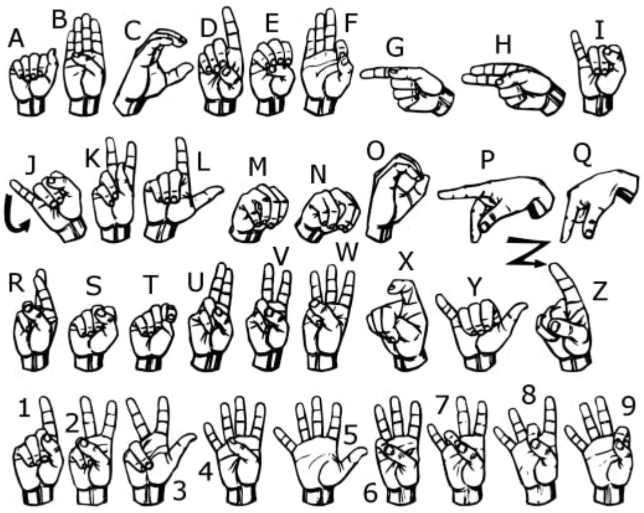

Fig. 1. Signs in American Sign Language

II. Literature Review

In literature, authors have worked on different sign languages like American, Greek, Indian etc. This section presents their work and also highlights the gaps.

Siming He [4] developed a network to locate hands from a common vocabulary dataset. To recognize a hand in the photo, the author employed Faster R-CNN. Author also proposed, a fusion model which was based on long and short time mem-

978-1-6654-3542-0/21/$31.00 © 2021 IEEE

ory (LSTM) methodology and a 3D CNN feature extraction network to achieve the best recognition accuracy. Nicholas et. al.[5] have done a comprehensive study about Sign Language Recognition Methods using a Greek sign language dataset. The analysis of their study was that the architectures based on 3D CNN-performed better in isolated SLR, while architectures based on 2D CNN models performed better on the majority of CSLR datasets. As part of their research, Aditya Das et.al[6] applied various pre-trained deep neural network models to the American Sign Language Dataset. The study concluded that Inception v3 is an appropriate methodology for recognizing sign language gestures which are static. An interesting research was done by G.Anantha Rao et.al.[7] based on Deep Neural Networks for Selfie mode SLR. CNN architecture gave best results when stochastic pooling was implemented. Lionel Pigou[8] conducted a similar study on Italian gestures. Two CNNs were used in the architecture of the model, one for selecting hands and one for selecting upper bodies. The local contrast normalization (LCN) was also used in the first two layers, and the activation function used was relu(rectified linear unit), which performed better than tanh.

Radha S. Shirbhate et.al.[9] applied Machine Learning Algorithms for Sign Language Recognition to an Indian Sign Language dataset. The work was done on dataset from UCI repository which had around 2,00,000 points obtained through skin segmentation. First, the linear kernel SVM model was trained to determine whether an alphabet was one-handed or two-handed. Following that, Multiclass SVM models with linear kernel were trained to classify one handed alphabets and two handed alphabets and then the system was put together. The former gave the best results. For continuous gesture recognition, Kumud Tripathi et. al[10] employed a gradient-based keyframe extraction technique. To extract features from preprocessed gestures, the orientation histogram (OH) was used, while Principal Component Analysis (PCA) was used for reducing dimensions obtained from OH. In this study, a continuous dataset of Indian Sign Language (ISL) was used as a data source. A variety of classifiers were tested for probes, including Euclidean distance, correlation, city block distance, Manhattan distance, etc. Results obtained from correlation and Euclidean distance was more accurate than other classifiers. The system proposed by Nada B. Ibrahim et.al. [11] Includes four stages: hand segmentation, tracking, feature extraction, and classification. A dynamic skin detector was used to segment the hands which depended on the color of the face. The head was used to track and identify hands through the blobs of segmented skin. The feature vector was constructed using geometric features. Finally in the classification step the Euclidean distance classifier was applied. The authors have worked with only 30 isolated words which children with hearing impairment use in their daily lives. For further details on various techniques applied for hand gesture recognition and SLR research work by Ming Jin Cheok et.al.[12] can be referred.

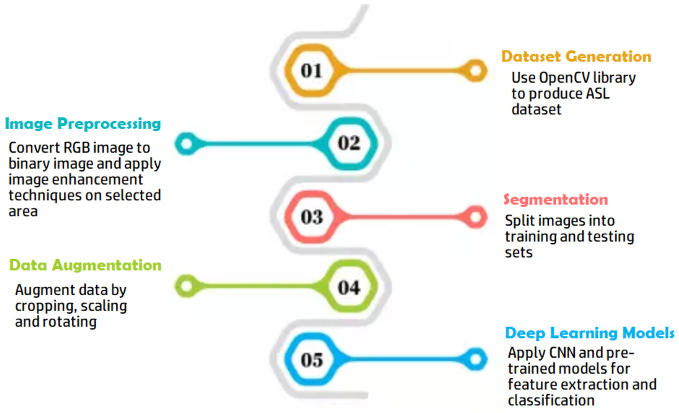
01
Dataset Generation
Use OpenCV library to produce ASL dataset
Image Preprocessing
Convert RGB image to binary image and apply image enhancement techniques on selected area
02
03
Segmentation
Split images into training and testing sets
Data Augmentation
Augment data by cropping, scaling and rotating
04
05
Deep Learning Models
Apply CNN and pre-trained models for feature extraction and classification

Fig. 2. General Framework of Proposed methodology

III. Proposed Methodology

A. Dataset Generation

Various sign language datasets are available online for SLR. Our search for ASL datasets found a number of limitations, such as poor lighting conditions, very few images in the dataset, and not a lot of symbols to train on. Unavailability of the right dataset prompted us to create our own ASL dataset. In order to produce this dataset, Open Computer Vision (OpenCV) an open source software library is used. This library is popularly used for applications related to machine learning and computer vision. There are 26 English alphabets which can be represented with one hand in American Sign Language. The dataset consists of 500 images per symbol for training purposes and 150 images per symbol for testing purposes. In each frame captured by the webcam, a region of interest (ROI) is defined. It created a sub region and only that portion of hand signing the symbol was captured and processed further.

B. Image Processing

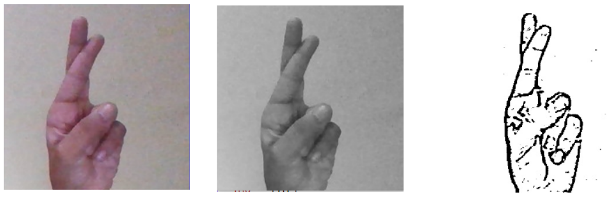

Fig. 3. Steps of Image Processing

Captured RGB images were converted into binary images to reduce the complexity of the model. For a clear image, removing noise was a daunting task. This obstacle was overcome by using Gaussian Blur filter and Adaptive Thresholding, both of which are considered to give excellent results when it comes to image de-noising. [13]. An image with Gaussian Blur systematically reduces image noise, minimizes the number of negligible details, and preserves low spatial frequencies. Using thresholding, an image can be segmented by assigning a foreground value to all pixels above its threshold and a background value to everyone else. For dramatic ranges of pixel intensity, adaptive thresholding was chosen, in which the threshold value is calculated for smaller regions, resulting in different threshold values for different regions.

C. Segmentation

After the images are finalized, they are divided into training dataset and testing dataset set. A ratio of 70:30 was maintained for the splitting. Out of a dataset of 17400 images, 11850 images (70% approximately) were used to train the model, while 5550 images (30%) were used to test it. The purpose behind taking the above mentioned ratio is to provide a sufficient number of images during the training process.

D. Augmented Data

Data is augmented to upsurge the diversity of our training set by applying random (but realistic) transformations. This includes various operations like; rescaling, cropping, flipping, rotating etc. This ensures that the classifier does not memorize a particular type of images which could lead to overfitting. In the data augmentation process, image flipping was excluded because it would have created a different symbol with a different meaning.

E. Augmented Data

This section explains proposed CNN architecture along with three Pre-trained CNN architectures (VGG16, Resnet50, Inception v3) applied for developing a learning model to recognize American Sign Language.

1) Proposed CNN model: The proposed CNN structure consists of two CNN blocks. Two Convolution Blocks are used sequentially, and one CNN block is depicted in Figure 4.

The model architecture comprises of following layers:

- Convolution Layer 1: The input image of 128x128 pixels is processed using 32 filter weights (3x3 pixels each). Each of the filter-weights will result in an image of 126X126 pixels.
- Pooling Layer 1: Max pooling technique with a filter size of 2x2 is applied to downsample the image. This reduces the size of image to 63x63 pixels.
- Convolution Layer 2: The output of first pooling layer which is images with size of 63x63 is passed as input to Convolution Layer 2. The second convolutional layer uses filter of 3x3 and 32 filter weights. This produces output images of size 60 x 60 pixel.
- Pooling Layer 2: Max pooling technique with a filter size of 2x2 is applied to downsample the image. This reduces the size of image to 30x30 pixels.
- Densely Connected Layer 1: The images received from pooling layer are then passed to 128 neurons. The output is then reshaped into a single dimensional array of size 30x30x32 = 28800 values. Dropout layer was used to overcome the problem of overfitting, the value taken was 0.5.
- Densely Connected Layer 2: In this step the output received from the previous step is forwarded to a fully connected layer. The number of neurons in this layer is set to 96.
- Final layer: The final layer takes as input the output obtained from densely connected layer 2. The output of this layer consists of same number of neurons as the number of classes needed to be detected in the American Sign language which is equivalent to 26.

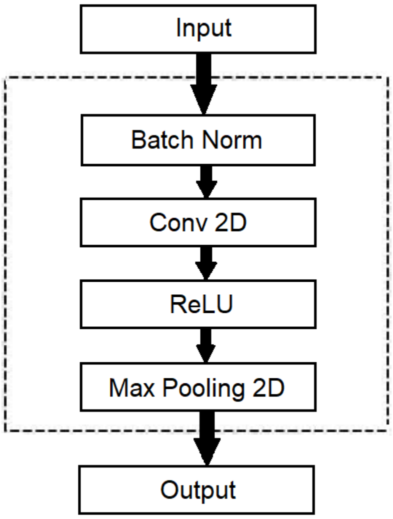

Fig. 4. CNN block architecture

Activation Function: In each layer (convolutional and fully connected neurons), ReLu (Rectified Linear Unit) was implemented as the activation function. This activation function evaluates max(x,0) for each pixel. ReLu activation function is able to handle nonlinear data also efficiently and supports in learning complicated features. By reducing the computation time, it removes the vanishing gradient problem and speeds up the training. In the final layer, softmax activation function was applied to classify multiple classes.

Dropout Layers: This layer suppresses a set of activation values by setting them to zero, in order to prevent overfitting. Even when some activations are missing, the network should be able to determine the right output or classification for a specific example. Adam also incorporates the second moments of the gradients (the uncentered variance) when adjusting the parameter learning rates instead of just using the mean (the first moment).

Optimizer: An Adam optimizer is used to update the weights of the neural network. It improves performance of the network as it is designed to bring together advantages of Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation Algorithm (RMSProp).

2) VGG16: VGG16 is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford [14]. It has 16 layers. The convolution layers in VGG16 have filter of size 3x3 with stride 1. It makes use of padding and follows max pooling with filter of size 2x2 and stride 2. Final output is passed through a softmax, which is a product of two FC (fully connected layers). The network is pretty big, with 138 million parameters (approx).

3) Resnet50: ResNet-50 is a model composed of 5 stages with 50 layers and 23 million trainable parameters[15]. Each stage consists of two blocks namely; convolutional and identity. Both these blocks have three convolution layers.

4) Inception v3: The Inception v3 [16] model is very popular for applications of image recognition. Inception v3 features the following characteristics. The Optimizer applied is RMSProp. It uses factorized convolutions of 7x7. BatchNorm is applied followed by Label smoothing to prevent overfitting.

IV. Experimental Results and Analysis

The proposed model of CNN and Pre-trained CNN architectures are applied to self-created American Sign Language dataset. The dataset consisted of 26 English alphabets which were put to training and testing in accordance to the ratio 70:30. The sign images are preprocessed by reducing the size of each image to 128x128. This helps in improving the computational speed of CNN. The number of training iterations is 30, and the variation trend of training and validation accuracies is shown in Fig. 5. According to the figure, no overfitting is noticed and both training and validation accuracies tend to merge after about 25 epochs. All the models are compared in terms of Accuracy performance measure. Accuracy is defined as ratio of correctly classified instances to the total number of instances. As observed from table 1 and Fig. 6, the proposed CNN model outperforms the Pre-trained architectures and achieves an accuracy of 98.11 %. Its computational speed is also faster than other models. Though this paper has achieved high accuracy, the data set is limited in scope and does not include numerics or common sign language gestures.

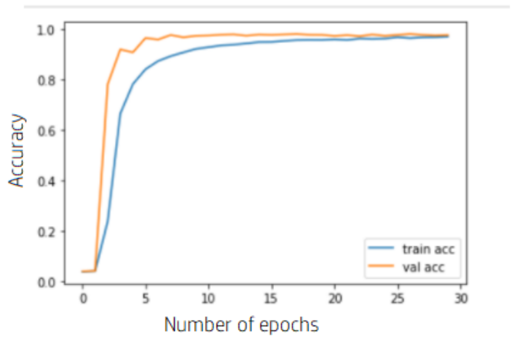

Fig. 5. Variation trend of training and validation accuracies

TABLE I
Accuracies of pre-trained and proposed CNN architectures

Deep Neural Network Models	Accuracy in %	Time in secs (for 30 epochs)
VGG16	91.16	174000
Resnet50	94.22	27000
InceptionV3	76.14	96000
Proposed CNN	98.11	12000

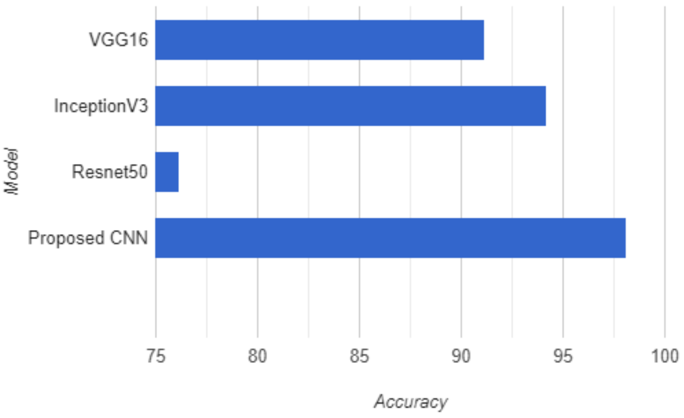

Fig. 6. Comparision between various models

V. Conclusion

This paper tests deep learning neural networks against the American Sign Language(ASL) for Sign Language Recognition (SLR). OpenCV was used to create the ASL dataset, and the images generated were noise-free and adequately lighted. Resnet50 was the best-performing pre-trained network, achieving 94.22 percent accuracy. The proposed CNN model achieved an accuracy of 98.11% (best out of all the networks) and correctly detected the majority of symbols. In the future, higher accuracy can be achieved by layering another model that focuses especially on the classes of alphabets that are becoming confused. Furthermore, work in recognizing signs in complex backgrounds can be done with the implementation of various background subtraction techniques.

References

[1] Z. Yang, Z. Shi, X. Shen, and Y.-W. Tai, "Sf-net: Structured feature network for continuous sign language recognition," , arXiv preprint arXiv:1908.01341, 2019.

[2] Belinda J. Hardin, Sheresa Boone Blanchard, Megan A. Kemmery, Margo Appenzeller, and Samuel D. Parker, "Family-Centered Practices and American Sign Language (ASL): Challenges and Recommendations" , Vol.81(1), Exceptional Children 2014, 2014, pp 107–123.

[3] Erik Drasgow, "American Sign Language as a Pathway to Linguistic Competence ", Vol 64 issue 3, Exceptional Children, 1998, pp 392–342.

[4] Siming He, “Research of a Sign Language Translation System Based on Deep Learning”, 2019 International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM), 2019, pp. 392-396.

[5] Nikolas Adaloglou, Theocharis Chatzis, Ilias Papastratis, Andreas Stergioulas, Georgios Th.Papadopoulos, Vassia Zacharopoulou, George J. Xydopoulos, Klimnis Atzakas, Dimitris Papazachariou, and Petros Daras, “A Comprehensive Study on Sign Language Recognition Methods”, arXiv preprint arXiv:2007.12530v2, 2020.

[6] Aditya Das, Shantanu Gawde, Khyati Suratwala,and Dr. Dhananjay Kalbande, “Sign Language Recognition Using Deep Learning on Custom Processed Static Gesture Images”, International Conference on Smart City and Emerging Technology (ICSCET), 2018.

[7] G.Anantha Rao, K.Syamala , P.V.V.Kishore, A.S.C.S.Sastry, “Deep Convolutional Neural Networks for Sign Language Recognition”, Conference on Signal Processing And Communication Engineering Systems (SPACES), 2018, pp. 194–197.

[8] Lionel Pigou(B), Sander Dieleman, Pieter-Jan Kindermans, and Benjamin Schrauwen, “Sign Language Recognition Using Convolutional Neural Networks”, Springer International Publishing Switzerland, 2015, pp. 572–578.

[9] Prof. Radha S. Shirbhate, Mr. Vedant D. Shinde, Ms. Sanam A. Metkari, Ms. Pooja U. Borkar, Ms. Mayuri A. Khandge, “Sign language Recognition Using Machine Learning Algorithm”, Volume: 07 Issue: 03, International Research Journal of Engineering and Technology (IRJET), 2020, pp. 2122–2125.

[10] Kumud Tripathi, Neha Baranwal and G. C. Nandi, “Continuous Indian Sign Language Gesture Recognition and Sentence Formation”, Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015), 2015, pp. 523–531.

[11] Nada B. Ibrahim, Mazen M. Selim, Hala H. Zayed, “An Automatic Arabic Sign Language Recognition System (ArSLRS)”, Journal of King Saud University – Computer and Information Sciences, 2018, pp. 470–477.

[12] Ming Jin Cheok, Zaid Omar, Mohamed Hisham Jaward, “A review of hand gesture and sign language recognition techniques”, Springer-Verlag GmbH Germany, 2017, pp. 131–153.

[13] Xinxin Xie, Wenzhun Huang, Harry Haoxiang Wang, Zhe Liu, “Image De-noising Algorithm based on Gaussian Mixture Model and Adaptive Threshold Modeling”, Proceedings of the International Conference on Inventive Computing and Informatics (ICICI 2017) , 2018, pp. 226–229.

[14] Shuying Liu, Weihong Deng. “Very Deep Convolutional Neural Network Based Image Classification Using Small Training Sample Size”, 2015 3rd IAPR Asian Conference on Pattern Recognition, 2015, pp. 730–734.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, “Deep Residual Learning for Image Recognition” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778.

[16] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, Zbigniew Wojna, “Rethinking the Inception Architecture for Computer Vision”, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2818-2826.