International Journal of Research in Engineering, Science and Management
Volume-3, Issue-7, July-2020
journals.resaim.com/ijresm | ISSN (Online): 2581-5792

American Sign Language Recognition Using CNN

U. Hari Priya[1], S. Krishna Prasad[2], Meba Meria Jacob[3*], R. Radhu Krishna[4], P. R. Vinod[5]
[1,2,3]*UG Student, Department of Computer Science and Engineering, College of Engineering, Chengannur, India*
[4,5]*Assistant Professor, Dept. of Computer Science and Engineering, College of Engineering, Chengannur, India*
*Corresponding author: mebameria103@gmail.com

***Abstract*: Speech impairment is a disability that affects person's ability to communicate using speech and by hearing. Non signers use other medium of communication such as sign language. Although sign language is ubiquitous, non-signers found it challenging to communicate with signers. This paper discusses some of the methods (SVM, KNN, Logistic regression and CNN) that can be used to implement a method to help make the communication of a non-signer with a signer much easier. At the end of the discussion, it was found that convolutional neural network is the most effective technique among the other methods. The main focus is to create a vision-based application which offers sign language recognition to text thereby enabling dynamic communication between them at real time.**

***Keywords*: Vision based techniques, Spatio-temporal features, Convolutional Neural Network (CNN).**

1. Introduction

Sign language is a form of communication used by speech and hearing impaired people. To express thoughts and emotion, people use sign gestures as a means of non-verbal communication. But the non-signers find it difficult to understand. The core idea of the project is to diminish this obstacle that the signers and non-signers face during their communication and use an ASL recognition system that uses Convolutional Neural Network(CNN) in real time to translate a video of a user's ASL signs into texts to enable dynamic communication. American Sign Language(ASL) substantially facilitates communication in the deaf community. The purpose of the work is to contribute to the field of automatic sign language recognition. This problem consists of three tasks to be done in real time:

1. Obtainng video of the user signing(input).
2. Classifying each frame in the video to a letter.
3. Reconstructing and displaying the most likely word from classification scores(output).

2. Related Works

One of the main approaches towards the sign recognition used includes Hidden Markov Models(HMM) that recognize facial expressions from video sequences combined with Bayesian Neural Network Classifies and Gaussian Tree Augmented Naïve Bayes Classifier. Francois also published a paper regarding human posture recognition in a video sequence using techniques based on 2D and 3D appearance. This work mentions using Principal Component Analysis(PCA) in order to recognize silhouettes from a static camera and then using 3D to model posture for recognition. But this approach has a disadvantage of having intermediate gestures which may lead to ambiguity in training and therefore gives lower accuracy in prediction. Apart from these approaches, there is another approach that involves the analysis of video segments using neural networks which involves extraction of visual information in the form of feature vectors. But the neural network faces lot of issues like difficulty in tracking of hands, segmentation of subject from the background and environment, variation, occlusion etc. The distance measure that this paper used is Euclidean distance and to split the dataset into segment, used K-nearest neighbor.

In a paper by Jie Huang [1], the authors recognized problems in Sign Language Recognition such as a problem in recognition when the signs are broken down to individual words and the issues regarding continuous SLR. They solved this problem without isolating individual signs by removing an extra layer of pre-processing (temporal segmentation) and another extra layer of post-processing. But, combined with the strenuous labelling of individual words added a huge challenge to SLR without the temporal segmentation. So, in order to solve this, they addressed this issue with a new framework called Hierarchical Attention Network with Latent Space (LS-HAN). This eliminates the preprocessing of temporal segmentation. The framework consists of a two-stream CNN for semantic gap bridging and a Hierarchical Attention Network for space-based recognition.

An approach to SLR by T.W. Chong includes using an external device such as Leap Motion controller to recognize the movements and gestures. What makes this different from other works is that it includes the complete grammar of the American Sign Language which consist of 26 letters and 10 digits. The experimental results were promising with accuracies of 80.30 percentage of Support Vector Machines (SVM) and 93.81 percentage for Deep Neural Networks(DNN). Research in the fields of hand gesture recognition also paved their way to aid SLR research such as in the work by Linqin. Here, they used RGB-D to recognize gestures for human-computer interaction. They used Euclidean distance measure to measure the distance between hand joints and shoulder features to generate a unifying feature descriptor. In order to get final recognition

results, a dynamic time warping (DTW) algorithm was proposed. This algorithm works by applying weighted distance and restricted search path to avoid major computation costs unlike conventional approaches. The result gave an average accuracy of 9 6.5 and better. The main idea is to develop real time gesture recognition which could also be extended to Sign Language Recognition system. The work done by Ronchetti on the Argentinean sign language uses a database of hand shapes of the Argentinean Sign Language and a technique for processing images, extracting descriptors and hand shape classification using ProbSom.

C. Hardie used an external device called Mayo armband to collect data about the position of a user's hands and fingers over time. These technologies were used along with sign language translation because they consider each sign as a combination of gestures. This paper used a dataset collected by a group at University of South Wales which consist of parameters like hand positions, hand rotation and finger bend for 95 various signs that are all unique. Every sign has an input stream and the system predicts to which sign the output stream falls. This classification adopted SVM and logistic regression models.

3. Methodology

A custom American Sign Language (ASL) dataset can be used for the project by capturing the images (only hand regions) through the computer webcam. The dataset consists of gestures for the first 18 English alphabets, excluding J. Different variations of a single gesture is also taken into consideration for properly training the model, so that it can achieve a good performance. During preprocessing the first step performed is creating a box in frame of camera where user can put hand to give as input. The gesture made by the user is captured in the frame and it is converted into grayscale image. The grayscale I image obtained from the preprocessing step is fed into the CNN.

The CNN used consist of 3 layers of convolution. The input layer takes image data as input. The 2D matrix is converted into 1D and this is performed in the input layer. The depth of this layer is the number of training images. Softmax activation function is used in the fully connected layer in combination with stochastic gradient optimizer and cross entropy loss function is used.

The fully connected layer with softmax activation gives the probabilities of multiple classes as output. The one with the maximum probability is the prediction made, this is performed by the output layer. This layer has the same number of neuron as the fully connected layer. The loss is calculated through cross entropy loss function and parameter update is performed through stochastic gradient optimization.

4. Proposed System

In this project, following softwares are used:

1. Tensorflow
2. Keras
3. OpenCV
4. CNN

A. *Collecting the Data*

1) This is the first step towards creating a working model.
2) The aim of this module is to collect the dataset and the dataset used here is from Kaggle consisting of 72025 images.
3) The user inputs are converted into processable format (here it is grayscale image). OpenCV can be used to serve this purpose.

B. *Convolutional Neural Network*

CNN or ConvNets are category of neural networks which are respectable in the field of image recognition and classification. CNN uses multilayer perceptron which requires minimal preprocessing to train the architecture to perform the task of recognition or classification effectively. CNN were modelled to perform biological process in terms of connectivity patterns between neuron in the visual cortex of animals. CNN tends to perform better than other image and video recognition algorithms in the field of image classification, medical image analysis and natural language processing.

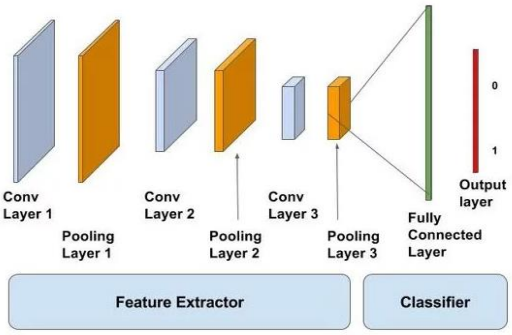

Fig. 1. Convolutional Neural Network

There are four key operation which forms the fundamental blocks of CNN:

1. Convolution
2. ReLU
3. Pooling
4. Classification

C. *Training a CNN*

The training phase of CNN can be summarized as follows:

1. Initialize all filters and parameters to perform the convolution step on the input image.
2. The system takes an input image and goes through all the above steps, shown in figure, in a sequential order and finds an output. Then the outputs are propagated backwards through the network in order to train the network.
3. Step 2 is repeated till the predicted outputs are close to the ground truth and cannot be modified further. These steps train the neural network to perform a specific task.

D. Cross Entropy Loss

Cross Entropy loss, also called log loss, measures the performance of a classification model whose output is a probability value which is between 0 and 1. Cross-entropy loss increases when the predicted probability diverges from the actual label. So predicting a probability of .012 when the actual observation is 1 would be bad and result in a high loss value. A perfect model would have cross entropy loss of 0.

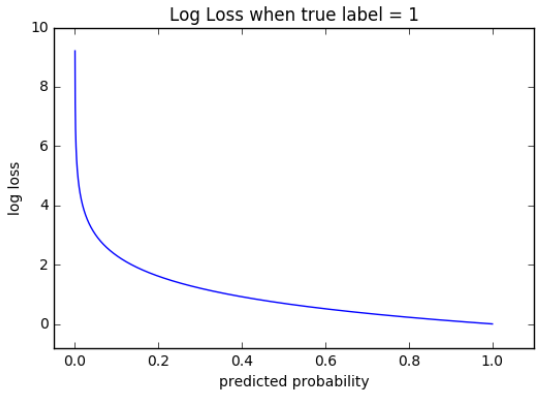

Fig. 2. Cross Entropy Loss

E. SGD Optimizer

A gradient is usually considered as the slope of a function. The more the gradient, the steeper is the slope. Gradient Descent is a convex function. SGD is an iterative method which is used for optimizing an objective function with appropriate smoothness properties (e.g. differentiable or sub-differentiable). It can be considered as a stochastic approximation of gradient descent optimization, since it replaces the actual gradient (calculated from the entire dataset) and hence the name Stochastic Gradient Decent Optimization. In SGD, since only one sample from the dataset is chosen at random and shuffled for each iteration, the path taken by the algorithm to reach the minima is usually noisier and faster than typical Gradient Descent algorithm.

5. System Requirements

A. Hardware Requirements

The minimum hardware configuration required for the proper functioning of the system can be outlined below:

- CPU: Intel Core I3 or above
- Operation Speed: 2.0GHz or above
- RAM: 4GB or above
- Hard disk: 1TB or above

B. Language

Python is the language used. It is a dynamic language which helps to write the code in fewer lines. It is widely used in data science and for producing deep learning algorithm. It includes libraries like NumPy, Scipy etc and other frameworks like Tensorflow, Keras etc.

C. Tools

1. Tensorflow: Tensorflow is a software library that is developed by Google Brain Team within Google's Machine Learning Intelligence research organization, with the purpose of conducting machine learning and deep neural network research. It offers APIs for beginners and experts to develop for desktop, mobile, web and cloud.
2. Keras: It is an open source neural network written in python that runs on top of tensorflow ot theano. It supports modularity and experimentation. It is fast and easy to use.
3. OpenCV: OpenCV (Open Source Computer Vision Library) is a machine learning software library. This contains more than 2500 optimized algorithms which can be used to detect and recognize faces, identify objects, classify human actions in videos etc. It is written natively in C++ and has a template interface that works seamlessly with STL containers.

6. Experimental Result

The experiment was done on the laptop with a webcam to record the signs. Figure given below shows the experiment result obtained by showing signs to the frame. The figure on the right side shows the hand gesture with the output alphabet and the figure on the left side shows its corresponding grayscale image.

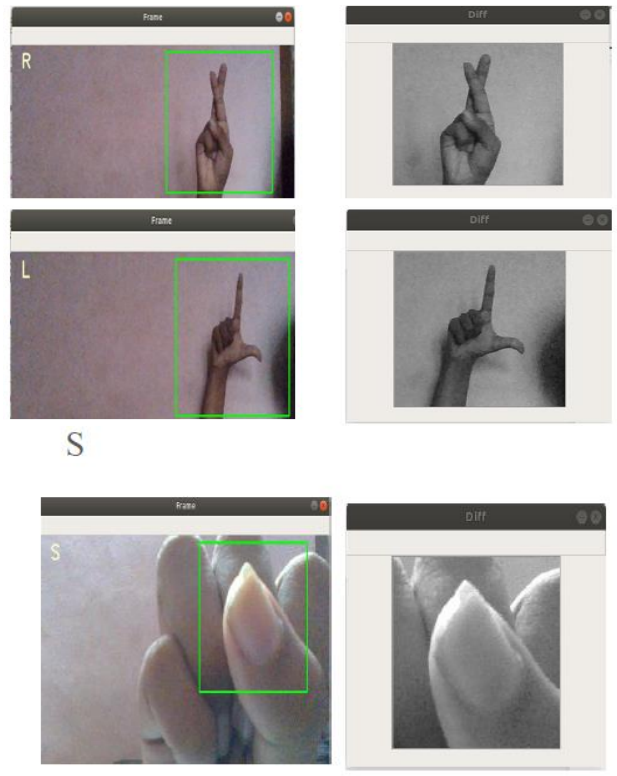

Fig. 3. A gesture is converted to its grayscale image using OpenCV and its corresponding alphabet is displayed

7. Conclusion

From the analysis above it is quite clear that the performance offered by convolutional neural network is far more superior than any of the other classifiers. With CNN being in use it can be said that the model produces accurate result and also simple to create and debug. The model takes real time input as a result it makes direct communication more easily and cause almost no delay. What makes the model unique is that here the alphabets recognized by model are appended together to form words which makes this model much more convenient for its users.

References

[1] J. Huang, W. Zhou and Q. Zhang, "Video based Sign Language Recognition without Temporal Segmentation", 2018.

[2] T. W. Chong and B. G. Lee, "American Sign Language Recognition using Leap Motion Controller with Machine Learning Approach", Sensors, vol. 18, 2018.

[3] V. Athitos, C. Neidle, S. Selaroff and J. Nash, "The American Sign Language Lexicon Video Dataset", 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2008.

[4] C. Hardie and D. Fahim, "Sign Language Recognition Using Temporal Classification," 2017.

[5] C. Linqin, C. Shaungjie and X. Min, "Dynamic Hand Gesture Recognition using RGB-D data for Natural Human Computer Interaction", Journal of Intelligent and Fuzzy Systems, 2017.

[6] Li Wan et al., "Regularization of Neural Networks using Drop Connect," ICML'13: Proceedings of the 30th International Conference on International Conference on Machine Learning, Volume 28, June 2013.