American Sign Language Recognition using Deep Learning and Computer Vision

Kshitij Bantupalli
Department of Computer Science
Kennesaw State University
Kennesaw, USA
kshitij.bantupalli@gmail.com

Ying Xie
Department of Computer Science
Kennesaw State University
Kennesaw, USA
yxie2@kennesaw.edu

***Abstract*—Speech impairment is a disability which affects an individuals ability to communicate using speech and hearing. People who are affected by this use other media of communication such as sign language. Although sign language is ubiquitous in recent times, there remains a challenge for non-sign language speakers to communicate with sign language speakers or signers. With recent advances in deep learning and computer vision there has been promising progress in the fields of motion and gesture recognition using deep learning and computer vision-based techniques. The focus of this work is to create a vision-based application which offers sign language translation to text thus aiding communication between signers and non-signers. The proposed model takes video sequences and extracts temporal and spatial features from them. We then use Inception, a CNN (Convolutional Neural Network) for recognizing spatial features. We then use a RNN (Recurrent Neural Network) to train on temporal features. The dataset used is the American Sign Language Dataset.**

***Index Terms*—computer science, machine learning, computer vision, sign language**

I. Introduction

Sign language is a form of communication used by people with impaired hearing and speech. People use sign language gestures as a means of non-verbal communication to express their thoughts and emotions. But non-signers find it extremely difficult to understand, hence trained sign language interpreters are needed during medical and legal appointments, educational and training sessions. Over the past five years, there has been an increasing demand for interpreting services. Other means, such as video remote human interpreting using high-speed Internet connections, have been introduced. They will thus provide an easy to use sign language interpreting service, which can be used, but has major limitations.

To address this, we use an ensemble of two models to recognize gestures in sign language. We use the custom American Sign Language Dataset for video data for training the model to recognize gestures. The dataset has different gestures performed multiple times giving us variation in context and video conditions. For simplicity, the videos are recording at a common frame rate. We propose to use a CNN (Convolutional Neural Network) model named Inception to extract spatial features from the video stream for Sign Language Recognition (SLR). Then, by using a LSTM (Long Short-Term Memory) [7], a RNN (Recurrent Neural Network) model, we can extract temporal features from the video sequences via two methods: Using the outputs from the Softmax and the Pool layer of the CNN respectively.

Rest of this research is organized as follows: Section 2 gives an overview of the literature study performed, Section 3 discusses the dataset. The architecture in the models used are described in Section 4. Section 5 highlights the experiments and cost to conduct the thesis. Finally, Section 6 shows the problems faced by the model followed by the proposed by possible improvements in Section 7.

II. Related Work

Literature review of the problem shows that there have been several approaches to address the issue of gesture recognition in video using several different methods. In [1] the authors used Hidden Markov Models (HMM) to recognize facial expressions from video sequences combined with Bayesian Network Classifiers and Gaussian Tree Augmented Naive Bayes Classifier.

Francois *et al.* [2] also published a paper on Human Posture Recognition in a Video Sequence using methods based on 2D and 3D appearance. The work mentions using PCA to recognize silhouettes from a static camera and then using 3D to model posture for recognition. This approach has the drawback of having intermediary gestures which may lead to ambiguity in training and therefore a lower accuracy in prediction.

Let's approach the analysis of video segments using Neural Networks which involves extracting visual information in the form of feature vectors. Neural Networks do face issues such as tracking of hands, segmentation of subjects from the background and environment, illumination of variation, occlusion, movements and position. The paper by Nandy *et al.* [3] splits the dataset into segments extracts features and classifies using Euclidean Distance and K-Nearest Neighbors.

Similar work by Kumud *et al.* [4] defines how to do Continuous Indian Sign Language Recognition. The paper proposes frame extraction from video data, pre-processing the data, extracting key frames from the data followed by extracting other features, recognition and finally optimization. Pre-processing is done by converting the video to a sequence

978-1-5386-5035-6/18/$31.00 ©2018 IEEE

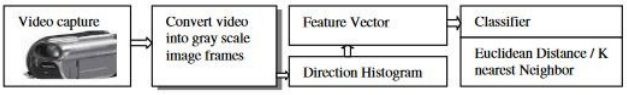

Fig. 1. Workflow followed by Nandy *et al.* [3].

of RGB frames. Each frame having the same dimensions. Skin color segmentation used to extract skin region, with the help of HSV. The images obtained were converted to binary form. The key frames were extracted by calculating a gradient between the frames. And the features were extracted from the key frames using oriental histogram. Classification was done by Euclidean Distance, Manhattan Distance, Chess Board Distance and Mahalanobis Distance.

III. Sign Language Data set

We created a data set of hundred different signs from the American Sign Language data set curated by Neidel *et al.* [5]. Each sign is performed five times by a single signer in varying lighting conditions and speed of signing. For the purpose of consistency, the background in all the videos is the same.

Fig. 2. Sample Gesture from Dataset.

The videos were recorded on an iPhone 6 camera on 60fps and at 720p resolution. Each video was broken down by frame to images and trimmed to 300 frames and then augmented to increase the data set for each sign to 2400 images. The images were re sized and rotated at random as part of the augmentations. The data set was further divided into training and test data sets, with 1800 as part of the training and the remaining as the test data set.

IV. System Architecture

As shown in Figure 2, the CNN model extracted temporal features from the frames which was used further to predict gestures based on sequence of frames. The CNN model used was Inception [6], which was a model developed by Google for image recognition as is widely regarded as the most image recognition neural network which exists right now.

We use two different approaches to classification: a. Using the predictions from the Softmax layer and b. Using the output of the global pool layer.

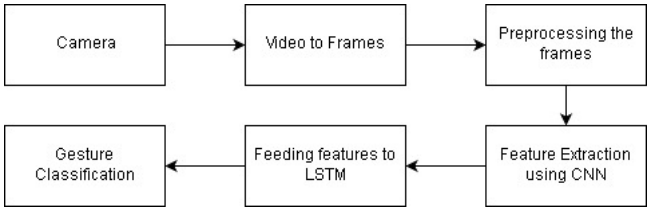

Fig. 3. High Level System Architecture.

The camera is used to record the hand gestures and then the videos are preprocessed to frame sequences which is fed individually to the CNN for two possible outputs. The global pool layer gives us a 2048 sized vector, which possibly allows for more features to be analyzed by the RNN.The feature sequence is then fed to an Long-Short Term Memory (LSTM) allowing longer time dependencies. RNN's suffer from the vanishing/exploding gradient problem, LSTM's deal with the problem allowing for higher accuracies on longer sequences of data. We used two different models of LSTM's which are depicted in Figure 3.

A. Gesture Detection

After creating the dataset of images for training the neural network to classify images, we can simply retrain the existing Inception model to work on our dataset. By using transfer learning, we can take advantage of previous training and use a relatively small training set.

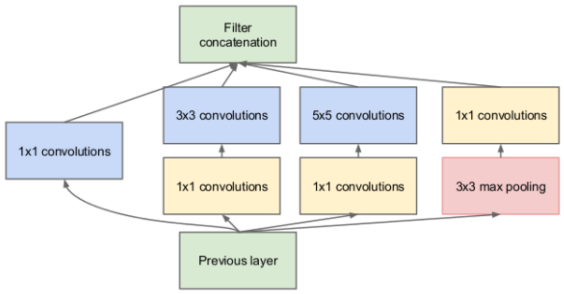

Fig. 4. Inception Module

Using the Inception model places the onus of choosing what type of convolution to use (3x3, 5x5) onto the model itself. Inception performs all the convolutions in parallel and concatenates the resulting feature maps before going to the next layer. The Inception model repeats the operations in Figure 3 several times to create a much deeper network [6][6].

Retraining the final layer of the network in Figure 4 took about 36 hours running on a GeForce GTX 920 GPU. After completion of the training steps, the model reported an incredibly high 99% accuracy on the training set.

We take the outputs of Softmax Layer and the Max Pooling layer for our architecture.

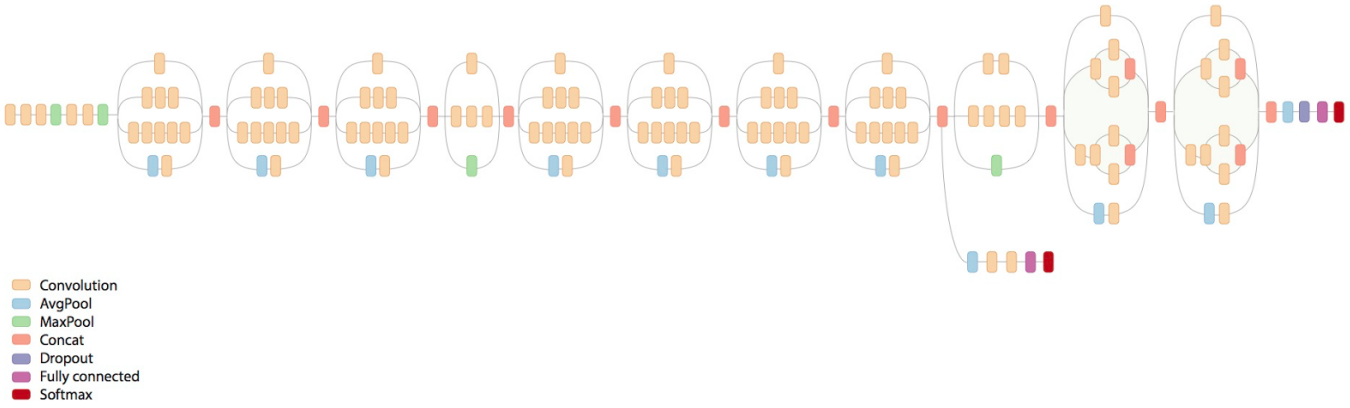

Fig. 5. Inception v3 Architecture.

B. Gesture Classification

We take the outputs of the Softmax Layer and the Max Pooling layer and feed it to the RNN architecture shown in Figure 4.

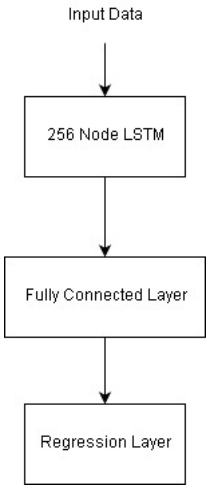

Fig. 6. RNN Architecture.

The gesture segments identified and processed by the CNN are classified by the LSTM [7] into one of the gesture classes using sequence data. Since the input segments have to be a fixed size, we trimmed the length of all the frame sequences. We use an LSTM because of their efficiency with longer sequences of data. We train the LSTM on the outputs from the CNN Softmax layer and the Pool layer and compare the results. After experimenting with deeper and wider RNN's we decided that this network architecture gave the best results in accuracy. The LSTM layer has a dropout of 0.3 to prevent over fitting. It uses ADAM optimizer [8] and a Softmax layer for predictions along with categorical cross-entropy loss.

# of Signs	Accuracy with Softmax Layer	Accuracy with Pool Layer
10	90%	55%
50	92%	58%
100	93%	58%
150	91%	55%

TABLE I
PERFORMANCE WITH VARYING SAMPLE SIZES.

V. EXPERIMENTAL EVALUATION

The CNN and RNN model are trained independently. The data set of 600 training samples of 300 frames each are shuffled and split with an 80-20 split into test and validation data, which gives us 54 samples for training and 6 for validation. We evaluate the CNN and RNN independently using the same training and test samples for both. This ensures that the test data is not seen during training by either the CNN and the RNN.

Both the models were trained to minimize loss by using cross-entropy cost function ADAM [8]. The models are trained with 64 batch sizes and 10 epochs.

Because the recordings are independent of each other, there is no continuation of gestures between two recordings. The gestures were restricted to 300 frames to ensure that all the crucial information was captured and wasn't excessive. The test data set was also augmented to double its size to gather more information from predictions about the model. The accuracy's shown in the table are over two variations of the same gesture.

The predictions for each label for the CNN Softmax and RNN model are as follows in Table 1.

VI. PROBLEMS FACED BY THE MODEL

One of the problems the model faced is with facial features and skin tones. While testing with different skin tones, the model dropped accuracy if it hadn't been trained on a certain skin tone and was made to predict on it.

The model also suffered from loss of accuracy with the inclusion of faces, as faces of signers vary, the model ends up training incorrect features from the videos. So the videos had to be trimmed to only include gestures which were only extended up till the neck.

The model also performed poorly when there was variation in clothing. Maybe using a ROI to isolate hand gestures from the images would help accuracy, but for the context of this paper, a consistent full-sleeved shirt was used in all the gesture recordings.

VII. Potential Improvements

One of the potential improvements would be to experiment with different RNN architectures for the output of the pool layer. Including GRU and Independent RNN's.

In terms of CNN improvements, using Capsule Networks [10] instead of Inception may yield better results than Inception.

References

[1] Ira Cohen, Nicu Sebe, Ashutosh Garg, Lawrence S, Chen and Thomas S. Huang (2003, February). Facial expression recognition from video sequences: temporal and static modeling. Computer Vision and Image Undertaking 91.

[2] Bernard Boulay, Francois Bremond, Monique Thonat, Human Posture Recognition in Video Sequence. IEEE International Workshop on VS-PETS, Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2003, Nice, France.

[3] Recognition of Isolated Indian Sign Language Gesture in Real Time, Anup Nandy, Jay Shankar Prasad, Soumik Mondal, Pavan Chakraborty, G. C. Nandi, Communications in Computer and Information Science book series (CCIS, volume 70)

[4] Continuous dynamic Indian Sign Language gesture recognition with invariant backgrounds by Kumud Tripathi, Neha Baranwal, G. C. Nandi at 2015 Conference on Advances in Computing, Communications and Informatics (ICACCI)

[5] Carol Neidle, Ashwin Thangali and Stan Sclaroff [2012] "Challenges in Development of the American Sign Language Lexicon Video Dataset (ASLLVD) Corpus," 5th Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon, LREC 2012, Istanbul, Turkey, May 27, 2012.

[6] Going Deeper with Convolutions,Szegedy et al. CVPR 2015, IEEE Explore.

[7] Long Short-Term Memory, Sepp Hochreiter et al., Neural Computation 9(8): 1735-1780, 1997

[8] Adam: A Method for Stochastic Optimization, Diederik P. Kingma, Jimmy Ba, Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015

[9] Dynamic Routing Between Capsules, Geoffrey Hindon et al., Nov 2017

[10] Independently Recurrent Neural Network (IndRNN): Building A Longer and Deeper RNN, Shuai Li, Wanqing Li, Chris Cook, Ce Zhu, Yanbo Gao, CVPR 2018