PROJECT REPORT

COMP-8590-1-2019F

SIGN LANGUAGE RECONGNITION USING CONVOLUTIONAL & RECURRENT NEURAL NETWORKS

Submitted By,

Rishav Chatterjee – SID: 110010348

Sanyam Sareen- SID: 105192200

Course Instructor: Dr. Alioune Ngom

Term: Fall 2019

School of Computer Science

University of Windsor

Abstract

According to the *Sacramento Daily Union* report, there are about 700,000 to 900,000 people with the inability to speak and hear. People with impaired hearing and speech use different modes to communicate with others. One of those methods of communication is with the use of 'sign language'. Developing a *Sign Language Recognition* (SLR) system for differently abled people can enable them to perceive & understand the environment in an easier way. This project aims to contribute towards filling the gap between overcoming technological considerations and real-life implementation of proposed sign language recognition system. The focus of this project is to build a dynamic, vision-based mechanism that identifies sign language gestures automatically. The individual frames from a video sequence is to be extracted using a deep learning algorithm. In this project report, forty-six different category of gestures have been taken care of.

It is conspicuous that a video sequence comprises of both the spatial and temporal features. Therefore, two deep learning algorithms, namely Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) respectively are to be used to train the model. The dataset used in this project consists of Argentinian Sign Language (LSA) Gestures, with more than 2000 videos referring to 46 gestures, thereby producing 50 videos per category. The validation accuracy of our proposed model is 93.3%. We further discuss some of the related work done by prominent authors such as the authors in (Kumar, Thankachan & Dominic, 2016), who proposed a vision-based SLR system using support vector machines as classifier; and the authors in (Pigou et al., 2015) also proposed a SLR system using the Microsoft Kinect, convolutional neural networks (CNNs) and GPU acceleration. In this report we try to iterate over these previously proposed systems to further improve the accuracy of the output.

Keywords: Sign Language Recognition, Gesture Recognition, Hand Gestures, Artificial Neural Network, Convolutional Neural Networks, Recurrent Neural Networks, Image Processing, Feature Extraction, Classification, Computer Vision, Machine Learning

Table of Contents

List of Figures

Chapter 1: Introduction

The method of communication between individuals by making shapes or movements with your hands with respect to the head or other body parts along with certain facial cues is commonly referred to as '*Sign Language*'. It is the basic means of conveying messages for those with hearing and vocal disabilities i.e. people with impaired hearing and speech. Sign language gestures are used as a means of non-verbal communication to express individual's thoughts and emotions. However, it is not a universal language i.e. it is not the same all over the world and differs from place to place. Some countries have unique sign languages e.g. American Sign Language (ASL) has its own grammar and rules and is not a visual form of the English alphabets. On the contrary, Portuguese Sign Language (PSL) combines both gestures of hand, body and facial expressions. The process of conveying meaningful information through physical gestures using techniques like data acquisition, pre-processing, transformation, feature extraction & classification is known as '*Sign Language Recognition*' (SLR). SLR is a multidisciplinary research area involving machine learning, pattern recognition, computer vision, natural language processing and psychology.

The recognition of human hand gestures using computer interaction is an extensive area of research in the field of Artificial Intelligence, more specifically in the field of computer vision, pattern recognition & machine learning. A recognition system would thus have to identify specifically the head and hand orientation or movements, facial expression and even body pose. In this report, our primary goal is to create a SLR system to provide an efficient and accurate way to convert sign languages using hand gestures into text or voice aids. Gestures from real-time video needs to be modelled in both spatial and temporal domains, where a hand posture is the static structure of the hand and a gesture is the dynamic movement of the hand and thus can be both static and dynamic. There are multiple approaches for hand gesture recognition mainly, vision-based approaches and data glove approaches. This report mainly focuses on the process of creating a vision-based system able to do real-time sign language recognition.

We propose the design for a basic yet extensible system that can recognize static and dynamic gestures of *Argentinian Sign Language* (LSA). LSA (*see Figure 1*) has been chosen since it is utilized by a vast majority of differently abled people. The dataset used, contains about 2000+ videos belonging to 46 gestures categories, thereby producing 50 videos per category.

The *problem statement* of our project can be stated as follows:

The problem can be simplified by generalizing it to execute three tasks in real-time; First step is obtaining video of the user signing (input). Next, classifying each frame in the video to a letter. Lastly, reconstructing and displaying the most likely word from classification scores (output).

However, from a computer vision perspective, this problem represents significant *challenges* due to a number of considerations, which includes *environmental concerns* (e.g. lighting sensitivity, background, and camera position), *occlusion* (e.g. some or all fingers, or an entire hand can be out of the field of view), *sign boundary detection* (when a sign ends and the next begins) & lastly, *co-articulation* (when a sign is affected by the preceding or succeeding sign). Since SLR implies conveying meaningful information using hand gestures, careful feature selection and extraction are very important aspects to consider and since visual features provide a description of the image content their proper choice for image classification is vital for the future performance of the recognition system.

The purpose of this work is to contribute to the field of automatic sign language recognition. We focus on the recognition of the signs or gestures. This part of the report is aimed towards providing a general idea on SLR systems. The rest of the report is as categorized as follows:

Chapter 2 discusses the related work that has been studied by other prominent authors. Chapter 3 describes our methods and procedure of implementation for creating a SLR system. Chapter 4 provides an in-depth discussion on the experimental design approaches. Conclusion and future work are drawn in Chapter 5. Final parts of this report contain references.

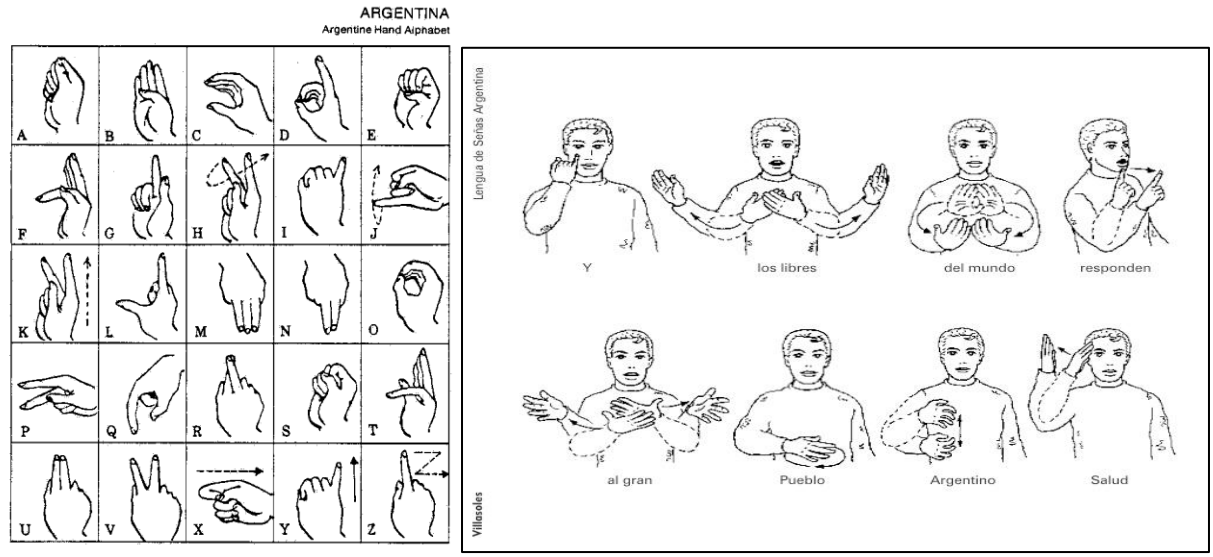

Figure 1: Diagram of Argentinian Hand Alphabet (left) & LSA (right)

Chapter 2: Related Work

2.1 Sign Language Recognition System using SVM as Classifier

In the paper, (Kumar, Thankachan & Dominic, 2016), the authors propose an algorithm to build an 'American Sign Language Recognition System' capable of extracting signs from video sequences using skin color segmentation. The algorithm distinguishes between static and dynamic gestures and extracts the appropriate feature vector. The extracted features were then classified using Support Vector Machines (SVM). The initial step starts with skin color sampling with the input being a video of two second at six frames per second. Then image pre-processing takes place followed by hand segmentation. After post-processing, the algorithm tracks the movement of gestures identifying if they are static or dynamic gestures, which is referred to as threshold comparison. Finally, important features are extracted through a number of steps and classified using SVM. The classification model is built using Java Machine Learning library. The proposed algorithm is also capable of speech recognition using a speech recognition engine called SPHINX. The system was an offline application which has minimal to zero-cost. The system proposed was to aid in communication for those having speech and vocal disabilities. However, further areas of improvement such as increasing the system performance under robust and unfavorable environment (lot of clutter, poor lighting) could be considered.

2.2 Sign Language Recognition System Using Neural Networks

The authors in (Pigou et al., 2015) propose a sign language recognition system using the Microsoft Kinect, convolutional neural networks (CNNs) and GPU acceleration. Since, CNNs are able to automate the process of feature extraction, the proposed system is able to recognize 20 Italian gestures with high accuracy. CLAP14 is used for dataset. The first step starts with image pre-processing where thresholding is used to reduce noise is in depth-maps. Then, CNN is used for classification. The proposed CNN model consists of 6 layers including input and output with 2 inputs in input layer. So, the shape would be $2\times2\times64\times64\times32$. Temporal segmentation is used to predict the begin and end frames of every video samples. During training, data augmentation is used in real-time to reduce overfitting. The predictive model has a cross-validation accuracy of 91.7%. The paper, (Pigou et al., 2015), thus proves that neural networks

can be used to accurately recognize different gestures of a sign language, with users and surroundings not included in the training set.

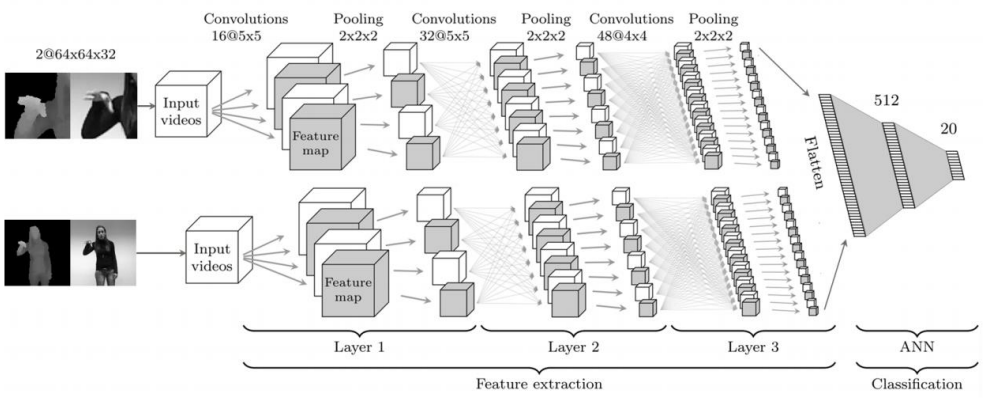

Figure 2: Architecture of Deep-Learning Model in (Pigou et al., 2015)

The authors in (Subharjito et al., 2018) propose a similar model to implement i3d inception model into their sign language recognition system and to analyze the result. A public dataset called LSA64 is used. The first step starts with data pre-processing that is turning videos to sequence of images. 3D CNN architecture is used which is also known as i3d inception. It consists of 67 convolutional layers including input & output. The authors then used 300 videos as training set i.e. 100 for validation set & 100 for testing set. The authors conclude that after several trainings the i3d inception without any modification suffers form too much overfitting. Although accuracy on training is 100% for 10 classes & 1 signer, the validation accuracy tends to lie between 50-80%.

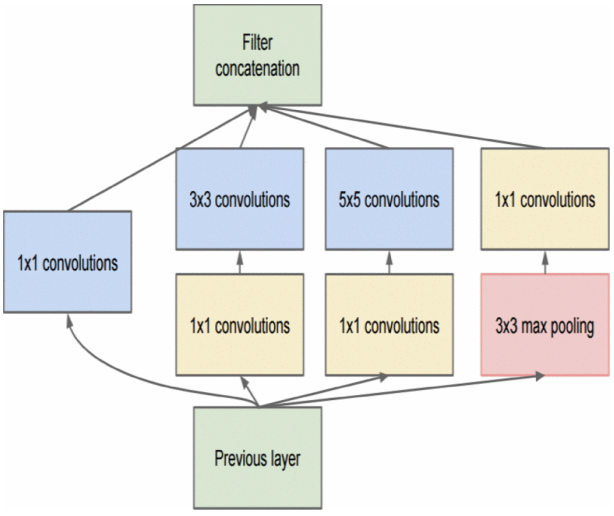

Figure 3: Diagram of inception module from (Subharjito et al., 2018)

Similarly, the authors in (Dogic & Karli, 2014), used the idea of neural networks to develop a Bosnian sign language sign language recognition system. The objective was to use digital image processing methods to develop a system that teaches a multilayer neural network using a back-propagation algorithm. The system operated in 3 phases. First, images were processed by feature extraction methods, and then by masking method (PCA) the data set was created. Lastly, training was done using cross validation method. However, a low validation accuracy of only 84.4% was achieved.

2.3 Sign Language Recognition Using Combination of Vision-Based Features

The authors in (Zaki & Shaheen, 2010) propose a combination of vision-based features in order to enhance the recognition of American sign languages. According to the authors, sign languages are based on four components hand shape, place of articulation, hand orientation, and movement. Three features i.e. kurtosis position, principal component analysis, (PCA) & motion chain code are selected to be mapped to these four components. PCA is used here as a descriptor that represents a global image feature to provide a measure for hand configuration and hand orientation. Kurtosis position is used as a local feature for measuring edges and reflecting the place of articulation recognition. Lastly, motion chain code represents movement of hand. The proposed system consists of three steps namely, hand detection & tracking, appearance-based feature extraction and Hidden Markov Model (HMM) classifier to classify data. The dataset is RWTH-BOSTON-50 database, sampled at 30 frames per second where the size of each frame is 312 X 242 pixels. An error rate of 10.9% was achieved using their proposed method. The authors conclude that parallel HMM and recurrent neural networks can be used to further improve the results.

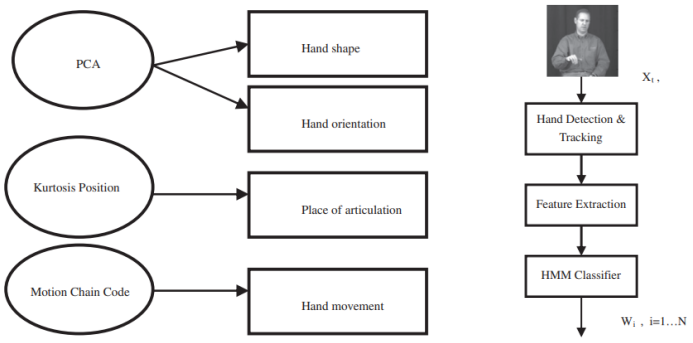

Figure 4: Diagram of proposed feature (left) & proposed system (right) by (Zaki & Shaheen, 2010)

Chapter 3: Methodology

In this section we discuss about our proposed algorithms our methodology and further explain the implementation of the system in the following section. Artificial Neural Networks have been used in general to realize the model. A detailed discussion is presented in the following sub-section.

3.1 Proposed Algorithm

Artificial Neural Networks (ANN) are a set of algorithms that are designed to recognize patterns in data samples & are modeled loosely after the human brain. They interpret sensory data through a kind of machine perception, labeling or clustering raw input. An ANN typically has 1 input and 1 output layer. There might be additional number of hidden layers between the input & output layers, however, that number varies across different network models depending upon the complexity of the problem to be solved. The patterns they recognize are numerical, contained in vectors, into which all real-world data, such as images, sound, text or time series, must be translated. Thus, for simplicity of understanding, neural networks help us cluster and classify data automatically and conveniently.

3.1.1 Convolutional Neural Networks (CNN)

Convolutional Neural Network (CNN) is one of the variants of ANN, used heavily in the field of Computer Vision. CNNs are automatic feature extraction models in deep learning and 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. Multiple filters are applied for each channel, and together with the activation functions of the neurons, they form feature maps. These network of learning units called neurons learn how to convert input signals (e.g. picture of a cat) into corresponding output signals (e.g. the label “cat”), forming the basis of automated recognition. This is followed by a pooling scheme, where only the interesting information of the feature maps are pooled together. These techniques are performed in multiple layers. The hidden layers of a CNN in our project consist of convolutional layers, subsampling, pooling layers, activation, fully connected layers, and normalization layers.

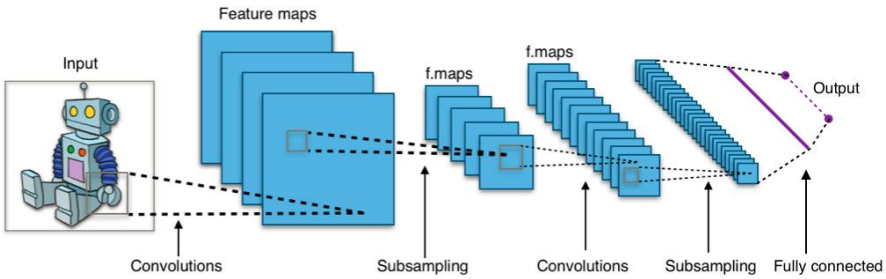

Figure 5: Diagram of a Convolutional Neural Network

3.1.2 Recurrent Neural Network (RNN)

Recurrent Neural Networks or RNN, are another variant of neural networks heavily used in Natural Language Processing. In a general artificial neural network, an input is processed through a number of layers and an output is produced, with an assumption that two successive inputs are independent of each other. RNNs are called recurrent because they perform the same task for every element of a sequence, with the output being depended on the previous computations. A RNN has loops in them that allow information to be carried across neurons while reading in input. The loop allows information to be passed from one step of the network to the next. The decision a recurrent net reached at time step t-1 affects the decision it will reach one moment later at time step t. So recurrent networks have two sources of input, the present and the recent past, which combine to determine how they respond to new data. In theory, RNNs can make use of information in arbitrarily long sequences, but in practice, they are limited to looking back only a few steps.

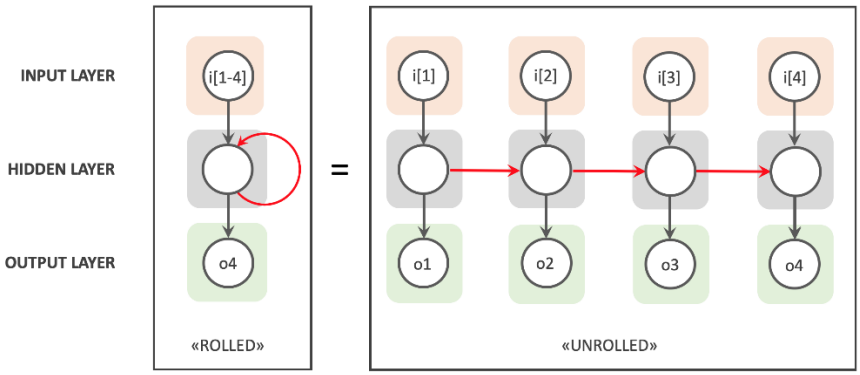

Figure 6: Diagram of Recurrent Neural Networks (left) & copies of a Feed Forward ANN executing in a chain (right)

Chapter 4: Experimental Design

In this section we present an in-depth discussion on the implementation of our proposed model and analyze the results obtained. We further provide the technical considerations and limitations we have encountered while developing the model.

4.1 Implementation

In this approach we extracted spatial features for individual frames using inception model (CNN) and temporal features using RNN. Each video (a sequence of frames) was then represented by a sequence of predictions made by CNN for each of the individual frames. These sequences of predictions were given as input to the RNN.

4.1.1 Data Set Used

The data set used for the approach consists of Argentinian Sign Language (LSA) Gestures, with around 2300 videos belonging to 46 gestures categories, thereby producing 50 videos per category or gesture. Out of the 50 gestures per category, 75% i.e. 40 were used for training and 25% i.e. 10 were used for testing.

4.1.2 Algorithm

The step by step procedure stated is as follows:

- In the first step, individual frames from multiple video sequences of each gesture is extracted.
- Then, any noise such as background and unwanted body parts are removed to highlight the important features from each frame.
- CNN model then trains on the spatial features using frames of the trained data.
- Train and test frame predictions are then stored.
- The inception model (CNN) was also used for predication of frames, which are then fed to the RNN model for training on temporal features.
- Long short-term memory (LSTM) which is an RNN architecture has been used for this purpose.

4.1.3 Frame Extraction and Background Noise Removal

Each video gesture video is broken down into a sequence of frames. Frames are then processed to remove all the noise from the image that is everything except hands. The final image consists of grey scale image of hands to avoid colour specific learning of the model. Figures 7 & 8 represents the extracted frames from video sequences and frames after feature extraction respectively.

Figure 7: One of the extracted frames

Figure 8: Frame after extracting hands

4.1.4 Training CNN for Spatial Features and Prediction of Frames

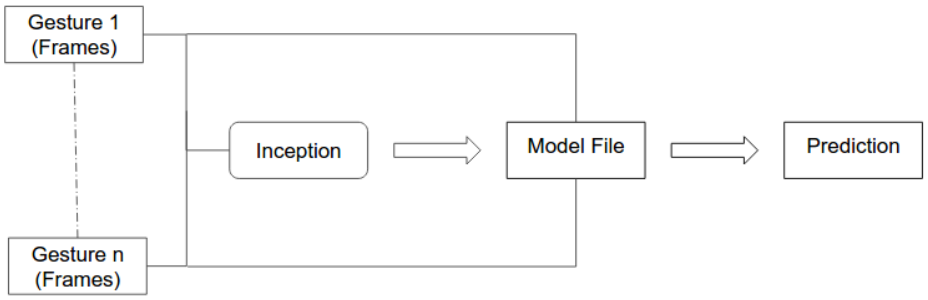

Figure 9: Diagram of CNN used for prediction of frames

The first row in Figure 10 illustrates the video of a gesture the word 'Elephant'. The second row shows the set of frames extracted from it and finally, third row represents the sequence of predictions for each frame by using CNN after training it.

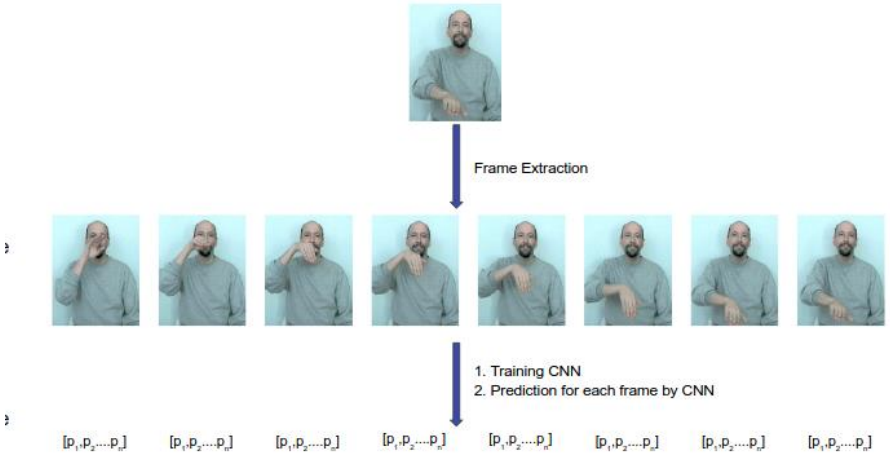

Figure 10: Diagram of working principle of CNN

4.1.5 Training RNN for Temporal Features

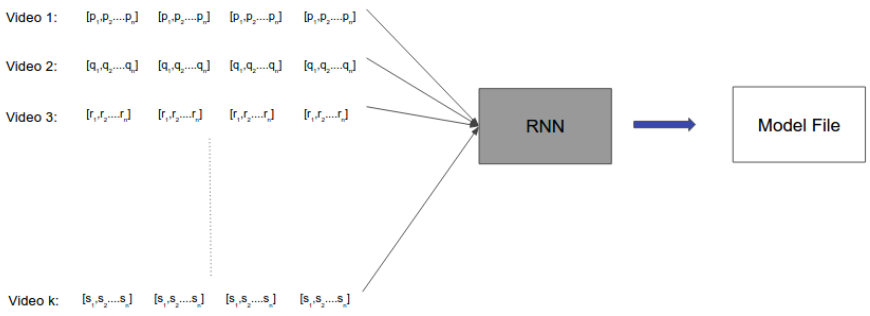

Figure 11: Diagram of RNN used for temporal feature extraction

From Figure 11, we can see the predication of frames obtained by using CNN is fed into the RNN model for training temporal feature extraction. The figure represents prediction of each frames from the sequences of videos labelled 1…k. The RNN model finally generates a model file consisting of the trained set.

4.2 Results

While implementing our approach we found an average accuracy of 93.3% was achieved while testing 438 gestures out of 460 gestures (10 per category). If RNN had more feature points to distinguish among different video the result could be further improved.

4.3 Limitations

The length of the probabilistic prediction by the CNN model in the sequences of predictions of frames is equivalent to the number of classes to be classified. In our case, it is equivalent to 46 as we have 46 classes to classify. Hence, the length of the feature vector of each frame for the RNN is dependent upon the number of classes to be classified. Lesser the number of classes lesser would be the length of the feature vector for each frame.

Chapter 5: Conclusion & Future Research

Sign language gestures are a powerful and effective way of communication for not only speech and hearing impaired individual but across people from different walks of life. In this project, we tried to develop a vision-based sign language recognition system capable of accurately and automatically recognizing signed gestures made by human hand movements. We mainly considered Argentinian Sign Language set as out data set. Dataset of many other sign language models are scarce and not available easily for training. However, in future different other models can be considered as a data set. We used CNN for classifying spatial features and RNN for classifying temporal features. The average accuracy of the model is 93.3%. This could be further improved using a different approach having more feature points to distinguish among different video sequences or through using some other approach.

Since, sign language recognition can be generally categorized as vision-based and sensor based, in the future we might consider implementing a sensor-based model and comparing the results between vision-based and sensor-based applications to obtain a better accuracy. This report summarizes and explains related work of prominent authors, discusses a new proposed system and further serves as a guideline for those who are interested in sign language recognition systems. To conclude, we would like to thank our course instructor for his guidance and support.

References

Agris, U.V., Zieren, J., Canzler, U., Bauer, B., & Kraiss, K. (2007). Recent developments in visual sign language recognition. Universal Access in the Information Society, Vol. 6(4), pages. 323-362.

Bantupalli, K., & Xie, Y. (2018). American Sign Language Recognition using Deep Learning and Computer Vision. 2018 IEEE International Conference on Big Data (Big Data), pages. 4896-4899.

Djogic, S., Karli., G. (2014). Sign Language Recognition using Neural Networks. Technology Education Management Journal, Vol.3(4), pages. 296-301

Hoque, O.B., Jubair, M.I., Islam, M.S., Akash, A., & Paulson, A.S. (2018). Real Time Bangladeshi Sign Language Detection using Faster R-CNN. 2018 International Conference on Innovation in Engineering and Technology (ICIET), pages. 1-6.

Kumar, A., Thankachan, K., & Dominic, M.M. (2016). Sign language recognition. 2016 3rd International Conference on Recent Advances in Information Technology (RAIT), pages. 422-428.

Nikam, A.S., & Ambekar, A.G. (2016). Sign language recognition using image-based hand gesture recognition techniques. 2016 Online International Conference on Green Engineering and Technologies (IC-GET), pages. 1-5.

Oz, C., & Leu, M.C. (2011). American Sign Language word recognition with a sensory glove using artificial neural networks. Engineering Applications of Artificial Intelligence, Vol. 24, pages. 1204-1213.

Pigou, L., Dieleman, S., Kindermans, P., & Schrauwen, B. (2014). Sign Language Recognition Using Convolutional Neural Networks. European Conference on Computer Vision Workshops, Vol. 8925, pages. 572-578.

Qutaishat, M., Habeeb, M., Takruri, B., & Al-Malik, H.A. (2007). American sign language (ASL) recognition based on Hough transform and neural networks. Expert System with Application, Vol. 32(1), pages. 24-37.

Stergiopoulou, E., & Papamarkos, N. (2009). Hand gesture recognition using a neural network shape fitting technique. Engineering Applications of Artificial Intelligence, Vol. 22, pages. 1141-1158.

Suharjito, Gunawan, H., Thiracitta, N., & Nugroho, A. (2018). Sign Language Recognition Using Modified Convolutional Neural Network Model. 2018 Indonesian Association for Pattern Recognition International Conference (INAPR), pages. 1-5.

Zaki, M.M., & Shaheen, S.I. (2011). Sign language recognition using a combination of new vision-based features. Pattern Recognition Letters, 32(4), pages. 572-577.