Sign Language Recognition System

Mr. Pramod Sunagar*, M Sneha, Mipsa Patel, Tilak S Naik and Vibha Karanth
Computer Science and Engineering, M. S. Ramaiah Institute of Technology, Bangalore
pramods@msrit.edu, sneham191@gmail.com,
mipsa.p.patel@gmail.com, s.tilak.naik@gmail.com, vibhakaranth@gmail.com

***Abstract*—Sign language is the language used by the hearing and speech impaired to communicate among themselves and others. Unless the concerned people know the sign language properly, there is a communication barrier between them. In most cases, an interpreter is required to carry out such a conversation. To reduce the dependency of the hearing and speech impaired people on interpreters, we develop a sign language detection system that identifies the gestures from images and videos, which can then be converted into grammatically correct sentences in any language, using various deep learning techniques.**

***Index Terms*—sign language recognition; CNN; RNN; OpenCV; Deep learning**

I. Introduction

Sign Language is a vision based language that uses the movements of the hands and facial expressions for communication. Subtle differences among different hand gestures can have a huge impact on the meaning that they convey. Along with this, it is important to prevent the background noise of the image from affecting the gestures being communicated. We consider such problems and challenges and develop an effective sign language recognition system. There exist models that convert American Sign Language to sentences. However, most of these techniques are not based on deep learning. These models also cannot handle the nuances of various sign languages. We use deep neural networks to develop a model that can identify gestures of sign languages and can later be converted into English text.

II. Background

There have been a lot of different approaches used to solve the problem of sign language recognition. However, most of these use image processing in MATLAB, upon which Machine Learning algorithms are applied. There has also been some work done using artificial neural networks.

Image and video processing is one of the important fields in Machine Learning and Deep Learning at present. The huge number of images and videos on social media and otherwise has led to an increased interest in the field. The algorithms and techniques developed as a result of this can be used in more impactful fields like helping the speech and hearing impaired lead more convenient lives.The hand signs along with the emotions displayed by the individual provide the content and tone of the sentence. A tool to convert their words into a widely used language enhances their ability to communicate.

*Assistant Professor at MSRIT, Bangalore

Since such applications will always be necessary, research and development of new algorithms to improve the efficiency of interpretation of sign language will be endless. Deep learning has a lot more to offer than what has already been explored. With such possibilities, there will always be scope to develop newer models that can be used in sign language recognition. The models used in this can also be used in other applications such as gesture controlled systems, tracking unusual activity, etc. with slight tweaks. With slight modification, it can also be used for recognition of other sign languages.

Moreover, the current work can be further extended to output speech instead of text. A similar generative model can be used to map speech or text to actions by stitching together pre-recorded videos or actually generate an interface with a skeletal structure. This model can be seamlessly integrated with various applications on different platforms to support high productivity.

III. Literature Survey

There have been a lot of different approaches used to solve the problem of sign language recognition. However, most of these use image processing in MATLAB, upon which Machine Learning algorithms are applied. There has also been some work done using artificial neural networks.Simulations may require a few hours to weeks. Scope and content vary greatly. However, similar principles apply to all simulations.

[1] deals with the double handed Indian Sign Language. It is captured as a series of images, processed with the help of MATLAB and then converted to speech and text. However this approach uses image processing techniques which are highly sensitive to lighting conditions. [2] proposes an image processing, computer vision and neural network based approach to identify the characteristics of the hand in images taken from a video through webcam. Identification of hand shapes from continuous frames is done by using series of image processing operations. Interpretation of signs and corresponding meaning is identified by using Haar Cascade Classifier. Finally displayed text is converted into speech using speech synthesizer. [3] proposes a system that consists of three phases, a training phase, a testing phase and a recognition phase. In the training phase, each class is trained with a multiclass support vector machine (MSVM). Hu invariant moment and structural shape descriptors are combined to make a combinational feature vector that are to be extracted from the input image in the testing phase after applying preprocessing.

In the recognition phase, different classes are used for testing an input gesture.

Deep neural networks have revolutionized how different Machine Learning problems are approached. Using them to solve this problem can give significant results, and that is precisely what this project explores.

IV. Risk Identification

Areas of risk can be one of the following:

1) Mis-recognition of gestures as the program returns before waiting for further gestures that might occur after a time gap. (No explicit way to signal the end of a sentence).
2) The same sentence might mean different things when said in different tones so the program also needs to take into account the expressions of the subject.
3) Missing frames in the video due to improper data transfer or improper recording.
4) Insufficient training.
5) Incorrect training data.
6) Insufficient lighting in the videos.

.

V. Design

This project has been designed such that it incorporates some of the major applications of machine learning and deep learning. Various concepts such as image and video processing, different neural network architectures, etc. are used. Simplicity and effectiveness are the key factors considered while designing and developing this application.

A. Architecture Design

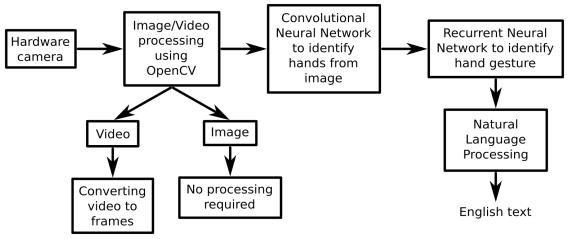

Fig. 1. Architecture

Figure 1 gives a brief overview of the architecture that the Sign Language Recognition application follows.

Input module
: A camera is required to record the entire conversation. Letters and digits can be considered as single frames, whereas words and sentences are recorded as videos.

Image and Video Processing
: Image and video processing is done using Python and OpenCV. Videos are converted into a series of frames to simplify the process of recognizing gestures.

Convolutional Neural Network
: These networks are a class of deep feedforward neural networks that are based on weight sharing. They are commonly used in image and video processing to identify manifolds. In this project, they are used to identify hands from the given frames.

Recurrent Neural Network
: Recurrent Neural Networks are used whenever memory of previous computations is required. It is used to identify the hand gesture by considering the gestures denoted by the current and the previous frames.

Natural Language Processing
: Natural Language Processing is used to convert the output of the Recurrent Neural Network into proper English text.

B. Data Flow Diagram

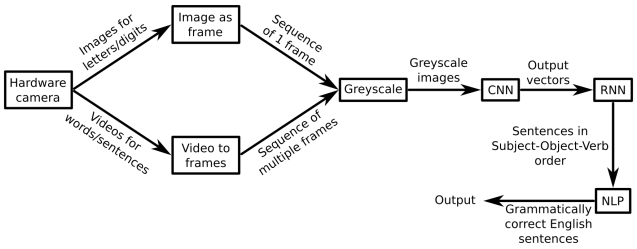

Fig. 2. Data Flow Diagram

Figure 2 gives the interaction between the different modules of the application and how data flows between them.

VI. Implementation

A. Technology Introduction

Python
: A high level programming language for general purpose programming. It is an interpreted language with a syntax that allows programmers to express concepts in fewer lines of code.

OpenCV
: *Open Source Computer Vision* is a library of programming functions mainly aimed at real-time computer vision. Here were using OpenCV for gesture recognition.

CUDA
: A parallel computing platform and application programming interface (API) model created by Nvidia. It allows software developers and software engineers to use a CUDA-enabled graphics processing unit (GPU) for general purpose processing

PyTorch
: An open source machine learning library for python, which is used for training and testing deep neural networks.

B. Algorithm

Convolution neural network
Use a filter that runs across the input images along with maxpooling to identify specific features in images that help identify gestures.

Recurrent neural network
The CNN can only be used on one frame but gestures are spread across time. It is often better to use RNN that retains the previous inputs in memory.

Optimizer
Adam optimizer combines the features of AdaGrad and RMSProp to learn adaptively, and is currently known to be the best learning algorithm.

C. Implementation of the Modules

Input
CNN is trained on individual frames while RNN is trained on videos. A common wrapper is used to load frames as tensors in order for RNN and the same are randomly accessed for CNN.

Image and Video Processing
The frames are extracted from the video using OpenCV and are converted to grayscale, cropped, scaled and converted to tensors.

Neural network
A common model is developed and RNN can be optionally disabled while training CNN.

VII. Results

The CNN RNN model gives a decent accuracy of 40% with very little training. The same model can be trained on a larger data for higher number of epochs to obtain better accuracies.

VIII. Conclusion

It is a challenge to convert hand gestures to text due to wide variety of signs and their associated meanings. It is important that a sign language recognition system is as accurate as possible, due to the number of people that could use it and benefit from it. In such a case, even if such applications already exist, there is always a possibility of developing more effective or advanced systems using newer technology, and the work presented here tests one such approach.Use of python and inbuilt libraries for it has greatly simplified implementation, training the neural networks for the purpose of this project was made faster with the help of CUDA enabled GPU.

IX. Future Work

Since such applications will always be necessary, research and development of new algorithms to improve the efficiency of interpretation of sign language will be endless. Deep learning has a lot more to offer than what has already been explored. With such possibilities, there will always be scope to develop newer models that can be used in sign language recognition. The models used in this can also be used in other applications such as gesture controlled systems, tracking unusual activity, etc. with slight tweaks. With slight modification, it can also be used for recognition of other sign languages.

Moreover, the current work can be further extended to output speech instead of text. A similar generative model can be used to map speech or text to actions by stitching together pre-recorded videos or actually generate an interface with a skeletal structure. This model can be seamlessly integrated with various applications on different platforms to support high productivity.

References

[1] K. K. Dutta, S. K. R. K, A. K. G. S, and S. A. S. B, "Double handed indian sign language to speech and text," in *Proceedings of the 2015 Third International Conference on Image Information Processing (ICIIP)*, ser. ICIIP '15. Washington, DC, USA: IEEE Computer Society, 2015, pp. 374–377. [Online]. Available: http://dx.doi.org/10.1109/ICIIP.2015.7414799

[2] K. Dabre and S. Dholay, "Machine learning model for sign language interpretation using webcam images," in *2014 International Conference on Circuits, Systems, Communication and Information Technology Applications (CSCITA)*, April 2014, pp. 317–321.

[3] K. Dixit and A. S. Jalal, "Automatic indian sign language recognition system," in *2013 3rd IEEE International Advance Computing Conference (IACC)*, Feb 2013, pp. 883–887.