RESEARCH ARTICLE OPEN ACCESS

Real-time Sign Language Identifier using Convolutional Neural Networks

Jerrin Mariam Mathew, Rebin Shaji, Sany Mary Thomas, Nilammudeen
Mar Baselios College of Engineering & Technology
Trivandrum, Kerala, India
sanimthomas15@gmail.com

****

Abstract:

In the realm of multi-modal communication, sign language is, and continues to be, one of the most understudied areas. With the recent advances in the field of deep learning, there are far reaching applications and implications that neural networks can have for sign language interpretation. Here, we propose a method for translating sign language using MobileNet to classify images to letters, digits and words in American Sign Language and Indian Sign Language standard.

Keywords **— sign language, MobileNet, classification**

****

I. INTRODUCTION

Sign Language is a communicating medium in the hearing-impaired community. However, only limited speakers are there which indeed results in limited amount of people that they can easily communicate with the deaf community. The alternative which is written communication can be impersonal, cumbersome and even impractical in a situation of emergency. We introduce a sign language detection system which uses Convolutional Neural Network in real time to translate a video format of a user's ASL or ISL signs into the corresponding text format in order to enable dynamic communication and to diminish this obstacle. Sign Language is a unique type of communication that often goes understudied. While interpretation can formally be known as the process of translation between a spoken or written language and signs.

The main objective of our system is the real time conversion of sign language to text format or video format. This is proposed to provide a helping-hand for the deaf to communicate with the society by using sign language. This helps in the removal of the middle person who commonly acts as a medium of translator. This system contains a user-friendly environment for the user by providing speech or text output for a sign gesture as input. This is proposed to allow deaf people to edit, make and give review video-based sign language contributions online, parallelly to the way in which people make text-based contributions on the Web.

II. LITERATURE SURVEY

As per the literature survey, the existing papers uses a 3D glove for hand detection and recognition. The gestures has also been limited so detection rate was less. Although it has custom user interface for efficient acquisition of training samples, finding right balance for obtaining high accuracy is crucial and difficult. Researches have done on extracting features from images and videos but they do not emphasis on recognition of ambiguous gestures under consideration. Several techniques have been used by various researches for recognizing hand gestures. Some researches worked with static hand gestures, while others with video only. Kinect have

been fast and accurate in 3D sign language recognition based on the depth and color images captures but Kinect controller is not common with everyone. Classifiers using Convolutional Neural Networks(CNNs) and k-Nearest Neighbors(k-NN) have been used for greater robustness and faster recognition rates but their performance have been relatively low.

III. PROPOSED SYSTEM

In this paper, we are proposing a system that can train the model, which classifies basic ASL and ISL words, letters and numbers. Then converting the model to TensorFlow Lite format for image extraction based on its features. Building a native Android Application for implementing TensorFlow Lite library. Transfer the converted model to the application built. Obtain input from the speaker through real time video. Classify each frame in the video to letter locally by the model trained. Display the output to the user real time in text format.

The advantages of the proposed system is that:
- It contains a real time character prediction
- It has faster classification with MobileNet
- It needs less processing power when compared
- It is easily accessible to everyone with ease

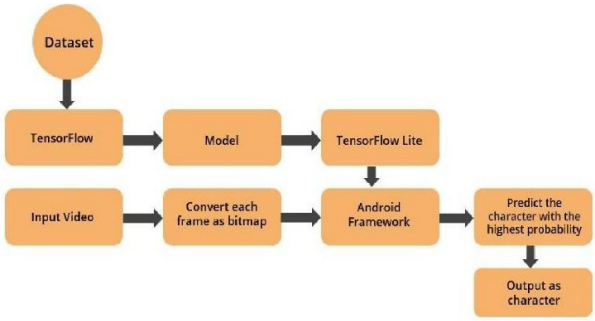

Fig. 1 Process Flow Diagram of Proposed System

A. Data Acquisition

The different approaches to acquire data about the hand gesture can be done in the following ways:

1. Use of sensory devices: It is accomplished by the use of electromechanical devices that offer precise hand configuration and position. To extract information, various glove-based methodologies were applied. However, it is both costly and inconvenient to use.
2. Vision based approach: In vision-based approaches, a mobile camera is utilised as an input device to observe data from hands or fingers. Vision-based solutions simply require a camera, allowing them to recognise natural human-computer interaction without the need for any additional hardware. The most difficult aspect of vision-based hand detection is dealing with the huge fluctuations in human hand appearance caused by a high number of hand movements. Also, consider the various skin colour options, as well as the camera's viewing locations and speed while filming the picture.

B. Data Acquisition

Data cleaning and data transformation are the two major sub-phases of the data preprocessing system. This phase's output is fed into the machine learning algorithms.

- Data cleaning: Data cleaning is done by removing many redundant data that occurs in the dataset.
- Data transformation: Data is transformed suitably to optimize performance and execution time. We tried using segmentation algorithms to undertake hand segmentation of an image, but as noted in the research papers, skin colour and tone are very dependent on lighting circumstances, therefore the segmentation we attempted produced lighter results. Furthermore, we have a large number of symbols to train for our project, and many of them look similar to each other, such as the gesture for letter "V" and digit "2," so rather than segmenting the hand out of a random background, we decided to keep the background of our hand at a stable single colour so that we don't have to segment it on the basis of skin color.

C. Feature Extraction

The image is represented as a three-dimensional matrix with height and width dimensions and depth values for each pixel.Furthermore, utilising CNN, these pixel values are used to extract valuable characteristics.

D. Gesture Classification

The classification of movements is done using Hidden Markov Models (HMM) in [1]. The dynamic features of gestures are addressed in this approach. The skin-color area corresponding to the hand is tracked into a body–facial space centred on the user's face to extract gestures from a succession of video images. The objective is to distinguish between two types of gestures: static and dynamic. Our approach uses two layers of algorithm to predict the final symbol of the user.

1) ***CNN Model:***

• 1st Convolution Layer: The resolution of the input image is 128x128 pixels. It is first processed using 32 filter weights to create the first convolutional layer (3x3 pixels each). This will produce a 126X126 pixel image for each of the filter-weights.

• 1st Pooling Layer: The images are sampled down using 2x2 max pooling, which means we keep the maximum value in the array's 2x2 square. As a result, our image has been reduced to 63x63 pixels.

• 2nd Convolution Layer: The 63 x 63 pixels from the first pooling layer's output are fed into the second convolutional layer. The second convolutional layer uses 32 filter weights to process it (3x3 pixels each). As a result, you'll get a 60 x 60 pixel image.

• 2nd Pooling Layer: The images are then downsampled again using a maximum pool of 2x2 and reduced to a resolution of 30 x 30 pixels.

• 1st Densely Connected Layer: These images are now fed into a 128-neuron fully connected layer, and the output of the second convolutional layer is reshaped into a 30x30x32 =28800-value array. The 2nd Densely Connected Layer receives the output of these layers. To avoid overfitting, we use a dropout layer with a value of 0.5.

• 2nd Densely Connected Layer: The 1st Densely Connected Layer's output is then used as an input to a completely connected layer.

• Final Layer: The 2nd Densely Connected Layer's output is used as an input for the final layer, which will contain the same number of neurons as the amount of classes we're classifying (alphabets + blank symbol + words).

2) ***Activation Function***: In each of the layers, we used ReLu (Rectified Linear Unit). For each input pixel, ReLu calculates max(x,0). This gives the formula some nonlinearity and makes it easier to learn more sophisticated features. It aids in the elimination of the vanishing gradient problem as well as the acceleration of training by lowering calculation time.

3) ***Pooling Layer****:* The input image was subjected to Max pooling with a pool size of (2, 2) and the ReLu activation function. This lowered the number of parameters, lowering computing costs and reducing overfitting.

4) ***Dropout Layers****:* By setting them to zero, this layer "drops out" a random collection of activations in that layer. Even if part of the activations are left out, the network should be able to produce the correct classification or output for a certain data [5].

5) ***Optimizer:*** We used the Adam optimizer to update the model in response to the loss function's output. Adam combines the benefits of two gradient descent extensions: adaptive gradient algorithm (ADA GRAD) and root mean square propagation (RMS Prop).

E. Result and Analysis

The majority of the research publications focus on employing kinect-like devices to detect hands. In [2] they use convolutional neural networks and kinect to create a recognition system for Dutch sign language with a 2.5 percent mistake rate. [4] uses a hidden markov model classifier with a vocabulary of 30 words to create a recognition model with a 10.90 percent error rate. In this research, we propose a functional real-time application for deaf and dumb persons based on identification of American and Indian sign languages. For the given dataset, we were able to reach an accuracy of around 80%. After constructing two layers of algorithms, we are able to improve our prediction by verifying and predicting symbols that are more similar to one another. We can recognise practically all symbols this way if they are displayed correctly, there is no background noise, and the illumination is acceptable.

F. Future Scope

By experimenting with various background removal methods, we hope to obtain improved

accuracy even in the situation of complicated backgrounds. We're also considering upgrading the preprocessing to better predict gestures in low-light situations.

IV. CONCLUSIONS

By using a CNN we are proposing an American Sign Language and Indian Sign Language translator in an Android application. This provides many people to understand sign language easily. New technology is giving rapid changes to our world. And barriers for deaf people are also no more. With the help of artificial intelligence, we are developing both hardware and software which promotes the deaf individuals communication and learning. These sign language helps in interpretation between deaf person and a person who is not deaf. A sign language interpreter who is fluent in both English and Sign Language is required for signing, finger spelling, and specific body language recognitions which is practically difficult in emergency situations. Both ASL and ISL is different from English and has its own grammar.

REFERENCES

[1] T. Yang, Y. Xu, and "A. Hidden Markov Model for Gesture Recognition", CMU-RI-TR-94 10, Robotics Institute, Carnegie Mellon Univ., Pittsburgh, PA, May 1994.

[2] Zaki, M.M., Shaheen, S.I.: Sign language recognition using a combination of new vision based features. Pattern Recognition Letters 32(4), 572–577 (2011).

[3] Byeongkeun Kang , Subarna Tripathi , Truong Q. Nguyen "Real-time sign language fingerspelling recognition using convolutional neural networks from depth map" 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR).

[4] Pigou L., Dieleman S., Kindermans PJ., Schrauwen B. (2015) Sign Language Recognition Using Convolutional Neural Networks. In: Agapito L., Bronstein M., Rother C. (eds) Computer Vision - ECCV 2014 Workshops. ECCV 2014. Lecture Notes in Computer Science, vol 8925. Springer, Cham.

[5] aeshpande3.github.io/A-Beginner%27s-Guide-To-UnderstandingConvolutional-Neural-Networks-Part-2/

[6] Brandon Garcia, Sigberto Alarcon Viesca "Real-time American Sign Language Recognition with Convolutional Neural Networks", 2016.

[7] https://en.wikipedia.org/wiki/TensorFlow

[8] https://en.wikipedia.org/wiki/Convolutional_neural_network

