Final Project: Cognitive Computing & Deep Neural Networks

Real-Time Interaction Using Sign Language

Interpretation of Sign Language to Text & Speech and Interaction with a Virtual Assistant

Dawna Grace Raj

Jai Soni

Nikhil Kohli

OVERVIEW

Communication is the most important part of life. Around 1% of the total population of the world is suffering from hearing impairment, and their life is not as easy as it is for human without limitations.

Deaf people can't hear. Most hearing people don't understand sign language.

Sign language are languages that use visual-manual modality to convey meaning. Sign languages are full-fledged natural languages with their own grammar and lexicon. ASL(Argentinian Sign Language) is a complete, complex language that employs signs made by moving the hands combined with facial expressions and postures of the body.

With this project, we propose a Computer Vision based model that recognizes ASL Letter and few specific words; and convert sign language to text & Speech using Sequence to Sequence models & Deep Neural Networks so that deaf people can communicate with other people in an effective way.

GOALS

Our Goal with this project is to achieve a real time conversation using Sign Language.

We will start with creating the prototype for recognizing signs from static gestures and will progress to interpret signs for individual letters to facilitate real-time recognition and interaction with a Virtual Assistant or another human who does not understand Sign language. We will do this by interpreting the letter into text and further into speech.

As a future goal, we will try to implement word recognition in video and face time conversations with Sentence-Level Sign Language Translation.

We will implement this using either using a real time web-cam demo or real time interaction with a Virtual Assistant (Alexa/Google Mini) using just Sign Language.

1. To recognize and interpret Sign Language letters from a Real-Time Image/Video/Feed/Web-cam into text
2. Convert the interpreted text into Speech to interact using the interpreted sign with a Virtual Assistant
3. A speech to text system to transcribe the response for the user
4. Providing the accuracy of the predicted text or speech outcome
5. Implementing the entire model in Keras or Tensorflow

APPROACH

We used the following approach for the classification of sign language gestures -

Video sequences contain both the temporal as well as the spatial features. We will use two different models to train both the temporal as well as the spatial features.

To train the model on the spatial features of the video sequences, we will use a deep CNN (convolutional neural network). CNN will be trained on the frames obtained from the video sequences of training data.

Then we will use RNN (recurrent neural network) to train the model on the temporal features. CNN model will be used to make predictions for individual frames to obtain a sequence of predictions or pool layer outputs for each video. Now this sequence of prediction or pool layer outputs will be given to RNN to train on the temporal features.

DATA

For Letter recognition, we will use a custom based Images of Signs and will try using Data Augmentation & GANs to increase the instances of each sign corresponding to the letter.

The data used consists set of Argentinian Sign Language (LSA) Gestures, with around 3200 videos belonging to 64 different types of words.

+ Argentinian Sign Language Video Dataset

http://facundoq.github.io/unlp/lsa64/

USE CASES

Facilitating the interface to have a conversation with a hearing disabled person without knowledge of Sign Language.

Real time Conversations –

Deaf people can't hear. Most hearing people don't understand sign language. Bridging this gap means a hearing person can interview a deaf person, or someone who is hard of hearing, via Skype or Google Hangout. They can hold a meeting or do job interview, and just communicate in a natural way.

Interaction with Virtual Assistants –

This will aid deaf people to interact with Virtual Assistants like Siri/Google Home/Alexa and be able to let them use the latest in technology in a useful way. They can ask Virtual Assistants to call 911 or to a hospital in an emergency situation.

Job Interviews and Consultants –

A hearing person can interview a deaf person, or someone who is hard of hearing, via Skype or Google Hangout. A hearing person can give consultation online to a deaf person.

ASL Learning Application –

Another application could be an ASL learning app, where those using American Sign Language would be able to evaluate their proficiency through the video captioning.

PROCESS OUTLINE

1. Data Pre-processing
    - Data Cleaning, Frame Capturing from Videos
    - Resizing the Frames as necessary
2. Exploratory Data Analysis
3. Study of Sequence to Sequence approaches and select the best model for interpretation of Sign languages
4. Design a Text to Speech model to interact with Virtual Assistant
5. Design of a pipeline and system to implement this approach
6. Deploy the Model on AWS or Google Cloud Computing Platform
7. Build a web application to demonstrate the usage of the application by interacting with a Web-cam or a virtual assistant using only Sign Language.

MILESTONES

Timeframe	Delivery
Week 1	1. Developing the Project Proposal 2. Studying about Sign language and Sequence to Sequence Modelling.
Week 2	1. Data Preprocessing and Exploratory Data Analysis

	2. Developing the Model, Training, Selection
Week 3	1. Training the model for accurate results 2. Connect the model with a virtual assistant for demo 3. Deployment of models on cloud and build web application

PERSONAS

1. Real time Conversations –
- Facilitating the interface to have a conversation with a hearing disabled person without knowledge of Sign Language.
2. Interaction with Virtual Assistants/Consultants –
- To aid deaf people in interaction with Virtual Assistants like Siri/Google Home/Alexa. They can ask Virtual Assistants to call 911 or to a hospital in an emergency situation.

DEPLOYMENT DETAILS

1) Language: Python
2) Container: Docker/Github
3) Cloud Tools/Platforms: AWS (Amazon Web Services) EC2, GCP
4) Other Considerations: Alexa/ Google Mini Dot, Flask

REFERENCES

http://facundoq.github.io/unlp/lsa64/