6.S198 Deep Learning Practicum Fall 2018

Project Proposal: ASL Translator

Students: Marwa Abdulhai & Adrian Meza

Purpose: A recent estimate claims that around 13 million people have some level of proficiency in American Sign Language, making it the third most commonly used language in the United States. The goal of our project is to create a system that will convert video feed of a person signing in American Sign Language into English text. We hope to encourage communication between those who have no other way of expressing their ideas other than using Sign Language and those who do not know how to sign. We hope to cater to the 500,000 deaf people in the US and Canada who's natural language is the American Sign Language (ASL), and work towards creating a more inclusive environment.

Framework & Libraries: We will be exploring techniques of using Convolutional Neural Networks in Tensorflow and Pytorch, and making a decision in the next week or so on which is best to use in our case.

Datasets: Some of the data sources that we hope to explore include the open-sourced American Sign Language Lexicon Video Dataset, a database of more than 3,300 ASL signs in citation form, each produced by 1-6 native ASL signers, for a total of almost 9800 tokens. The dataset includes multiple synchronized videos showing the signing from different angles, with annotations including gloss labels, sign start and end time codes, start and end handshape labels for both hands, morphological and articulatory classifications of sign type. For compound signs, the dataset includes annotations for each morpheme. To facilitate computer vision-based sign language recognition, the dataset also includes numeric ID labels for sign variants, video sequences in uncompressed-raw format, and camera calibration sequences.

Some of our preliminary processing steps include clipping the videos by the start and end time tags provided. In addition, we will need to figure out how we will feed our video data to our Convolutional Neural Network. Some preliminary steps include resizing videos to preserve aspect ratio and using the TV-L1 optical flow algorithm to capture the pattern of apparent motion. For the Flow stream, we first sample the videos at 25 frames per second, convert the videos to grayscale, and apply the TV-L1 optical flow algorithm. Pixel values are truncated to the range [-20, 20], then rescaled between -1 and 1. We only use the first two output dimensions, and apply the same cropping as for RGB.

We will look at this blog post to understand how continuous video classification is done as a model for our network: https://blog.coast.ai/continuous-video-classification-with-tensorflow-inception-and-recurrent-nets-250ba9ff6b85

Other preprocessing techniques are detailed here:
https://www.usenix.org/legacy/publications/library/proceedings/es99/full_papers/lee/lee_html/node5.html

Dividing the Work: We have currently set up a time to meet every Tuesday from 1:00 - 3:00 pm to ensure constant progress on our project. Our goals broken down into safe, target, and stretch goals are as follows:

Reqs for data review: data structure, where we obtained, good enough(?), storage, preprocessing

1. Safe Goals (Simple App; small sign understanding capabilities):
    1.1. Storage and Preprocessing figured and tested by 10/17
        1.1.1. Investigate how much storage needed (Adrian)
        1.1.2. Collect data Storage Options (Marwa)
        1.1.3. Test Preprocessing Techniques on Video (Adrian & Marwa)
    1.2. CNN Structure Figured out by 11/01 (in tandem with Training & Testing)
        1.2.1. Investigate viable NN structures for video (Adrian & Marwa)
    1.3. Training , Validation, and Testing by 11/14
        1.3.1. Organize data for Training & Testing (Marwa & Adrian)
        1.3.2. Run data through Neural Net
        1.3.3. Assess performance of net & investigate other options if necessary (Adrian & Marwa)
    1.4. Working app by 12/5
        1.4.1. Figure out how to transfer model to JS (Marwa)
        1.4.2. Use Tensorflow JS to build a simple web app (Adrian)
2. Target Goals (A little more detailed; medium sign understanding capabilities):
    2.1. Storage and Preprocessing figured and tested by 10/12
    2.2. CNN Figured out by 10/25
    2.3. Training , Validation, and Testing by 11/01
    2.4. Working app by 11/14
3. Stretch Goals (Formal App; Large sign understanding capabilities):
    3.1. Storage and Preprocessing figured and tested by 10/5
    3.2. CNN Figured out by 10/18
    3.3. Training , Validation, and Testing by 11/14
    3.4. Working app by 11/21
    3.5. Work on NLP component by 12/5
    3.6. Implement More Sign understanding capabilities by 12/5

We are interested in using the potential of deep learning to cause a social impact and work on applications in the sphere of accessibility.