

**6.S198 Deep Learning Practicum Fall 2018**

Bridging the Gap Between People

Student: Marwa Abdulhai & Adrian Meza

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 to encourage communication between those who have no other way of expressing their ideas other than signing and those who cannot 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).

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

Some of the data sources that we hope to explore include the 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.

There are a couple of misconceptions that we will need to deal with while processing videos

Dealing with Misconceptions

* ASL is a distinct language with it’s own syntax and grammar. ASL signs used to convey ideas and concepts rather than actual words
* Facial expressions and body language are essential to convey the full meaning of the sign.
* Word order is flexible in ASL: ”I am hungry” can be signed as “I hungry” or “Hungry I”. Articles are not signed. Different grammatical forms for nouns, adjectives, and adverbs are also not distinguished
* Need to speak with people who know ASL

Finding Data

* [American Sign Language Lexicon Video Dataset (ASLLVD)](http://www.bu.edu/av/asllrp/dai-asllvd.html) : Videos of > 3,300 ASL signs in citation form, each produced by 1-6 native ASL signers
* Using Amazon Turk to gather data?

Literature Review & TensorFlow Tutorials

* [Language Recognition Using 3D Convolutional Neural Networks](https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7177428)
* [Video Based Approach to Translation ASL to Sentences](https://pdfs.semanticscholar.org/8545/22ab41b16a7ea82deaedd8d78a0b4ad4d313.pdf)
* TensorFlow [Tutorials](https://github.com/Hvass-Labs/TensorFlow-Tutorials)

Building CNN:

Using Convolutional Neural Networks to process videos into text (does not have to be coherent sentences)

Other techniques: <https://sites.google.com/site/autosignlan/home> including PCA, K Nearest Neighbors

Dealing with Video As Input – Have only handled image data previously. Previous methods have only worked with ASL Alphabet or numbers

Getting Data - Do I have enough or will I need to look for more via Amazon Turk?

* Recognition of alphabet and letter images in ASL

**Written Deliverables**

Submit one per a 1-2 person team:

**A proposal** detailing the project you’ve decided to do. It should be about one page long, detailing the purpose of the project and outlining the work to be done (and how you will split that work with your partner, if applicable). You should have pinned down the details of your project by this point:

* What **framework and libraries** you will be using (DeepLearnJS, Tensorflow, etc.).
* What **datasets** you will be using, how you will acquire them, and what preprocessing you have to do.
* If you are working with a partner, how you will evenly divide up the work.
* Your **internal milestones and timeline with specific dates**. Keep in mind that everything will take longer than you expect, so make sure to have intermediate finishing points in case you don’t accomplish your main goal. Have a safe goal, a target goal, and a stretch goal.

You should go over these proposals with your LA/TA and industry mentor before submitting to make sure that the scope of the project is appropriate and the roadmap is well-planned. Please submit the proposal as an attachment to Gradebook/Learning Modules (this will be the only assignment for the class not submitted through a Google Form).

**A data review** describing the datasets that you have acquired for training on your final project. This should be approximately two pages long with the following information:

* **What your data looks like** with a few examples. Detail the number of classes/labels, the number of samples per label, the dimensions of each sample (100x100 pixels, 5 seconds per sound clip...etc), and the storage size of the dataset (how many GB/MB?).
* **How/from where you obtained this data**
* **Why you think this data is “good enough”** for what you want to do for your final project.
* **Licensing**: Is this data open sourced or only for restricted use? What restrictions does it have? Does your dataset contain PII (personally identifiable information)?
* **Where you are storing the data**. This cannot just be a link to a kaggle dataset or some website where you plan to download the data. A huge challenge in ML is storing, parsing through, and computing on large datasets that may not fit on your computer. We want to see that you have mostly gotten past this hurdle.
* If transfer learning (using a pre-trained neural network like SqueezeNet or MobileNet) is an important part of your final project, you also need to: (1) describe the base neural network you are working off of, (2) why it is a good fit for your project, and (3) why you think the additional data you have for your project will play well on top of it.
* What/whether **preprocessing** is required to get the data into a form that you can input into your system. You do not have to have this implemented yet, but you do need a reasonable idea of how to do this.

This should be a near-finalized and complete version of the data you will use to train your system. The needs will vary for each project, but your datasets should be at least several thousand samples per label if you are training something from scratch.

**A blog post** written for a technical audience about what you achieved, the steps you took toward completing your project, the challenges you faced, and what you learned. This will serve as the equivalent of a project writeup. You can see examples from Spring 2018 in the next section.

**An online demo and the source code** demonstrating your end-product. This should be linked to your blog post. You can include videos or images of your system in action.

**Project Ideas**

Here are the types of projects that you might do. We encourage you to talk to your LA/TA or industry mentor for more ideas. Although we want your final projects to have some technical depth, we welcome novel applications.

**Application.** Design a real-world application that relies significantly on a deep learning component to perform some task. For instance, you might find a health records dataset and attempt to use deep learning to predict medical outcomes, or you might create your own facial recognition door lock hardware interface.

**Exploratory.** Investigate the properties of neural networks, similar to the style of the assignments. For instance, you might look at ML interpretability, investigating how certain learned features indicate bias.

**Replication.** Take a recent deep learning paper, implement their architecture carefully, and replicate the results. Try to extend or improve their results further.

[Select final projects from the Spring 2018 version of 6.S198](https://docs.google.com/document/d/1EKMtZ-9ei2mJXpXInWDX-4q70FIvnr_WCyokMJIlZ5g/edit?usp=sharing)

**Resources**

There will be two lectures that cover (1) how to obtain large quantities of data for ML and (2) how to connect to and use a GPU to train a more complex model. Here are some resources to look at while you’re getting started:

Public datasets:

<https://github.com/awesomedata/awesome-public-datasets>

<http://archive.ics.uci.edu/ml/index.php>

<https://www.kaggle.com/datasets>

Compute resources:

AWS credits for students: <https://aws.amazon.com/education/awseducate/>

Google Cloud credits for students: <https://cloud.google.com/edu/>

Microsoft Azure credits for students: <https://azure.microsoft.com/en-us/education/>