Interim Report for Computer Vision Final Project

Shyam Narasimhan, Satendra Varma, Scott Mathews

Due March 30, 2018

1 Introduction

As the power of machine learning techniques continues to improve, the number of interfaces between machine and man continue to grow. For example, assistants such as Alexa, Siri, Cortana, and Google Assistant have popularized a speech based interface with computers.

Our goal in this project is to prototype a gesture based interface with computers, with the goal of being able to recognize a predefined series of hand based gestures from a live video feed. To this end, we apply cutting edge computer vision techniques to identify the maximal

2 Background and Related Work

Gesture recognition is a fairly broad area, which can be interpreted in several ways. For example, one prior work in the area included recognizing shapes that the user drew with their finger ⟨1⟩. The other major source of prior work is in American Sign Language recognition ⟨4⟩.

What makes our contribution novel, is that we have yet to identify an end to end neural network based method for gesture recognition. Our model will be easily trained, so long as it is provided sufficient data.

Additionally, we have yet to find benchmarks for this task. One of the outcomes for our project will be to establish an easily interpretable benchmark for the task of hand gesture recognition, as a baseline for future work.

3 Progress so far

So far, we have gathered most of the data we will need for the project, and established a baseline in performance against which our future efforts will be

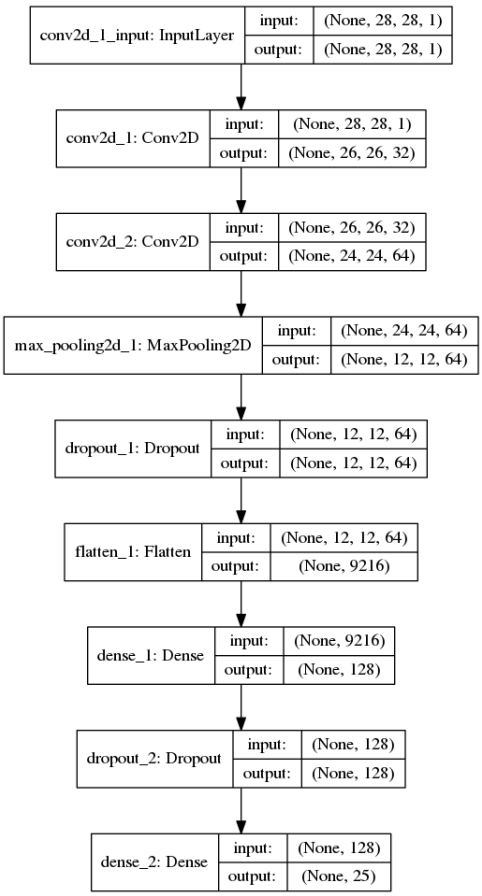

Figure 1: Baseline Model Architecture

compared.

First, we gathered data from several sources. $\langle 2-4 \rangle$, and wrote the boilerplate code to compile them together into one dataset. Since we were dealing with image data, we took the approach of simply scaling all larger images down to 28x28 and making them black and white. Thus far, we have approximately 30,000 data points for training, and 10,000 data points for testing.

The model we have built so far is quite rudimentary, built based on example neural networks we found $\langle 5 \rangle$. Nevertheless, after only 10 epochs of training our model achieves 90% accuracy on our dataset.

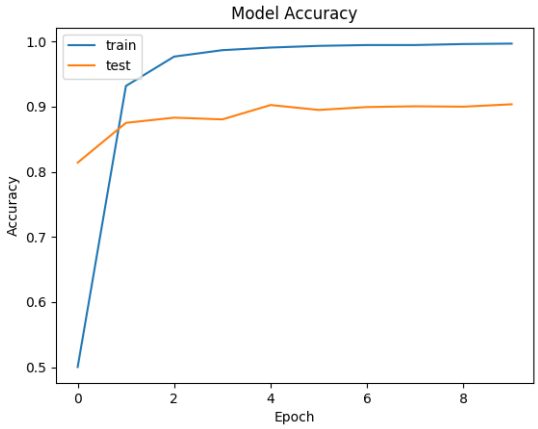

Figure 2: Baseline Model Accuracy

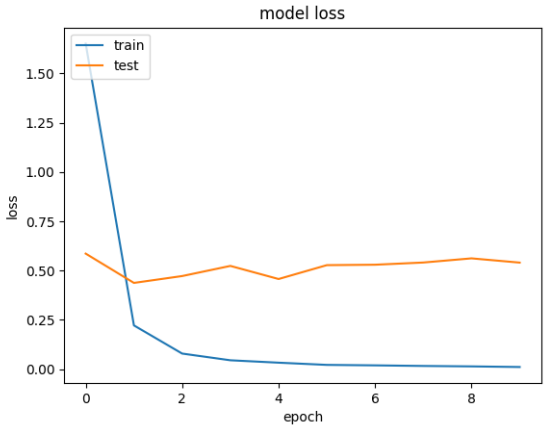

Figure 3: Baseline Model Loss

4 Revised research plan

In this section we enumerate the specific tasks which need to be done, as well as our time line for completing these tasks.

4.1 Tasks

1. Data Collection
   (a) Compile various datasets into one
   (b) Investigate adding time-series image data to improve accuracy
2. Model Building
   (a) Tweak parameters to improve convolutional model performance as high as possible
   (b) If time-series data is added, compare current method against convolutional LSTM
3. Application Development
   (a) Use OpenCV live video feed API to demonstrate our model in real-time
   (b) extract model parameters for use in application
   (c) Build user interface for visualizing results on live video

4.2 Schedule

April 1 - April 7

- Video data gathering - manually collect and label at least 100 video data samples. (1b)
- Investigate hand gesture video datasets already in existence (1b)
- Work on improving model as much as possible, investigate scaling up model to work on larger images to improve accuracy. (2a)
- Prototype live video feed through OpenCV, assessing whether a Python based application is fast enough for our processing needs, our backup plan is using C++ for the application. (3a)

April 8 - April 14

- Using newly collected video data, build a model utilizing time-series information, and compare accuracy to that of original model. (2b)
- Build prototype complete pipeline for application, from video capture to video display with label. (3c)
- Continue to iterate on static recognition model. (2a)

April 15 - April 21

- Make production configuration neural network run on CPU with acceptable efficiency for livestream. (3b)
- Integrate model into application
- Begin working on final paper and poster: since all research should be done, compile references into bibtex.

April 22 - April 28

- Buffer time for any work that has fallen behind
- Finalize model to use in application
- Design and print poster for poster session

5 References

1. http://www.cs.toronto.edu/ aamodkore/more/reports/glive/itsp.html

2. https://www.faceplusplus.com/gesture-recognition/

3. https://github.com/mon95/Sign-Language-and-Static-gesture-recognition-using-sklearn/blob/master/Dataset.zip

4. https://www.kaggle.com/datamunge/sign-language-mnist/data

5. https://github.com/keras-team/keras/tree/master/examples