Austin Spurlock, David Morris, Chris Rodriguez

Dr. Hooper

Artificial Intelligence

10/30/20

American Sign Language: Where Does AI Fit In?

We decided to explore utilizing computer vision to read American Sign Language finger spellings and transcribe what is signed into a text format for our semester project. We discovered this problem when one of our group members noticed a real-life example of a deaf person's difficulties trying to communicate with a blind person. One of these individuals tried to use sign language to communicate, but this was ultimately futile as the blind person could not see the gestures. The communication problem between these two groups is significant because it creates a language barrier that prevents the visually impaired and hard of hearing from communicating with each other. According to the National Institute on Deafness and Other Communication Disorders, 2 out of every 1,000 children in the US are born with some form of detectable hearing loss (NIH). The National Federation of the Blind’s 2017 report reveals that over several hundred-thousand people have some form of visual disability, which further shows how important the destruction of this language barrier is (NFB). This observation made our group think of ways that artificial intelligence can be utilized to minimize this language barrier and allow these individuals to communicate unhindered. By allowing a hard of hearing person to sign naturally, they can communicate at their regular tempo with a visually impaired person as they listen to the output of their own pace.

The history of artificial intelligence and sign language can be traced as far back as 1988, where James Kramer and Larry Leifer, researchers at Stanford, developed a glove that utilizes hand gestures to process and verbalize sign language (Stanford, 1988). However, this translation method did not become too popular, as it did not translate what the hearing person was saying back to the deaf individual. Some more recent developments in translating sign language come from SignAll, which utilizes machine learning and computer vision to translate sign language into text (SignAll). The system has two monitors, one for both parties, to read each other’s messages and respond accordingly. This was a significant upgrade from the last technologies, as it allowed both parties to communicate seamlessly. However, the deaf individual still has to wear special gloves, which still has a small downside when considering this form of communication. The last notable development is by KinTrans, which uses 3D datasets and machine learning (KinTrans). This analysis style is more powerful than relying on gloves, as it allows for a more commercial use for the translator. Rather than having the deaf individual put on special gloves to translate, they can sign as they normally would, which feels more natural and adheres to a more sanitized practice. Like SignAll, the hearing individual can type their response on a screen so the deaf individual can respond accordingly. These breakthroughs in the translation of sign language have had a significant impact on the deaf community, and more developments are being reached every day.

Our project utilizes neural networks to translate American Sign Language into English. During our research, we found a slew of helpful tutorials to understand what we need to create our neural network. Eventually, we found a Digital Ocean tutorial by Alvin Wan, which gave us the building blocks we needed to get our project started. Alvin broke down precisely what we required step by step and understood what each part of the code does. We use the OpenCV python library, which is what we use for our image utilities. Beyond OpenCV, we are using PyTorch to create a deep neural network and Onnx to export it. The program uses these to translate what letter the user is holding up to the English equivalent. The program works by training data with PyTorch’s tensors and exporting the data using Onnx. Each letter of the alphabet is assigned a value 0-25, and we use camera.py to analyze the pixels that our webcams capture. It compares what our webcams see and outputs the letter that it thinks it recognizes. However, his building blocks were not perfect, and we had to add some code ourselves to improve its functionality. The first thing we had to do is add trained data for the letters J and Z. This is because Alvin did not want to implement letters that required hand movements into the training data. Furthermore, we had to find a way to fix the program's read rate, as it tried to read what letters were being signed too quickly. Lastly, we needed to create a pretty UI because every good translator needs a stunning UI to go along with it.

We successfully implemented OpenCV to detect both gestures and letters when using ASL to communicate. On top of allowing our programming to translate ASL to letters and words, we were able to adjust the read rate to enable the user to hold their sign for a minimum of 0.75 seconds before the program would record the input. This would allow for fewer mistakes, and the user would have more time to adjust their hands to sign a specific letter or gesture. We also decided to store each session into a text file that would allow users to share the results where they liked. The results of each session are stored in a .txt file where the program environment is located. The program also contains an intuitive interface that allows users to navigate and use the program with ease. To wrap our project up, we implemented voice feedback after the user signs each letter, along with reading the entire word, words, or sentence upon completion. This feature was put into the program for the visually-impaired audiences we had in mind before starting this project.

Our future vision for this project is to enable the program to read the user’s screen visually. For example, the visually-impaired user would be able to run the program while on zoom so that the OpenCV could read the other participant’s sign language intuitively. This would create an alternative solution for the visually-impaired and ultimately is more inclusive. It would allow people to feel more comfortable participating in video calls. For this to work, we would have to train the program thoroughly and decrease the error rate. Beyond this, we would like to make a more intuitive user interface to utilize the program.

Work Cited:

Kramer, James, and Larry Leifer. “The Talking Glove.” *ACM SIGCAPH Computers and the Physically Handicapped*, no. 39, 1988, pp. 12–16., doi:10.1145/47937.47938.

Wan, Alvin. “How To Build a Neural Network to Translate Sign Language into English.” *DigitalOcean*, DigitalOcean, 9 Oct. 2020, www.digitalocean.com/community/tutorials/how-to-build-a-neural-network-to-translate-sign-language-into-english.

“Product Info.” *SignAll*, www.signall.us/product\_info/.

“Home.” *Kintrans*, www.kintrans.com/.

“Quick Statistics About Hearing.” *National Institute of Deafness and Other Communication Disorders*, U.S. Department of Health and Human Services, 2 Oct. 2020, www.nidcd.nih.gov/health/statistics/quick-statistics-hearing.

“Blindness Statistics”, *National Federation of the Blind* , Jan. 2020, https://www.nfb.org/resources/blindness-statistics