HelpingHand: The ASL Translator

Colin Cumming[1], Dennis Huynh[2], Griffin Clark[3], Alexia Quinn[4], Nicolas Wills[5]

QMIND – Queen's AI Hub
Queen's University, Kingston, Ontario K7L 3N6, Canada.

1 e-mail: 16cmc21@queensu.ca
2 e-mail: 16dh24@queensu.ca
3 e-mail: 17gsc2@queensu.ca
4 e-mail: 18amq@queensu.ca
5 e-mail: 17nvw@queensu.ca

Abstract: *We began this project with the intention of bridging the communication gap between the hearing impaired that require on American Sign Language (ASL) for communication and those that do not know ASL. There is a problem in society where people are uneducated in ASL, even though it is estimated 500,000 people natively speak in ASL* [1]. *The project hoped to increase the accessibility of the hearing impaired and improve inclusivity in day-to-day life. To solve the problem, a convolutional neural network was designed to translate ASL alphabet signs in real time and display the translated output to the user via a web application. We were successful in developing a model which was 93% accurate in a controlled environment but performed at 64% accuracy in real time.*

1. INTRODUCTION

1.1 Motivation

There are currently over 500,000 people in the United States and Canada who use American Sign Language as their natural language [1]. This is only a portion of the one million adults in Canada who have a hearing-related disability, a number which is projected to increase by over 20% in the next few years [2]. There is an abundance of programs translating text to ASL; however, resources in translating hand signs are limited. Everyone deserves to be able to communicate with others, regardless of the presence of a hearing disability. According to Communication Services for the Deaf, 72% of deaf children's families do not use sign language and 98% of people who are deaf do not receive education in sign language [3]. The gap in communication between users and non-users of ASL needs to be addressed in a way that is easy to use and accessible to the multitude of people who want to communicate with users of ASL.

1.2 Related Works

There have been several projects that have attempted to make a sign language translator. First, Google has developed AI software that tracks the shape and motions of hands. The developers, Valentin Bazarevsky and Fan Zhang, say that the freely published hand-tracking software serves as the basis for sign language detection. However, it will be up to the developers to create an application that does sign language translation. Another invention is haptic gloves that translates sign language into an Android application which then reads the text aloud [4]. Similarly, DeepASL, a glove-less technology developed by a team from Michigan State University, translates hand signs with a camera, then sends that video feed through a deep learning algorithm to identify the word, and finally, it constructs a sentence. Both of these inventions translate hand gestures to words, which leads to the problem of expression gained from facial expressions and speed of signing. Also, translating words or sentences with these methods would not include the sign language vernacular. Finally, a Hungarian company called SignAll has developed full translation through the use of cameras that considers body language, facial features, speed, and hand gestures per person with the use of computer vision and AI [5]. However, it only translates to text that displays on a screen.

1.3 Problem Definition

For ASL to be implemented into a conversation, both the speaker and the listener require knowledge of the language. This leaves many Americans and Canadians unable to effectively communicate with people that have hearing loss. In order to bridge the communication gap, we set out to create an ASL real time translator. In order for something like this to be effective in day to day use it needs to be portable, easy to use and be able to translate sign language quickly and accurately. We are not the first team to come up with the idea of making ASL more accessible. The multitude of projects that are related to this field gave us a good understanding of what would be possible in the time period allotted. The published work also allowed us to do valuable research to choose the type of neural network that would work best for our project. As mentioned in *Section 1.2*, Google recently developed a software to track the movement of hands in a live video. After looking into this feat of engineering, we decided that using this level of AI would become too complex for our project. DeepASL, a Michigan State research project is another amazing example of communication barriers being broken. The team used a HB-RNN as well as a probabilistic framework to create single word as well as sentence translation. Although this was much deeper dive into the problem than what we were planning, the published paper gave us a step-by-step demonstration of how to go about our problem. After going through all our research, we decided that with the time given, translating single letters and numbers would be the best course of action.

2. METHODOLOGY

2.1 Dataset generation

The beginning of the project began with sourcing a dataset. This dataset needed to be extensive enough for the model to be able to determine signs in multiple lighting conditions and hands at different angles. A dataset was procured from an online dataset repository website Kaggle [6]. This dataset consisted of 87,000 images of the ASL alphabet. This included the 26 letters of the alphabet and three additional signs for *space, delete,* and *nothing*. This provided 3,000 images of varied lighting and angles for the model to be trained on.

The images had to be preprocessed using OpenCV and Numpy before they could be provided as input to the model. The preprocessing involved converting the images to greyscale (rational described in the following paragraphs) as well as resizing them to improve the speed of the training. The images are paired with a label that identifies the corresponding ASL alphabet sign to the image which is represented as a one-hot encoded value signifying a specific letter of the alphabet. This label is coupled with the Numpy matrix data that represents the image. These values together create data that is readable by the model allowing it to be trained on the image data.

2.2 Model Creation

The proposed solution involved creating a convolutional neural network (CNN) that was capable of translating ASL to text and displayed to the user. The CNN was developed using Keras, a machine learning library build as an additional interface to the well-known library TensorFlow. Keras was chosen as the machine learning library of choice due to its ease of use and power. We concluded on using a CNN for the solution as they work well with unstructured data and are commonly used for image classification purposes.

After the model was trained on the data, it was evaluated on a separate testing set of data. This data was prepared in the same way as the training data but was excluded from the training set. Results from the evaluation can be found in section *3. Results and Discussion*.

2.3 Additional Analysis

To improve the accuracy of the model, additional procedures were considered. A recurrent neural network (RNN) was created to perform background subtraction on the images. This was never tested to see if the background subtracted images created an improved model. Multiple pre-processing techniques were applied on the images including Gaussian blur filters, edge enhancing, and greyscale. Minimal accuracy difference was observed, and the accuracy differences were not documented. The design team applied a greyscale filter in the final model as it reduced the colour dimension of the image, improving training times. Feature engineering and modification of hyper-parameters were applied in an iterative fashion to create the model.

3. RESULTS AND DISCUSSION

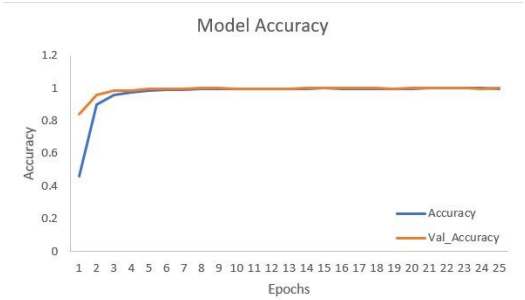

Figure 1 - Graph depicting the model accuracy over time represented in epochs, or rounds trained.

Figure 1 shows the model's accuracy after a given number of epochs. The accuracy plateaus at 93% accuracy. Despite this value, practical accuracy was much lower when the user inputted data had very noisy backgrounds. From this, we learned that the image preprocessing technique used must work well in very noisy environments, which was not considered during the development phase of the project.

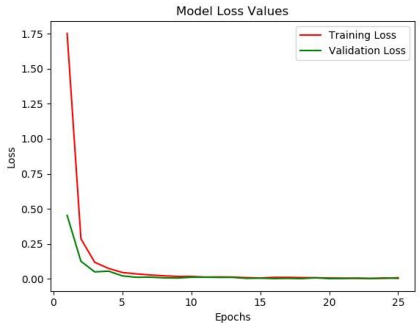

Figure 2 - Graph showcasing the trend of the mean loss values over time represented in epochs, or rounds trained.

Figure 2 shows that the sum of the errors between the train and test guesses plateaus at zero as the number of epochs increases. Ideally, the loss would exhibit a more gradual decline towards zero. This figure is indicative of overfitting which may have contributed to the lower practical accuracy.

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 50, 50, 32)	320
conv2d_2 (Conv2D)	(None, 50, 50, 64)	18496
max_pooling2d_1 (MaxPooling2	(None, 25, 25, 64)	0
conv2d_3 (Conv2D)	(None, 25, 25, 128)	73856
max_pooling2d_2 (MaxPooling2	(None, 12, 12, 128)	0
conv2d_4 (Conv2D)	(None, 12, 12, 64)	73792
max_pooling2d_3 (MaxPooling2	(None, 6, 6, 64)	0
conv2d_5 (Conv2D)	(None, 6, 6, 32)	18464
max_pooling2d_4 (MaxPooling2	(None, 3, 3, 32)	0
dense_1 (Dense)	(None, 3, 3, 1024)	33792
dropout_1 (Dropout)	(None, 3, 3, 1024)	0
flatten_1 (Flatten)	(None, 9216)	0
dense_2 (Dense)	(None, 26)	239642

Total params: 458,362
Trainable params: 458,362
Non-trainable params: 0

Figure 3 - Document showing the model structure and parameters.

Finally, Figure 3 shows the mode layers that were used. The model alternated between convolutions and max pooling layers with a dropout layer to prevent overfitting. This network design was decided upon after rounds of feature engineering and model retraining. This network structure produced the best results. There is then a final dense output layer to create a one-dimensional array which represents the model's guess for the letter of the alphabet.

4. CONCLUSIONS AND FUTURE WORK

In the future, we hope to expand the database to include numbers, words, and the entire alphabet for HelpingHand as it can only translate 24 letters (it did not include J or Z because they required movement). Another feature that would be implemented is ensuring that images of signs can be translated accurately in any environment as the current model can only translate when there is no noise in the background. Finally, we hope to be able to translate video and in real-time through a mobile application for Android and iOS to make it more convenient than accessing HelpingHand through a web application.

REFERENCES

[1] Michelle Jay, "American Sign Language | Start ASL," *Start ASL*, 20-Mar-2020. (accessed Mar. 25, 2020).

[2] "Facts and figures," *Canadian Hearing Services*, 18-Nov-2013. (accessed Mar. 25, 2020).

[3] Sophia Waterfield, "Today is #ASLDay, but how is it used In today's society?," *Newsweek*, 15-Apr-2019. (accessed Mar. 25, 2020).

[4] Emily Matchar, "Sign Language Translating Devices Are Cool. But Are They Useful? | Innovation | Smithsonian Magazine," *Smithsonian Magazine*, 26-Feb-2019. (accessed Mar. 25, 2020).

[5] Devin Coldewey, "SignAll is slowly but surely building a sign language translation platform," *TechCrunch*, 14-Feb-2018. (accessed Mar. 25, 2020).

[6] Akash Nagaraj, "ASL Alphabet," 22-Apr-2018.