ASL Translator

Sarkis Bouzikian
University of California, San Diego
San Diego, California, USA
sbouzikian@ucsd.edu

Ryan Eveloff
University of California, San Diego
San Diego, California, USA
reveloff@ucsd.edu

Connor Kuczynski
University of California, San Diego
San Diego, California, USA
ckuczyns@ucsd.edu

ABSTRACT

We propose the development of an American Sign Language (ASL)-to-English interpreter that aims to assist deaf and hard-of-hearing individuals in understanding classroom interactions. Our goal is to create a runnable program capable of converting ASL signs captured through images, videos, and live feeds into real-time translated ASL letters. We intend to expand the program to include full words and phrases and integrate text-to-speech technology. Our research has successfully resulted in the development of a functional ASL-to-English interpreter capable of translating the full alphabet of ASL signs in real-time with 99.27% accuracy. This achievement demonstrates the potential of our program to enhance communication accessibility for deaf and hard-of-hearing individuals in classroom settings.

KEYWORDS

Machine Learning, Computer Vision, Real Time Computing, Classifier, Overfitting, Dimensionality reduction, Accessible Technology

ACM Reference Format:
Sarkis Bouzikian, Ryan Eveloff, and Connor Kuczynski. 2023. ASL Translator. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

1.1 Motivation

The use of remote learning, particularly at the collegiate level, offers increased flexibility and the ability to reach a larger audience. The long-term impact of COVID-19 on the education system remains uncertain, but it is clear that remote learning will continue to be an important avenue for college education. Remote learning allows for a more diverse student population, including those who previously faced financial constraints or had limited access to education. Additionally, the customization of the learning environment in remote settings benefits special needs students. In this project, our goal is to enable educators who are deaf or hard of hearing to use American Sign Language (ASL) for remote teaching, leveraging the existing remote learning technologies.

In many universities, lecture halls and classrooms are equipped with cameras for recording lectures. These cameras, along with microphones and screen capture capabilities, enable lecturers to record all important aspects of a lecture. This existing infrastructure can be utilized to facilitate ASL-based education without the need for an interpreter. We propose using computer vision and a classification neural network to interpret ASL signs from the recorded video data. By processing the video on a server, our ASL translator acts as an interface between the educator and the student, eliminating the need for a third party to be present.

1.2 Computer Vision and Classifier Neural Networks

Computer vision, in our context, refers to the ability to process images or frames from videos to identify and extract objects. In our case, the objects of interest are hands, which serve as the medium of communication in ASL. The challenge lies in filtering out noise from the images and accurately identifying the hand performing the sign. Computer vision techniques, such as edge detection and dimensionality reduction, are employed to preprocess the images and focus on the relevant hand information.

Classifier neural networks, on the other hand, are learnable functions that map input data (in our case, images of hands) to specific outputs (corresponding letters). We utilize supervised learning to train the model, where the network learns the relationship between input data and labeled examples. Convolutional neural networks, which excel in processing spatial data, are predominantly used in our network architecture. These networks leverage locality to learn features from the input images and are suitable for our ASL recognition task, as the exact position of the hand within the image is not relevant.

2 RELATED WORK

Scalable ASL Sign Recognition using Model-based Machine Learning and Linguistically Annotated Corpora (2018)[5]

A computational approach for sign recognition video similar to ours, here they model and recognize the different parts of the sign by focusing on the hand. They differ in their ability to recognize a 350-sign vocabulary. Additionally, they are able to recognize one-two-two-handed signs, whereas ours are trained on only one-hand signs. Another interesting difference is that the code uses conditional fields (CRF) instead of a pure classifier approach. CRF relaxes the assumption that samples are independent and finds global correlations.

A new 2D static hand gesture color image dataset for ASL gestures (2011)[2]

Use a similar multiple-sub-system approach to detect ASL gestures. First, the image is processed, reducing noise and extracting the hand position. We use a similar approach when using computer vision to identify features before they are imputed into a neural network. The most unique part of this paper is the novel dataset.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2425 images were used compared to our 87000; however, their hand gestures specifically targeted diverse light conditions from 5 different individuals.

American Sign Language Recognition Using Leap Motion Controller with Machine Learning Approach (2018)[4]

The study aimed to recognize all 26 letters and 10 digits. This is similar to what this achieved, where we were able to recognize individual letters. An interesting problem that both we and this study encountered was that static gesture recognition is possible by purely looking at one frame or image at a time. However, if we want to be able to recognize the gesture "i" from "j," the primary difference is in the motion of the pinky with the letter "j." These dynamic gestures require a method of classification beyond just classifying a single image. The novel part of this study is their prototype "Leap Motion Controller," which is a low-cost, small device designed to track hand and finger motion in 3D.

Comparing ANN, SVM, and HMM based Machine Learning Methods for American Sign Language Recognition using Wearable Motion Sensors (2019)[6]

Unlike [3], the proposed device for gesture recognition is worn by the signer. Its design derives from wearable motion sensors, whose sensor data is fed into a support vector machine. Support vector machines are another type of supervised learning that finds a hyperplane that maximizes the separation between the two classes of data. This differs from our approach of using a traditional neural network for classification. The most interesting part of the study is that they found artificial neural networks to be the most accurate method for classifying ASL words.

Real-Time American Sign Language Recognition System Using Surface EMG Signal (2015)[3]

Similar to [4], a support vector machine is used for classification by finding the hyperplane that best separates one class of data (each class is a specific gesture) from the rest of the classes. The novel technique of this paper is the use of surface electromyography, which measures the electrical activity of muscles. Electrodes are placed on the surface of the skin and then filtered and amplified before feeding into the SVM. This new way of representing gestures saw success in recognizing all 26 letters.

LeapASL: A platform for design and implementation of real-time algorithms for translation of American Sign Language using personal supervised machine learning models (2021)[9]

The Leap Motion Sensor is a device used to capture hand data quickly and accurately. They use a combination of Unity and Python to create an artificial neural network that can do real-time translation. Their approach to using supervised machine learning models is similar to this paper's. The most novel aspect of this paper is that this device learns individual words that are personalized to the signer themselves, allowing for higher levels of accuracy compared to our generation-based model that uses many different signers.

Sign-to-speech translation using machine-learning-assisted stretchable sensor arrays (2020)[8]

A yarn-based stretchable sensor array with a circuit board is another ASL translator that interprets signs through a wearable device similar to [3] and [5]. However, the novel part that differentiates this device from others is that it was able to interpret 660 sign language gestures in real time with minimal delay (less than one second). Additionally, they are able to obtain a very high accuracy of 98.63%. This study is one of the most successful attempts at creating an ASL translator that can be extended to hundreds of signs.

Hand Gesture Recognition of Static Letters American Sign Language (ASL) Using Deep Learning (2020)[7]

The study leverages deep learning models to statistically classify ASL signs. The technique used is similar to ours, where they use edge detection on images before using a convolutional neural network (CNN). The novel part of this study is that they were able to train their CNN very quickly (under an hour) compared to other approaches that use support vector machines and artificial neural networks. They limit themselves to only the 26 letters of the ASL alphabet and use static images, which fail to capture more dynamic signs.

The American Sign Language Lexicon Video Dataset (2008)[1]

A dictionary of ASL words using video was created as a method of documenting ASL signs. The problem they tackled is: given some ASL sign, what is the English translation? A computer vision system is proposed that will enable people to easily look up signs using a camera.

Real Time Conversion of American Sign Language to text with Emotion using Machine Learning (2022)[10]

This study also uses MediaPipe and a CNN model similar to our approach to building a system that can do real-time conversion of ASL to English. The most interesting aspect of this study is that they did not just take into account the hands of the signer but also the facial expression. This differs from many other studies (ours included) that use video as the method of interpreting the sign.

3 TECHNICAL MATERIAL

3.1 Methods

First, we need to experiment with the best technique for interpreting ASL gestures. Other approaches require custom hardware like electrodes on the skin or the use of physical sensor-based techniques. One such approach involves the use of the "Leap Motion Controller". However, we want to focus on primarily simple camera-based approaches. This is because the long-term target in mind is the camera stationed in lecture halls, which is already widely distributed with the software infrastructure to support it. An example of these cameras is shown in Figure 1.

Therefore, we need to first develop a method of correctly classifying static images of ASL gestures and measure the accuracy of these signs. Our minimum viable product (MVP) is the successful, high-accuracy detection of a few ASL gestures. The primary purpose of our MVP is to be able to show that yes, it is possible to extract hands out of a static image by filtering out noise and then correctly classifying the sign. The target goal for this project

Figure 1: Video camera used in lecture halls to record lecture

is to create a system that can correctly identify the entire 26-letter alphabet A-Z, but instead of static images, it uses live video.

Now we will briefly examine the two approaches we had to processing the image of the ASL gesture. Both approaches used the Tensorflow library to build a convolutional neural network (CNN) to do supervised learning. The CNN utilized five 2D convolution layers interleaved with maximum pooling to do nonlinear feature extraction. The activation function for each convolution layer is ReLU, which is a nonlinear operation used for learning more complex relationships between features in an image. Finally, there are two fully connected layers used to determine the likelihood of the given ASL gesture. Dropout is also performed after the convolution layers to help mitigate overfitting. The primary difference between the two approaches, however, is how the images were processed before being used for training the CNN.

Approach (1) involved modifying the images of gestures by overlaying them with a wireframe. The reasoning behind this is that the wireframe reduces the number of dimensions to make the neural network easier to train. It also attempts to prevent overfitting by filtering out noise from the image. The wireframe works by identifying 21 points on the hand with lines showing their connectivity. As seen below, the wireframe emphasizes the key aspects of the hand that differentiate different signs. The neural network is encouraged to learn patterns with these wireframes in mind.

The second approach uses the grayscale version of the raw image without the wireframe. The reasoning for gray scaling the image is that it reduces data, which helps speed up the training process, and it improves edge detection, which is important for correctly extracting hands. Correctly identifying the image, especially a noisy image, is essential for the neural network to be able to function at all. The reason is that the neural network is fed a subset of the image containing the squared hand instead of the whole image. We discuss the comparable success of these two approaches in the Results section.

To accomplish our target goal, we need a method of extracting frames from live video, performing grayscale conversion and detection, and then passing them forward through the neural network. One key problem to keep in mind is that live video requires

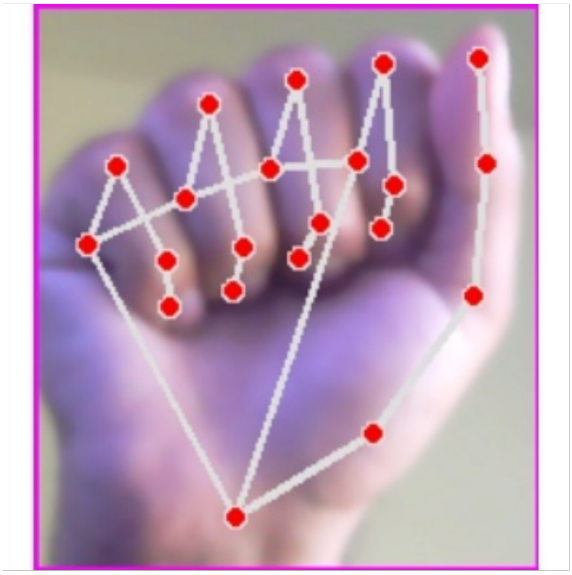

Figure 2: ASL gesture of the letter A overlaid with wireframe

real-time computing, so the size of our network needs to be manageable to get a reasonable response time. In the scenario where our system is used to translate ASL into audible English, the system must be responsive enough to facilitate this task. However, in the case where the video is recorded like a lecture, time constraints are more flexible and can perform computationally more intensive tasks to achieve better accuracy of the sign. With these ideas in mind, we decided on using MediaPipe as our method for quickly sampling frames from live video that are grayscale, then passing them through edge detection to detect the hand, and finally through the CNN.

3.2 Data Set

Early development of the system utilized a homebrew dataset for the ASL signs A, B, and C. The minimum viable product (MVP) for this system is a proof of concept that ASL sign translation for a single letter is possible given a set of static images of signs. We learned fairly quickly that scaling our dataset from just a few letters to the whole alphabet would require a new approach. Furthermore, if we want to be able to translate entire words, we need even more data, which would be extremely time-consuming to create. We were also concerned that training the data on just a limited number of people doing signs would create bias in the system overall. For example, hand shape and signing technique vary from person to person; therefore, we decided to look into using a more comprehensive dataset.

The dataset we decided on has 87,000 images with 29 classes. These classes consist of the 26 letters A–Z and three classes for space, delete, and The reason we used this dataset is the abundance of data, which helps the neural network better map images of signs

to their corresponding letters. One key problem we need to be wary of is overfitting the dataset. We discuss our findings in the quantitative results section. Additionally, we did not partition the dataset into training and testing because we instead used a live video feed of performers performing signs to verify correctness. The dataset can be found on Kaggle[1].

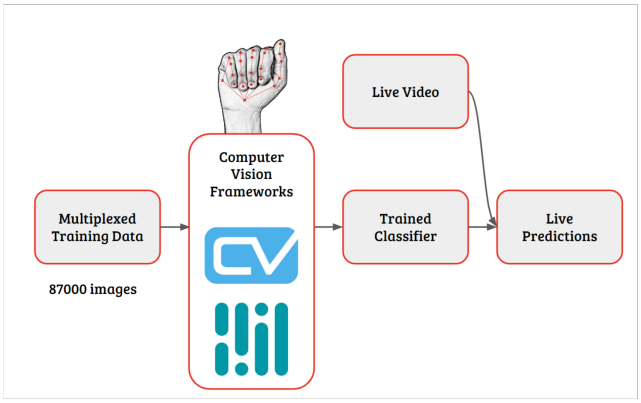

Figure 3: System Level Design using CV Zone and MediaPipe to enable live video parsing and image processing

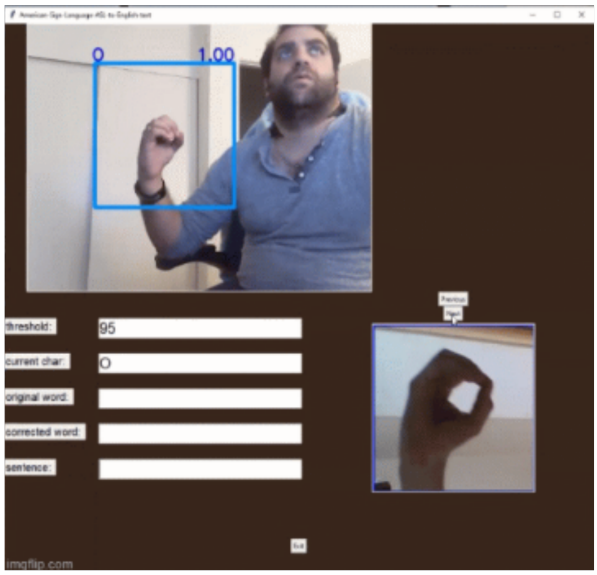

Figure 5: Sample of interface for real-time video translation of full ASL alphabet.

3.3 Results

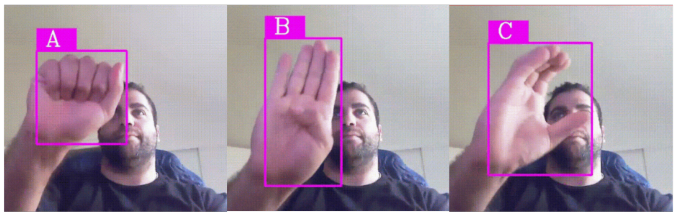

Figure 4: Live video translation of letters A, B, C

3.3.1 Qualitative. Both approaches (1) and (2) were capable of correctly extracting the hand from the image by squaring it while also correctly classifying the sign as A, B, or C. This is proof that the MVP is working as intended. Furthermore, we surpassed our goal of identifying static images. Live video translation of three letters is possible using MediaPipe. The homebrew dataset for training worked well enough that it was able to quickly and confidently classify each of the three signs. The largest problem with scaling up the number of classes is that the classifier needs to be able to learn distinct features that differentiate the different classes. The letters A, B, and C are all fairly diverse in finger position and orientation, making them easier to classify.

However, when scaling the dataset from 3 classes to 29 classes, approach (1) - the one overlaying a wireframe over the image - failed to produce consistently accurate results. The system would flip between two or more classes between frames. This meant that small changes in the sign orientation led to different sign colors. Furthermore, the system failed to meaningfully differentiate between similar signs. For instance, the signs "i" and "j" appear similar. One explanation for the system's inaccuracy is that the wireframe reduced the dimensionality, causing a failure of complex pattern learning.

Approach (2), utilizing the grayscale of the raw image data, performed much better and is what was used to train the system seen in Figure 5. This is very interesting because it goes against our original idea that the wireframe approach would work better. The grayscale images combined with edge detection were able to better extract the relevant features used for sign classification compared to just the wireframe. The wireframe was used when there were only 3 classes; however, it was scaled to 29 classes.

A proof-of-concept text-to-speech method was devised that accepts the translated letter in audible English. This method has tolerable latency, meaning that it is capable of keeping up with the signal introduced by the neural network. All models and figures are also available on our GitHub[2].

3.3.2 Quantitative. Our model showed high validation and training accuracy that scaled similarly, indicating that we did not overfit the data. The final accuracy achieved by our model is significant. However, it is important to note that the development of our model involved several iterations where we tuned different hyperparameters, such as the number of epochs and learning rate, to improve its performance.

[1]https://www.kaggle.com/datasets/grassknoted/asl-alphabet

[2]https://github.com/oplikos/American-Sign-Language-ASL-to-English-text#-american-sign-language-asl-to-english-text--

Figure 6 presents the loss and accuracy curves during the training process. As depicted in the figure, the loss decreases gradually over the epochs, while the accuracy steadily increases. Overall, the quantitative results validate the effectiveness of our model in achieving high accuracy on the given task. These results provide confidence in the robustness and reliability of our approach.

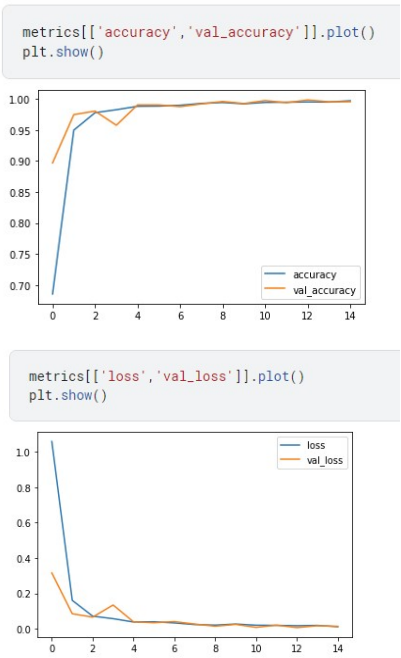

Figure 6: Accuracy (top) and loss (bottom) plots for the validation and training sets.

4 MILESTONES

Each team member is assigned specific responsibilities for each deliverable. Sarkis is responsible for wireframing, testing, live demo testing, and Raspberry Pi deployment. Ryan is responsible for still image IO, recorded video IO, live video IO, report/slides, and text-to-speech. Connor is responsible for the classifier, wireframing, and ASL word integration. This division of labor can be observed in table 1.

5 DELIVERABLES

5.1 Deliverable 1: Single Letter Classification (MVP)

We were able to successfully complete the MVP of extracting the hand from the image and then correctly identifying a letter. This was surpassed by creating two methods to preprocess the images (wireframe and grayscale with edge detection) that were capable of identifying three letters.

5.2 Deliverable 2: Full Alphabet Still Image

We successfully created a system that could identify 29 ASL gestures using a dataset of 87,000 images. One of the largest problems was finding a workable dataset that was accurate and robust enough to prevent overfitting. We resolved this problem using the dataset mentioned in Section 3.1.2. Another challenge faced was the limited computational resources. The problem with using such a large dataset is that it requires many hours to create the wireframe images from the given dataset. The most challenging part of this project was that we had to wait for training to be complete before we knew if the new adjustments made had improved accuracy, leading to a slow development time.

5.3 Deliverable 3: Live Video Sign Classification (Final Product)

The live video portion expanded on Deliverable 2. Once the model was trained on still images, the next challenging task was finding a method for sampling live video so that it could be broken down into frames. We were able to complete this task using MediaPipe. Live video classification was successful, as seen in the results section.

5.4 Deliverable 4: Writeup/Presentation

We successfully presented our project to a technical audience through a live presentation and also completed a video providing an overview of their overall project. We also considered stretch goals that would extend the project to a broader audience (like using the UCSD podcasting platform).

5.5 Deliverable 5: Text-to-Speech, Closed Captioning, Distribution (Stretch Goals)

We were able to successfully develop a method that is capable of translating letters into audible English. We also developed closed captioning using the 29-letter alphabet using a GUI. Distributing the system onto a server for others to use was not completed due to time constraints.

6 CONCLUSION

Overall, we were able to show that ASL translation is possible without the need for custom hardware. Computer vision techniques combined with a CNN proved to be successful methods of accomplishing this task, as also seen in studies [**?**] and [**?**]. Further work needs to be done to extend this to entire words, as shown in the study [**?**], with a moderate level of success. Additionally, further work will need to be done to distribute this system so that lecturers can use it to do live translation without the need for a third party to be present. Advances in machine learning technology will help break down communication barriers and ensure that deaf and hard of hearing people have equal access to the same opportunities that normal-hearing people have.

REFERENCES

[1] The american sign language lexicon video dataset. https://ieeexplore.ieee.org/abstract/document/4563181, 2008.
[2] A new 2d static hand gesture color image dataset for asl gestures. https://mro-ns.massey.ac.nz/handle/10179/4514, 2011.
[3] Real-time american sign language recognition system using surface emg signal. https://ieeexplore.ieee.org/abstract/document/7424365, 2015.

Deliverable	Date	Sarkis	Ryan	Connor
Deliverable 1	May 2, 2023	Wireframing	Still Image IO, Classifier	Wireframing, Classifier
Deliverable 2	May 9, 2023	Testing	Recorded Video IO	Classifier
Deliverable 3	May 25, 2023	Testing	Live Video IO	Classifier
Deliverable 4	May 30, 2023	Live Demo Testing	Report/Slides	Report/Slides
Deliverable5	Jun 12, 2023	Raspberry Pi Deployment	Text-to-Speech	ASL Word Integration

Table 1: Divison of labor.

[4] American sign language recognition using leap motion controller with machine learning approach. https://www.mdpi.com/1424-8220/18/10/3554, 2018.
[5] Scalable asl sign recognition using model-based machine learning and linguistically annotated corpora. https://www.sign-lang.uni-hamburg.de/lrec/pub/18005.pdf, 2018.
[6] Comparing ann, svm, and hmm based machine learning methods for american sign language recognition using wearable motion sensors. https://ieeexplore.ieee.org/abstract/document/8666491, 2019.
[7] Hand gesture recognition of static letters american sign language (asl) using deep learning. https://www.iasj.net/iasj/download/df4561780ca5a6ef, 2020.
[8] Sign-to-speech translation using machine-learning-assisted stretchable sensor arrays. https://www.nature.com/miscs/s41928-020-0428-6, 2020.
[9] Leapasl: A platform for design and implementation of real-time algorithms for translation of american sign language using personal supervised machine learning models. https://www.sciencedirect.com/science/misc/pii/S2665963822000434, 2021.
[10] Real time conversion of american sign language to text with emotion using machine learning. https://ieeexplore.ieee.org/abstract/document/9987362, 2022.