ASL Finger Spelling

Yendluri Lohith Jayasurya[1], Vivek Anand Thoutam[2], Saranya Somepalli[3], and Srinivas Siripurapu[4]

[1]lyendlur@asu.edu
[2]vthoutam@asu.edu
[3]ssomepa1@asu.edu
[4]ssiripu1@asu.edu

December 9, 2021

Abstract

This project creates an application that takes a video of a person signing the ASL alphabet as input and tries to guess what the person will say next. the video's alphabet Image processing and other similar methods to implement machine learning, Create an ASL Fingerspelling system that works in real time. application. Previously, the majority of the ASL alphabet was written in ASL. if the outcome is as expected, Individuals can be predicted by the application. that term contains alphabets. The software has been a high level of precision in letter recognition words as well

Keywords
Convolutional neural network,Deep Learning, Image processing, PosNet, Depth feature, FingerSpelling.

1 Introduction

Deaf people commonly converse using American Sign Language (ASL). It's a full language with grammar and linguistic elements that's similar to English. Hands and facial expressions are used to depict alphabets in ASL. Because of the widespread use of ASL, we decided to create an application that can take an ASL input and output the meaning.

The 26 alphabets of English are supported by American Sign Language using basic hand motions that are normally used for FingerSpelling. It's a method of borrowing alphabets from one language and using them in another. The letters 'J' and 'Z' are the only ones out of the 26 that are represented by static motions. Figure 1 shows a visual representation of these hand gestures.

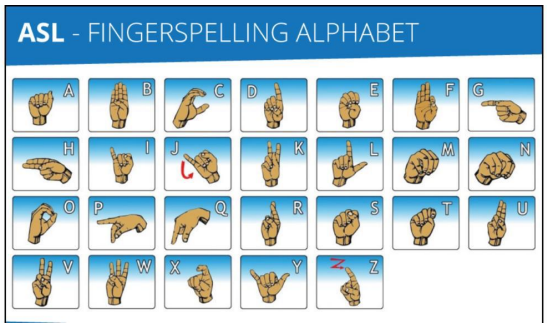

Figure 1: ASL Fingerspelling

Automated gesture detection could make computer-human interaction easier and more accessible, especially for the disabled. Human position and facial interpretation could also help with behavior analysis. Feature extraction techniques and machine learning models have been used to achieve fingerSpelling in the past. The method we picked employs image processing to extract a frame from a movie, which is then analyzed, and the image's complex features are extracted using a Convolutional Neural Networks (CNN) model. These characteristics are used to train the model and forecast what ASL means.

2 Model Architecture

This is an application that uses American Sign Language and has been trained using alphabets to guess what a gesture in a video means. Using the ASL alphabet videos, a model is built and trained. The wrist points are obtained by the palm detection technique using Posenet, a deep learning model. The section of the image with only the wrist is retrieved using the cropping method that was developed. After that, the

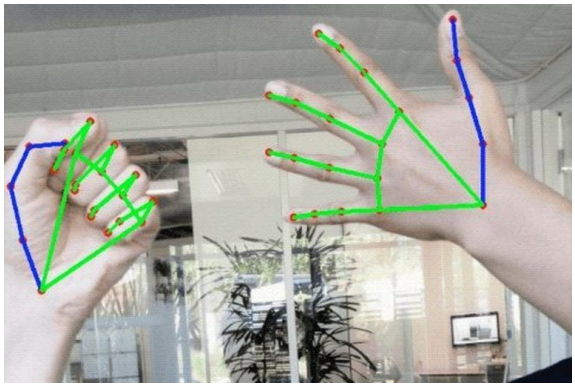

Figure 2:

CNN model is trained using these images. Similarly, we use films to teach the machine to understand language. Posenet assists in the creation of the keypoint series using photos from the videos. Separating the alphabets in the video clipping requires a distinct method called the segmentation algorithm, which has been created. Another algorithm is also built to merge the individual alphabets to produce the word.

3 How the Model Works

We carried out the project's tasks according to the following work categories:
1) Extracting frames from films containing either the alphabet or words to be predicted.
2) Posenet is used to create a keypoints json file from the extracted frames.
The output of PoseNet could be easily comprehended by referring to Figure2.
3) The key points.csv file is created from the json file.
4) Based on key points coordinates for the right and left wrists. The csv file and the frames extracted are cropped to remove all but the hand portion of the frames.

We'll get leftwristscore, leftwrist x coordinate, left wrist y coordinate, rightwristscore, rightwrist x coordinate, rightwrist y coordinate for each of the frames. We'll extract the x and y coordinates of that hand for that frame based on the leftwristscore and rightwristscore, whichever is greater. We'll make a box with x-d, x+d, y-d, y+d (where d is a constant that varies depending on the width and height of video frames) based on the x and y coordinates. Only the box portion of the frame will be segmented out to retrieve only the hand portion of a certain frame.

The cropped frames (just the hand section is

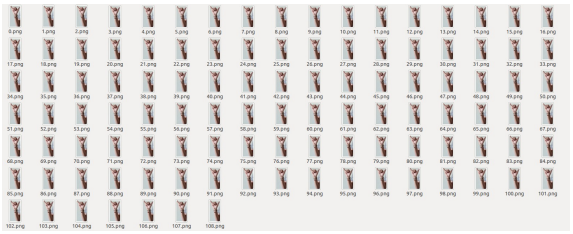

Figure 3:

clipped) for the ASL alphabet "Y" are presented in figure 3: 5) The cropped image is fed into a CNN model that has already been trained using Kaggle's ASL data and predicts the ASL alphabet/word: The cropped frames (hand part only) are fed to the CNN model, which predicts the alphabet using a python program. The following is a screenshot of the output of the python application for ASL alphabet detection:

4 ASL Word Algorithm

Using the Posnet, we will extract keypoints JSON from the frames of the ASL word video and convert them to CSV. As a result, we'll have all of the ASL word video frames' keypoints. From the keypoints csv file, this method will track the current and prior x and y coordinates of the Left or Right wrist. If the absolute value of the difference between current x and previous x coordinates in either hand exceeds a threshold value, an alphabet transition occurs, and all frames from the current frame number to the transition frame number are fed to the pretrained CNN model to determine that alphabet. This procedure is repeated until the video's last frames.
6) Precision and recall for both ASL Alphabet and Word detection have been provided based on the True value and projected value F1 score. The following is a snapshot of the output of the python program for the alphabets their true and predicted value shown in figure 4:

7) Finally, the result is saved as a CSV file. There are only two columns in the csv: projected value and true value.

Conclusions

We obtained a good grasp of how ASL may be translated into another language like English utilizing algorithms as a result of implementing the project. We received an awareness of the current research activities in the subject of language translation, as well as the various machine

learning techniques and implementations. To improve the accuracy, a variety of methods were investigated. Posenet, a deep learning model for analyzing poses by detecting body components, was used to teach us.

5 Task Completion

Table 1: Task Completion Table

S.No	Task	Assignee
1	Record Alphabet videos	Lohith
2	Palm Algorithm Using posenet	Srinivas,Vivek
3	Validation palm algorithm	Saranya
4	Configure CNN model	Srinivas,Vivek
5	Reporting Metrics	Lohith, Saranya
6	Record Words	All Members
7	Create Key points	Srinivas,Vivek
8	Segmentation Algorithm	Lohith, Saranya
9	CNN model for Alphabets	Srinivas,Vivek
10	Algorithm for words	Lohith, Saranya
11	Pipelining	Srinivas,Vivek
12	Calculating accuracy	Lohith, Saranya
13	Final Report	All members

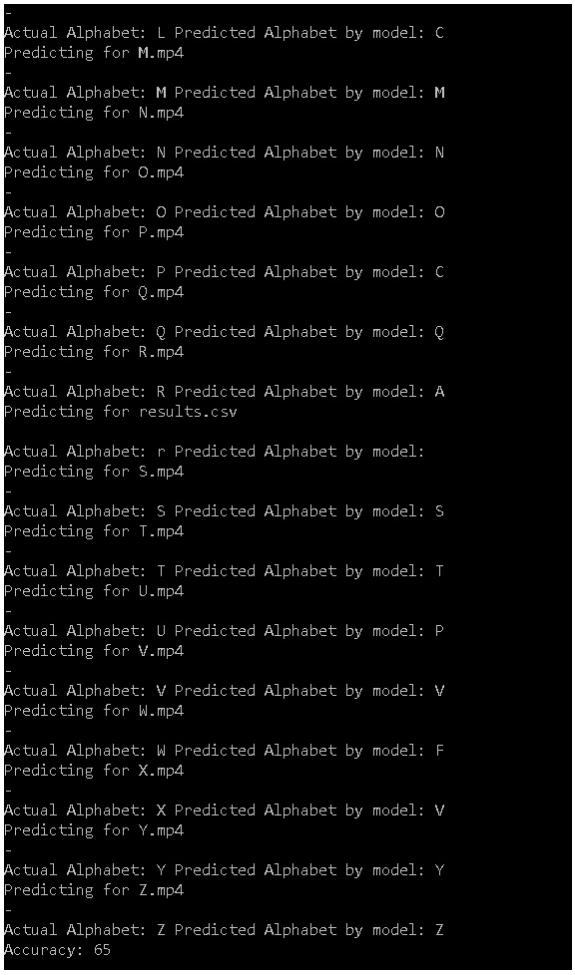

Figure 4:

6 ACKNOWLEDGMENT

We'd like to express our gratitude to Dr. Ayan Banerjee for his encouragement and assistance with our questions. We'd also want to thank the writers whose work has aided us in developing and understanding earlier study ideas. We'd like to express our gratitude to everyone of our team members for their efforts and contributions to the project.

References

[1]Lucas Rioux-Maldague and Philippe Giguère (2014). Using a Deep Belief Network, classify sign language fingerspelling from depth and color images. 10.1109/CRV.2014.20. Proceedings - Conference on Computer and Robot Vision, CRV 2014. 92-97.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. FeiFei. Imagenet: A large-scale hierarchical image database. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pages 248–255. IEEE, 2009

[3] Chris T., White J., Dougherty B. , Albright A. and Schmidt DC.," WreckWatch: Automatic Traffic Accident Detection and Notification with Smartphones ", International Journal of mobile network and application, Springer, Hingham, MA, USA., Vol. 16, Issue 3, PP. 285-303, March 2011.

[3] A. Bergamo, S. N. Sinha, and L. Torresani. Leveraging structure from motion to learn discriminative codebooks for scalable landmark classification. In Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, pages 763– 770. IEEE, 2013.

[4] M. Cummins and P. Newman. FAB-MAP: Probabilistic localization and mapping in the space of appearance. The International Journal of Robotics Research, 27(6):647–665, 2008.