ACTIVITY RECOGNITION USING STATISTICAL MODELS COUPLED WITH DIGITAL TECHNOLOGIES

Computer Vision Final Project: 16:198:534

Netid: hu33
Harish Udhayakumar
harish.udhayakumar@rutgers.edu
Project link

Abstract

This project aims to reduce the time and space complexity of the existing sign language recognition model. Sign language is a set of gestures using which mute people communicate. However, ordinary people don't understand it, which creates a communication gap that needs to be filled. Our work demonstrates a novel approach that overcomes these issues by using hand landmark detection over time as a feature representation fed into an LSTM model. This model uses less than half a million parameters, promising fast deployment into ubiquitous and digital settings for remote sign language detection with a validation accuracy of about 98%. This work aims to demonstrate the feasibility of deep learning technologies for detection of hand signs from unstructured home videos as a first step towards validation of whether statistical models coupled with digital technologies can be leveraged to aid in automatic behavioral analysis of Sign language.

Introduction

Sign language is the only tool to express what mute people need. Without an interpreter, communication between the deaf and others is a barrier that prevents them from having an everyday life. So what is sign language? Sign language is a set of hand gestures, facial expressions, and body motions representing words. Each country has different sign language gestures. This poses a significant difficulty for any trial to use modern technology in developing a competent and
efficient tool to help this community of deaf people. Also, this project aims to eliminate the need for a deaf person to have a human interpreter everywhere, with higher accuracy and computational efficiency.

Related work

In recent days, modern technologies in handheld and intelligent devices facilitate many processes in computer vision tasks. Research in sign language recognition has one of three approaches: sensor-based gloves, 3-D skeletons, or computer vision.

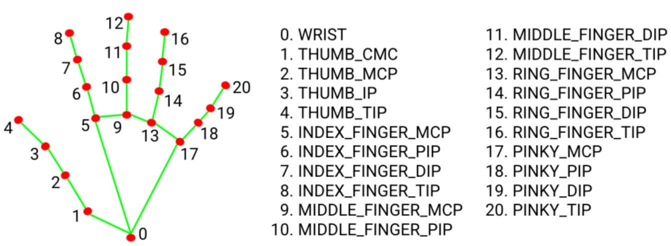
Although the performances in each paper cannot be directly compared due to a lack of standard datasets, they showcase the feasibility of using automated computer vision methods to detect sign language.

The first two approaches neglect facial estimations, which play a massive role in sign language recognition. On the other hand, computer-vision systems can capture the whole gesture, not to mention the mobility that differentiates them from glove-based systems.

For two decades, deep learning has been used in sign language recognition by researchers worldwide. Convolutional Neural Networks (CNNs) have been used for video recognition and achieved high accuracy last years.

Proposed Method

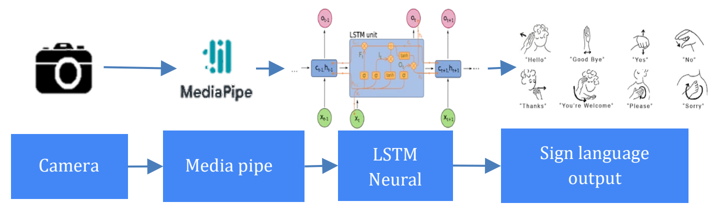
To design a model for use on a device, the number of model parameters must be low enough to run efficiently. Hence, we use the numerical coordinates of the detected hand landmarks as the primary feature representations. We use google's MediaPipe, to extract the hand coordinates [3]. The model provides the (x, y, z) coordinates of each of the 21 landmarks it detects on a hand. The x and y coordinate describes how far the landmark is on horizontal and vertical dimensions. The z coordinate provides an estimation of depth from the visual source. The three coordinates are on a scale of 0 to 1. Figure 2 shows the 21 hand landmarks that MediaPipe detects. We take the first 90 frames of a video and, for each frame, concatenate the detected coordinates of a set amount of landmarks into a single vector that is then fed as input into an LSTM model. We experiment with various subsets of landmarks provided by MediaPipe; we try with all 21 landmarks. Although we have good accuracy with a subset of landmarks, we get the best performance (highest accuracy) when we use all 21 landmarks for our prediction.



Model Architecture

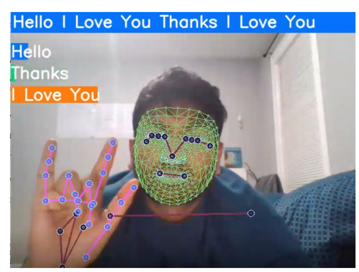
The network architecture has an LSTM layer with an output of 64 dimensions passed into a fully-connected layer with a sigmoid as an activation function. To reduce overfitting, we also insert a dropout layer with a dropout rate of 30%.

I experimented with other model architectures before selecting this model. It was found that adding more than one LSTM or fully connected layer did not make any significant change in performance; thus, we removed the additional layers to reduce the probability of model overfitting. I also experimented with the output dimensionality of the LSTM: I tried 8, 16, 32, and 64. It was found that using 32 and 64 performed very similarly, with 64 usually performing slightly better.

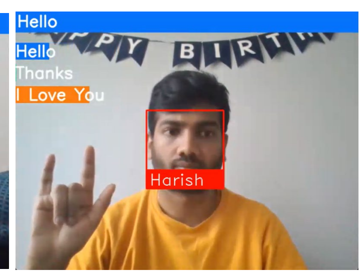


Results

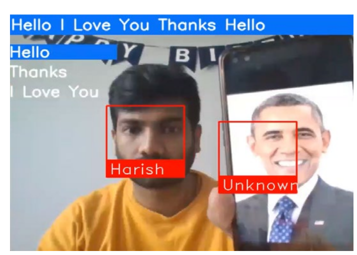
The model predicts with an accuracy of 98% on an average for data of 3 signs. Using CNN, the model requires more than a million parameters and is computationally intensive. However, using media pipe and LSTM, we reduce the parameters to more than half. Another estimate is the number of epochs; a superficial LSTM layer with the media pipe computations requires almost half the epochs to train the model than required for a traditional CNN or RNN.



Before Enhancement



With face recognition



Recognizing multiple faces

Model Type	Model Summary	Epochs	Accuracy
Simple LSTM	Total parameters: 562,115 LSTM Units: 64	60	99.9999%
Bidirectional LSTM	Total parameters: 1124227 Bidirectional units: 128	100	60%
Stacked LSTM	Total parameters: 716483 Stacked Units: (30 x 64), (30 x 120), 64, 64	100	98.99%

Table showcasing architectures and their accuracies based on LSTM layers.

Model Type	Model Summary	Epochs	Accuracy
Simple LSTM	Total parameters: 562,115 LSTM Units: 64	60	99.9999%
Bidirectional LSTM	Total parameters: 1124227 Bidirectional units: 128	100	60s%
Stacked LSTM	Total parameters: 716483 Stacked Units: (30 x 64), (30 x 120), 64, 64	100	98.99%

Enhancements

For face recognition, we use the 'face_recognition' library, which further increases the capacity of the sign language recognition model without adding further complexity to the neural network.

We read an image of the person to detect while recognizing the signs and we compare this numerical array with the visual feed and perform face detection. This doesn't need additional training.

Cons: With this enhancement, there is a lag in sign language detection compared to performing it without face recognition. However, it is a negligible lag.

References

[1] arXiv:2108.07917v1 [cs.CV] 18 Aug 2021 Anish Lakkapragada, Peter Washington, Dennis Wall

[2] L. Pigou, S. Dieleman, P. Kindermans, B. Schrauwen. "Sign Language Recognition using Convolutional Neural Networks"

[3] C.Lugaresi, J.Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee, et al., "Mediapipe: A framework for building perception pipelines," arXiv preprint arXiv:1906.08172, 2019.