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
Video 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 a deep learning 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 efficiency of deep learning technologies for detection of hand signs from unstructured live videos as a first step towards validation of whether statistical models paired with digital technologies can be leveraged to facilitate 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 communicating efficiently in everyday life. So what is sign language? Sign language recognition 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

Holistic visual appearance based approach and 2D human pose based Vision-based sign language recognition are two existing approaches. pose-based and appearance-based models achieve slightly comparable performances up to 62.63%.

These sign recognition approaches primarily consists of three steps: the feature extraction, temporal-dependency modeling and classification. Prior works first employ different hand-crafted features to represent static hand poses, such as SIFT-based features [4, 5, 6], HOG-based features [7, 8, 9] and features in the frequency domain [10, 11]. Hidden Markov Models (HMM) [12, 13] are

then employed to model the temporal relationships in video sequences. Dynamic Time Warping (DTW) [14] is also leveraged to handle differences of sequence lengths and frame rates. Classification algorithms, such as (SVM) Support Vector Machine [15], are used to describe the signs with the corresponding words. These approaches demonstrate the effectiveness of using human poses in the sign recognition task. Recent works also use 3D CNNs [28, 76] to capture spatial & temporal features together instead of encoding the
spatial and temporal information separately. However, these methods are only tested on small datasets.

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 1 shows the 21 hand landmarks that MediaPipe detects. We retrieve the first 90 frames of a video and, for each frame, append the coordinates of a set amount of landmarks into one vector that is then fed as input into an LSTM model. I experimented with various subsets of landmarks provided by MediaPipe and with various types of neural networks; 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.

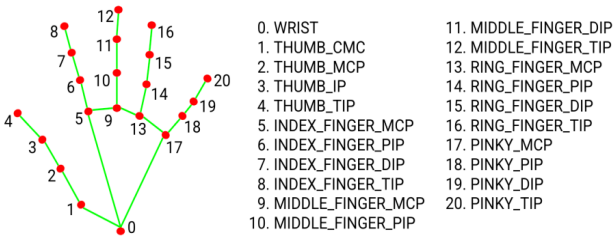


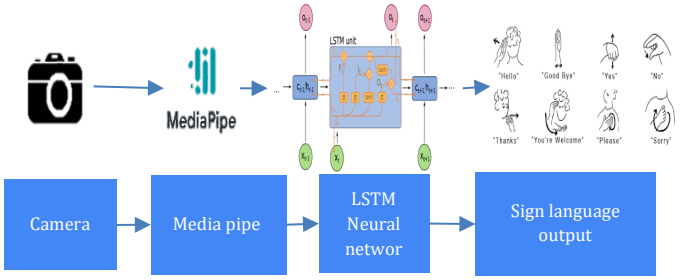
Figure :1

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%.

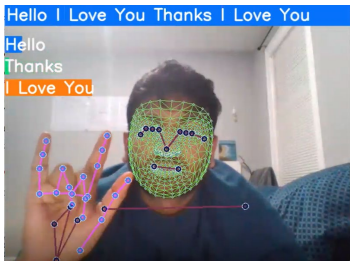
I experimented with other model architectures before selecting this model. It was found that adding more than one LSTM or fully connected layer did not make any significant change in performance; thus, we removed the additional layers to reduce the probability of model overfitting. I also experimented 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.

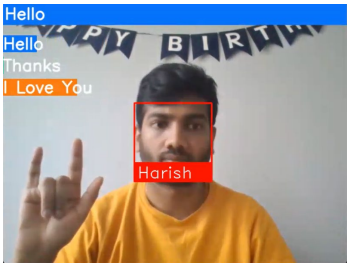


Results

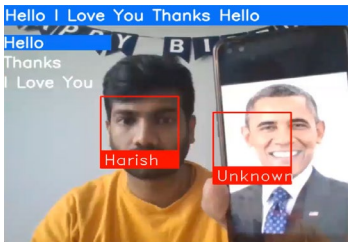
The model predicts with an accuracy of 98% on an average for data of 3 signs. Using CNN, the model requires more than a million parameters and is computationally intensive. However, using media pipe and LSTM, we reduce the parameters to more than half. Another estimate is is the number of epochs; a superficial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



(a) Before Enhancement



(b)With face recognition



Recognizing multiple faces

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	98%
Bidirectional LSTM	Total parameters: 1124227 Bidirectional units: 128	60	87%
Stacked LSTM	Total parameters: 716483 Stacked Units: (30 x 64), (30 x 120), 64, 64	60	90.99%

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 delay in computation when face recognition is involved due to accessing two different libraries in sign language detection compared to performing it without face recognition. However, In sign language usecase, this 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, F. Zhang, E. Uboweja, M. Hays, C. McClanahan, C.-L. Chang, J. Lee, M. G. Yong, et al., "Mediapipe – A framework for building the

perception pipelines," arXiv preprint arXiv:1906.08172, 2019.

[4] Q. Yang. Chinese sign language recognition using video sequence appearance modeling. In 2010 5th IEEE Conference about Industrial Electronics and Applications, pages 1537–1542. IEEE, 2010.

[5] A. Alsadoon , P. C. Prasad, F. Yasir, and A. Elchouemi. Sift based approach on bangla sign recognition. In 2015, IEEE 8th International Workshop about Computational Intelligence and Applications (IWCIA), pages 35–39. IEEE, 2015

[6] A. Tharwat, T. Gaber, A. E. Hassanien, M. K. Shahin, and B. Refaat. Sift-based arabic sign language recognition system. In Afro-european conference for industrial advancement, pages 359–370. Springer, 2015

[7] S. Liwicki and M. Everingham. Automatic recognition of fingerspelled words in british sign language. In 2009 IEEE computer society conference about computer vision & pattern recognition workshops, pages 50–57. IEEE, 2009.

[8] P. Buehler, A. Zisserman, and M. Everingham. Learning sign language by
watching tv (using weakly aligned subtitles). In 2009 IEEE Conference about Computer Vision and Pattern Recognition, pages 2961–2968. IEEE, 2009.

[9] H. Cooper, E.-J. Ong, N. Pugeault, and R. Bowden. Sign language recognition with sub-units. Journal on Machine Learning Research, 13(Jul):2205–2231, 2012.

[10] M. Al-Rousan, K. Assaleh, and A. Talaa. Video-based signer-independent arabic sign recognition using HMM. Applied Soft Computing, 9(3):990– 999, 2009.

[11] P. C. Badhe and V. Kulkarni. Indian sign language translator with gesture recognition algorithm. In 2015 IEEE International Conference on CG, CV and Information Security (CGVIS), pages 195–200. IEEE, 2015.

[12] T. Starner, J. Weaver, and A. Pentland. Real-time american sign language recognition utilizing desk and wearable computer based video. IEEE Transactions on pattern analysis & machine intelligence. 20(12):1371–1375, 1998.

[13] T. E. Starner. Visual recognition of american sign language using (HMM) hidden markov models. Technical report, Massachusetts Institite Of Tech Cambridge Dept Of Brain & Cognitive Sciences, 1995.

[14] J. F. Lichtenauer, E. A. Hendriks, and M. J. Reinders. Sign recognition using statistical dtw & independent classification. IEEE about oan Pattern Analysis & Machine Intelligence, 30(11):2040–2046, 2008.

[15] S. Nagarajan and T. Subashini. Static hand gesture recognition for sign alphabets with edge oriented histogram and multi class svm. International Journal on Computer Applications, 82(4), 2013.