MACHINE LEARNING BASED AMERICAN SIGN LANGUAGE TRANSLATOR

Aryyama Kumar Jana
Arizona State University
Tempe, USA
akjana@asu.edu

Lakshya Garg
Arizona State University
Tempe, USA
lgarg@asu.edu

Krishna Sriharsha Gundu
Arizona State University
Tempe, USA
kgundu1@asu.edu

Rupika Peela
Arizona State University
Tempe, USA
rpeela1@asu.edu

ABSTRACT

Sign language translators are used to translate hand gesture-based sign language to basic words/ alphabets or actions for efficient communication with deaf people. An automated translator can act as an efficient way to convert the sign languages that deaf people use so that it can be understood properly by normal human beings. Here, we have implemented state-of-the art Convolutional Neural Networks for translating ASL hand gestures to meaningful letters / words. These hand gestures can be recorded through our mobile app. Here we have used Posenet based pose estimation model for keeping track of hand positions and predicted the hand gesture using a trained CNN based machine learning model, which outputs the predicted alphabet that the hand-gesture in the frame corresponds to. These alphabets are then combined into respective words.

Author Keywords

Sign Language Detection, Machine Learning, Posenet, Palm Detection, Convolutional Neural Network

INTRODUCTION

ASL or American Sign Language is a completely designed hand-language which has almost every letter or action of the English alphabet that is required for communication by the deaf people, it was made with an attempt to cover as much of the communication as possible through hand-gestures. ASL is expressed by hand expressions and movements and used by deaf people to communicate amongst themselves as well as with other normal people.

The American sign language is not universal and people from different countries might have their own sign languages. It is said that ASL and is as old as 200 years and has been improving and upgrading since then. ASL is completely distinct from the English language, it has its own rules of grammar and word formation. ASL may vary from region to region. Fingerspelling is a part of ASL which uses finger gestures to depict actions/signs or alphabets. ASL is taught in different schools, especially schools dedicated to deaf people and other people who want to learn the language for their known one's who might be deaf, ASL is also taught by various organizations.

The goal of our project is to use Convolution Neural Networks and Pose Estimation using Posenet to determine sign languages by taking videos of people performing ASL gestures using a phone video camera and cropping out the hand-gestures using video preprocessing and image processing techniques using OpenCV and then depicting what the signs means.

TECHNOLOGIES USED

Laptop with below specifications has been used for this project:

OS: MACOS Big Sur Version 11.2.3

Processor: 2.2 GHz Quad-Core Intel Core i7

Disk: 250.79 GB SSD

Python 3.9

Running Instructions

1. Install Nodejs in system and related dependencies.
2. Configure dependencies as per Requirements.txt
3. Collect the alphabet ASL test videos and put them in the folder "./Videos_letters/Demo_videos"
4. Collect the alphabet ASL test videos and put them in the folder "./Videos_Words/Demo_videos"
5. Run "python3 ASLpredict.py"
6. Choose option 1 for letters and option 2 for words prediction.
7. results.csv are saved in respective video folders for letters and words

Brief Explanation of Terminologies

1. **Open CV:** Open-Source Computer Vision Library Is a library of programming functions for real-time computer vision applications
2. **PoseNet:** It is a deep-learning model in TensorFlow which allows one to estimate and track poses by detecting key coordinates of various body parts like elbows, hips, wrists, knees, and ankles, etc.
3. **NodeJS:** We used NodeJS to handle our API calls to PoseNet Model and get key point estimates for our videos for each time-unit or frame of the

video. Node.js is a cross-platform runtime environment in JavaScript.

4. **Tensorflow**: An open-source machine learning library, it is used for a varied range of tasks, primarily used for training the Convolution Neural Networks and in Deep Learning. Some of the most important applications of Tensorflow include RankBrain by Google.
5. **CNN:** Convolution Neural Network is a Deep Learning algorithm which takes images as input, extract features of importance out of these images which helps it in distinguishing one image from the other. Convolution Neural Networks are better and faster as compared to the other classification algorithms because of faster preprocessing. Convolution Networks can be thought of as neurons in a human brain. Convolution Neural Networks are used to capture the spatial and temporal dependencies in any image. The Convolution Neural Networks consists of an Input layer, several hidden layers (i.e., any middle layers are basically called hidden layers) and output layers, outputs generally have an activation function.

IMPLEMENTATION

Collecting Videos for Testing

Every member in the group had to record videos for all English alphabets (around 6-7 each) and a total of 40 words (10 words each) for testing purpose.

Implementation of Palm Detection and cropping

Palm was identified from the body in the video using various key coordinates and a rectangular box was drawn around the detected hand for each frame in the videos. The portion of the hand inside the rectangular frame was cropped and saved in the Headframes folder.

Generation of Keypoint Time series from the video

We implemented the pose-net Nodejs source-code provided by the professor along the with the converttocsv.py file also provided by the professor. First frames were generated from the video and saved in a folder. From every frame, keypoints were estimated using the Posenet model and added to a json file. Finally, this json file which consisted of the keypoints time series (or keypoints pertaining to every time frame) was converted to csv and saved in keypoints_Posenet folder.

CNN model for Prediction

We used a CNN model trained on the given Kaggle data set to estimate the ASL gesture from the cropped-out hand frame which was obtained using the palm detection and cropping algorithm.

Alphabet Prediction Algorithm

We are taking the statistical "Mode" of the predicted alphabets for all the time frames to predict the alphabet.

Word Prediction Algorithm

We are estimating change of alphabets in a word by keeping a track of the change in distance between various keypoints in the hand and arm obtained through Posenet. Now for each duration of letter, we are finding the statistical MODE to predict each letter. Finally we concatenate the obtained MODEs to get the predicted word.

TASK DISTRIBUTION

The contributions of various group members in the project are given below:

1. Aryyama Kumar Jana
   - (i) CNN Model Implementation and configuration
   - (ii) Implementation of Posenet to obtain wrist points
   - (iii) Development of palm detection algorithm
   - (iv) Palm cropping algorithm
   - (v) Validation of palm detection and palm cropping algorithm on video taken by us.
   - (vi) Implementation of alphabet recognition algorithm
   - (vii) Implementation of word detection algorithm
   - (viii) Implementation of Posenet to develop a key point time series from the video in json and CSV.
   - (ix) Created segmentation algorithm to separate out each letter in a word
   - (x) Algorithm to combine individual letters to get the final predicted word.
   - (xi) Generation of classification report consisting of precision recall NFL score for both letters and words.
   - (xii) Debugging, testing and execution of the above code.
   - (xiii) Video recording for 7 alphabets
   - (xiv) Video recording for 10 words
   - (xv) Making of Project Report & Documentation
   - (xvi) Code Integration

2. Lakshya Garg
   - (i) Implementation of Posenet to obtain wrist points
   - (ii) Implementation of Palm Cropping algorithm
   - (iii) Generation of Key point time series from the video.
   - (iv) Project Report & Documentation
   - (v) Video Recording for 5 alphabets
   - (vi) Video recording for 10 words

(vii) Debugging and execution of the code.
(viii) Code Testing

3. Krishna Sriharsha Gundu
   (i) Conversion of Key points to csv
   (ii) Initial stage Implementation of Posenet and related debugging
   (iii) Video Recording for 7 alphabets
   (iv) Video recording for 6 words
   (v) Word Detection Algorithm
   (vi) Execution of the code.
   (vii) Code Testing

4. Rupika Peela
   (i) Initial stage Implementation of Posenet and related debugging
   (ii) Video Recording for 7 alphabets
   (iii) Video recording for 6 words
   (iv) Execution of the code.
   (v) Code Testing

RESULTS

The following figures clearly exhibit that our implemented palm detection and cropping algorithm has worked very efficiently in cropping out the hand frames from each video frame. We are getting almost >90% accuracy in cropping out the hand.

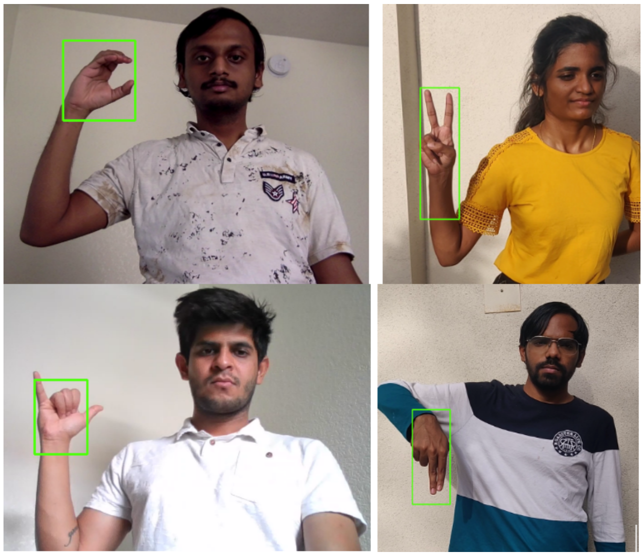
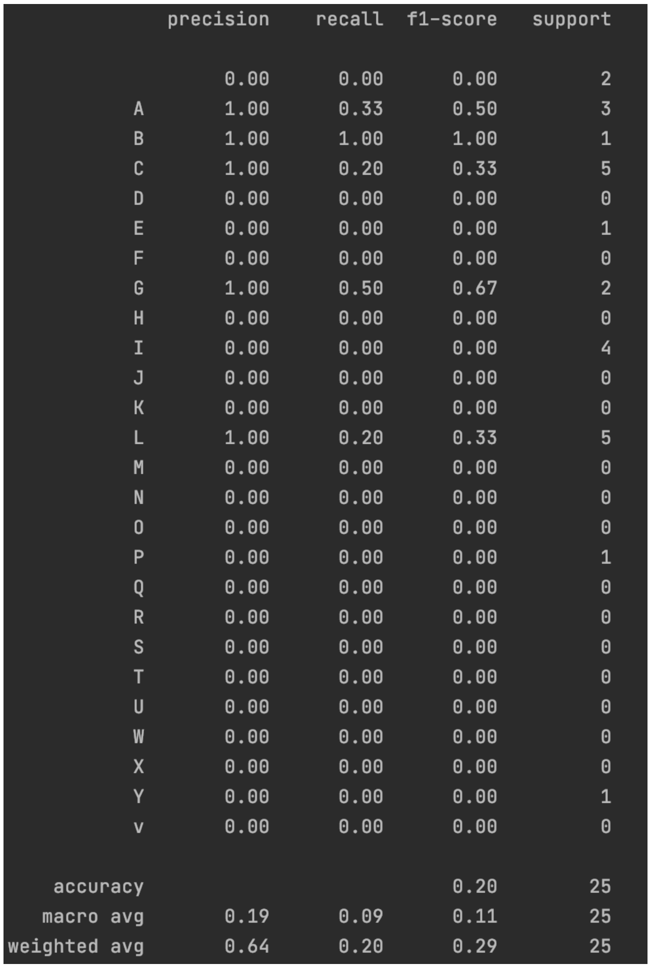


Fig 1

We tested our model on all alphabets and got a total accuracy of 20% and weighted average overall F1 score of 29% as is evident from the classification report screenshot attached in Figure 2.



```
              precision    recall  f1-score   support

                   0.00      0.00      0.00         2
           A       1.00      0.33      0.50         3
           B       1.00      1.00      1.00         1
           C       1.00      0.20      0.33         5
           D       0.00      0.00      0.00         0
           E       0.00      0.00      0.00         1
           F       0.00      0.00      0.00         0
           G       1.00      0.50      0.67         2
           H       0.00      0.00      0.00         0
           I       0.00      0.00      0.00         4
           J       0.00      0.00      0.00         0
           K       0.00      0.00      0.00         0
           L       1.00      0.20      0.33         5
           M       0.00      0.00      0.00         0
           N       0.00      0.00      0.00         0
           O       0.00      0.00      0.00         0
           P       0.00      0.00      0.00         1
           Q       0.00      0.00      0.00         0
           R       0.00      0.00      0.00         0
           S       0.00      0.00      0.00         0
           T       0.00      0.00      0.00         0
           U       0.00      0.00      0.00         0
           W       0.00      0.00      0.00         0
           X       0.00      0.00      0.00         0
           Y       0.00      0.00      0.00         1
           v       0.00      0.00      0.00         0

    accuracy                           0.20        25
   macro avg       0.19      0.09      0.11        25
weighted avg       0.64      0.20      0.29        25
```

Fig 2

The accuracy metrics for words videos are attached below. Overall accuracy for word prediction is 30.09%.

S.No.	Actual Word	Predicted Word	Accuracy
1	FIG	BIB	33%
2	BAG	CAG	66%
3	WON	GID	0%
4	WIN	KIK	33%
5	BAD	BAC	66%
6	SUN	GIG	0%
7	ACE	ACA	66%
8	HE	HA	50%
9	BED	BCL	33%
10	RUN	RIG	33%
11	YOU	GIG	0%
12	CAB	CAA	66%

13	CAFE	CABC	50%
14	RAT	IKD	0%
15	CAGE	CAGC	75%
16	BE	BA	50%
17	POT	CGC	0%
18	AGE	AGA	66%
19	LEG	HIG	33%
20	HAT	HAG	66%
21	RAT	IKD	0%
22	TO	DO	50%
23	HOW	CLC	0%
24	OH	LL	0%
25	SHE	YPY	0%
26	SO	LY	0%
27	TRY	KLB	0%
28	UP	UK	50%
29	WHY	LGL	0%
30	WAS	YAS	66%
31	SAW	YOY	0%
32	OK	KK	50%
33	OF	OI	50%
34	ME	MS	50%
35	HIT	AGL	0%
36	HIM	CFA	0%
37	HI	CI	50%
38	GUM	YAY	0%
39	EYE	ELP	33%
40	AT	SL	0%
41	DO	LL	0%

CONCLUSION

The accuracies are low because of the following constraints:

1. Ambient light
2. Low training data
3. Noise
4. Distance from camera
5. Low Image resolution

We got a higher accuracy for word prediction as many accurate letters were repeated in words.

REFERENCES

1. E. Anderson Moryossef A., Tsochantaridis I., Aharoni R., Ebling S., Narayanan S. (2020) Real-Time Sign Language Detection Using Human Pose Estimation. In: Bartoli A., Fusiello A. (eds) Computer Vision – ECCV 2020 Workshops. ECCV 2020. Lecture Notes in Computer Science, vol 12536. Springer, Cham
2. https://www.kaggle.com/grassknoted/asl-alphabet
3. https://www.nidcd.nih.gov/health/american-sign-language
4. https://heartbeat.fritz.ai/real-time-human-pose-estimation-with-tensorflow-js-51da62b68d3a