

ASL FingerSpelling

Ankit Sharma

Arizona State University
Tempe, AZ, USA
ashar263@asu.edu

Avinash Patnala

Arizona State University
Tempe, AZ, USA
apatnala@asu.edu

Avirup Biswas

Arizona State University
Tempe, AZ, USA
abiswa15@asu.edu

Sai Madhuri Molleti

Arizona State University
Tempe, AZ, USA
smolleti@asu.edu

ABSTRACT :

This project implements an application that takes a video of a person showing the ASL alphabet sign as input and will try to predict the alphabet being displayed in the video. Techniques such as Image processing and machine learning are implemented to develop a real time ASL Fingerspelling application. Once most of the ASL alphabet is getting predicted correctly then the application can also predict individual alphabets in that word. The Application has a significant accuracy in recognizing letters and words.

INDEX TERMS: Deep Learning, Image processing, Convolutional neural network, Depth feature, FingerSpelling, PosNet

INTRODUCTION:

The American Sign Language (ASL) is prominently used to communicate among the deaf communities. It is a complete language similar to English which has grammar and linguistic features. ASL alphabets are represented using hands and facial expression. The wide usage of ASL has inspired us to develop this application which can take the ASL input and produce the corresponding meaning as output.

The American Sign Language supports the 26 alphabets in English by using simple hand gestures which are otherwise used for FingerSpelling. It is a form of borrowing alphabets from one language to another. Of the 26 letters, 24 are represented using static gestures the others being 'J', 'Z'. An illustration of these hand gestures can be seen in Figure 1.



Figure 1

The automated recognition of gestures may facilitate the interaction between computer and humans and could be an alternative way of interaction with the system, especially for the disabled community. Interpreting the human poses and faces could also enhance analysis of human behaviour.

The fingerSpelling has been implemented previously using different techniques of

feature extraction and employing machine learning models. The approach that we have chosen makes use of image processing for extracting the frame from a video, which is analysed and using Convolutional Neural Networks (CNN) model we extract the complex features from the image. These features are used to train the model and predict the meaning of ASL.

PROJECT SETUP:

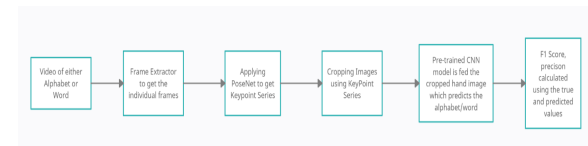
We have developed our project with the following components -

- ASL Data on Kaggle <https://www.kaggle.com/grassknoted/asl-alphabet>
- TensorFlow
- Node 8
- Posenet
- Python3.8
- Keras

SYSTEM ARCHITECTURE:

This is an application using American Sign Language that is trained using the alphabets such that it can predict what the gesture in video corresponds to. A model is created and trained using the ASL alphabet videos. The palm detection algorithm uses the Posenet which is a deep learning model to obtain the wrist points. By developing the cropping algorithm, the part of the image which has only the wrist is extracted. These images are then used to train the CNN model. Similarly we develop the model to comprehend words using videos to train the model. Posenet helps develop the keypoint series from the images obtained from the

videos. A separate algorithm called the segmentation algorithm is developed that can separate the alphabets in the video clipping. Another algorithm to combine the individual alphabets to form the word is also implemented.

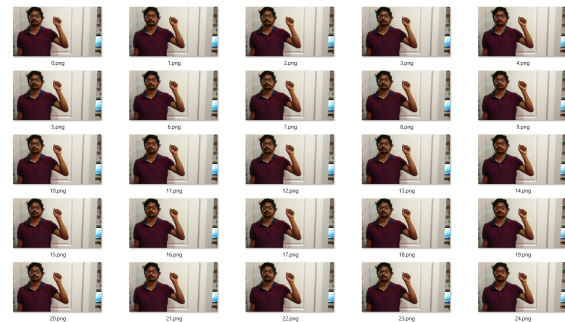


IMPLEMENTATION:

We have implemented the tasks of the project based on the category of the work which is as follows:

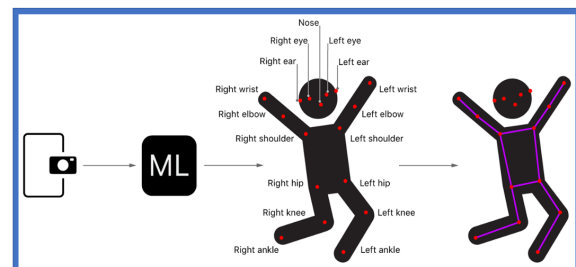
1) Extracting the frames from the videos of either alphabet or words which should be predicted.

For example for ASL alphabet “A” the frames are shown below on a sample video:



2) Keypoints json file is obtained from the extracted frames using Posenet

The output of PoseNet could be easily comprehended by referring to Figure2.




```
Python 3.9.4 (tags/v3.9.4:1f2e308, Apr 4 2021, 13:27:16) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:\Users\abhiswa15\ASL-Fingerspelling-Prediction-main\prediction.py =
Choose a recognition model:
1. Alphabets
2. Words
Choose an option: 1
Running for A.mp4
-----
test_data:../alphabetsframes/demo/A.mp4
-----
True Value: A Prediction: A
```

A screenshot of the output of python program for ASL word detection is shown below:

```
Python 3.9.4 (tags/v3.9.4:1f2e308, Apr 4 2021, 13:27:16) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:\Users\abhiswa15\ASL-Fingerspelling-Prediction-main\prediction.py =
Choose a recognition model:
1. Alphabets
2. Words
Choose an option: 2
Running for ARE.mp4
-----
Selection of Frame is Done

Predicting alphabets from frames extracted.
-
-
-
generating keypoint timeseries for the word from posenet.csv
-
-
True Value: ARE Prediction: ARE
Running for DIG.mp4
-----
Selection of Frame is Done

Predicting alphabets from frames extracted.
-
-
-
generating keypoint timeseries for the word from posenet.csv
-
-
True Value: DIG Prediction: DIG
```

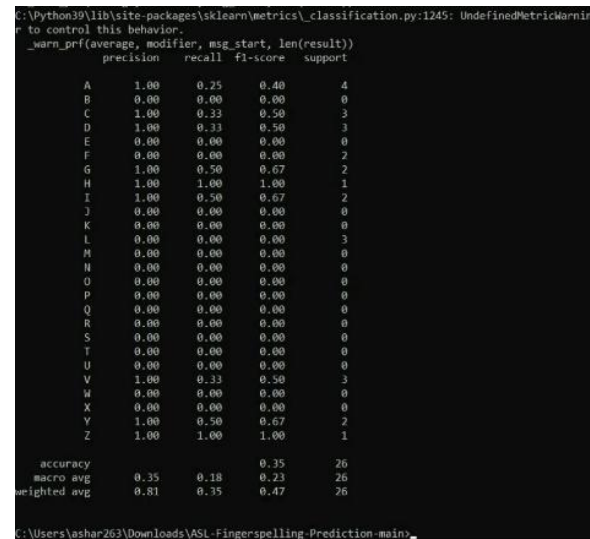
ASL Word Detection Algorithm:

We will get keypoints JSON of the ASL word video's frames using the Posnet and we will convert the keypoints JSON to CSV. Thus we will have all the keypoints of the ASL word video's frames. This algorithm here will track current and previous x and y coordinates of Left wrist or Right wrist from the keypoints csv file. If the **absolute value of the difference in current x coordinate and previous x coordinate** or the **absolute value of the difference in current y coordinate and previous y coordinate** in any hand goes beyond a threshold value then a transition of a alphabet takes place and all the frames from the current frame number till the transition frame number are being fed to the pretrained CNN model to determine that alphabet. This process continues till last frames of the video.

6) Based on the True value and predicted value F1 score, precision, recall has been

displayed for both ASL Alphabet and Word detection

A screenshot of the output of python program for F1 score, precision, recall of ASL Alphabet detection is shown below:



	precision	recall	f1-score	support
A	1.00	0.25	0.40	4
B	0.00	0.00	0.00	0
C	1.00	0.33	0.50	3
D	1.00	0.33	0.50	3
E	0.00	0.00	0.00	0
F	0.00	0.00	0.00	2
G	1.00	0.50	0.67	2
H	1.00	1.00	1.00	1
I	1.00	0.50	0.67	2
J	0.00	0.00	0.00	0
K	0.00	0.00	0.00	0
L	0.00	0.00	0.00	3
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	0
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
V	1.00	0.33	0.50	3
W	0.00	0.00	0.00	0
X	0.00	0.00	0.00	0
Y	1.00	0.50	0.67	2
Z	1.00	1.00	1.00	1
accuracy			0.35	26
macro avg	0.35	0.18	0.23	26
weighted avg	0.81	0.35	0.47	26

A screenshot of the output of python program for F1 score, precision, recall of ASL word detection for words is shown below:

	precision	recall	f1-score	support
ALA	0.00	0.00	0.00	1
ARE	0.00	0.00	0.00	0
DIG	1.00	1.00	1.00	1
accuracy			0.50	2
macro avg	0.33	0.33	0.33	2
weighted avg	0.50	0.50	0.50	2

7) Finally a CSV file named result.csv has been created with only two column predicted value and true value

Screenshot of the result.csv for ASL alphabet detection can be seen below:

	A	B	C
1		pred	TRUE
2	0	A	A
3	1	V	B
4	2	C	C
5	3	D	D
6	4	F	E
7	5	L	F
8	6	G	G
9	7	H	H
10	8	I	I
11	9	C	J
12	10	V	K
13	11	A	L
14	12	D	M
15	13	A	N
16	14	A	O
17	15	L	P
18	16	C	Q
19	17	F	R
20	18	Y	S
21	19	L	T
22	20	I	U
23	21	V	V
24	22	G	W
25	23	D	X
26	24	Y	Y
27	25	Z	Z

Screen of the result.csv for ASL word detection:

word	pred	TRUE
A		
V		
C		
D		
F		
L		
G		
H		
I		
C		
V		
A		
D		
A		
L		
C		
F		
Y		
L		
I		
V		
G		
D		
Y		
Z		

LINKS:

1) Alphabet and Word videos: The video recordings of 26 alphabets and 10 words made by each of our team members is at the following link. The link has 4 folders with alphabet and word videos of each member of the team.

<https://drive.google.com/drive/folders/1MCuiF7p4rZJD6E1V0TEvjVekucMXpEDD?usp=sharing>

2) Demo links:

(80%) before pipelining input/output and with only ASL alphabet detection-

<https://youtu.be/G09-q2FL2Ik>

(100%) after pipelining input/output and with ASL alphabet and word detection-

<https://youtu.be/vrjuAeqgwQs>

TASK COMPLETION:

S.no	Task	Assignee
1	Record 26*4 ASL alphabets videos.	Ankit, Avinash, Avirup, Madhuri
2	Develop palm cropping algorithm using wrist points obtained from posenet.	Ankit, Avirup
3	Validating palm detection algorithm	Avinash, Avirup
4	Configuring the 3D CNN model	Madhuri, Avinash
5	Reporting F1 Metrics	Ankit, Avirup
6	Record 10*4 word	Ankit,

	videos using ASL	Avirup, Avinash, Madhuri
7	Developing Keypoint Series	Avirup, Avinash
8	Implementing Segmentation Algorithm	Ankit, Madhuri
9	Using 3D CNN to recognize Alphabets	Avirup, Ankit, Avinash
10	Developing algorithm to recognize words	Ankit, Avinash
11	Automation pipelining	Avinash, Avirup
12	Calculating the word recognition accuracy	Madhuri, Avirup
13	Final Report	Ankit, Avirup, Avinash, Madhuri

CONCLUSION :

By implementing the project, we have gained a clear understanding of how ASL can be translated into another language such as English by using algorithms. We have gained insights into the present research activity taking place in this field of language translation, developed a strong understanding of the different machine learning algorithms and their implementations. A range of different approaches were explored to improve the accuracy. We have learnt using Posenet

which is a deep learning model that analyzes poses using detection of body parts.

ACKNOWLEDGMENT:

We would like to thank Dr. Ayan Banerjee for encouraging us and helping out with our queries. We would also acknowledge the authors whose research has helped us develop as well as understand previous research ideas. We would like to thank our team members for their efforts and contribution to the project.

REFERENCES:

[1] Rioux-Maldague, Lucas & Giguère, Philippe. (2014). Sign Language Fingerspelling Classification from Depth and Color Images Using a Deep Belief Network. Proceedings - Conference on Computer and Robot Vision, CRV 2014. 92-97. 10.1109/CRV.2014.20.

[2]“https://web.stanford.edu/class/ee368/Project_Autumn_1617/Reports/report_ranmuthu_ewald_patil.pdf”

[3]“https://en.wikipedia.org/wiki/American_Sign_Language”