ASL Fingerspelling

Aditi Shashank Joshi
1222838916
ajoshi64@asu.edu

Jayagauri Adinath Sunke
1219715991
jsunke@asu.edu

Sai Sruthi Mareedu
1220282531
smareed1@asu.edu

Sree Valindha Maddineni
1223698866
smaddin5@asu.edu

***Abstract*— In this project, the application designed takes a video input which is of a person displaying/signing the ASL alphabets and it will try to predict and guess the alphabet shown in the video. Image processing and machine learning techniques are used to develop the real time ASL Fingerspelling application. ASL words can also be detected by detecting the individual letters in the word, and this prediction can be made accurately once most of the alphabets are getting predicted correctly. The Application has a significant accuracy in recognizing letters and words.**

Keywords*— *Deep Learning, Image processing, Convolutional neural network, Depth feature, Fingerspelling, PoseNet

I. Introduction

According to the World Health Organization, 466 million individuals worldwide suffer from hearing loss that is incapacitating. Hearing loss is clearly a more widespread condition than most people believe. There are several forms of hearing loss, including sensorineural hearing loss, conductive hearing loss, and mixed hearing loss, the first of which is the most frequent. There might be a variety of reasons for something to exist. Aging, excessive noise exposure, injuries, and pre-existing health issues such as diabetes and viral infections are just a few examples. If current trends continue, by 2050 there will be approximately 900 million people with similar issues.

The American Sign Language (ASL) is widely utilized by the deaf community to communicate. It is a full language with grammar and linguistic traits, similar to English. Hands and facial expressions are used to portray ASL alphabets. Because of the widespread use of ASL, we decided to create an application that can accept an ASL input and provide the matching meaning as an output. The 26 alphabets of English are supported by the American Sign Language by employing basic hand motions that would otherwise be utilized for Fingerspelling. It's a method of borrowing alphabets from one language and applying them to another. 24 of the 26 letters are represented by static motions, with the letters 'J' and 'Z' being the exceptions. Figure 1 shows an illustration of these hand motions.

Figure 1 ASL Fingerspelling

Automated gesture recognition might improve computer-human interface and provide an alternate means of interacting with the system, particularly for the handicapped people. Human postures and faces might be used to aid in the understanding of human behaviour. Fingerspelling has already been achieved using a variety of feature extraction techniques and machine learning models.

We designed this program based on this premise, which essentially allows a person to capture video using American Sign Language. Before passing these movies through the pipeline to accomplish their eventual purpose, we undertake some pre-processing on them. After the recordings are filmed and generated, a python script is used to transform them to frames [4]. The number of frames generated for a video is determined by the video's quality and length. A five-second video usually yields 150 frames. They are then sent into an algorithm that determines the main locations in the video picture, allowing for data gathering and analysis. We transmit these critical spots to a pretrained convolutional neural network after they've been identified.

Convolutional Neural Networks are a sort of Deep Learning Algorithm that is mostly used to analyse visual pictures. CNN essentially employs a set of learnable weights and biases to give priority to pictures and then use these to discriminate between them. The filters are

specified for general purposes, but if we have a large enough number of training samples, a CNN can learn these filters as well. It's important to note that the construction of a CNN is based on the organization of neurons in the human brain, and it's nothing more than a completely linked collection of neurons from one layer to the next. The American Sign Language dataset was used to train our convolutional neural network.

When a video is provided, it is first divided into two second videos, one for each letter. The whole video size may range from 6 to 8 seconds depending on the length of the word. This is how prediction works: the letters are joined to produce a word.

We may check the test and train loss for our initial training and adjust the convolution neural network settings to better fit our objectives. Dropout layers might be added, which could be quite useful in some situations. Dropout is a crucial strategy for improving the performance of neural networks, and it also saves a lot of time when compared to training with other layers. Dropout is just disregarding a few of the weights during training the neural network, if we examine attentively. In other words, these neurons are not taken into account in either the forward or backward passes.

People with hearing loss would benefit greatly from such an application, which has a wide range of applications. We essentially give an end-to-end solution for deaf people's convenience and comfort. This method recognizes American sign language and generates a text representation of the words said in the videos. It has the potential to assist individuals in overcoming obstacles in their daily lives if performed effectively and with great precision. If scaled up, the application might theoretically convert a natural language into a text format that is easy to read and comprehend [2].

II. Technology Used

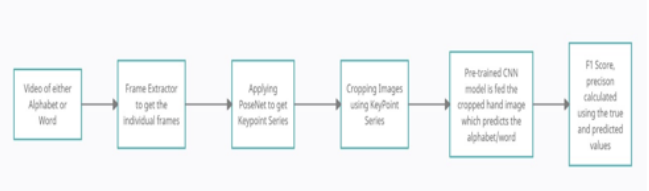
For the development of this project the following technologies and components were used:

1. ASL Data on Kaggle.
2. Node 8
3. PosNet
4. Python 3.8
5. TensorFlow
6. Keras

III. System Architecture

This is an application that uses American Sign Language and has been trained using the alphabets to guess what a gesture in a video represents. The ASL alphabet clips are used to develop and train a model.

To extract the wrist points, the palm identification technique uses PoseNet, a deep learning model. The section of the image that just includes the wrist is retrieved by constructing the cropping algorithm. After that, the CNN model is trained using these frames. Similarly, we use films to teach the machine to understand language. PoseNet assists in the creation of the key point 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 method is also built to merge the separate alphabets to produce the word.

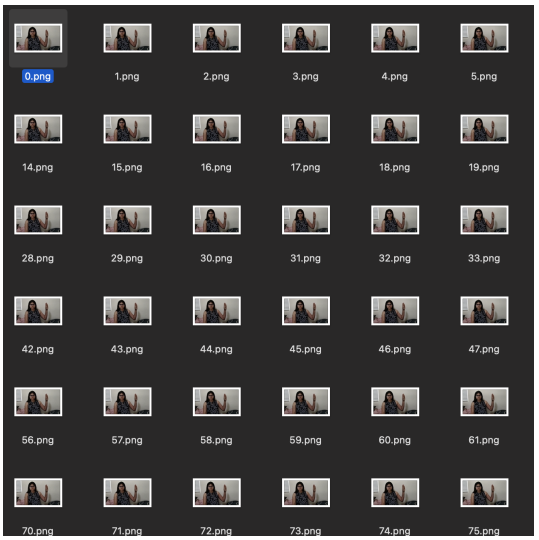


Implementation:

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 “B” 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.

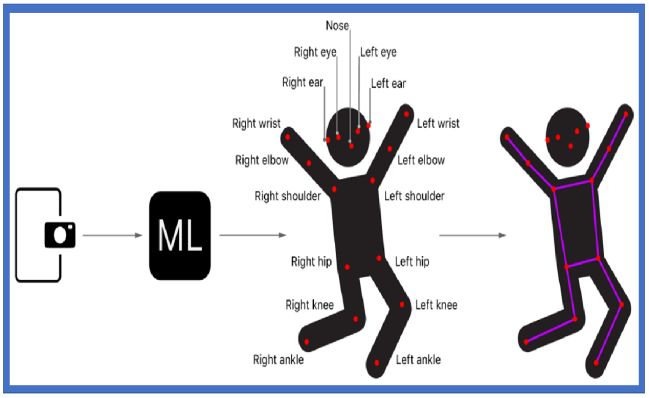
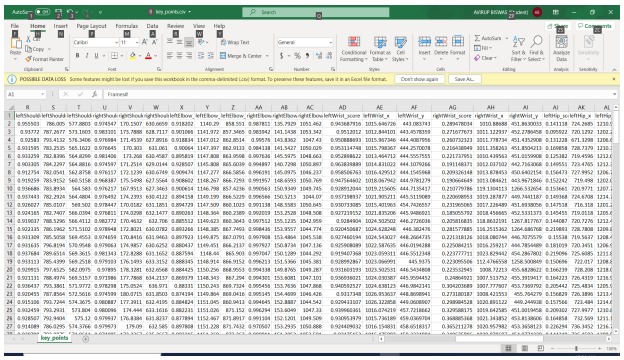


Figure 2

For example, for ASL alphabet “A” the key points generated are as follows:

3) key_points.json file is converted to key_points.csv file

For example, for ASL alphabet “A” the key points CSV is shown below:



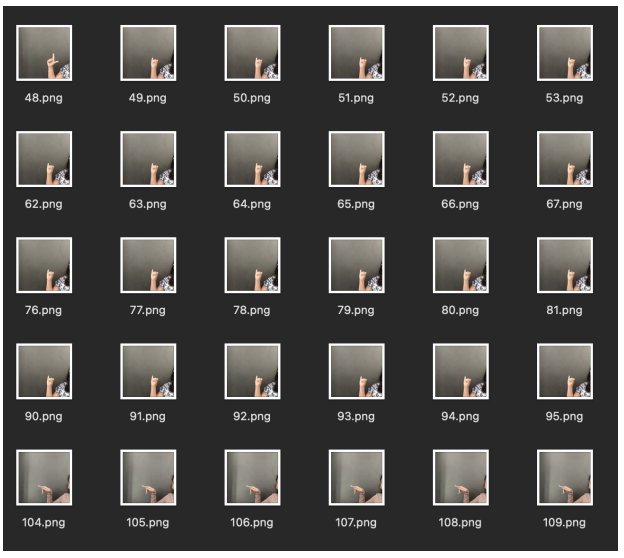
4) Based on the Right wrist and Left wrist coordinates from keypoints.csv file and the frames extracted is being cropped to get only the hand part from the frames.

Segmentation Algorithm:

For Each of the frames we will get leftwristscore,leftwrist x coordinate, left wrist y coordinate, rightwristscore, rightwrist x coordinate, rightwrist y coordinate. Based on the leftwristscore and rightwristscore whichever is greater we will get x and y coordinates of that hand for that frame. Based on the x and y coordinates we will create a box with x-d, x+d, y-d, y+d (Where d is a constant and it vary depending on video frame’s width and height).We will segment out only the box portion from the frame to get only the hand portion from a particular frame.

For example for ASL alphabet “B” the cropped frames (only the hand portion is being cropped) are shown below:

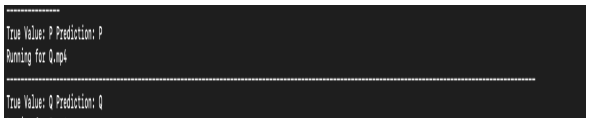
For example for word “LIP” using ASL the cropped frames(only the hand portion is being cropped) are shown below:



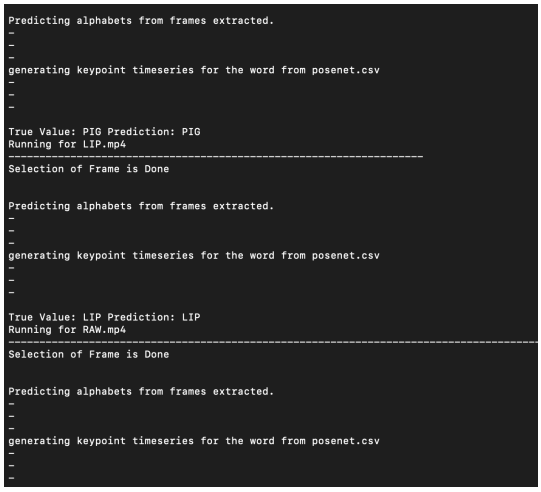
5) Cropped image is being fed to a pre trained CNN model (already trained using the ASL data from Kaggle) **and it predicts ASL alphabet/word:**

A python program has been created that feeds the cropped frames(hand part only) to the CNN model and the CNN model predicts the alphabet.

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



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



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 word detection for words is shown below:

	precision	recall	f1-score	support
GYPL	0.00	0.00	0.00	1
LIP	1.00	1.00	1.00	1
PIG	1.00	1.00	1.00	1
RAW	0.00	0.00	0.00	0
accuracy			0.67	3
macro avg	0.50	0.50	0.50	3
weighted avg	0.67	0.67	0.67	3

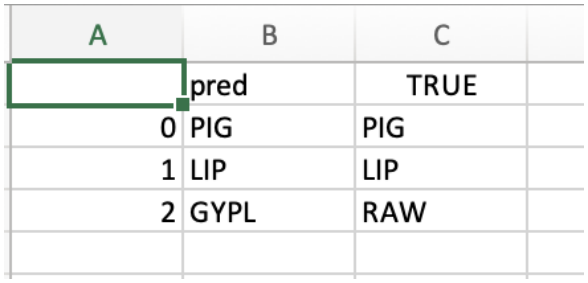
A screenshot of the output of python program for F1 score, precision, recall of ASL Alphabet detection is shown below:

	precision	recall	f1-score	support
	0.00	0.00	0.00	4
A	0.00	0.00	0.00	2
B	0.00	0.00	0.00	1
C	0.00	0.00	0.00	1
D	0.00	0.00	0.00	0
E	0.00	0.00	0.00	0
F	0.00	0.00	0.00	0
G	1.00	0.14	0.25	7
H	0.00	0.00	0.00	0
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	1.00	0.33	0.50	3
M	0.00	0.00	0.00	0
N	1.00	1.00	1.00	1
O	0.00	0.00	0.00	0
P	1.00	0.50	0.67	2
Q	1.00	1.00	1.00	1
R	0.00	0.00	0.00	1
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	0.00	0.00	0.00	0
W	0.00	0.00	0.00	2
X	0.00	0.00	0.00	0
Y	1.00	1.00	1.00	1
Z	0.00	0.00	0.00	0
r	0.00	0.00	0.00	0
s	0.00	0.00	0.00	0
accuracy			0.25	28
macro avg	0.24	0.15	0.18	28
weighted avg	0.61	0.25	0.32	28

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

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



	A	B	C
		pred	TRUE
0		PIG	PIG
1		LIP	LIP
2		GYPL	RAW

Task Completion

S.no	Task	Assignee
1	**Record 26*4 ASL alphabets videos.**	**sruthi, Aditi, Gauri Valinda**
2	**Develop palm cropping algorithm using wrist points obtained from posenet.**	**Sruthi, Gauri**
3	**Validating palm detection algorithm**	**Sruthi, Aditi**
4	**Configuring the 3D CNN model**	**Aditi , Valinda**
5	**Reporting F1 Metrics**	**Sruthi , Valinda**
6	**Record 10*4 word videos using ASL**	**sruthi, Aditi, Gauri Valinda**
7	**Developing Keypoint Series**	**Aditi Gauri**
8	**Implementing Segmentation Algorithm**	**Sruthi, Valinda**
9	**Using 3D CNN to recognize Alphabets**	**Aditi , Gauri , Sruthi**
10	**Developing algorithm to recognize words**	**Valinda, Gauri**
11	**Automation**	**Sruthi ,**
	pipelining	**Aditii**
12	**Calculating the word recognition accuracy**	**Valinda, Aditi**
13	**Final Report**	**sruthi, Aditi, Gauri Valinda**

Conclusion

We obtained a clear knowledge of how ASL may be translated into another language such as English utilizing algorithms by completing the project. We gained an awareness of the current research activities in the subject of language translation, as well as a thorough comprehension of the various machine learning algorithms and their implementations. To increase the accuracy, a variety of ways were investigated. Posenet, a deep learning network that analyzes poses by detecting body components, was used to teach us.

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] 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”

Links

Link to the word videos and alphabet videos.

https://drive.google.com/drive/folders/11d3reK_MslP3As9ahB4qrk8UnH66BLYe

Demo : https://youtu.be/KLW9tm_jPGc