ASL FINGERSPELLING PROJECT

Nithish nnallamj@asu.edu

Navitha nvemired@asu.edu

Shivani sdevula4@asu.edu

Kavya Sree Bachina kbachina@asu.edu

ABSTRACT

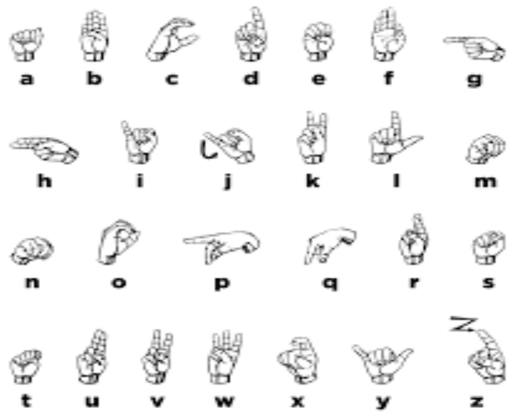
The idea of this project is to develop an application which can capture the videos of a person showing ASL Alphabet signs and predicting the correct ASL sign.Machine learning algorithms and image processing libraries are implemented to develop ASL Fingerspelling application and inferring the type of ASL Alphabet signs which are being displayed.After predicting the ASL alphabet correctly application can also predict individual alphabets in that particular word.This application has good accuracy in predicting the right set of ASL Alphabet

INDEX TERM

Fingerspelling, Posenet, Deep learning, Image processing, Convolutional Neural Network, Depth feature.

INTRODUCTION

Communication among Deaf communities is made easy by American Sign Language(ASL). It has all the components of a complete language similar to other languages. ASL consists of linguistics and grammar.ASL alphabets are represented using hands and facial expressions. This wide usage of ASL has made us to develop this application which takes the ASL signs as input and produce the meaning as the output.26 Alphabets in English are supported by ASL using simple hand gestures. This borrowing of Alphabet makes the task of developing Fingerspelling application easier. Of the 26 letters, 24 are represented using static gestures the others being ‘J’, ‘Z’ which are represented dynamically. An illustration of these hand gestures can be seen in Figure 1.



Recognition of gestures automatically will improve the interaction between the computer and people and be a great help for the deaf communities.This implementation can be used to analyse human behaviour.

The fingerSpelling project can be implemented in many ways possible.But the approach that we have chosen makes use of image processing libraries for extracting the frames from a video of a person showing gestures, which are then processed and using Convolutional Neural Networks (CNN) model we extract the complex features from the image. These features are then used to train the Machine learning model and predict the ASL signs.

PROJECT SETUP

We have used the following components for our project:

- ASL Data on Kaggle https://www.kaggle.com/grassknoted/asl-alphabet
- Node 8
- TensorFlow
- Posenet
- Keras
- Python 3.8

SYSTEM ARCHITECTURE

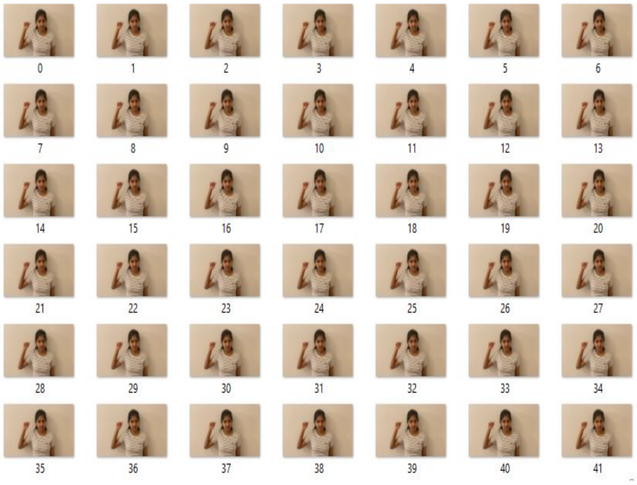
This application predicts the gestures in the videos and predicts the correct ASL sign corresponding to that Alphabet.A Machine learning model is developed and trained using the ASL alphabet videos. Posenet(Palm detection algorithm) a deep learning model is used to obtain the wrist points from the frames. By developing and implementing the cropping algorithm, the part of the image consisting of only the wrist is extracted. These images are then used to train the Convolutional Neural Network model. Later similarly we update the existing model to comprehend words using videos which are taken to train the model .Posenet helps develop the keypoint series from the images which are obtained from the videos. Segmentation algorithm is then developed which separates the alphabets in the video clipping. Later another algorithm is developed to combine the individual alphabets to form words.

IMPLEMENTATION

We have implemented the following tasks for our project:

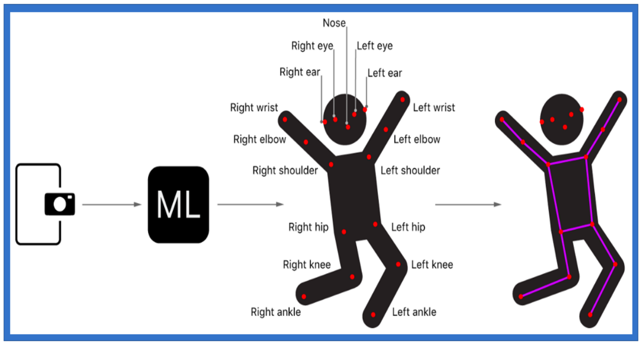
1) **Extraction of frames from the videos of either the alphabet or words which are supposed to be predicted.**

For ASL alphabet "A" the following frames are shown below based on a sample video:

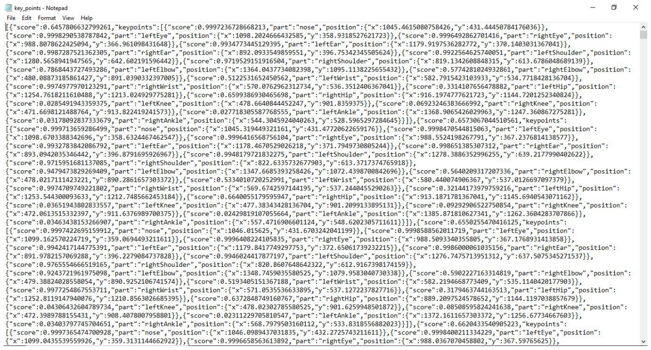


2) **Keypoints json file is obtained by using Posenet from the extracted frames**:

The output of the PoseNet model can be easily understood by referring to Figure2.



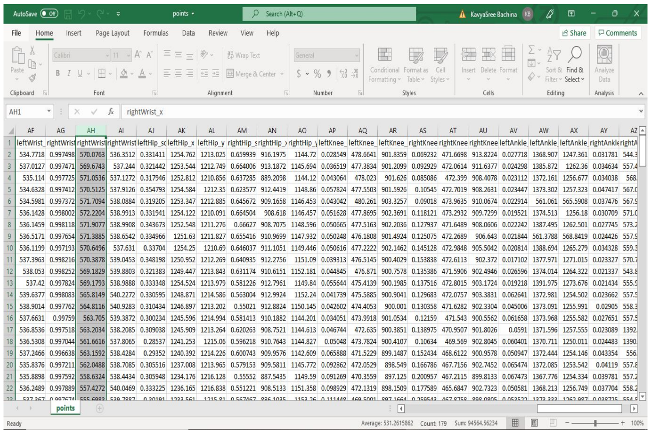
For ASL alphabet "A" the following key points generated are as follows:



```
[{"score":0.6457806632799261,"keypoints":[{"score":0.9997236728668213,"part":"nose","position":{"x":1045.4615080758426,"y":431.44450784176036}},
{"score":0.9998290538787842,"part":"leftEye","position":{"x":1098.2024666432585,"y":358.9318527621723}},{"score":0.9996492862701416,"part":"rightEye","position":
{"x":988.8078622425094,"y":366.961098431648}},{"score":0.9934773445129395,"part":"leftEar","position":{"x":1179.9197536282772,"y":370.1403031367041}},
{"score":0.9987287521362305,"part":"rightEar","position":{"x":892.0933549859551,"y":396.75342345505624}},{"score":0.9922564625740051,"part":"leftShoulder","position":
{"x":1280.5658941947565,"y":642.602191596442}},{"score":0.9719529151916504,"part":"rightShoulder","position":{"x":819.1342608848315,"y":613.6786048689139}},
{"score":0.7868443727493286,"part":"leftElbow","position":{"x":1364.0437734082398,"y":1095.1138225655432}},{"score":0.5774281024932861,"part":"rightElbow","position":
{"x":480.08873185861427,"y":891.0390332397005}},{"score":0.5122531652450562,"part":"leftWrist","position":{"x":582.7915423103933,"y":534.7718428136704}},
{"score":0.9974977970123291,"part":"rightWrist","position":{"x":570.0762962312734,"y":536.3512406367041}},{"score":0.3314107656478882,"part":"leftHip","position":
{"x":1254.7618211610488,"y":1213.02492977528l}},{"score":0.6599386930465698,"part":"rightHip","position":{"x":916.1974777621723,"y":1144.720125234O824}},
{"score":0.0285491943359375,"part":"leftKnee","position":{"x":478.6640844452247,"y":901.8359375}},{"score":0.06923246383666992,"part":"rightKnee","position":
{"x":471.6698121488764,"y":913.822419241573}},{"score":0.027718305587768555,"part":"leftAnkle","position":{"x":1368.9065426029963,"y":1247.360867275281}},
{"score":0.03178092837333679,"part":"rightAnkle","position":{"x":544.3045924040263,"y":528.5965297284645}}]},{"score":0.6573067044510561,"keypoints":
[{"score":0.999713659286499,"part":"nose","position":{"x":1045.319449321161,"y":431.47720622659176}},{"score":0.9998470544815063,"part":"leftEye","position":
{"x":1098.6703388342696,"y":358.6324467462547}},{"score":0.9996416568756104,"part":"rightEye","position":{"x":988.5524198267791,"y":367.2376814138577}},
{"score":0.9932783842086792,"part":"leftEar","position":{"x":1178.4670529026218,"y":371.7949730805244}},{"score":0.998651385307312,"part":"rightEar","position":
{"x":893.0942035346442,"y":396.8791695926967}},{"score":0.9948179721832275,"part":"leftShoulder","position":{"x":1278.3886352996255,"y":639.2177990402622}},
{"score":0.9715951681137085,"part":"rightShoulder","position":{"x":822.6335732677903,"y":613.3717374765918}},
{"score":0.9479473829269409,"part":"leftElbow","position":{"x":1347.6685393258426,"y":1072.4398700842696}},{"score":0.5640209317207336,"part":"rightElbow","position":
{"x":478.0217111423221,"y":890.2861657303372}},{"score":0.5334010720252991,"part":"leftWrist","position":{"x":580.440074906367,"y":537.0126697097379}},
{"score":0.9974709749221802,"part":"rightWrist","position":{"x":569.6742597144195,"y":537.2440455290263}},{"score":0.32144173979759216,"part":"leftHip","position":
{"x":1253.544300093633,"y":1212.7485662453184}},{"score":0.6640055179595947,"part":"rightHip","position":{"x":913.1871781367041,"y":1145.6940543071162}},
{"score":0.03651943802833557,"part":"leftKnee","position":{"x":477.38343428136704,"y":901.2099133895131}},{"score":0.09292906522750854,"part":"rightKnee","position":
{"x":472.0613515332397,"y":911.6376989700375}},{"score":0.02429819107055664,"part":"leftAnkle","position":{"x":1385.871810627341,"y":1262.3604283707866}},
{"score":0.03463438153266907,"part":"rightAnkle","position":{"x":557.4716906601124,"y":548.6202305711611}}]},{"score":0.6550255470416125,"keypoints":
[{"score":0.9997422695159912,"part":"nose","position":{"x":1046.015625,"y":431.6703242041199}},{"score":0.9998588562011719,"part":"leftEye","position":
{"x":1099.162570224719,"y":359.0694493211611}},{"score":0.9996408224105835,"part":"rightEye","position":{"x":988.5093340355805,"y":367.1768931413858}},
{"score":0.9942417144775391,"part":"leftEar","position":{"x":1179.8417749297753,"y":372.65061739232215}},{"score":0.9986000061035156,"part":"rightEar","position":
{"x":891.9782157069288,"y":396.2279084737828}},{"score":0.9946024417877197,"part":"leftShoulder","position":{"x":1276.7475713951312,"y":637.5075345271537}},
{"score":0.9765554666519165,"part":"rightShoulder","position":{"x":820.8607648642322,"y":612.9167398174159}},
{"score":0.924372196197509,"part":"leftElbow","position":{"x":1348.7459035580525,"y":1079.9583040730338}},{"score":0.5902227163314819,"part":"rightElbow","position":
{"x":479.38824028558054,"y":890.9252106741574}},{"score":0.5193405151367188,"part":"leftWrist","position":{"x":582.2194668773409,"y":535.114042017790З}},
{"score":0.9977254867553711,"part":"rightWrist","position":{"x":571.0535536633895,"y":537.1272237827716}},{"score":0.3179463744163513,"part":"leftHip","position":
{"x":1252.8119147940076,"y":1210.8563026685395}},{"score":0.6372848749160767,"part":"rightHip","position":{"x":889.2097524578652,"y":1144.1197038857679}},
{"score":0.04306432604789734,"part":"leftKnee","position":{"x":478.02302785580525,"y":901.6259948501872}},{"score":0.08508595824241638,"part":"rightKnee","position":
{"x":472.3989788155431,"y":908.4078007958801}},{"score":0.02311229705810547,"part":"leftAnkle","position":{"x":1372.1611657303372,"y":1256.6734467603}},
{"score":0.03403797745704651,"part":"rightAnkle","position":{"x":568.7979503160112,"y":533.8318556882023}}]},{"score":0.6620433540905223,"keypoints":
[{"score":0.9997365474700928,"part":"nose","position":{"x":1046.0989437031835,"y":432.2725743211611}},{"score":0.9998400211334229,"part":"leftEye","position":
{"x":1099.0435539559926,"y":359.313114466292}},{"score":0.9996658563613892,"part":"rightEye","position":{"x":988.0367070458802,"y":367.59765625}},
```

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

For ASL alphabet "A" the following key points CSV is shown below:



	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV	AW	AX	AY	AZ
1	leftWrist_	rightWrist	rightWrist	rightWrist	leftHip_sc	leftHip_x	leftHip_y	rightHip_s	rightHip_x	rightHip_y	leftKnee_	leftKnee_	leftKnee_	rightKnee	rightKnee	rightKnee	leftAnkle_	leftAnkle_	leftAnkle_	rightAnkle	rightA
2	534.7718	0.997498	570.0763	536.3512	0.331411	1254.762	1213.025	0.659939	916.1975	1144.72	0.028549	478.6641	901.8359	0.069232	471.6698	913.8224	0.027718	1368.907	1247.361	0.031781	544.3
3	537.0127	0.997471	569.6743	537.244	0.321442	1253.544	1212.749	0.664006	913.1872	1145.694	0.036519	477.3834	901.2099	0.092929	472.0614	911.6377	0.024298	1385.872	1262.36	0.034634	557.4
4	535.114	0.997725	571.0536	537.1272	0.317946	1252.812	1210.856	0.637285	889.2098	1144.12	0.043064	478.023	901.626	0.085086	472.399	908.4078	0.023112	1372.161	1256.677	0.034038	568.
5	534.6328	0.997412	570.5125	537.9126	0.354793	1254.584	1212.35	0.623577	912.4419	1148.86	0.057824	477.5503	901.5926	0.10545	472.7019	908.2631	0.023447	1373.302	1257.323	0.047417	567.0
6	534.5981	0.997372	571.7094	538.0884	0.319205	1253.347	1212.885	0.645672	909.1658	1146.453	0.043042	480.261	903.3257	0.09018	473.9635	910.0674	0.022914	561.061	565.5908	0.037476	567.9
7	536.1428	0.998002	572.2204	538.9913	0.331941	1254.122	1210.091	0.664504	908.618	1146.457	0.051628	477.8695	902.3691	0.118121	473.2932	909.7299	0.019521	1374.513	1256.18	0.030709	571.0
8	536.1459	0.998118	571.9077	538.9908	0.343673	1252.548	1211.276	0.66627	908.7075	1148.596	0.050665	477.5163	902.2036	0.127937	471.6489	908.0606	0.022242	1387.495	1262.501	0.027745	573.2
9	536.5171	0.997654	571.3885	538.6542	0.334966	1251.63	1211.827	0.655416	910.9699	1147.932	0.050248	476.1808	901.4924	0.125075	472.2689	906.643	0.021844	561.3788	568.8419	0.024426	557.9
10	536.1199	0.997193	570.6496	537.631	0.33704	1254.25	1210.69	0.646037	911.1051	1149.446	0.050616	477.2222	902.1462	0.145128	472.9848	905.5042	0.020814	1388.694	1265.279	0.034328	559.3
11	537.3963	0.998216	570.3878	539.0453	0.348198	1250.952	1212.269	0.640935	912.2756	1151.09	0.039313	476.5145	900.4029	0.153838	472.6113	902.372	0.017102	1377.971	1271.015	0.023327	570.7
12	538.053	0.998252	569.1829	539.8803	0.321383	1249.447	1213.843	0.631174	910.6151	1152.181	0.044845	476.871	900.7578	0.135386	471.5906	902.4946	0.026596	1374.014	1264.322	0.021337	543.8
13	537.42	0.997824	569.1793	538.9888	0.333348	1254.524	1213.979	0.581226	912.7961	1149.84	0.055644	475.4139	900.1985	0.137516	472.8015	903.1724	0.019218	1391.975	1273.676	0.021434	555.9
14	539.6377	0.998083	565.8149	540.2272	0.330595	1248.871	1214.586	0.563004	912.9924	1152.24	0.041739	475.5885	900.9041	0.129683	472.0757	903.3831	0.062641	1372.981	1254.502	0.023662	557.5
15	538.9014	0.997762	564.8116	540.9283	0.310434	1246.897	1213.202	0.55021	912.8824	1150.145	0.042602	474.4053	900.001	0.130358	471.6282	902.3304	0.045006	1373.091	1255.991	0.02905	558.3
16	537.6631	0.99759	563.705	539.3872	0.300234	1245.596	1214.994	0.581413	910.1882	1144.201	0.034051	473.9918	901.0534	0.12159	471.543	900.5562	0.061658	1373.968	1255.582	0.027651	557.5
17	536.8536	0.997518	563.2034	538.2085	0.309038	1245.909	1213.264	0.620263	908.7521	1144.613	0.046744	472.635	900.3851	0.138975	470.9507	901.8026	0.0591	1371.596	1257.555	0.023089	1392.
18	536.5308	0.997044	561.6616	537.8065	0.28537	1241.253	1215.06	0.596218	910.7643	1144.827	0.05048	473.7824	900.4107	0.10634	469.569	902.8045	0.060401	1370.711	1250.011	0.024483	1390.
19	537.2466	0.996638	563.1592	538.4284	0.29352	1240.392	1214.226	0.600743	909.9576	1142.609	0.065888	471.5229	899.1487	0.152434	468.6122	900.9578	0.050947	1372.444	1254.146	0.043354	556.
20	535.8376	0.997211	562.0488	538.7085	0.305516	1237.008	1213.965	0.579153	909.5811	1145.772	0.092862	472.0529	898.549	0.166786	467.7156	902.7452	0.065474	1372.085	1253.542	0.04119	557.8
21	535.8898	0.997592	558.6324	538.4434	0.305948	1234.176	1216.128	0.55552	887.5435	1149.59	0.091269	470.3559	897.125	0.200957	467.2115	899.8133	0.067473	1367.776	1254.334	0.039781	557.2
22	536.2489	0.997889	557.4272	540.0469	0.333225	1236.165	1216.838	0.551221	908.5133	1151.358	0.098929	472.1319	898.1509	0.177589	465.6847	902.7323	0.050581	1368.213	1256.749	0.037704	558.2
23	537.367	0.997674	555.6983	539.7887	0.30191	1233.561	1215.81	0.567467	886.1025	1153.26	0.111448	469.5001	897.1664	0.259543	467.8758	898.0805	0.053522	1373.333	1262.987	0.038725	554.5

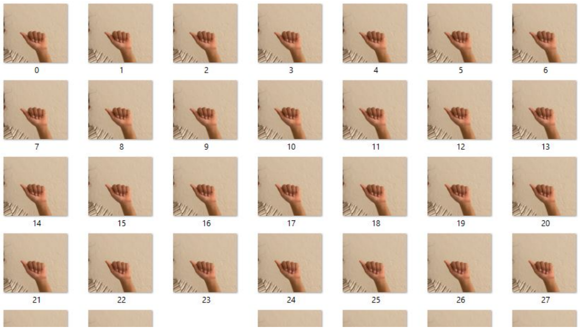
4) **Based on the left wrist and right wrist coordinates from key_points.csv file and the frames are extracted for cropping to get only the hand part from the frames.**

Segmentation Algorithm:

We will get the left wrist score,left wrist x coordinate, left wrist y coordinate, right wrist score, right wrist x coordinate, rightwrist y coordinate for each of the frames generated from the videos. Based on the left wrist score and right wrist score we get x and y coordinates for that frame.

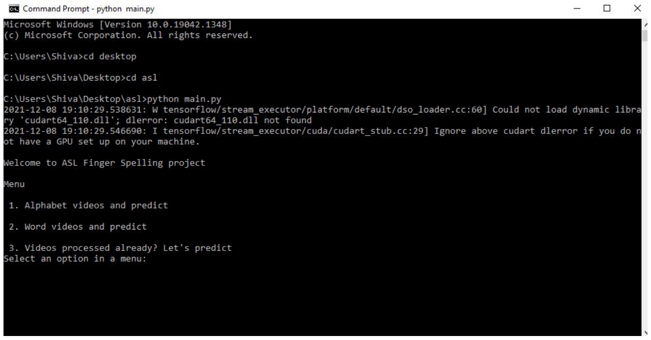
Based on the x and y coordinates we have we are going to generate a box with x-d , x+d , y-d , y+d (where d is a constant and it varies on the width and height of the video frame)

We are going to segment only the hand palm portion using the boxes we have developed.For ASL alphabet "A" the cropped frames(only hand part is cropped) are shown below in the figure.



For word “YOU” using ASL the cropped frames(only the hand portion is being cropped) are shown below in the following figure:

5)Images which are cropped are then fed to a pre trained Convolutional Neural Network model (already trained using the ASL data which is extracted from Kaggle) **and it predicts ASL alphabet/words:**
We developed a python program which feeds the cropped frames(hand part only) to the CNN model and then CNN model correctly predicts the alphabet.
Below is the output of the python program which ASL alphabet prediction:



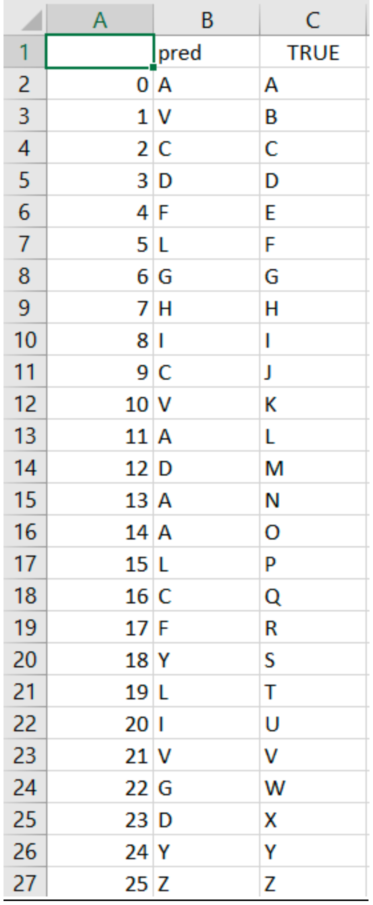
```
Microsoft Windows [Version 10.0.19042.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Shiva>cd desktop

C:\Users\Shiva\Desktop>cd asl

C:\Users\Shiva\Desktop\asl>python main.py
2021-12-08 19:10:29.538631: W tensorflow/stream_executor/platform/default/dso_loader.cc:60] Could not load dynamic libra
ry 'cudart64_110.dll'; dlerror: cudart64_110.dll not found
2021-12-08 19:10:29.546690: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do n
ot have a GPU set up on your machine.

Welcome to ASL Finger Spelling project

Menu

 1. Alphabet videos and predict

 2. Word videos and predict

 3. Videos processed already? Let's predict
Select an option in a menu:
```

ASL Word Detection Algorithm: Posenet generates key points JSON file of the ASL word video’s frames and later we convert those JSON files to CSV files. By this we will get all the key points of the ASL word video’s frames. This algorithm is used to track current and previous x and y coordinates of Left wrist or Right wrist from the key points csv file we have generated from JSON file .If the absolute value of the difference in current x coordinate and previous x coordinate or absolute value of the difference in current y coordinate and previous y coordinate in any hand gesture crosses beyond a fixed threshold value then alphabet transition takes place and all the frames from the current frame number till the transition frame numbers are being fed to the pre trained Convolutional Neural Network model to determine that particular alphabet sign. This continues till the end of the frames

6) Precision and recall are displayed based on the True value and predicted value F1 for both ASL Alphabet and Word detection

7) Finally we create a CSV file named result.csv with only two column predicted value and true value

Figure for the result.csv for ASL word detection is shown 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

S no	Task	Members
1	Record 26*4 ASL alphabets videos	KavyaSree, Nithish, Shivani, Navitha
2	Developing palm cropping algorithm using wrist points	KavyaSree,Nithish
3	Palm detection algorithm	KavyaSree,Nithish
4	Configuring the 3D CNN model	Nithish, Shivani
5	F1 Metrics Reporting	Nithish, Shivani, Navitha
6	Recording 10*4 word videos	KavyaSree, Nithish, Shivani, Navitha
7	Developing Keypoint series files	KavyaSree, Shivani
8	Implementation of Segmentation Algorithm	KavyaSree, Nithish, Shivani, Navitha
9	Using 3D CNN to recognize Alphabet	KavyaSree, Navitha
10	Development of algorithm to recognise words	Nithish, Shivani, Navitha, KavyaSree
11	Automation pipeling	Shivani, Navitha
12	Calculating the word recognition accuracy	Navitha, Shivani
13	Final Report	KavyaSree, Nithish

CONCLUSION:

Implementing this project made us understand how to convert the ASL signs into machine understood language.We have also come to know about various image processing techniques and machine learning models to extract the features from the videos and predict the correct ASL signs.We have also learnt about the Posenet which is a deep learning model which detects the body parts and converts the frames into keypoints.

ACKNOWLEDGEMENT:

We would like to heartly thank Dr. Ayan Banerjee for encouraging us and helping out with our queries and helping us in each step. We would also like to acknowledge the authors whose research helped us in developing as well as understanding previous research. Last but not least we would also like to thank our team members for their constant efforts and contribution to this 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”