ASL Fingerprinting

Sandhya Balu
Arizona State University
Tempe, AZ, USA
sbalu3@asu.edu

Vinodh Pothapala
Arizona State University
Tempe, AZ, USA
vpothapa@asu.edu

Sai Ganesh Gunda
Arizona State University
Tempe, AZ, USA
sgunda3@asu.edu

Rajesh Badam
Arizona State University
Tempe, AZ, USA
rbadam@asu.edu

ABSTRACT:

This project develops a real time American Sign Language (ASL) fingerspelling translator using Deep learning which is a branch of Machine Learning based on neural networks. Using smart gestures application capture a video of hand gestures with all 26 different symbols and interpret all these videos into online server. Using Pose-net develop a palm detection algorithm to obtain the main wrist points. Also use an algorithm which can crop the videos from the videos data. Input this data into 3D Convolutional Neural Networks (CNNs) which is one of the most popular neural network architectures and extremely successful in the field of image processing. This CNN has to train with the sample ASL data. With all these techniques and algorithms ASL fingerspelling application predict the alphabets provided by the user. By utilizing this reasonable accuracy feed the application with different words, this model will recognize all the letters individually and combines each letter result to develop a recognition of a word.

KEY WORDS: Image processing, Deep Learning, Convolutional neural network, Posenet, Keras, TensorFlow, Fingerspelling.

INTRODUCTION:

American Sign Language (ASL) is a natural language that serves as the cardinal sign language of Deaf communities. It is organized visual language which can be shown by hands movements. This prompted us to develop a translator-based application that could recognize hand symbols and give us the corresponding meaning of the input. ASL has 26 unique symbols, among them 24 alphabets are static and remaining two letters ('J' and 'Z') are movement-based gestures as shown in Figure 1. It is used to spell out the 26 different letters of the standard English language by using particular hand gestures. This hand gesture technology is also used for the computers and machines. Interpreting hand actions may also end up a manner of logging and studying human behavior.

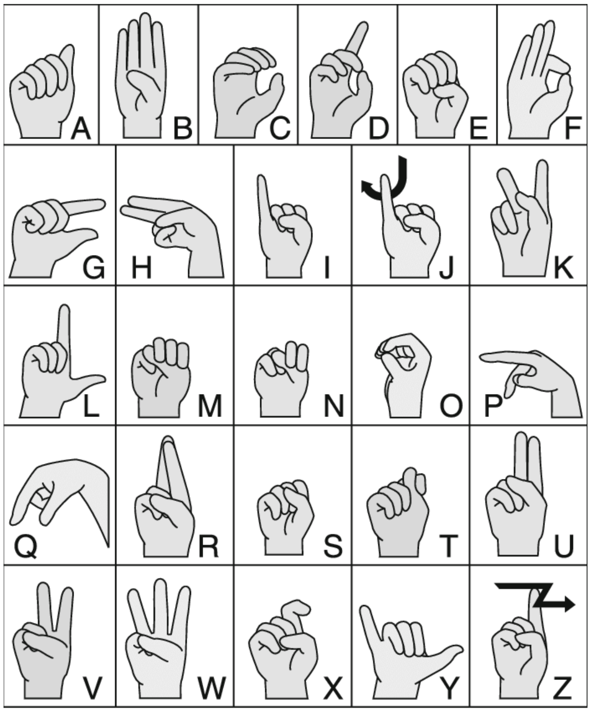


Figure 1

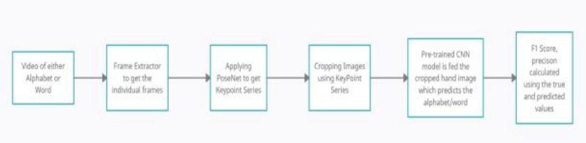
PROJECT SETUP:

Development of project needs following software’s and components:

- ASL Data on Kaggle
- Keras
- Posenet
- TensorFlow
- Node 12
- Python 3.8

SYSTEM ARCHITECTURE:

American uses a series of steps to predict words or alphabets from a video. A set of training videos are recorded from mobile application and frames for each video are extracted. Video frames once extracted are then passed as a input to posenet which is a prediction technique to detect human beings gestures in videos and images. It has capable of both single mode and multimode detection. Single mode is nothing but the single person detection and multi-mode is multiple persons pose detection. In fact, it is a deep learning based TensorFlow model that permits us to recognize the human pose by detecting the hands and form a skeleton by joining the multiple points. Using Posenet key points are extracted for wrist and video frames generated are cropped to form hand cropped frames. The hand frames thus generated are served to train the CNN model. Similarly for extracting the word videos, a segmentation algorithm is run on the key points generated by posenet to separate alphabets for each video. Once the segmentation algorithm is done, all the separated alphabets are predicted individually and combined using an algorithm and accuracy is predicted.


Video of either Alphabet or Word
Frame Extractor to get the individual frames
Applying PoseNet to get Keypoint Series
Cropping Images using KeyPoint Series
Pre-trained CNN model is fed the cropped hand image which predicts the alphabet/word
F1 Score, precison calculated using the true and predicted values

IMPLEMENTATION:

Six tasks need to be implemented to complete this project.

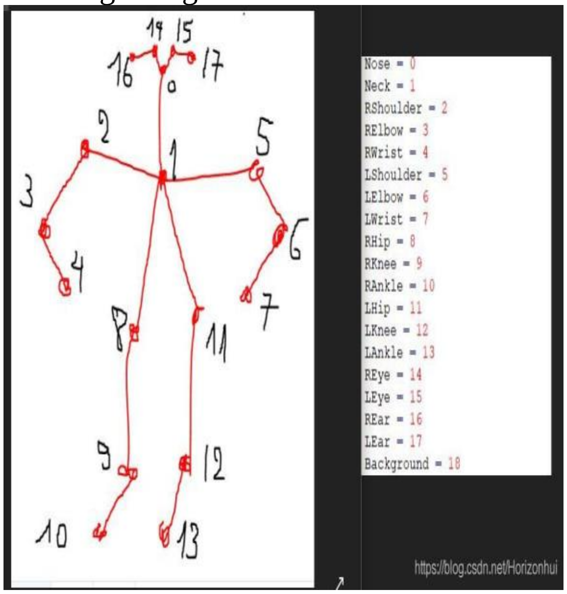
1)Extracting the frames from alphabet videos:

American should extract number of frames from a single alphabet video. Initially it will extract the frames then it will crop the extracted frames to frames which contains only hand gestures.

A sample video frames extracted are as below for alphabet ‘L’:

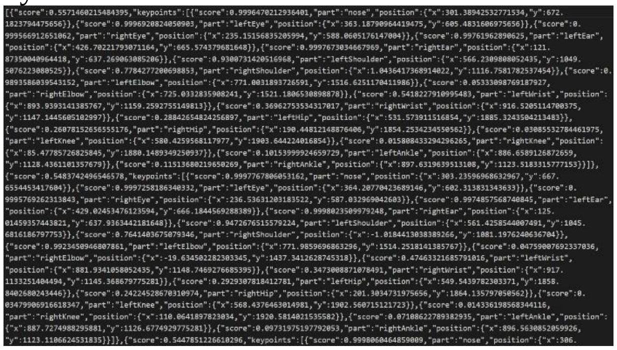
2) Using Posenet to create keypoints from the video frames extracted in step(1)

The output of PoseNet could be easilycomprehended by referring to Figure2.



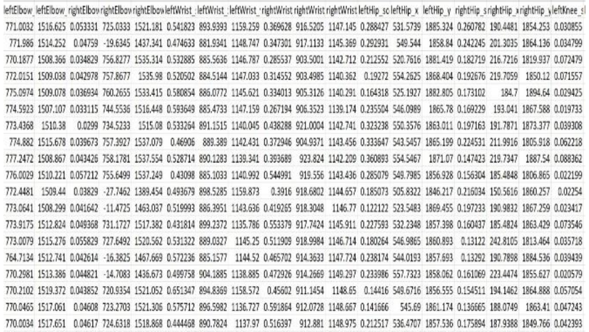
```
Nose = 0
Neck = 1
RShoulder = 2
RElbow = 3
RWrist = 4
LShoulder = 5
LElbow = 6
LWrist = 7
RHip = 8
RKnee = 9
RAnkle = 10
LHip = 11
LKnee = 12
LAnkle = 13
REye = 14
LEye = 15
REar = 16
LEar = 17
Background = 18
```

A sample JSON file generated by ‘L’ is as below:



```
[{"score":0.5571460215484395,"keypoints":[{"score":0.9996470212936401,"part":"nose","position":{"x":301.38942532771534,"y":672.
1823794475656}},{"score":0.9996920824050903,"part":"leftEye","position":{"x":363.18790964419475,"y":605.4831606975656}},{"score":0.
999566912651062,"part":"rightEye","position":{"x":235.15156835205994,"y":588.0605176147004}},{"score":0.99761962890625,"part":"leftEar",
"position":{"x":426.70221793071164,"y":665.574379681648}},{"score":0.9997673034667969,"part":"rightEar","position":{"x":121.
87350040964418,"y":637.269063085206}},{"score":0.9300731420516968,"part":"leftShoulder","position":{"x":566.2309808052435,"y":1049.
5076223080525}},{"score":0.7784277200698853,"part":"rightShoulder","position":{"x":1.0436417368914022,"y":1116.7581782537454}},{"score":0.
9893586039543152,"part":"leftElbow","position":{"x":771.0031893726591,"y":1516.6251170411986}},{"score":0.05333098769187927,
"part":"rightElbow","position":{"x":725.0332835908241,"y":1521.1806530898878}},{"score":0.5418227910995483,"part":"leftWrist","position":
{"x":893.939314138576 7,"y":1159.2592755149813}},{"score":0.36962753534317017,"part":"rightWrist","position":{"x":916.520511470037 5,
"y":1147.1445605102997}},{"score":0.28842654824256897,"part":"leftHip","position":{"x":531.57391151685 4,"y":1885.3243504213483}},
{"score":0.26078152656555176,"part":"rightHip","position":{"x":190.44812148876406,"y":1854.2534234550562}},{"score":0.03085532784461975,
"part":"leftKnee","position":{"x":580.4259568117977,"y":1903.644224016854}},{"score":0.015808433294296265,"part":"rightKnee","position":
{"x":85.47785726825845,"y":1880.1489349250937}},{"score":0.10153999924659729,"part":"leftAnkle","position":{"x":886.6589126872659,
"y":1128.4361101357679}},{"score":0.11513680219650269,"part":"rightAnkle","position":{"x":897.6319639513108,"y":1123.5183315777153}}]},
{"score":0.5483742496546578,"keypoints":[{"score":0.9997767806053162,"part":"nose","position":{"x":303.23596968632967,"y":667.
6554453417604}},{"score":0.9997258186340332,"part":"leftEye","position":{"x":364.20770423689146,"y":602.313831343633}},{"score":0.
999576926231384 3,"part":"rightEye","position":{"x":236.53631203183522,"y":587.032969042603}},{"score":0.9974857568740845,"part":"leftEar",
"position":{"x":429.02453476123594,"y":666.1844569288389}},{"score":0.9998023509979248,"part":"rightEar","position":{"x":125.
01459357443821,"y":637.9363442181648}},{"score":0.9472676515579224,"part":"leftShoulder","position":{"x":561.4258544007491,"y":1045.
6816186797753}},{"score":0.7641403675079346,"part":"rightShoulder","position":{"x":-1.0184413038389266,"y":1081.1976240636704}},
{"score":0.9923450946807861,"part":"leftElbow","position":{"x":771.9859696863296,"y":1514.2518141385767}},{"score":0.04759007692337036,
"part":"rightElbow","position":{"x":-19.634502282303345,"y":1437.3412628745318}},{"score":0.4746332168579101 6,"part":"leftWrist",
"position":{"x":881.9341058052435,"y":1148.7469276685395}},{"score":0.3473008871078491,"part":"rightWrist","position":{"x":917.
1133251404494,"y":1145.368679775281}},{"score":0.2929307818412781,"part":"leftHip","position":{"x":549.5439782303371,"y":1858.
8402680243446}},{"score":0.24224528670310974,"part":"rightHip","position":{"x":201.3034731975656,"y":1864.135797050562}},{"score":0.
03479906916618347,"part":"leftKnee","position":{"x":568.4376463014981,"y":1902.560715121723}},{"score":0.01433619856834411 6,
"part":"rightKnee","position":{"x":110.0641897823034,"y":1920.5814021535582}},{"score":0.07108622789382935,"part":"leftAnkle","position":
{"x":887.7274988295881,"y":1126.6774929775281}},{"score":0.09731975197792053,"part":"rightAnkle","position":{"x":896.5630852059926,
"y":1123.1106624531835}}]},{"score":0.5447851226610296,"keypoints":[{"score":0.9998060464859009,"part":"nose","position":{"x":306.
```

Key points json file generated is converted to csv for making it easy to read the key points data. A sample of JSON to CSV file converted is as below:



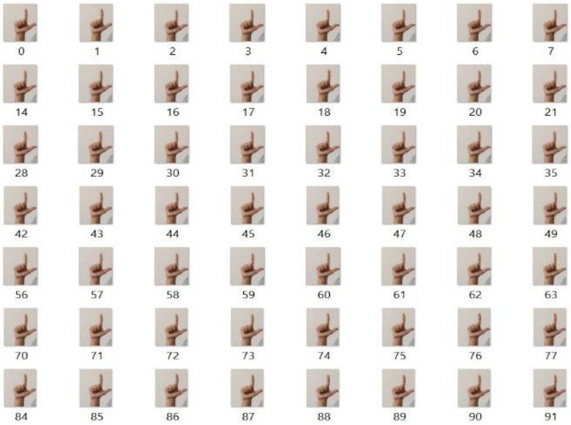
leftElbow_	leftElbow_	rightElbow	rightElbow	rightElbow	leftWrist_	leftWrist_	leftWrist_	rightWrist	rightWrist	rightWrist	leftHip_sc	leftHip_x	leftHip_y	rightHip_s	rightHip_x	rightHip_y	leftKnee_s
771.0032	1516.625	0.053331	725.0333	1521.181	0.541823	893.9393	1159.259	0.369628	916.5205	1147.145	0.288427	531.5739	1885.324	0.260782	190.4481	1854.253	0.030855
771.986	1514.252	0.04759	-19.6345	1437.341	0.474633	881.9341	1148.747	0.347301	917.1133	1145.369	0.292931	549.544	1858.84	0.242245	201.3035	1864.136	0.034799
770.1877	1508.366	0.034829	756.8277	1535.314	0.532885	885.5636	1146.787	0.285537	903.5001	1142.712	0.212552	520.7616	1881.419	0.182719	216.7216	1819.937	0.072479
772.0151	1509.038	0.042978	757.8677	1535.98	0.520502	884.5144	1147.033	0.314552	903.4985	1140.362	0.19272	554.2625	1868.404	0.192676	219.7059	1850.12	0.071557
775.0974	1509.078	0.036934	760.2655	1533.415	0.580854	886.0772	1145.621	0.334013	905.3126	1140.291	0.164318	525.1927	1882.805	0.173102	184.7	1894.64	0.029425
774.5923	1507.107	0.033115	744.5536	1516.448	0.593649	885.4733	1147.159	0.267194	906.3523	1139.174	0.235504	546.0989	1865.78	0.169229	193.041	1867.588	0.019733
773.4368	1510.38	0.0299	734.5233	1515.08	0.533264	891.1515	1140.045	0.438288	921.0004	1142.741	0.323238	550.3576	1863.011	0.197163	191.7871	1873.377	0.039308
774.882	1515.678	0.039673	757.3927	1537.079	0.46906	889.389	1142.431	0.372946	904.9371	1143.456	0.333647	543.5457	1865.199	0.224531	211.9916	1805.918	0.062218
777.2472	1508.867	0.043426	758.1781	1537.554	0.528714	890.1283	1139.341	0.393689	923.824	1142.209	0.360893	554.5467	1871.07	0.147423	219.7347	1887.54	0.088362
776.0029	1510.221	0.057212	755.6499	1537.249	0.43098	885.1033	1140.992	0.544991	919.556	1143.436	0.285079	549.7985	1856.928	0.156304	185.4848	1806.865	0.022199
772.4481	1509.44	0.03829	-27.7462	1389.454	0.493679	898.5285	1159.873	0.3916	918.6802	1144.657	0.185073	505.8322	1846.217	0.216034	150.5616	1860.257	0.02254
773.0641	1508.299	0.041642	-11.4725	1463.037	0.519993	886.3951	1143.636	0.419265	918.3048	1146.77	0.122122	523.5483	1869.455	0.197233	190.9832	1867.259	0.023417
773.9175	1512.824	0.049368	731.1727	1517.382	0.431814	899.2372	1135.786	0.553379	917.7424	1145.911	0.227593	532.2348	1857.398	0.160437	185.4824	1863.429	0.073546
773.0079	1515.276	0.055829	727.6492	1520.562	0.531322	889.0327	1145.25	0.511909	918.9984	1146.714	0.180264	546.9865	1860.893	0.13122	242.8105	1813.464	0.035718
764.7134	1512.741	0.042614	-16.3825	1467.669	0.572236	885.1577	1144.52	0.465702	914.3633	1147.724	0.238174	544.0193	1857.693	0.13292	190.7898	1884.536	0.039439
770.2981	1513.386	0.044821	-14.7083	1436.673	0.499758	904.1885	1138.885	0.472926	914.2669	1149.297	0.233986	557.7323	1858.062	0.161069	223.4474	1855.627	0.020579
770.2102	1519.372	0.043852	720.9354	1521.052	0.651347	894.8369	1158.572	0.45602	911.1454	1148.65	0.14416	549.6716	1856.555	0.154511	194.1462	1864.888	0.057054
770.0465	1517.061	0.04608	723.2703	1521.306	0.575712	896.5982	1136.727	0.591864	912.0728	1148.667	0.141666	545.69	1861.174	0.136665	188.0749	1863.41	0.047243
770.0034	1517.651	0.04617	724.6318	1518.868	0.444468	890.7824	1137.97	0.516397	912.881	1148.975	0.212517	536.4707	1857.536	0.175894	187.9388	1849.766	0.042393

3) Crop the extracted frame to obtain palm region:

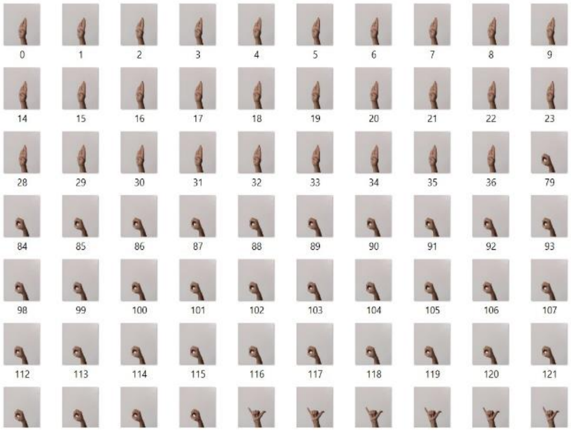
In CSV file, we have right wrist and left wrist coordinates which is used for extracting the frames from it which has only the palm part. We just need hand part to recognize the ASL alphabets. In this task we use palm detection algorithm which has been included with cropping algorithm.

Segmentation Algorithm:

Each and every frame has left wrist x and y coordinates along with left wristcore and similarly right wrist x and y coordinates of particular frame. By utilizing these coordinates, algorithm creates a box like structure with x+d, x-d and similarly y-d, y+d. Here, d is constant and may vary based on the video frame features such as height and width. This segmentation algorithm will segment only the portion which has hand part. We have 26 different symbols, this algorithm segments different frame portions for different alphabetical videos.An example of hand frames extracted using the key points is as below for alphabet ‘L’

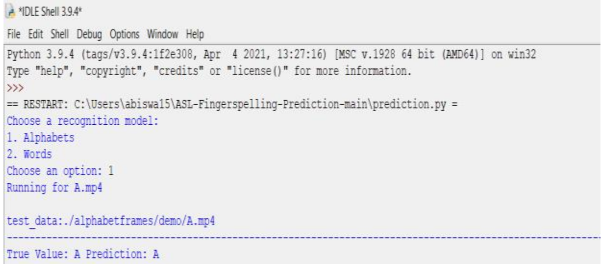


Similarly, word frames are cropped for word “BOY” which generates the hand frames as below:



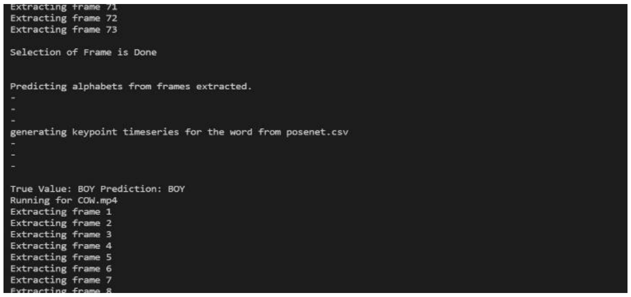
4) **CNN Model is fed with cropped image:**

Convolutional Neural Network (CNN) is a part of deep neural networks, which is widely used in image processing. Convolution is the special technique used in the CNN. It contains multiple layers of artificial neurons which are mathematical functions that measures the sum of different types of inputs and outputs. When we provide an image to the Convent, each layer provides many activation functions which can pass on to the next layer. Here, the Initial layer just extracts diagonal or horizontal edges. The outcome of first layer is given to the next layer which extracts complex edges like corners or combinational edges. Likewise, it will deep dive and recognize more complicated features like hands, objects etc. CNN model is already trained with the Kaggle and ASL data. Use Python programming language to feed the cropped image frames to the CNN model, this CNN model uses different image processing techniques along with train data and recognize the alphabet. An output of alphabet prediction is as 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\abiswa15\ASL-Fingerspelling-Prediction-main\prediction.py =
Choose a recognition model:
1. Alphabets
2. Words
Choose an option: 1
Running for A.mp4

test_data:./alphabetframes/demo/A.mp4
---------------------------------------------------------------------------------------------------
True Value: A Prediction: A
```

Output of words prediction is as below :



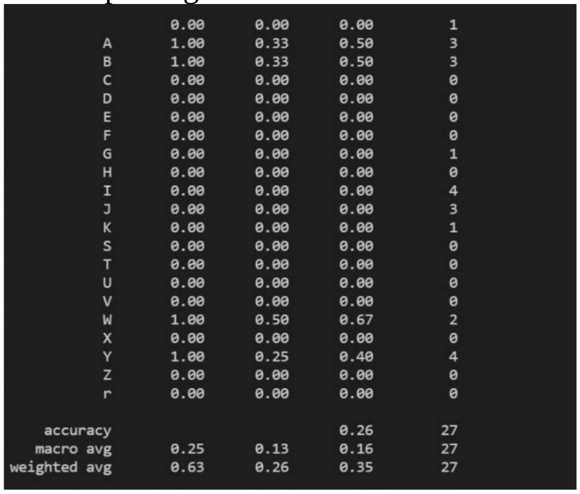
```
Extracting frame 71
Extracting frame 72
Extracting frame 73

Selection of Frame is Done


Predicting alphabets from frames extracted.
-
-
-
generating keypoint timeseries for the word from posenet.csv
-
-
-

True Value: BOY Prediction: BOY
Running for COW.mp4
Extracting frame 1
Extracting frame 2
Extracting frame 3
Extracting frame 4
Extracting frame 5
Extracting frame 6
Extracting frame 7
Extracting frame 8
```

<u>ASL Word Detection Algorithm:</u>

For the detection of words, we will be using the key points csv file generated in the step above. Once key points have been generated to check the transition of one alphabet to another alphabet, we check the current and previous x and y coordinates. If the difference between current and previous x and y coordinates reaches a threshold value, then it is considered that a transition has occurred from one alphabet to another alphabet and the frames till the current frame are considered as a single alphabet and similarly calculated till the end of the frames and each alphabet is predicted individually which is combined to generate word as output.

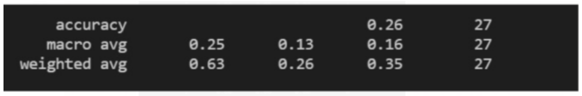
6) According to the true and prediction value F1 score, precision, recall has been showed:

A classification report has been generated using the sklearn metrics function which gives a report containing label, precision, recall, f1-score and support. An example of how output is generated is shown as below:



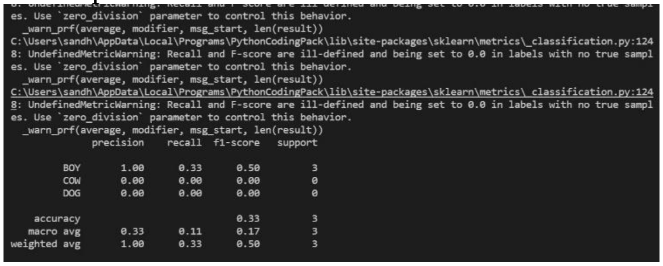
```
                  0.00      0.00      0.00         1
           A      1.00      0.33      0.50         3
           B      1.00      0.33      0.50         3
           C      0.00      0.00      0.00         0
           D      0.00      0.00      0.00         0
           E      0.00      0.00      0.00         0
           F      0.00      0.00      0.00         0
           G      0.00      0.00      0.00         1
           H      0.00      0.00      0.00         0
           I      0.00      0.00      0.00         4
           J      0.00      0.00      0.00         3
           K      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      1.00      0.50      0.67         2
           X      0.00      0.00      0.00         0
           Y      1.00      0.25      0.40         4
           Z      0.00      0.00      0.00         0
           r      0.00      0.00      0.00         0

    accuracy                          0.26        27
   macro avg      0.25      0.13      0.16        27
weighted avg      0.63      0.26      0.35        27
```

A screenshot of the output of python program for classification is shown as below:



```
    accuracy                          0.26        27
   macro avg      0.25      0.13      0.16        27
weighted avg      0.63      0.26      0.35        27
```

Similarly below is the attached screenshot of the output for classification of words:

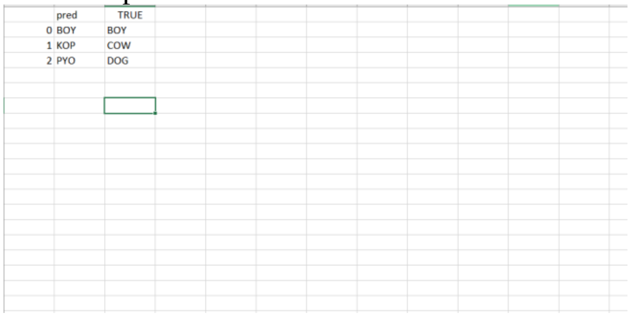


```
8: UndefinedMetricWarning: Recall and F-score are ill-defined and being set to 0.0 in labels with no true sampl
es. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\sandh\AppData\Local\Programs\PythonCodingPack\lib\site-packages\sklearn\metrics\_classification.py:124
8: UndefinedMetricWarning: Recall and F-score are ill-defined and being set to 0.0 in labels with no true sampl
es. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\sandh\AppData\Local\Programs\PythonCodingPack\lib\site-packages\sklearn\metrics\_classification.py:124
8: UndefinedMetricWarning: Recall and F-score are ill-defined and being set to 0.0 in labels with no true sampl
es. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
              precision    recall  f1-score   support

         BOY       1.00      0.33      0.50         3
         COW       0.00      0.00      0.00         0
         DOG       0.00      0.00      0.00         0

    accuracy                           0.33         3
   macro avg       0.33      0.11      0.17         3
weighted avg       1.00      0.33      0.50         3
```

7) CSV file has been generated with True and predicted value:

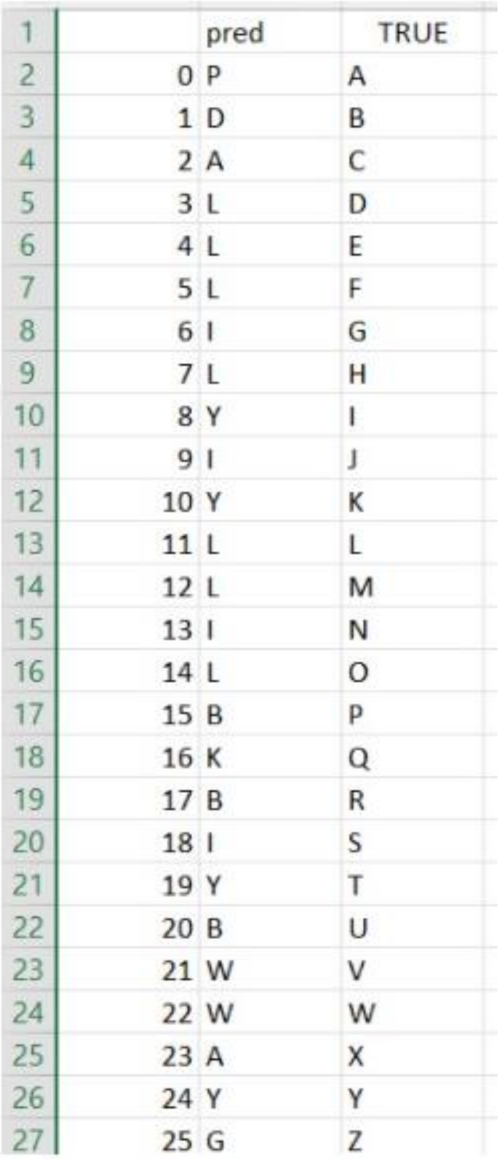
Once the frames are processed and prediction are generated, all the results are stored in a csv file in the same folder where the training videos are stored.

Below is the attached results.csv screenshot for word prediction:



	pred	TRUE
0	BOY	BOY
1	KOP	COW
2	PYO	DOG

Below is the attached results.csv screenshot for alphabet prediction



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

LINKS:

1)Alphabet and Word videos: All the alphabets and videos each 26 by 4 team members are
Uploaded in the following drive folder https://drive.google.com/drive/folders/1Rs6GKdX8tpFFY0fjJWbbHyszQ6Qz4bOB?usp=sharing

2) Demo Links:
80% before pipelining input/output andwith only ASL alphabet detection-https://www.youtube.com/watch?v=br4QLSxXg

TASK COMPLETION:

S.no	**Task**	**Assignee**
1	Record 26 alphabets by each person (26 *4)	Rajesh, Sandhya , Vinodh, Ganesh
2	Develop palm cropping algorithm using wrist points obtained from posenet.	Sandhya, Vinodh
3	Validating palm detection algorithm	Rajesh, Vinodh
4	Configuring the 3D CNN model	Ganesh, Rajesh
5	Reporting F1 Metrics	Sandhya, Vinodh
6	Record 10*4 word	Sandhya,
	videos using ASL	Vinodh, Rajesh,

		Ganesh
7	Developing Keypoint Series	Vinodh, Rajesh
8	Implementing Segmentation Algorithm	Sandhya, Ganesh
9	Using 3D CNN to recognize Alphabets	Vinodh, Sandhya, Rajesh
10	Developing algorithm to recognize words	Sandhya, Rajesh
11	Automation pipelining	Rajesh, Vinodh
12	Calculating the word recognition accuracy	Ganesh, Vinodh
13	Final Report	Sandhya, Vinodh, Rajesh, Ganesh

CONCLUSION:

ASL fingerspelling project has helped in gaining how one can convert ASL language to English with the help of deep learning algorithms. We have also learned how on how efficiently an algorithm can be used to predict the accuracy and approaching different algorithms and their algorithms. Approaches were explored to improve the accuracy. We have learnt using Posenet to generate key points and identifying each of part of the body.

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-MaldagueLucas & Giguère, Philippe. (2014). Sign Language Fingerspelling Classification from Depthand 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 ”