UE18CS390B – Capstone Project Phase – 2

SEMESTER - VII

END SEMESTER ASSESSMENT

Project Title : Indian Sign Language Translation
Project ID : PW22SS202
Project Guide : Dr.Shylaja SS
Project Team : Neeli Krishna Dheeraj (PES1201800182)
Alluri Nayan Varma (PES1201800208)
Akhil Chaduvula (PES1201800282)

Outline

- Abstract
- Team Roles and Responsibilities.
- Summary of Requirements and Design (Capstone Phase - 1)
- Summary of Methodology / Approach (Capstone Phase - 1)
- Design Description
- Modules and Implementation Details
- Project Demonstration and Walkthrough
- Test Plan and Strategy
- Results and Discussion
- Lessons Learnt
- Conclusion and Future Work
- References

Abstract

- The aim of this project is to build an automatic ISL translator - i.e. given an input video of a specially abled person signing a word, the model should be able to predict which word is being signed.
- This translator is highly beneficial for the not just the deaf and dumb people, but also for the people trying to communicate with them.
- The model built should be reasonably robust to variations in camera angle, video quality, distance from the camera to the subject, variations in lighting etc.
- The goal of this project is to provide(coverage) a means of communication for deaf anddumb people living in India to communicate with anyone in ISL.
- This Application provides support for only one sign Language (i.e ISL).

Team Roles and Responsibilities

Akhil Chaduvula		
Task Done	**Timeline**	**Description**
Read Papers (To begin research about the project) and Wrote Reports.	4th March 2021 - 29th April 2021	Few of the papers read were 1. Text2Sign: Towards Sign Language Production Using Neural Machine Translation and Generative Adversarial Networks 2. Indian Sign Language Translator Using Gesture Recognition Algorithm 3. Sign Language to Text and Speech Translation in Real Time Using Convolutional Neural Network
Worked on Finger Keypoint Detection and Searched for Appropriate Dataset	2nd May 2021 - 25th May 2021	Wrote about 100 lines of Code and finalised upon using INCLUDE Dataset
Worked On Preprocessing	15th August 2021 - 4th September 2021	Preprocessed 1/3rd of the the dataset using the Openpose Method
Worked on Frame Extraction	3rd October 2021 - 15th October 2021	Converted the Openpose preprocessesd model to frames
Worked on the Front End	15th October 2021 - 15th November 2021	Created 2 Web Pages that would be used to integrate with the backend Model
Worked on the documentation	15th November 2021 - 28th November 2021	Worked on the project report, ppts and poster

Team Roles and Responsibilities

Alluri Nayan varma		
Task Done	**Timeline**	**Description**
Read Papers (To begin research about the project) and Wrote Reports.	4th March 2021 - 29th April 2021	Few of the papers read were 1. Neural Sign Language Translation based on Human Keypoint Estimation 2. Word-level Deep Sign Language Recognition from Video: A New Large-scale Dataset and Methods Comparison 3. Sign Language Transformers: Joint End-to-end Sign Language Recognition and Translation
Worked on Openpose Method and Searched for Appropriate Dataset	2nd May 2021 - 25th May 2021	Wrote about 300 lines of Code and finalised upon using INCLUDE Dataset
Worked On Preprocessing	15th August 2021 - 4th September 2021	Preprocessed 1/3rd of the the dataset using the Openpose Method
Built Models	3rd October - 15th November	Worked on building the Resnet Model and LSTM model
Worked on Documentation	15th November - 28th November	Worked on the IEEE paper

Team Roles and Responsibilities

Neeli Krishna Dheeraj

Task Done	Timeline	Description
Read Papers (To begin research about the project) and Wrote Reports.	4th March 2021 - 29th April 2021	A few of the papers read were 1. Sign Language Recognition and Translation: A Multidisciplined Approach From the Field of Artificial Intelligence 2. Neural Sign Language Translation 3. Translating Sign Language in Real Time With AI
Worked on Human Body segmentation and Searched for Appropriate Dataset	2nd May 2021 - 25th May 2021	Wrote about 100 lines of Code and finalised upon using INCLUDE Dataset
Worked On Preprocessing	15th August 2021 - 4th September 2021	Preprocessed 1/3rd of the the dataset using the Openpose Method
Worked on transformers	3rd October 2021 - 15th October 2021	Worked on transforming the frames to a input embedding
Worked on Model	15th October 2021 - 15th November 2021	Worked on the RNN model as well as Adaboost Model
Worked on the documentation	15th November 2021 - 28th November 2021	Worked on the Project Report

Summary of Requirements and Design

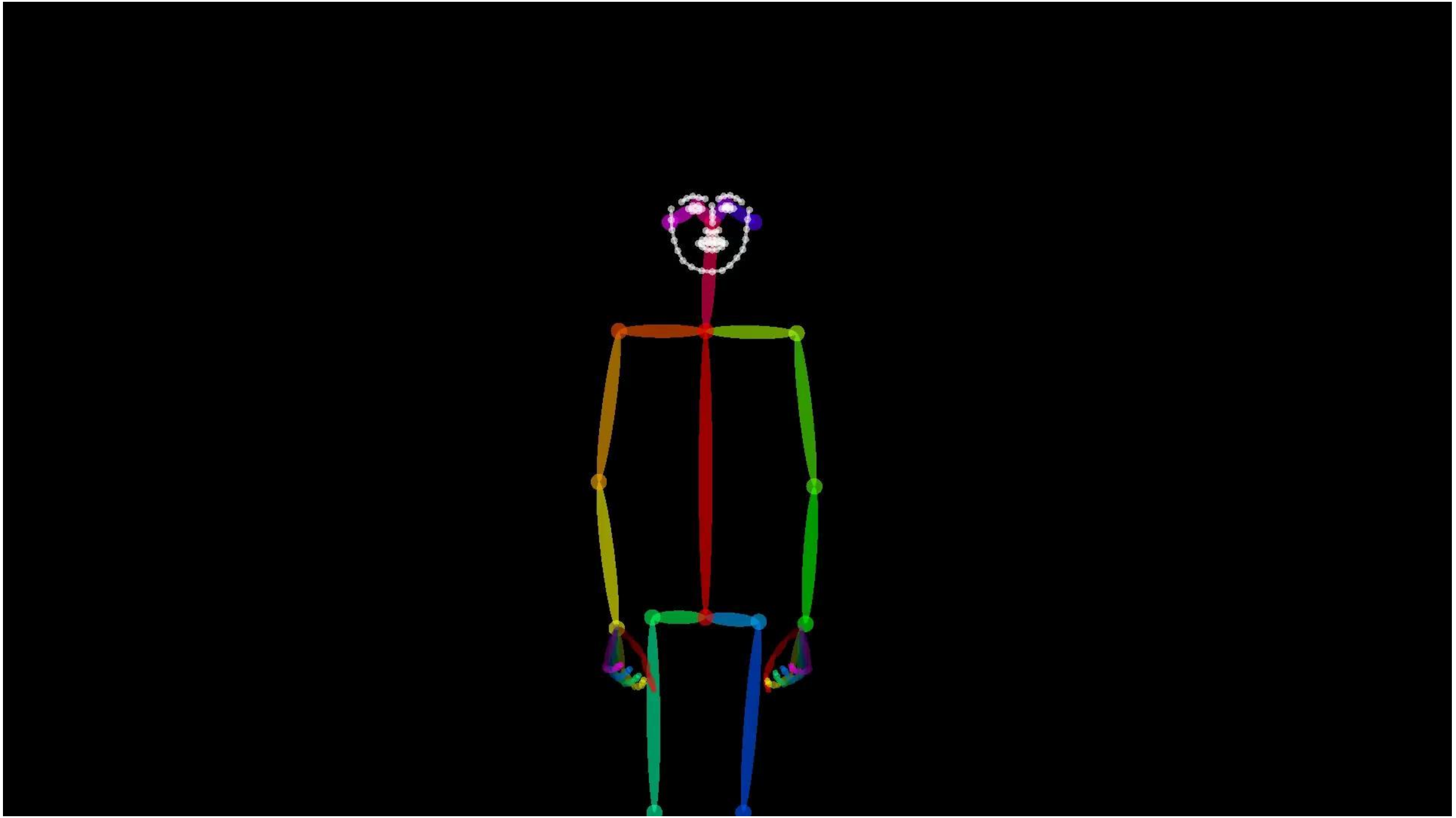
The requirements of the project are:-

1) OS:- A mobile operating system was what was planned.
2) Performance requirements:- The application must reliable and robust
3) Input and output :- On Providing and input of a person Signing, our model must be able to predict the word that the person describes

Summary of Methodology / Approach

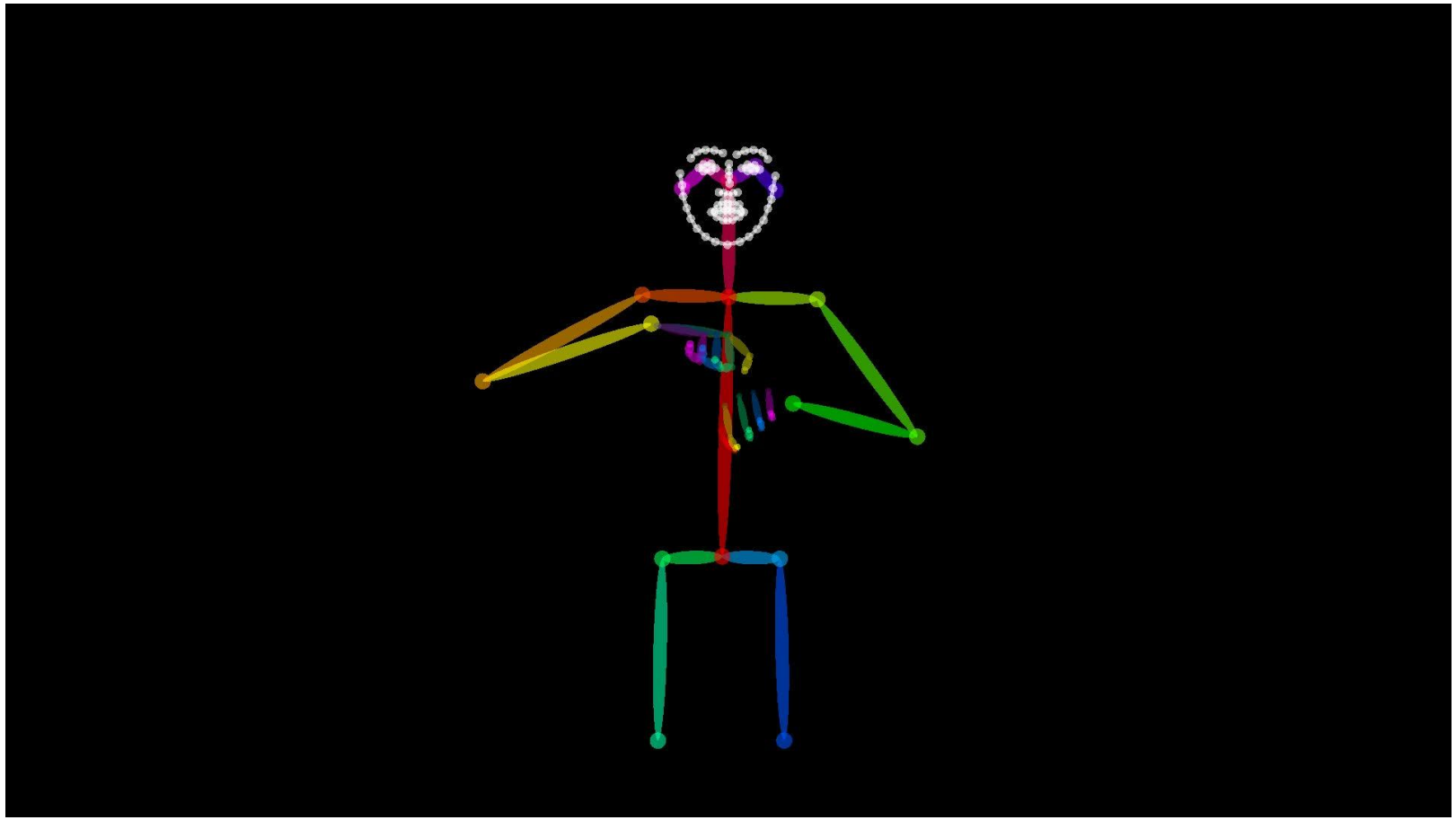
- The implementation steps of the project were:

1) OpenPose Preprocessing: This was the preprocessing Phase, wherein we preprocessed all our data of videos into stickfigure like videos wherein we could see out all the Key points in the video.



Summary of Methodology / Approach

2) The videos have then been converted into frames



Summary of Methodology / Approach

3) These frames have then been embedded into a json file which was then used to build a few models. We first worked on an RNN model..later on we developed an LSTM model and achieved a higher accuracy.

```
7/7 [==============================] - 0s 13ms/step - loss: 0.1429 - accuracy: 0.9602
Epoch 732/750
7/7 [==============================] - 0s 11ms/step - loss: 0.1632 - accuracy: 0.9516
Epoch 733/750
7/7 [==============================] - 0s 12ms/step - loss: 0.1517 - accuracy: 0.9528
Epoch 734/750
7/7 [==============================] - 0s 13ms/step - loss: 0.1544 - accuracy: 0.9496
Epoch 735/750
7/7 [==============================] - 0s 12ms/step - loss: 0.1562 - accuracy: 0.9528
Epoch 736/750
7/7 [==============================] - 0s 13ms/step - loss: 0.1530 - accuracy: 0.9528
Epoch 737/750
7/7 [==============================] - 0s 12ms/step - loss: 0.1488 - accuracy: 0.9528
Epoch 738/750
7/7 [==============================] - 0s 11ms/step - loss: 0.1559 - accuracy: 0.9525
Epoch 739/750
7/7 [==============================] - 0s 11ms/step - loss: 0.1827 - accuracy: 0.9433
Epoch 740/750
7/7 [==============================] - 0s 12ms/step - loss: 0.1498 - accuracy: 0.9525
Epoch 741/750
7/7 [==============================] - 0s 14ms/step - loss: 0.1605 - accuracy: 0.9499
Epoch 742/750
7/7 [==============================] - 0s 11ms/step - loss: 0.1717 - accuracy: 0.9457
Epoch 743/750
7/7 [==============================] - 0s 12ms/step - loss: 0.1521 - accuracy: 0.9522
Epoch 744/750
7/7 [==============================] - 0s 12ms/step - loss: 0.1490 - accuracy: 0.9555
Epoch 745/750
7/7 [==============================] - 0s 11ms/step - loss: 0.1670 - accuracy: 0.9451
Epoch 746/750
7/7 [==============================] - 0s 12ms/step - loss: 0.1531 - accuracy: 0.9543
Epoch 747/750
7/7 [==============================] - 0s 12ms/step - loss: 0.1584 - accuracy: 0.9525
Epoch 748/750
7/7 [==============================] - 0s 11ms/step - loss: 0.1522 - accuracy: 0.9505
Epoch 749/750
7/7 [==============================] - 0s 12ms/step - loss: 0.1471 - accuracy: 0.9579
Epoch 750/750
7/7 [==============================] - 0s 10ms/step - loss: 0.1579 - accuracy: 0.9508
<keras.callbacks.History at 0x7ff73664dbd0>
```

```
[ ]  1 model.evaluate(X_test,y_test)
```

```
27/27 [==============================] - 0s 3ms/step - loss: 1.8562 - accuracy: 0.6418
[1.8562036752700806, 0.6417556405067444]
```

```
[ ]  1 model.save("/content/gdrive/MyDrive/Capstone_Project/model_conv1D_h5")
```

```
Epoch 734/750
[290] 7/7 [==============================] - 0s 11ms/step - loss: 0.0687 - accuracy: 0.9780
Epoch 735/750
7/7 [==============================] - 0s 11ms/step - loss: 0.0716 - accuracy: 0.9745
Epoch 736/750
7/7 [==============================] - 0s 13ms/step - loss: 0.0725 - accuracy: 0.9780
Epoch 737/750
7/7 [==============================] - 0s 10ms/step - loss: 0.0744 - accuracy: 0.9769
Epoch 738/750
7/7 [==============================] - 0s 11ms/step - loss: 0.0648 - accuracy: 0.9778
Epoch 739/750
7/7 [==============================] - 0s 11ms/step - loss: 0.0781 - accuracy: 0.9727
Epoch 740/750
7/7 [==============================] - 0s 11ms/step - loss: 0.0663 - accuracy: 0.9786
Epoch 741/750
7/7 [==============================] - 0s 11ms/step - loss: 0.0803 - accuracy: 0.9751
Epoch 742/750
7/7 [==============================] - 0s 11ms/step - loss: 0.0788 - accuracy: 0.9739
Epoch 743/750
7/7 [==============================] - 0s 11ms/step - loss: 0.0723 - accuracy: 0.9778
Epoch 744/750
7/7 [==============================] - 0s 12ms/step - loss: 0.0822 - accuracy: 0.9730
Epoch 745/750
7/7 [==============================] - 0s 11ms/step - loss: 0.0717 - accuracy: 0.9757
Epoch 746/750
7/7 [==============================] - 0s 12ms/step - loss: 0.0657 - accuracy: 0.9810
Epoch 747/750
7/7 [==============================] - 0s 12ms/step - loss: 0.0741 - accuracy: 0.9763
Epoch 748/750
7/7 [==============================] - 0s 11ms/step - loss: 0.0767 - accuracy: 0.9748
Epoch 749/750
7/7 [==============================] - 0s 12ms/step - loss: 0.0590 - accuracy: 0.9807
Epoch 750/750
7/7 [==============================] - 0s 9ms/step - loss: 0.0649 - accuracy: 0.9780
<keras.callbacks.History at 0x7f33881bddd0>
```

```
1 model.evaluate(X_test,y_test)
```

```
27/27 [==============================] - 0s 3ms/step - loss: 1.4268 - accuracy: 0.7331
[1.4268007278442383, 0.7330960631370544]
```

Clear output

executed by Akhil Chaduvula
12:23 PM (5 minutes ago)
executed in 0.021s

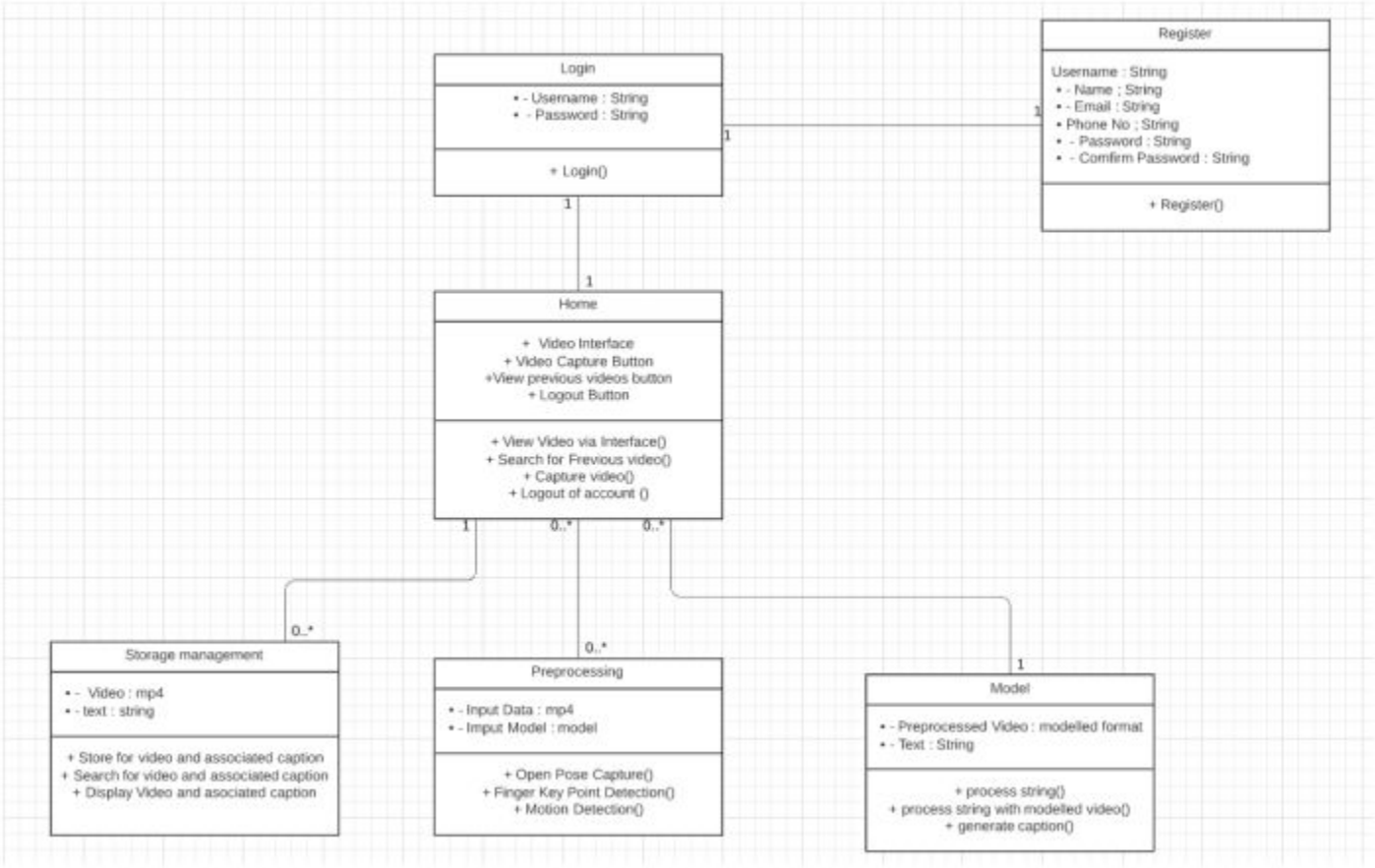
```
[29 /gdrive/MyDrive/Capstone_Project/model_conv1D.h5")
```

```
1
```

0s completed at 12:27 PM

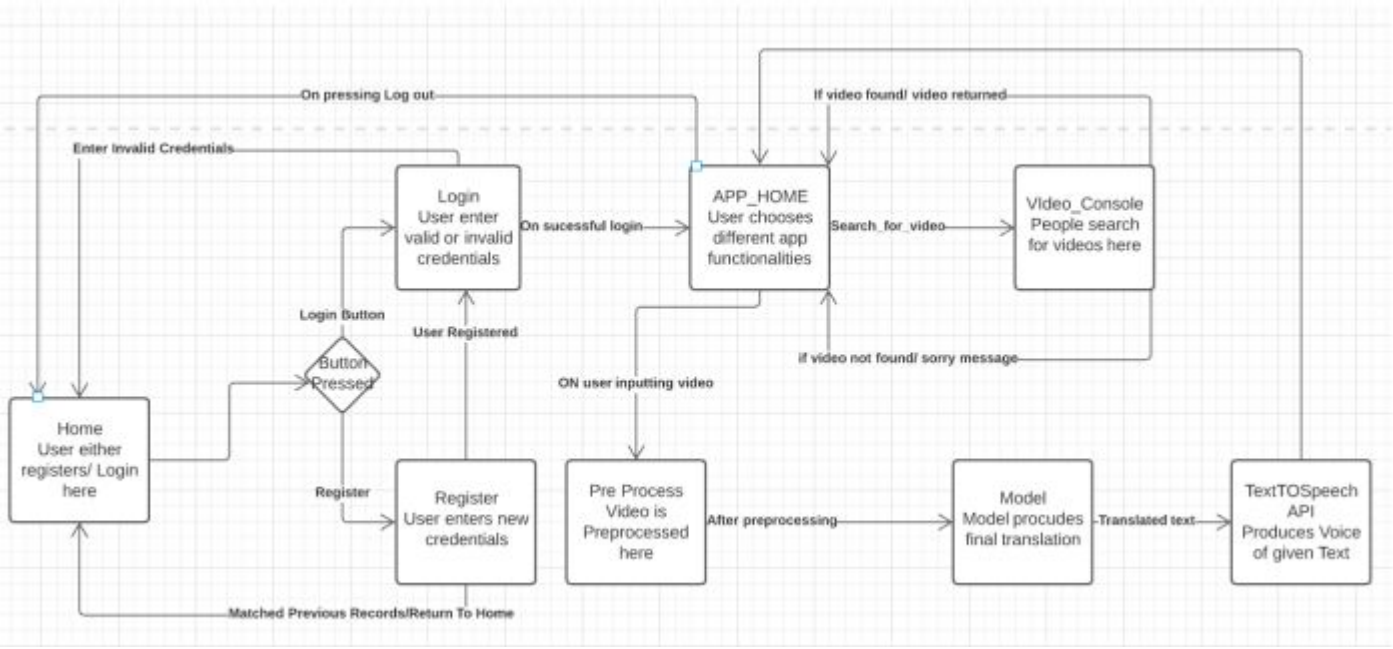
Design Description

Master Class Diagram



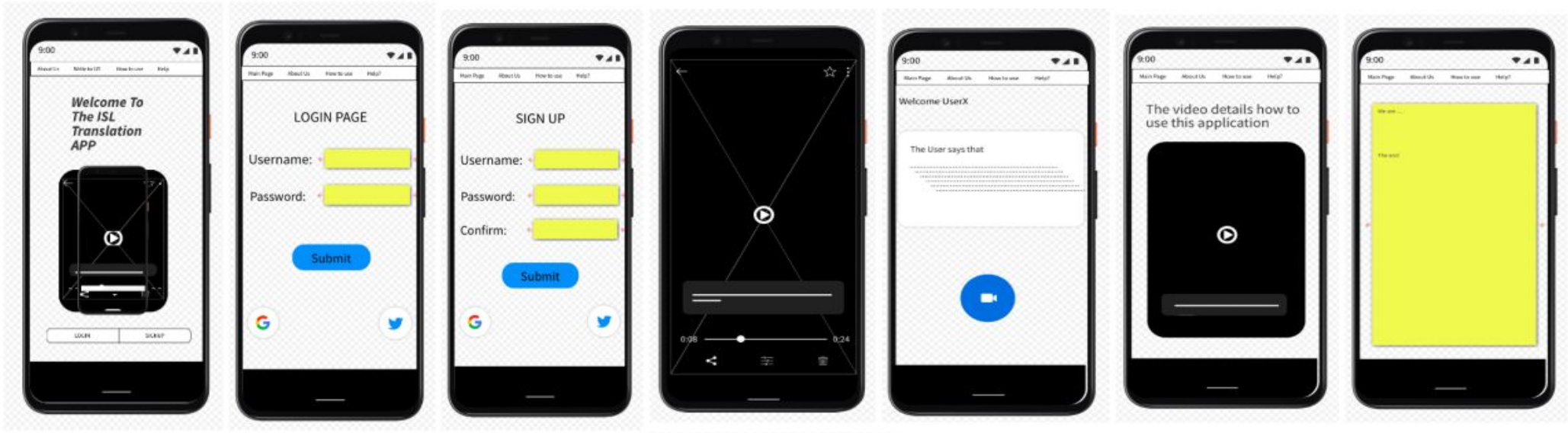
Design Description

State Diagram



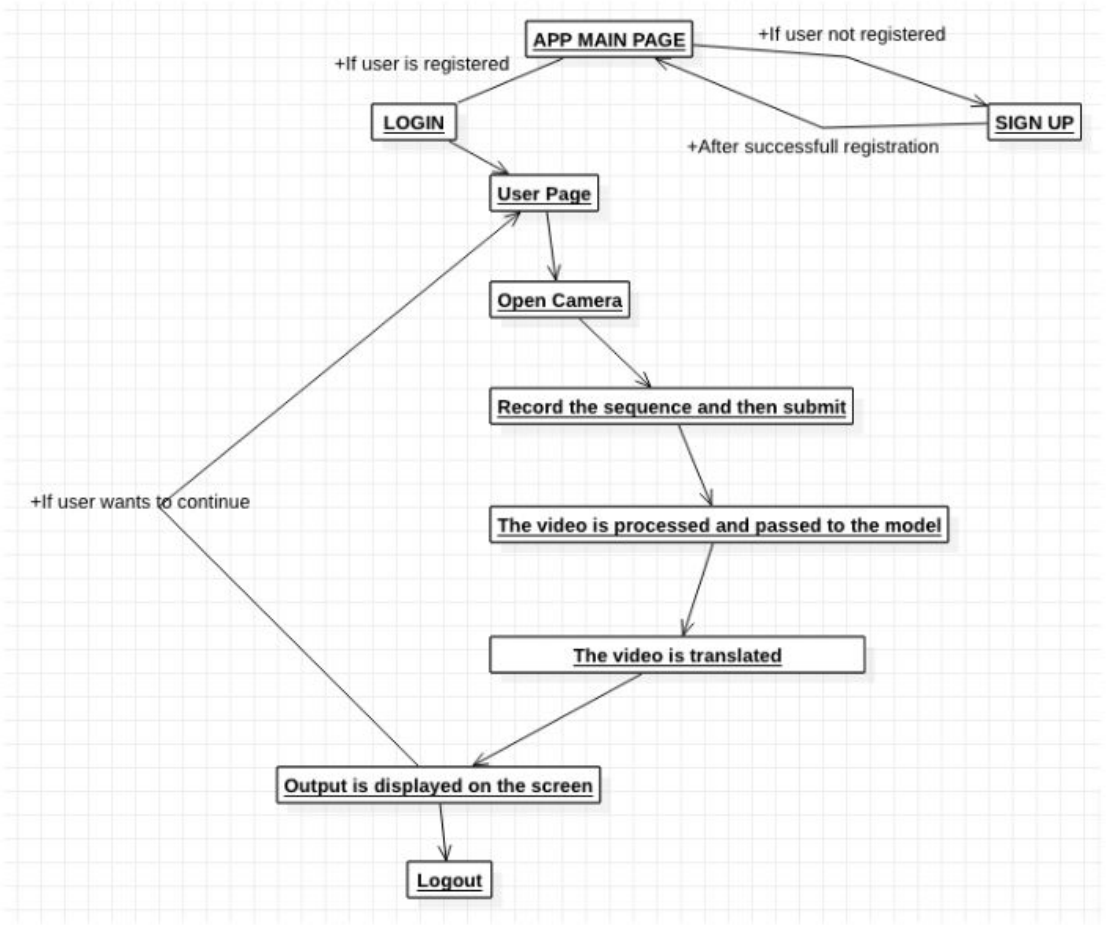
Design Description

User Interface Diagram



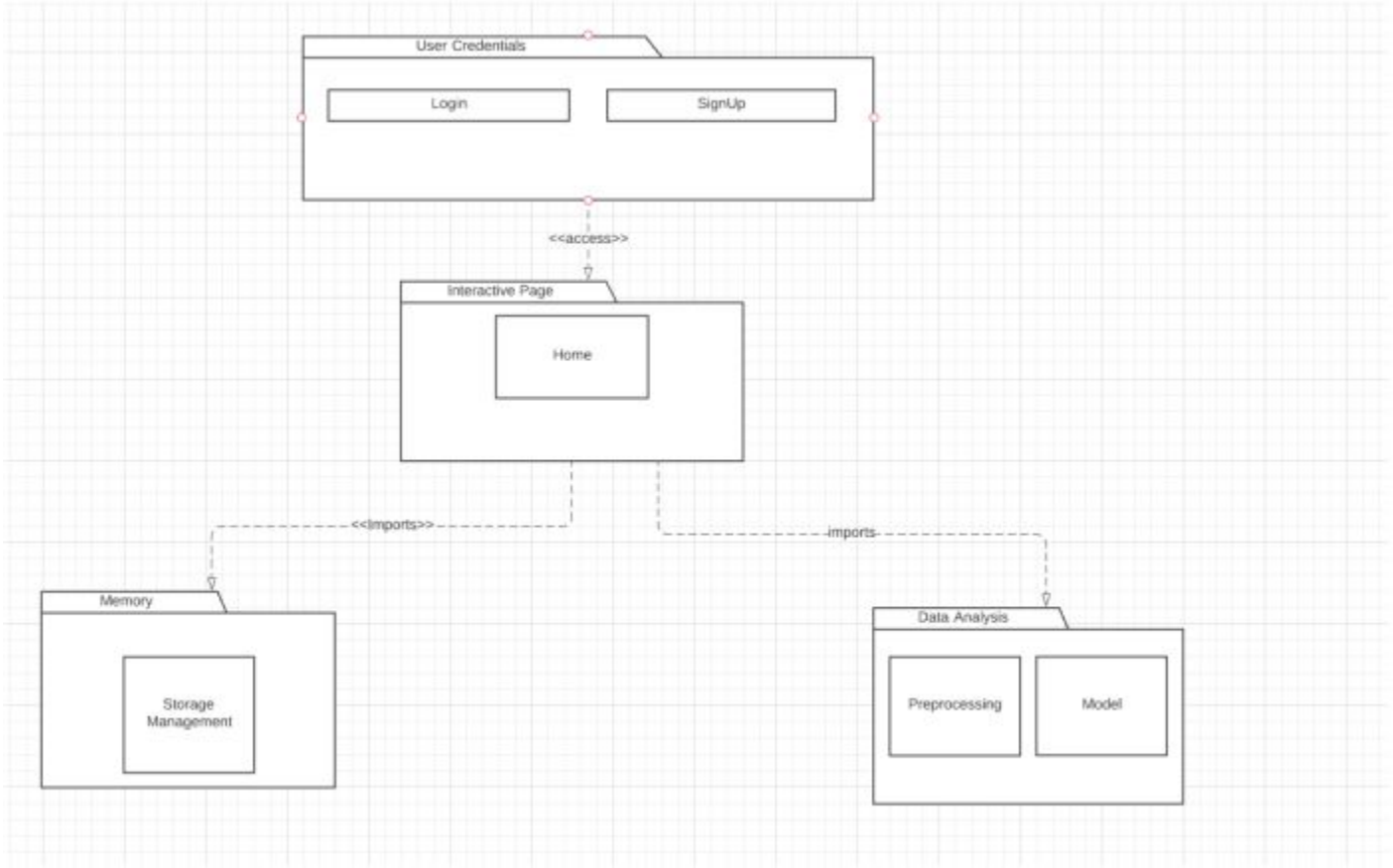
Design Description

External Interfaces



Design Description

Packaging and Deployment Diagram



Modules and Implementation Details

1) Login :- This Module is used to help a user log into their account. Using this module every user can Login to their account so that they can access the videos that they recorded. This contains the username and Password Fields.
2) Preprocessing :- In this module the video that is captured from the home screen is preprocessed and the output i.e a openpose Stick Diagram with keypoints is pushed to the next phase
3) Frame Extraction :- The preprocesses video files along with the names are bought into this module where the video is divided into 15 frames.
4) Feature Extraction :- The features are extracted from the frames and are saved in a dictionary with the name of the video as its key, so that classification can be done successfully.
5) Model :- This is the most important model of the entire project. In this step the features are being trained. The methods of modelling used here are RNN, LSTM, ResNet and Random Forest Model wherein the model all the models are combined together and are

Project Demonstration

- Exhibit the working demonstration of complete project.
- It has to be complete in all aspects
- Data set creation needs to be showed wherever applicable.

Note :

1. If any special facility required, please make sure that you have that set up. For example, if Internet connectivity is required, please have your mobile or hotspot set up.

Walkthrough

Demo and Product walk thru

Test Plan and Strategy

The testing Methodologies that we have implemented were

1. Unit testing: We tested all the semi functionalities in code and checked if they are working fine by trying with different inputs and parameters .We tested during implementation upon which we started moving on to the next step i.e Functional testing.
2. Functional Testing: In this step we tested various functionalities like openpose ,keypoint detection , video conversion to frames , model training, testing are working fine with various inputs and parameters and semi functionality integrity checks
3. 3. Integration Testing: Here we checked all the above modules integrated is working fine and producing the expected results without any dependency mismatches which finally resulted in a model with accuracy of 75 % which we are further trying to improve .

Results and Discussion

Results and discussions on the experimentation conducted after testing.

Are the results same as expected? Is it as per initial estimates planned? If there is a deviation, give the reasons for the change.

Results obtained in comparison other with other technology/methodology including graphs/charts (if applicable).

Product based projects can explain how your product meets the requirements. Clearly tie each test to the requirement (forward and backward traceability).

Documentation

Show the evidences, status of the below documents:

- Project report finalized by Guide?
- IEEE (similar) Format of Paper ready for submission or current status? **Which Conferences are you targeting? Have you submitted to any conference/journal? List out.**
- Video (2-3 minutes) of your project? Please Play.
- Add the Github repository link.
- A3 size Poster of your project to be shown.
- All artifacts of your project uploaded in the CSE Project repository?

Lessons Learnt

- We learned that building a model with 260 classes is very difficult and a good accuracy is hard to achieve.
- Integration of the backend to the frontend needs time.

Conclusion and Future work

As a part of the project we were able to complete the frontEnd that is interactive and the Backend Parts which is supporting the frontend parts.

We really are looking forward to working on improving the services provided in the App. We plan to simultaneously generate captions for the signs so that the process would appeal to more real time usage. Also instead of using stick figures for converting the captions to video we would like to use GAN's to generate an artificial human so that it would be more appealing to the end users.

References

1) Danielle Bragg, Oscar Koller, Mary Bellard, Larwan Berke, Patrick Boudreault, Annelies Braffort, Naomi Caselli, Matt Huenerfauth, Hernisa Kacorri, Tessa Verhoef,Christian Vogler, and Meredith Ringel Morris.2019. Sign Language Recognition, Generation, and Translation: An Interdisciplinary Perspective. In The 21st International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '19). Association for Computing Machinery, New York, NY, USA, 16–31. DOI:https://doi.org/10.1145/3308561.3353774

2) P. C. Badhe and V. Kulkarni, "Indian sign language translator using gesture recognition algorithm," 2015 IEEE International Conference on Computer Graphics, Vision and Information Security (CGVIS), Bhubaneswar, India, 2015, pp. 195-200, doi: 10.1109/CGVIS.2015.7449921.

3) Parton, Becky. (2006). Sign Language Recognition and Translation: A Multidisciplined Approach From the Field of Artificial Intelligence. Journal of deaf studies and deaf education. 11. 94-101. 10.1093/deafed/enj003.

4) N. C. Camgoz, S. Hadfield, O. Koller, H. Ney and R. Bowden, "Neural Sign Language Translation," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 7784-7793, doi: 10.1109/CVPR.2018.00812.

5) Akyol, S., & Alvarado, P. (2001). Finding relevant image content for mobile signlanguage recognition. Retrieved June 17, 2005, from http://www.techinfo.rwth-aachen.de/Veroeffentlichungen/V001_2001.pdf

6) Word-level Deep Sign Language Recognition from Video: A New Large-scale Dataset and Methods Comparison DONGXU LI, Cristian Rodriguez, Xin Yu, HONGDONG LI; Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 2020, pp. 1459-1469

Thank You