SLIIT
FACULTY OF COMPUTING

E-learning System For Hearing-Impaired Students

Project ID : 2021-176

SLIIT
FACULTY OF COMPUTING

Team Member

STUDENT ID	STUDENT NAME
IT18144772	Niroshan K
IT18069600	Accash R
IT18068610	Pirathikaran V
IT18152074	Sangeeth Raj A

Introduction

- Today Education has shifted from traditional learning style to Online. With this sudden shift many group of people encountered some problems.
- Online Learning for Hearing-impaired was a major problem.
- The Education sector needed a proper solution for hearing-impaired people to continue their learning.

Research

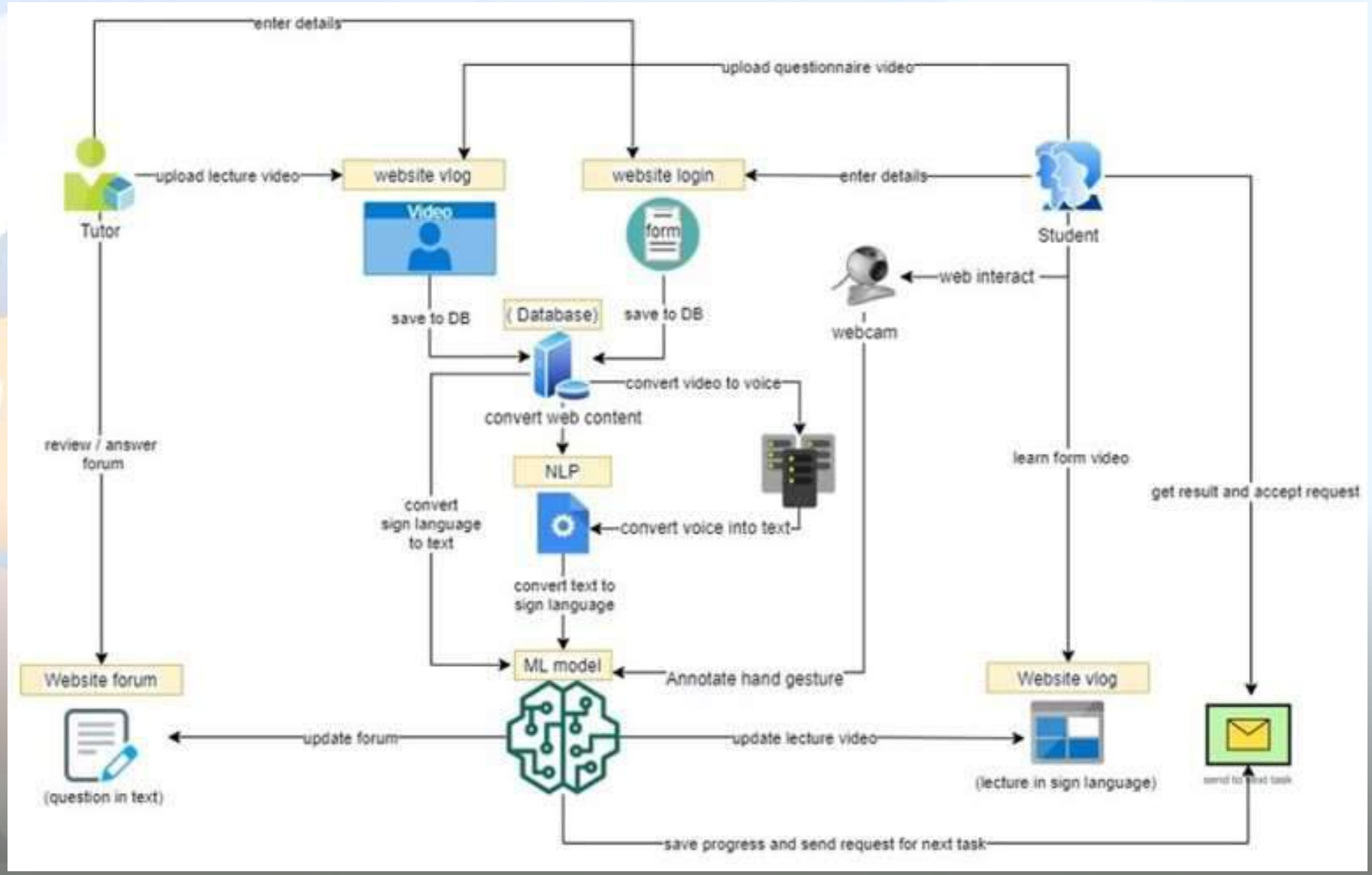
Implementing a learning environment which can be used by Hearing impaired students as well as tutors.

Objective

Objectives

- Enhancing the low-light videos and providing subtitles or Transcription in real time.
- Using the Transcription and generating sign language interpretation.
- Students clear doubts using sign language which can be converted into meaningful sentences.
- Teaching sign language for general Audience.

System Overview

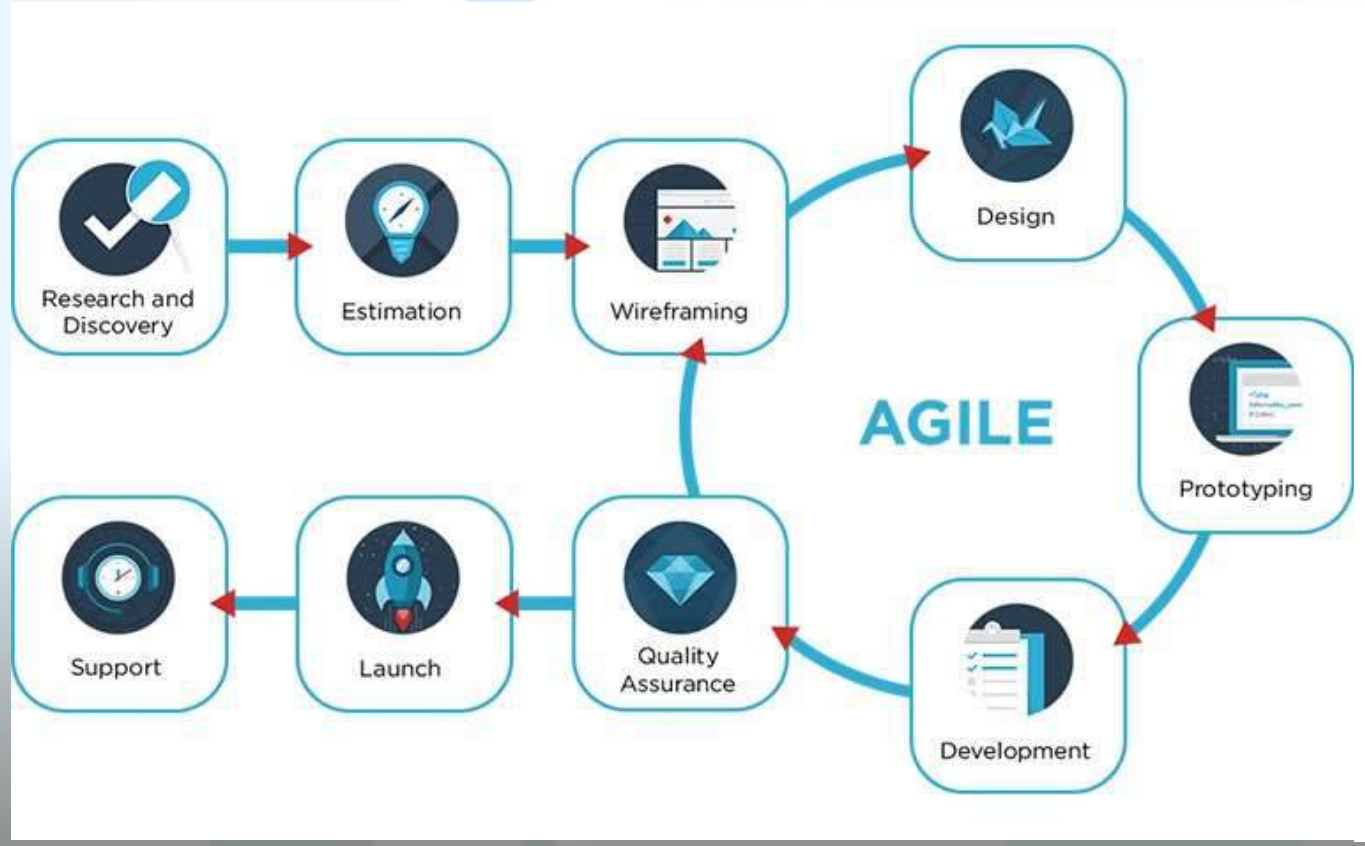


Methodology

The proposed LMS is divided into some main components

- **Enhancing uploaded video and producing caption for video content.**
- **Converting the captioned text to sign-language.**
- **Converting the Hearing-impaired student's video to meaningful text.**
- **Detecting user motion and analyzing the sign-language gesture for quiz.**

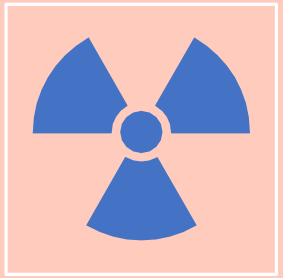
System Development



Business Potential

Provide as SaaS with one time subscription.

Can monetize the website using advertisements



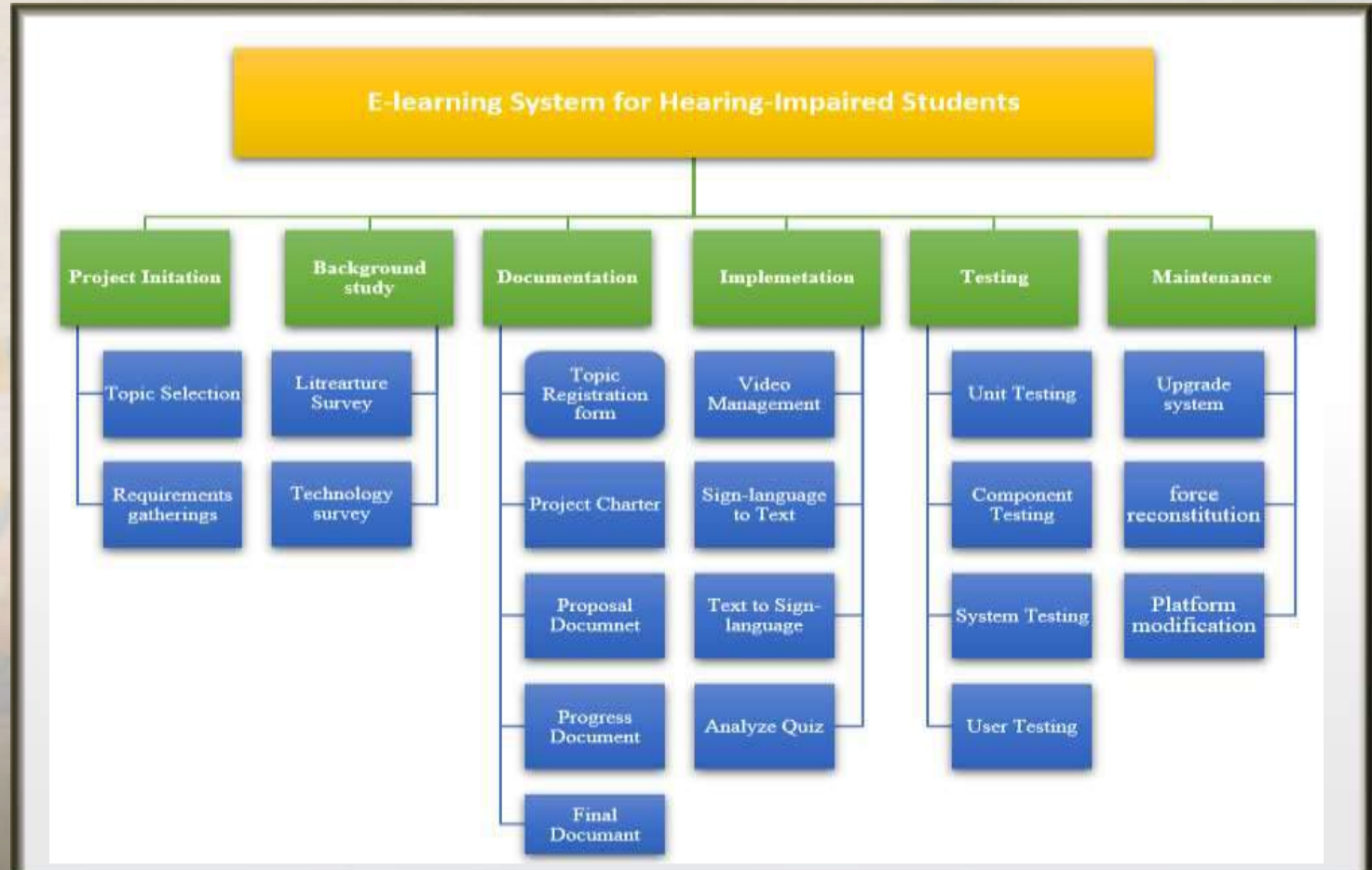
A Freemium plan.

Future Scope

Can be developed for other sign languages.

Can be developed for omni-platform.

WORK BREAK DOWN



SLIIT
FACULTY OF COMPUTING

Gantt Chart

Task Name | Timeline

Description	December	January	February	March	April	May	June	July	August	September	October	November	December
Project Initiation													
Evalation													
Topic Assessment form													
Charter													
Proposal Draft													
Proposal Presentation													
Project Phase													
System Planning													
Collecting Required Data													
Selecting Algoritham techonologies													
Implementation Phase													
Video Management													
Sign-language to Text													
Text to Sign-language													
Analyze Quiz													
Exprimental Analysis													
Testing Phase and Evaluation													
Testing													
Final Report and Research paper													
Final Evaluation													

SLIIT
FACULTY OF COMPUTING

IT18144772 – Niroshan K

Bachelor of Science (Hons) in Information Technology Specializing in Software Engineering

Research

- **Implementing low light algorithm for normal light images result in over exposed bright images.**

- **Sign Language cannot be directly interpreted from sound it needs to be converted to text format**[2].

Objective

Main Objective

- **Enhance the uploaded video and provide captions or transcripts for them.**

Sub Objective

Automated Video Enhancement

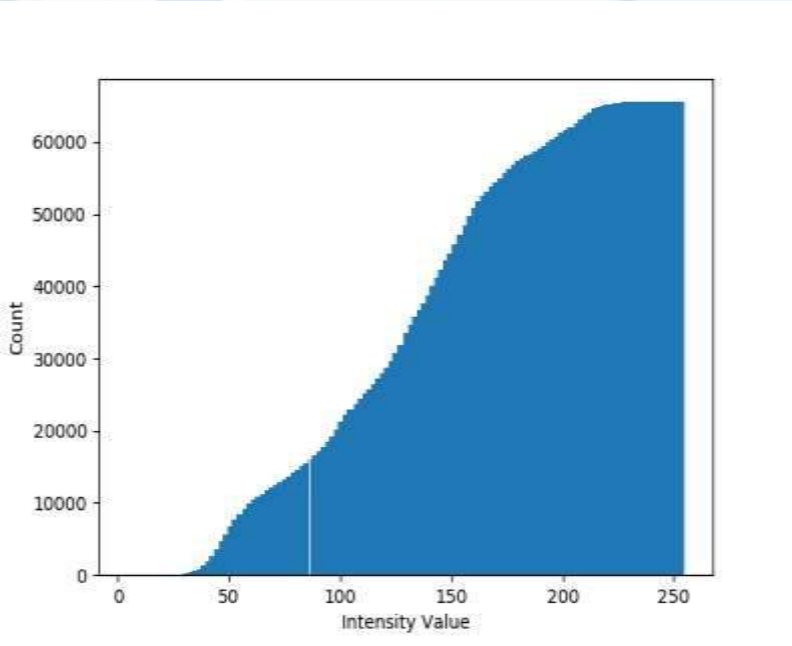
- **Use an algorithm to identify low light videos.**
- **Enhance the low light videos and reduce the noise in them**

Automated Captioning

- **Extracting Audio from the video content.**
- **Convert the Speech to Text using Speech-to-Text Model and the generated text as captions to the video**

Methodology

- **Construct cumulative intensity histogram for the image or frame of a video.**
- **Identify a threshold to differentiate low-light images and normal light images.**
- **Create an algorithm which can intelligently identify the low light images and normal light images separately.**
- **Use Low light enhancement techniques to improve the intensities of the low-light images and frames.**



Methodology

- For Automated Captioning, Audio is extracted from the video.
- Google's Speech-to-Text model will be used to extract the text output of the speech.
- Select an accurate timestamp to divide the transcript into individual sentences.

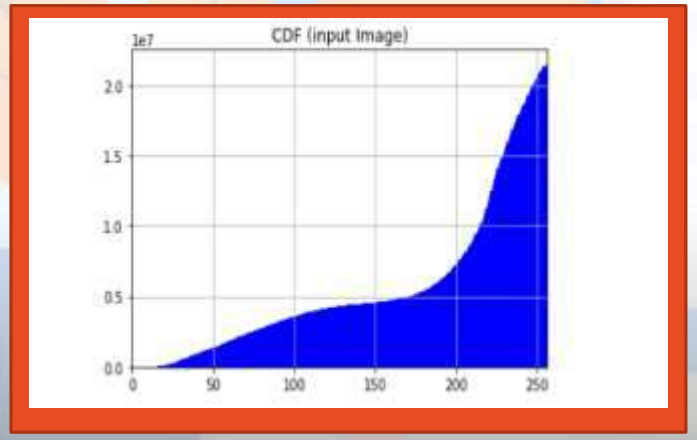
TESTING

- I used special low-light videos recorded using webcam and images taken from webcam as well as mobiles.

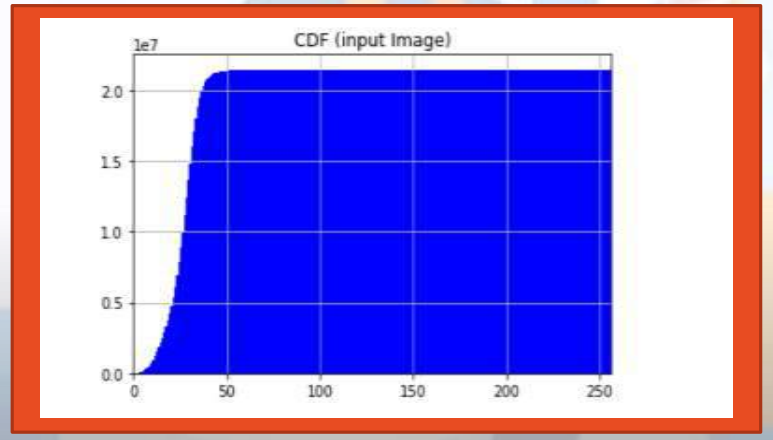
Completion of Project

Enhancement Technique

- **Initially Creating cumulative intensity histograms for images**



Normal Light Image



Low light Image

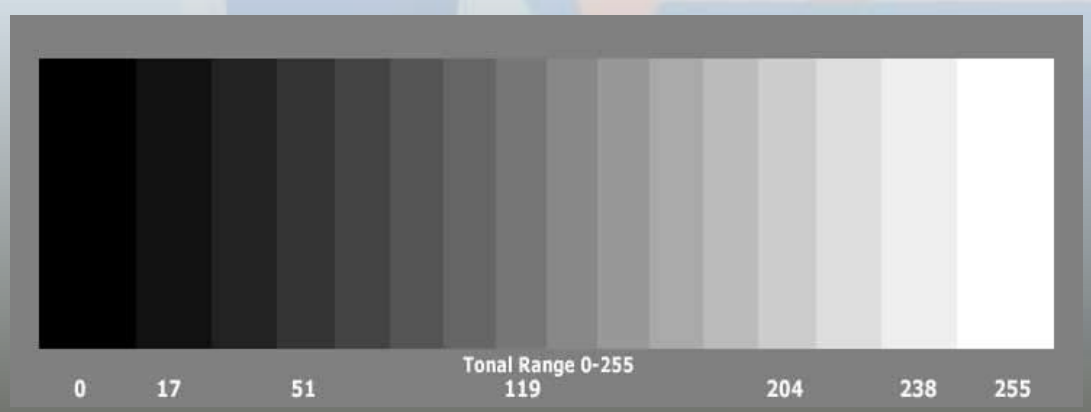
Completion of Project

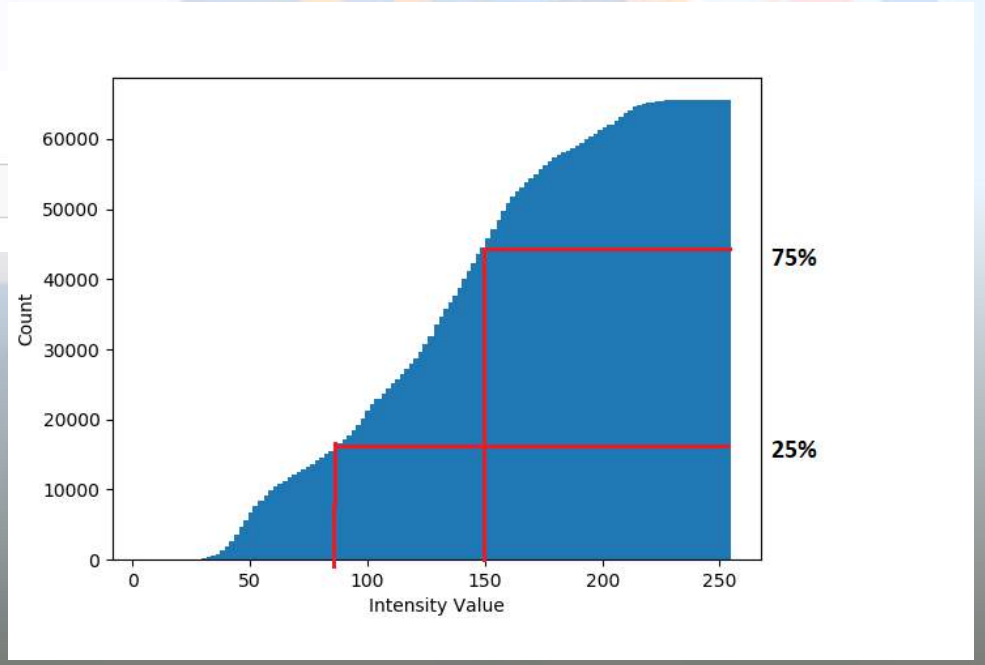
Enhancement Technique

- **Identifying a threshold value to separate the low light images from normal light images.**

Thresholding value to identify low light frames

```
In [126]: threshold = 103
```

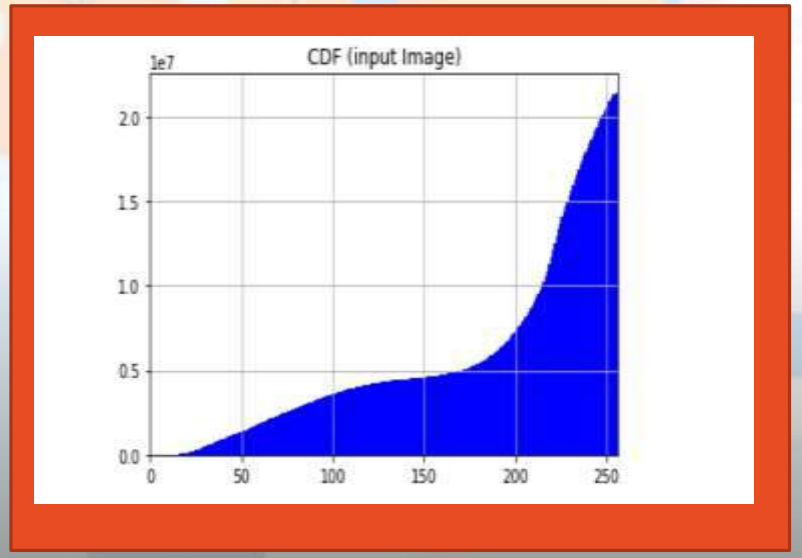




Completion of Project

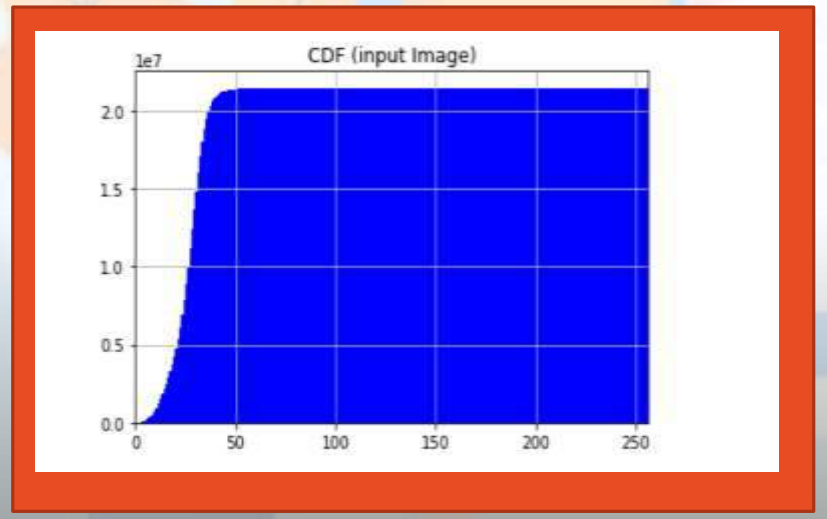
Enhancement Technique

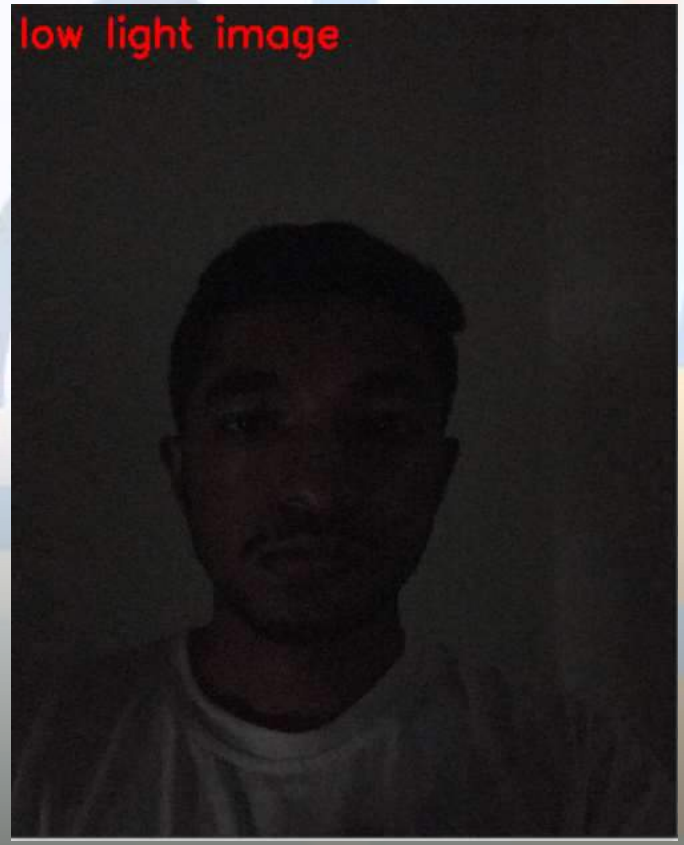
- **Implementing the algorithm to identify low-light images and normal light images.**



Completion of Project

Enhancement Technique





Completion of Project

Enhancement Technique

- **Enhancing low light images using the Gamma correction technique.**

This algorithm is then adopted for Videos.

Completion of Project

Captioning Technique

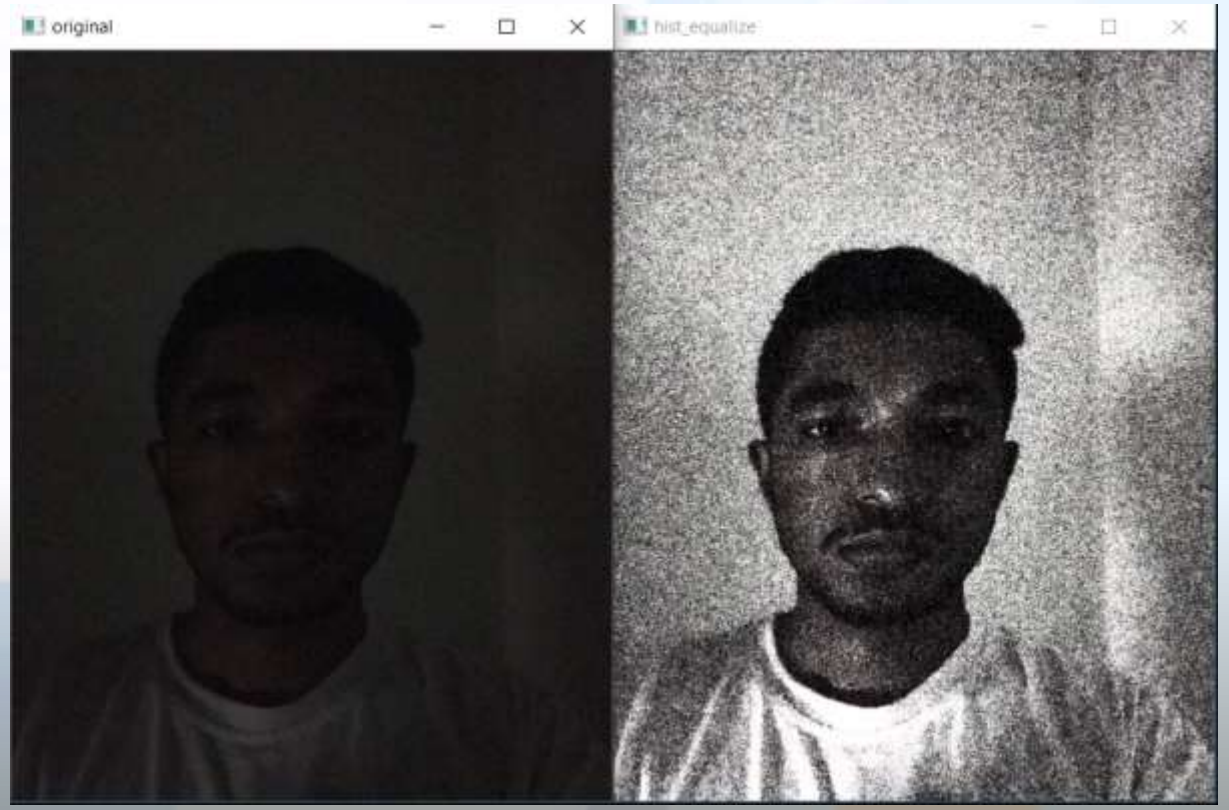
- **Audio File is extracted from the video file**
- **Transcription is generated using the Google STT model**
- **Algorithm for transcription to divided into sentences of 6 words using timestamps.**
- **Average time to read a word is taken as 0.5s [4].**

Completion of Project

Failed Attempts

- **Enhancing low light images using histogram equalization produces high noise in the image.**

- **Using gamma correction for videos in HSV format destroyed the low light parts of the video.**



SLIIT
FACULTY OF COMPUTING

Gantt Chart

Task Name		Timeline											
Description	December	January	February	March	April	May	June	July	August	September	October	November	December
Project Initiation													
Evalation													
Topic Assessment form													
Charter													
Proposal Draft													
Proposal Presentation													
Project Phase													
System Planning													
Collecting Required Data													
Selecting Algoritham techonologies													
Implementation Phase													
Implementing Algorithm													
Applying Video Enhancement													
Getting Captions for Videos													
Adding Captions for Video													
Research Paper													
Testing Phase and Evaluation													
Testing													
Final report													
Final Evaluation													

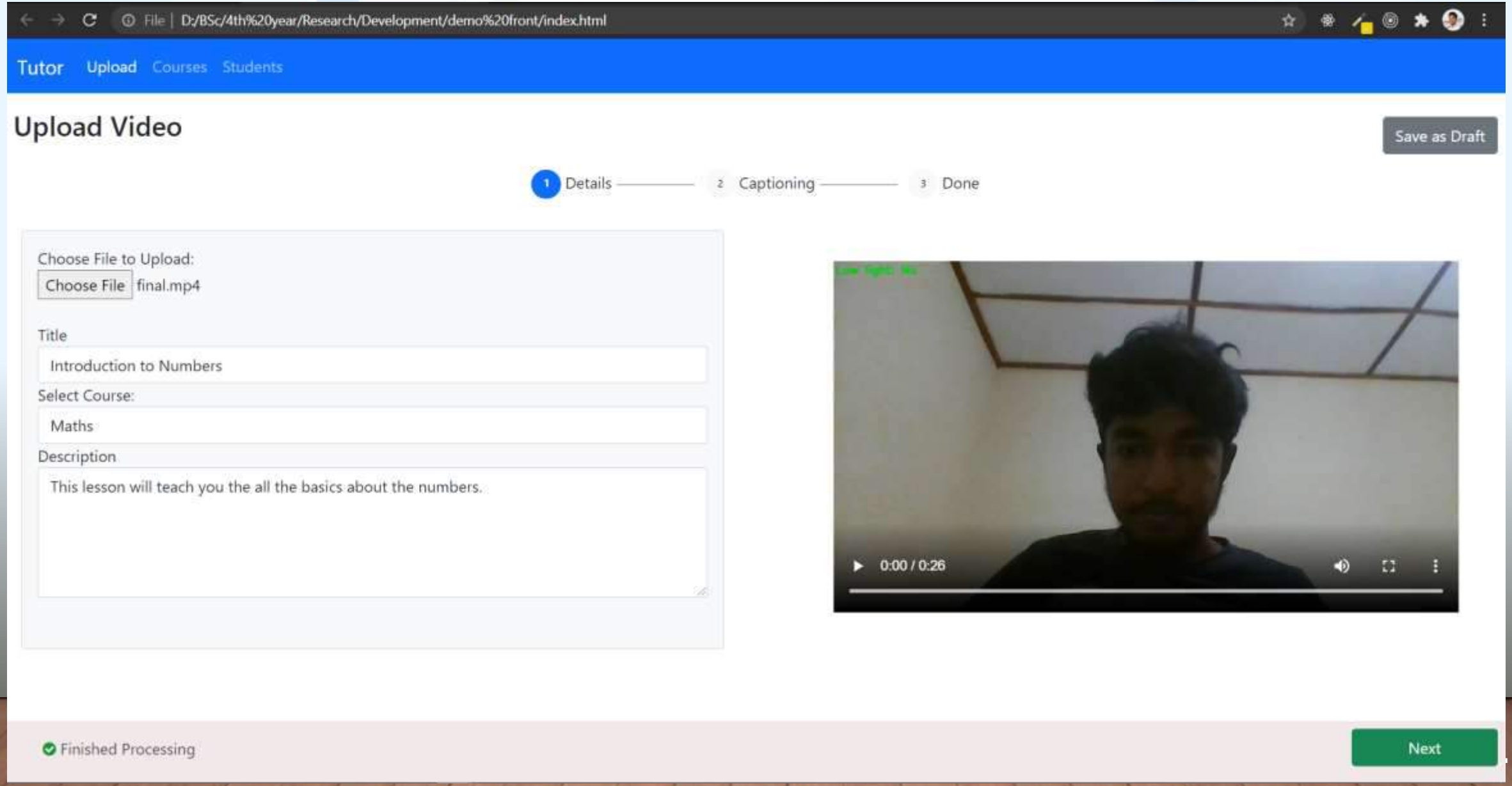
IT18144772 – Niroshan K.

Future Work

- **Working on Backend Services**
- **Working on Front End**

SLIIT
FACULTY OF COMPUTING

Sample User Interface



Technology & Tool Selection

Technologies

- Image/Video Processing
- Speech Recognition

Tools

- For Video Processing– OpenCV, Skimage
- For Speech Recognition– GCP STT
- For version controlling – GitLab
- Project Management – MS Planner

References

- [1] W. Farhan and J. Razmak, "A comparative study of an assistive e-learning interface among students with and without visual and hearing impairments," in *Disability and Rehabilitation Assistive Technology*, 2020.
- [2] R.Ranchel, Teresa, Y. Guo and K. Bain, "Using speech recognition for real-time captioning and lecture transcription in the classroom," in *IEEE Transactions of Learning Technologies*, 2013.
- [3]R. Krutsch and D. Tenorio, "Histogram Equalization," F*ree. Semicon. Doc. Number AN4318, Appl. Note,* 2011.
- [4]https://capitalizemytitle.com/reading-time/10-words/

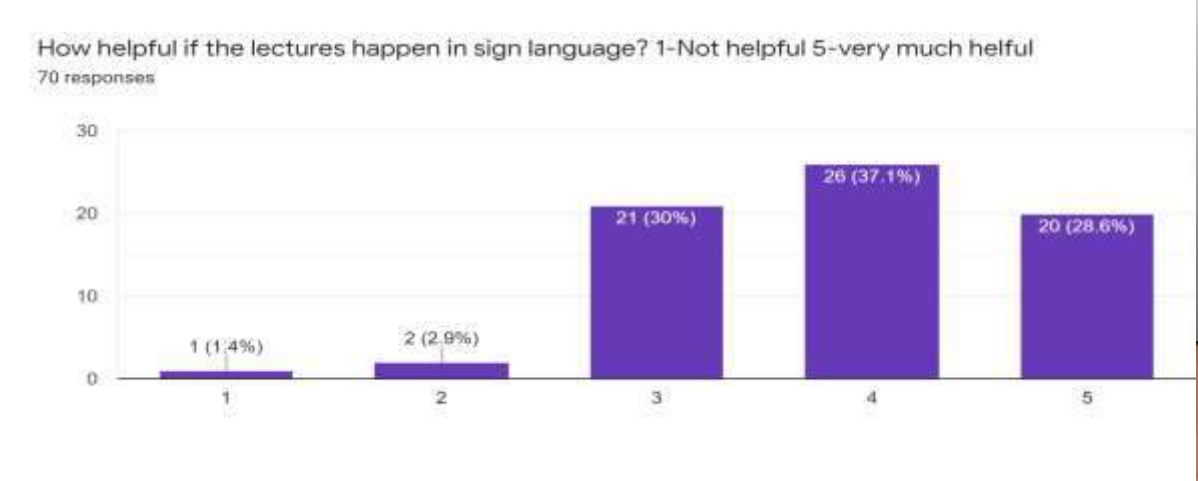
SLIIT
FACULTY OF COMPUTING

IT18069600 – Accash R.

Bachelor of Science (Hons) in Information Technology Specializing in Data Science

Research

- **Understanding the study content through the usual lecture videos in the LMS is difficult for the hearing-impaired students.**
- **Therefore, it is required that an automated mechanism for translation to sign language is developed.**
- **The module for translation will help hearing disabled people to understand in an efficient and easy way by providing them with a video to convey them the message of text.**



Objective

Main Objective

- **Converting lecture videos into sign language through the video captions (text) using Natural Language Processing technique.**

Sub Objective

- **Perform a complete analysis of the most used sign language which will be helpful to implement in the system.**
- **Design a user-friendly system to improve user interaction and user experience.**
- **Build a reliable translator within the system to ensure correct conveyance of the study material.**

Methodology

1. I used MS-ASL to download the video clips of each and every word.

2. The video clips of sign languages for each words will be found online and downloaded. They will be manually labelled and sorted.

3. The algorithm design will take place. Here,

4. a parser will be used to parse the English text.
   - The sentences will be reordered based on the ASL grammar rules.
   - An eliminator will be used for stop words removal.
   - Stemming will be done for getting the root words and replacing the synonyms.
   - Video conversion will be done.

Methodology

- **Video conversion**

In this final step, the ASL transformed text will be made to find matches from the downloaded video data set available for each word, using its label.

Then, a set of videos will be displayed as a sequence on the screen, representing the captions of the lecture video.

Project Requirements

Functional requirements

- Converting the extracted text / captions to the sign language.

Non-Functional requirements

- Less manual work to translate into sign language.
- Take less time to covert text to the sign language.
- Accurate translation.

Technology & Tool Selection

Technologies

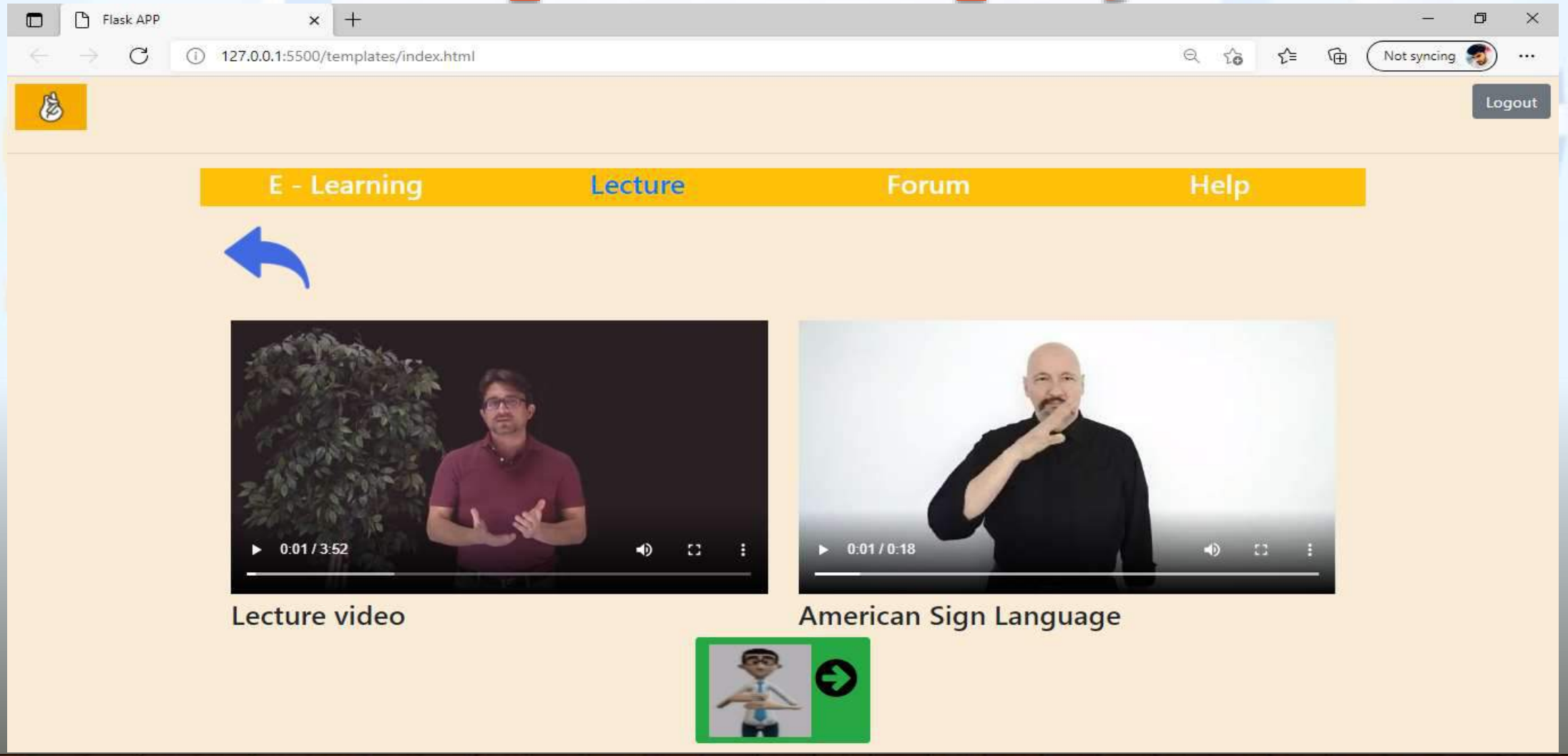
- Natural Language Processing

Tools

- Natural Language Processing - NLTK
- For version controlling – GitLab
- Project Management – MS Planner

SLIIT
FACULTY OF COMPUTING

Completion of project



Completion of project

```
inputString = " "

java_path = "C:\\Program Files\\Java\\jdk-13.0.1\\bin\\java.exe"
os.environ['JAVAHOME'] = java_path

for each in range(1, len(sys.argv)):
    inputString += sys.argv[each]
    inputString += " "

# inputString = raw_input("Enter the String to convert to ISL: ")
inputString = "I am going to School to do my Presentation tomorrow."

# D:\accash\stanford-postagger-full-2015-12-09\models
parser = StanfordParser(
    model_path='D:/accash/stanford-parser-full-2015-12-09/edu/stanford/nlp/models/lexparser/englishPCFG.ser.gz')

o = parser.parse(inputString.split())

englishtree = [tree for tree in parser.parse(inputString.split())]
parsetree = englishtree[0]

dict = {}
```

Completion of project

```
(NP (PRP$ my) (NNP Presentation) (NN tomorrow.))
3
1
(PRP$ my)
1
0
(NNP Presentation)
1
0
(NN tomorrow.)
1
0
(PRP$ my)
1
0
(NNP Presentation)
1
0
(NN tomorrow.)
1
0
school present tomorrow. go
PS D:\accash>
```

Future work

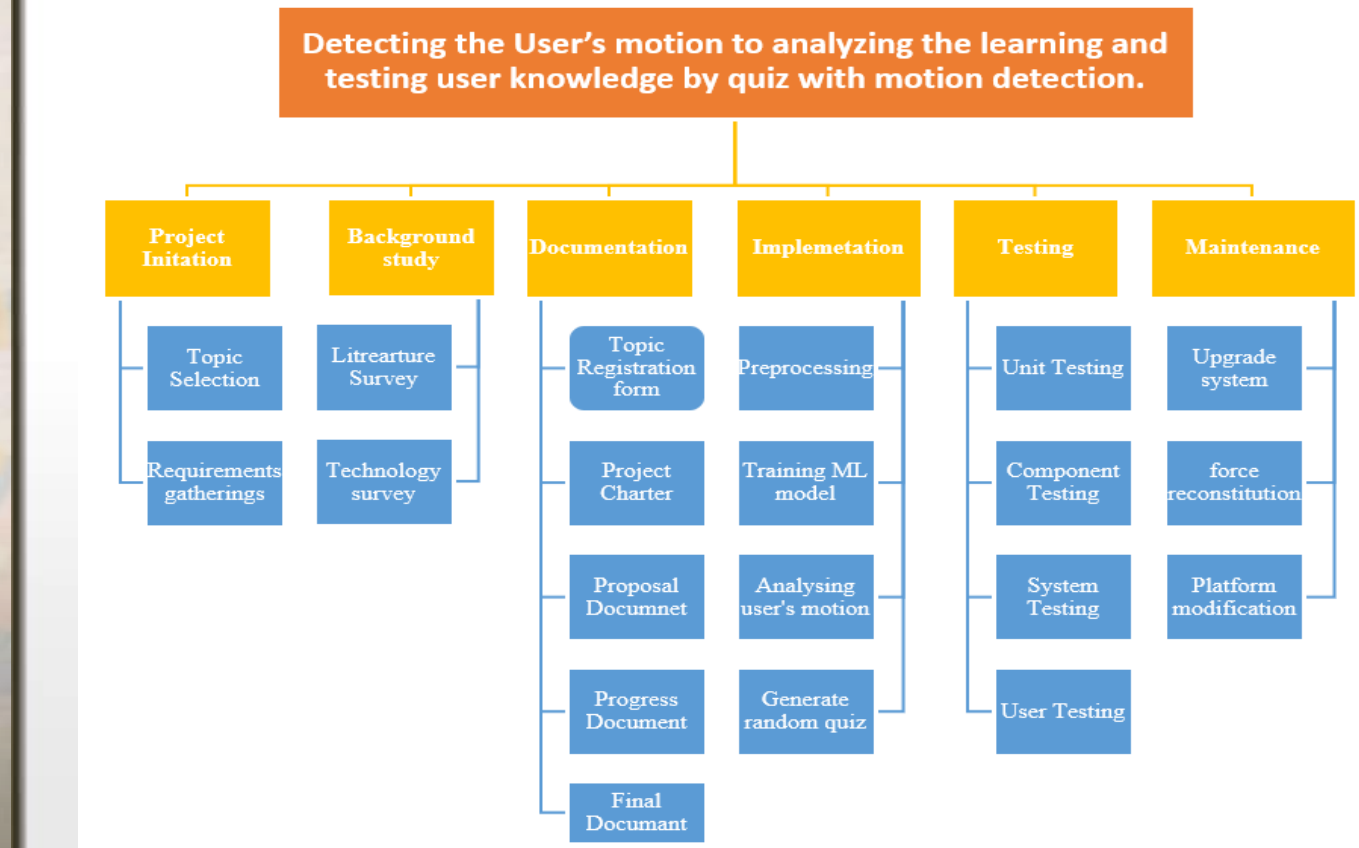
- Working on Backend Services
- Working on Front End

SLIIT
FACULTY OF COMPUTING

Gantt Chart - Function

Description	December	January	February	March	April	May	June	July	August	September	October	November	December
Project Initiation													
Evalation													
Topic Assessment form													
Charter													
Proposal Draft													
Proposal Presentation													
Project Phase													
System Planning													
Collecting Required Data													
Selecting Algorithm technologies													
Implementation Phase													
NLP Preprocessing													
Model Creation													
Connecting Models													
Experimental Analysis													
Testing Phase and Evaluation													
Research Paper													
Testing													
Final report													
Final Evaluation													

FUNCTION WORK BREAK DOWN



References

[1] T. Jamil, "Design and Implementation of an Intelligent System to translate Arabic Text into Arabic Sign Language," 2020. [Online]. Available:

Design and Implementation of an Intelligent System to translate Arabic Text into Arabic Sign Language - IEEE Conference Publication

[2] M. S. Nair, N. A. P and S. M. Idicula, "Conversion of Malayalam Text to Indian Sign Language Using Synthetic Animation," 2016. [Online]. Available:

Conversion of Malayalam text to Indian sign language using synthetic animation - IEEE Conference Publication

[3] M. M. Nasr, "An Enhanced e-Learning Environment for Deaf/HOH Pupils," 2010. [Online]. Available:

An enhanced e-learning environment for Deaf/HOH pupils - IEEE Conference Publication

[4] A. Drigas, D. Kouremenos, S. Kouremenos and J. Vrettaros, "An e-Learning System for the Deaf people," 2005. [Online]. Available:

An e-learning system for the deaf people - IEEE Conference Publication

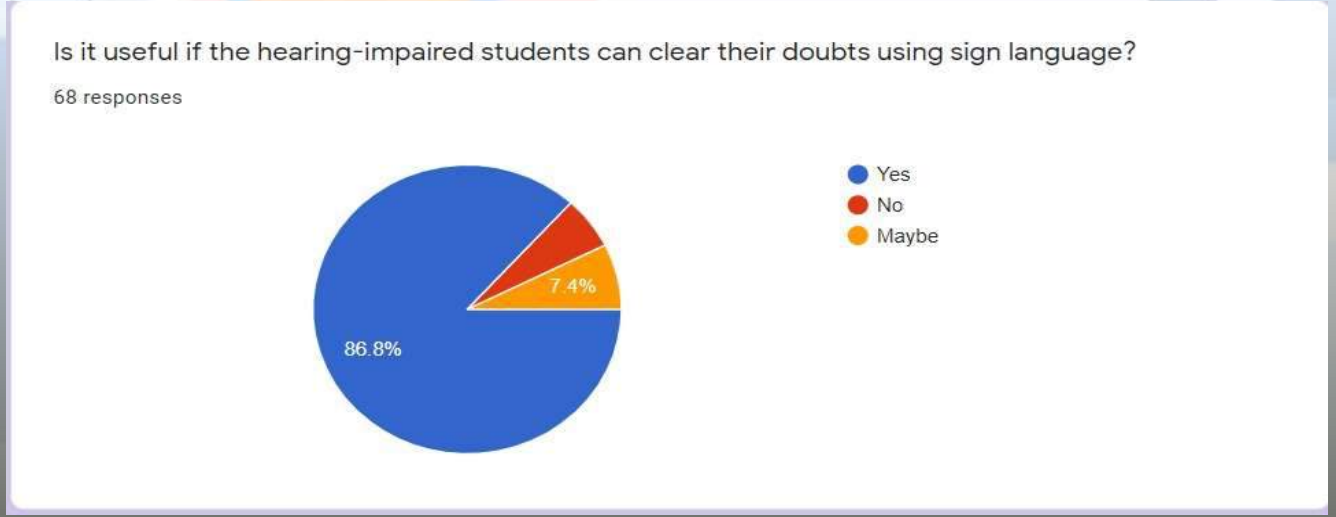
SLIIT
FACULTY OF COMPUTING

IT18068610 – Pirathikaran V.

Bachelor of Science (Hons) in Information Technology Specializing in Software Engineering

Introduction

- **Hearing-impaired students want to communicate with others.**
- **Hearing-impaired can ask a question and clarify with the tutor.**
- **Deaf and dumb students can overcome their education issues and, all students encourage to follow their education.**



Research Gap & Problem

- **Students who are deaf and dumb have not yet fully utilized ways to voice their doubts.**
- **Two-way communication is not yet in use in e-learning platform.**

Paper	Tasks	Limitation	Our system
[1]	Using Two-way hand gesture	Developed but Not accuracy	Yes Accuracy level High
[2]	Sign language gestures detect word	Only detect Letters and Numbers	Yes Detect word and letters also
[3]	Make Two way communication	Developed but Cannot used in e-Learning Platform	Yes We using two way communication
[4]	Easy to use	Only using glove can detect sign language	Yes Without glove using video only

Research Gap & Problem

- **A tutor cannot understand sign language.**
- **Deaf and dumb student only way to communicate with ordinary people through sign language.**
- **Each country has unique sign languages.**

Objective

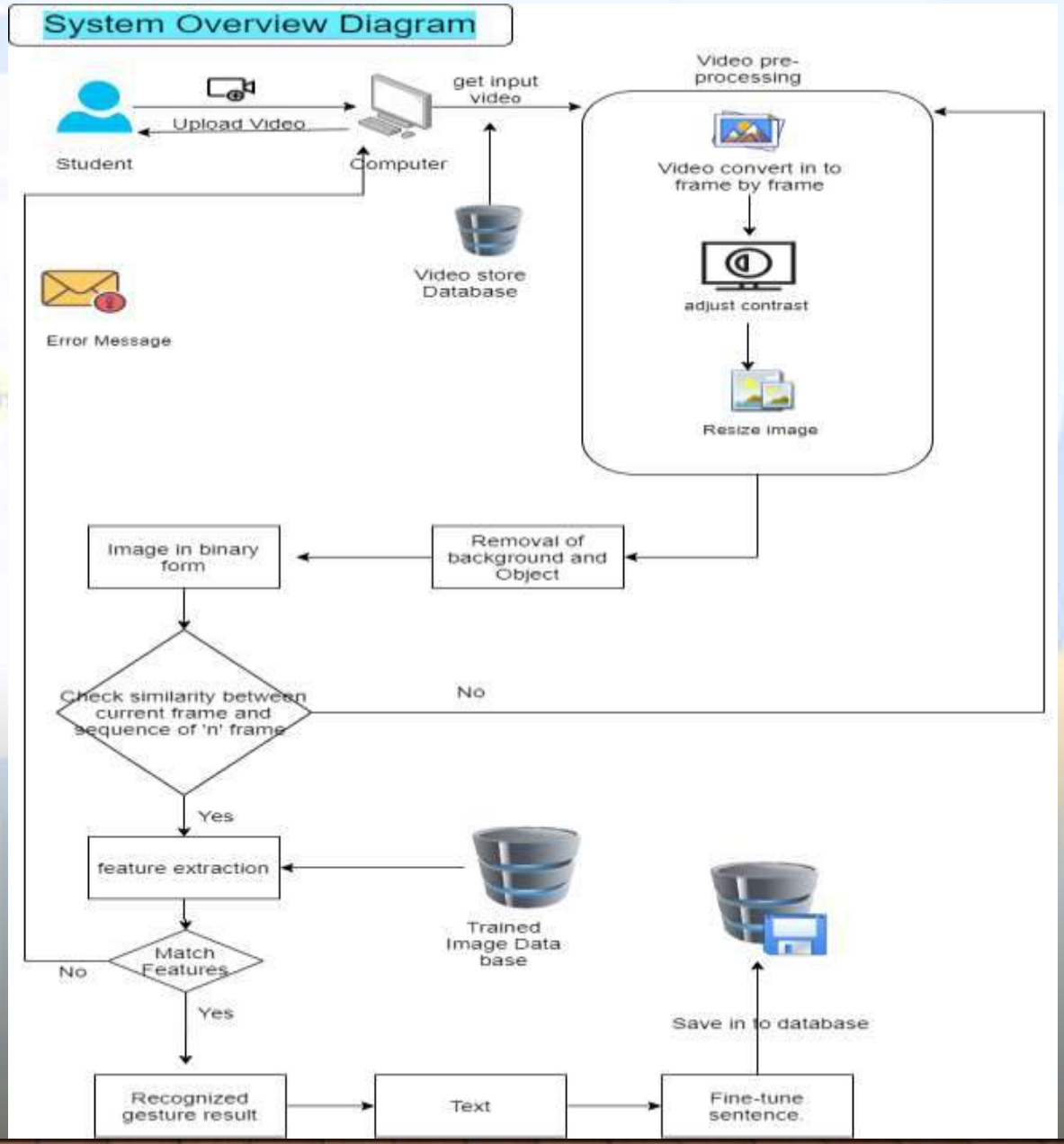
Main Objective

- Recognize sign language and convert it into Text.

Sub Objective

- Do the preprocessing video and get frame by frame.
- Removal of background and objects
- Convert Image in binary form.
- Feature Extraction
- Recognize text and fine tune to native language.

Function Overview



Methodology

- Taking input video and do the video pre-processing.
- In the pre-processing video convert into frame by frame
- Adjust contrast
- Image resize
- Image background and object removal
- Image into binary form
- Feature Extraction
- Training ML model data set from own dataset
- Recognize gestures text
- Fine tune the text to native language

Project Requirements

Functional requirements

- Converting sign language into text and fine-tune.

Non-Functional requirements

- Less manual work to translate sign language.
- Take less time to covert the sign language into text.
- Accurate recognition.

Technology & Tool Selection

Technologies

- **Video Processing**

Tools

- **For Video Processing– OpenCV**
- **For version controlling – GitLab**
- **Project Management – MS Planner**

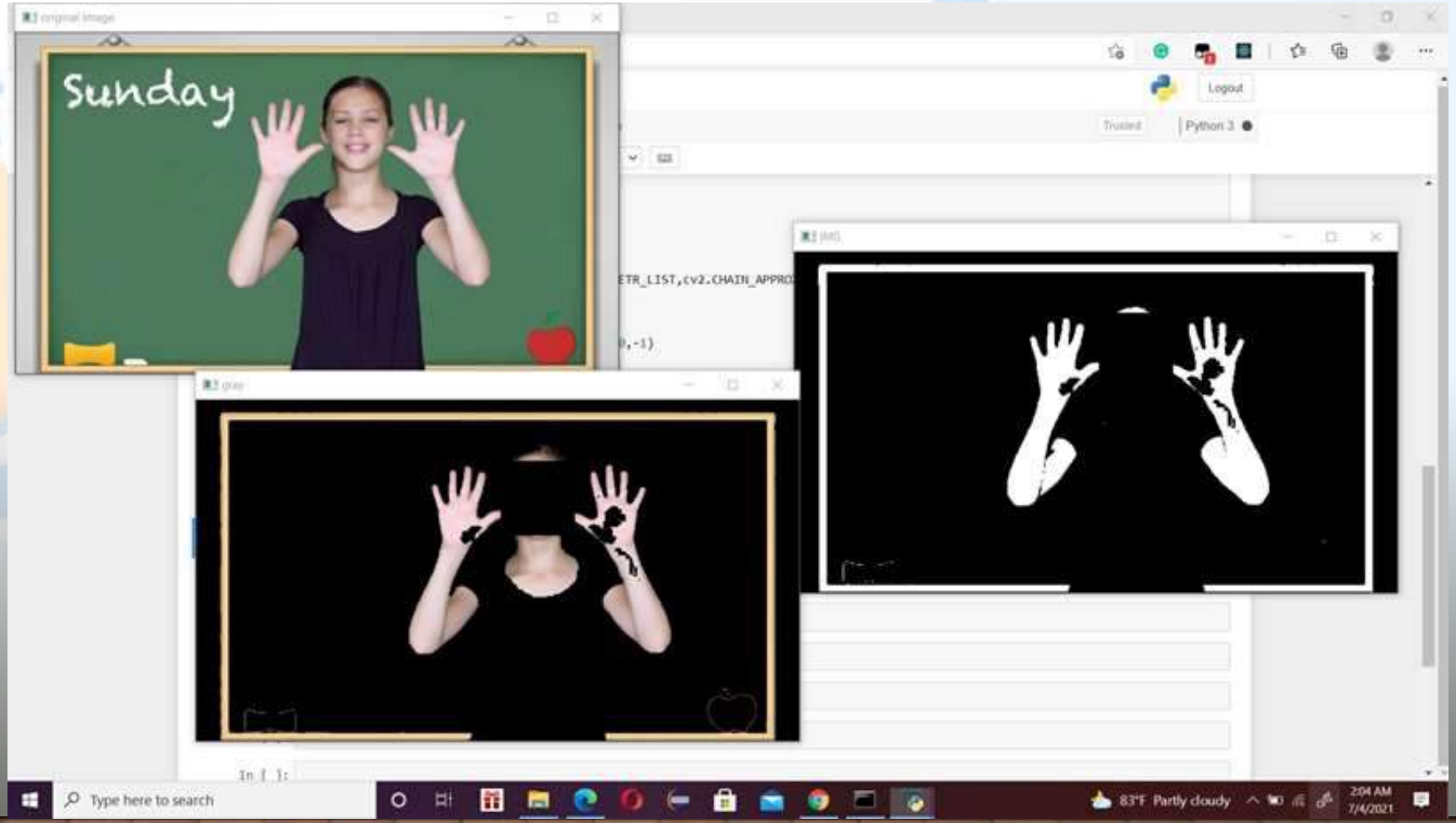
Completion of the project

Pre processing

Resize Image

Find skin color area

Hide face



Failed attempts

- **MS ASL data set not accurate**

Model accurancy

In [180]:
```
cr = sklearn.metrics.classification_report(y_test,y_pred_test,output_dict=True)
pd.DataFrame(cr).T
```

Out[180]:

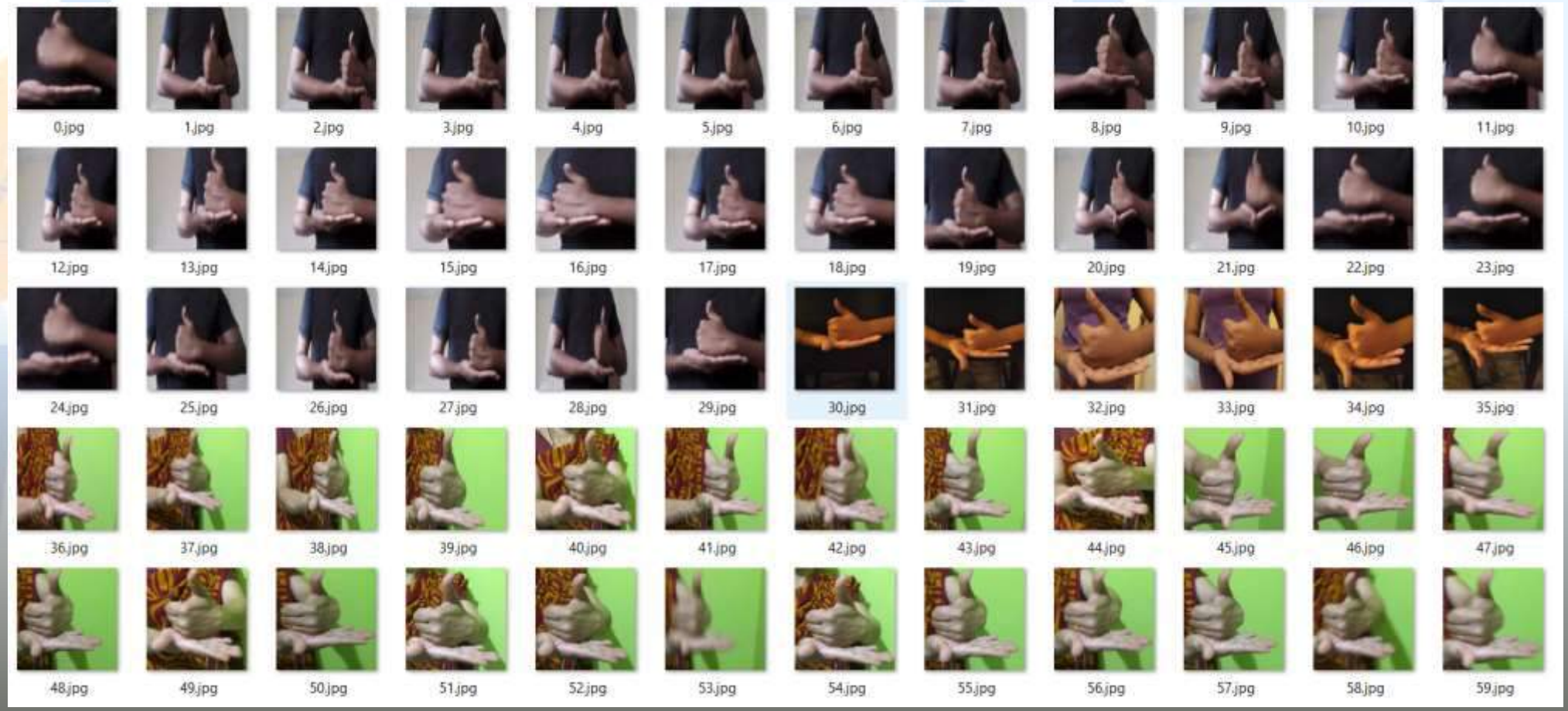
	precision	recall	f1-score	support
book	1.0	1.0	1.0	20.0
boring	1.0	1.0	1.0	17.0
easter	1.0	1.0	1.0	26.0
fail	1.0	1.0	1.0	21.0
germany	1.0	1.0	1.0	12.0
library	1.0	1.0	1.0	33.0
like	1.0	1.0	1.0	14.0
phone	1.0	1.0	1.0	31.0
signlanguage	1.0	1.0	1.0	23.0
accuracy	1.0	1.0	1.0	1.0
macro avg	1.0	1.0	1.0	197.0
weighted avg	1.0	1.0	1.0	197.0

In [179]:
```
metrics.cohen_kappa_score(y_test,y_pred_test)
```

Out[179]: 1.0

Completion of the project

Own dataset



Completion of the project

Labelling

```
In [21]: data['data'].shape

Out[21]: (486, 200, 200, 3)
```

```
In [22]: plt.figure(figsize=(12,6))
         for i,c in enumerate(data['labels']):
             index=data['target'].index(c)
             img=data['data'][index]

             plt.subplot(3,10,i+1)
             plt.imshow(img)
             plt.xticks([]), plt.yticks([])
             plt.title(c)
         plt.show()
```

Completion of the project

Model create and Training

Model Evaluation

In [41]:
```
cr = sklearn.metrics.classification_report(y_test,y_pred_test,output_dict=True)
pd.DataFrame(cr).T
```

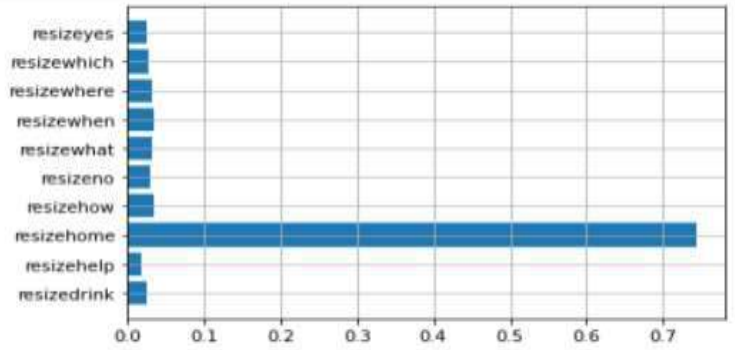
Out[41]:

	precision	recall	f1-score	support
resizedrink	1.000000	1.000000	1.000000	5.000000
resizehelp	1.000000	1.000000	1.000000	15.000000
resizehome	0.500000	0.200000	0.285714	5.000000
resizehow	0.750000	1.000000	0.857143	6.000000
resizeno	0.636364	0.700000	0.666667	10.000000
resizewhat	0.857143	0.923077	0.888889	13.000000
resizewhen	0.928571	0.866667	0.896552	15.000000
resizewhere	0.800000	0.800000	0.800000	10.000000
resizewhich	0.800000	0.923077	0.857143	13.000000
resizeyes	0.500000	0.333333	0.400000	6.000000
accuracy	0.826531	0.826531	0.826531	0.826531
macro avg	0.777208	0.774615	0.765211	98.000000
weighted avg	0.814644	0.826531	0.814130	98.000000

In [42]:
```
metrics.cohen_kappa_score(y_test,y_pred_test)
```

Out[42]:
```
0.8034218289085546
```

In [17]:
```
plt.barh(labels,prob_value)
plt.grid()
```



In [18]:
```
# top five probabilty values
top_5_prob_ind = prob_value.argsort()[::-1][:5]
```

In [19]:
```
top_5_prob_ind
```

Out[19]:
```
array([2, 6, 3, 5, 7], dtype=int64)
```

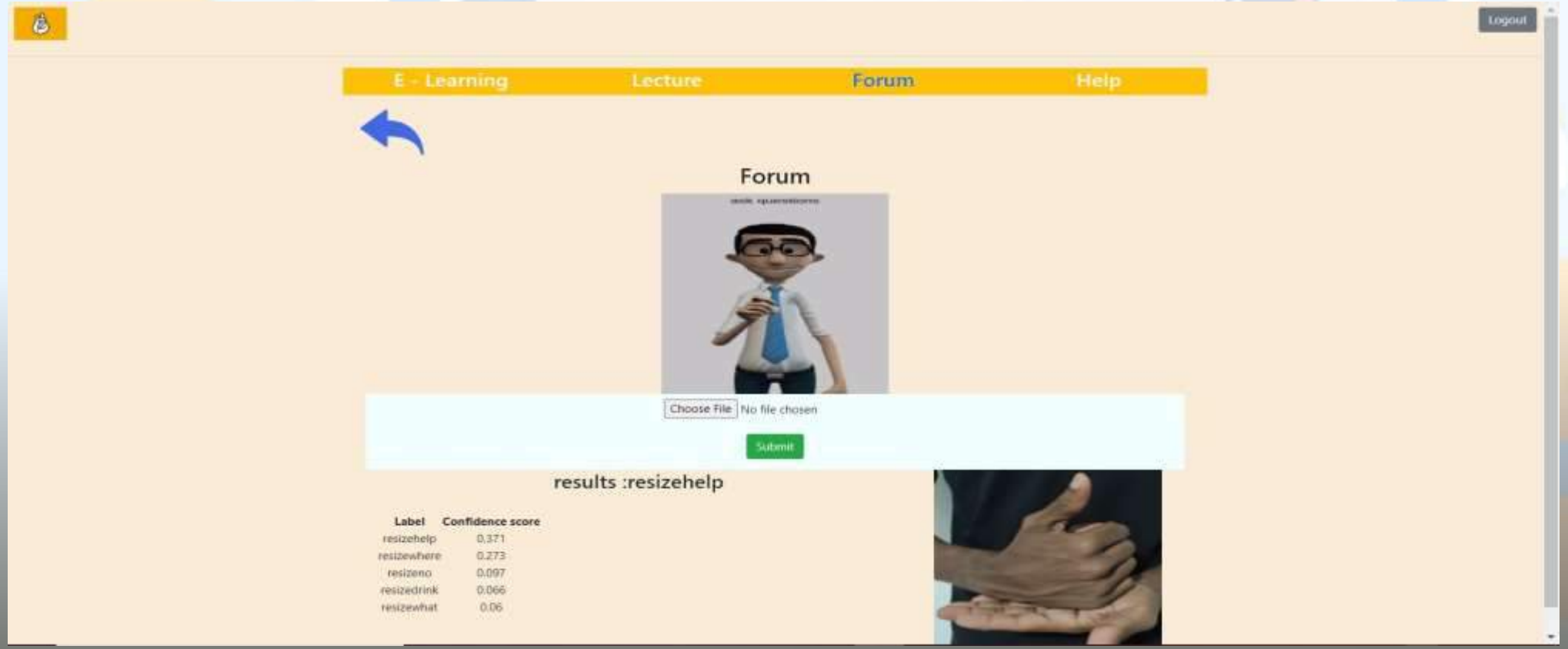
In [20]:
```
top_labels = labels[top_5_prob_ind]
top_prob = prob_value[top_5_prob_ind]
```

In [21]:
```
top_prob,top_labels
```

Out[21]:
```
(array([0.74374656, 0.03406137, 0.03361478, 0.03219125, 0.03138241]),
 array(['resizehome', 'resizewhen', 'resizehow', 'resizewhat',
        'resizewhere'], dtype='<U11'))
```

Completion of the project

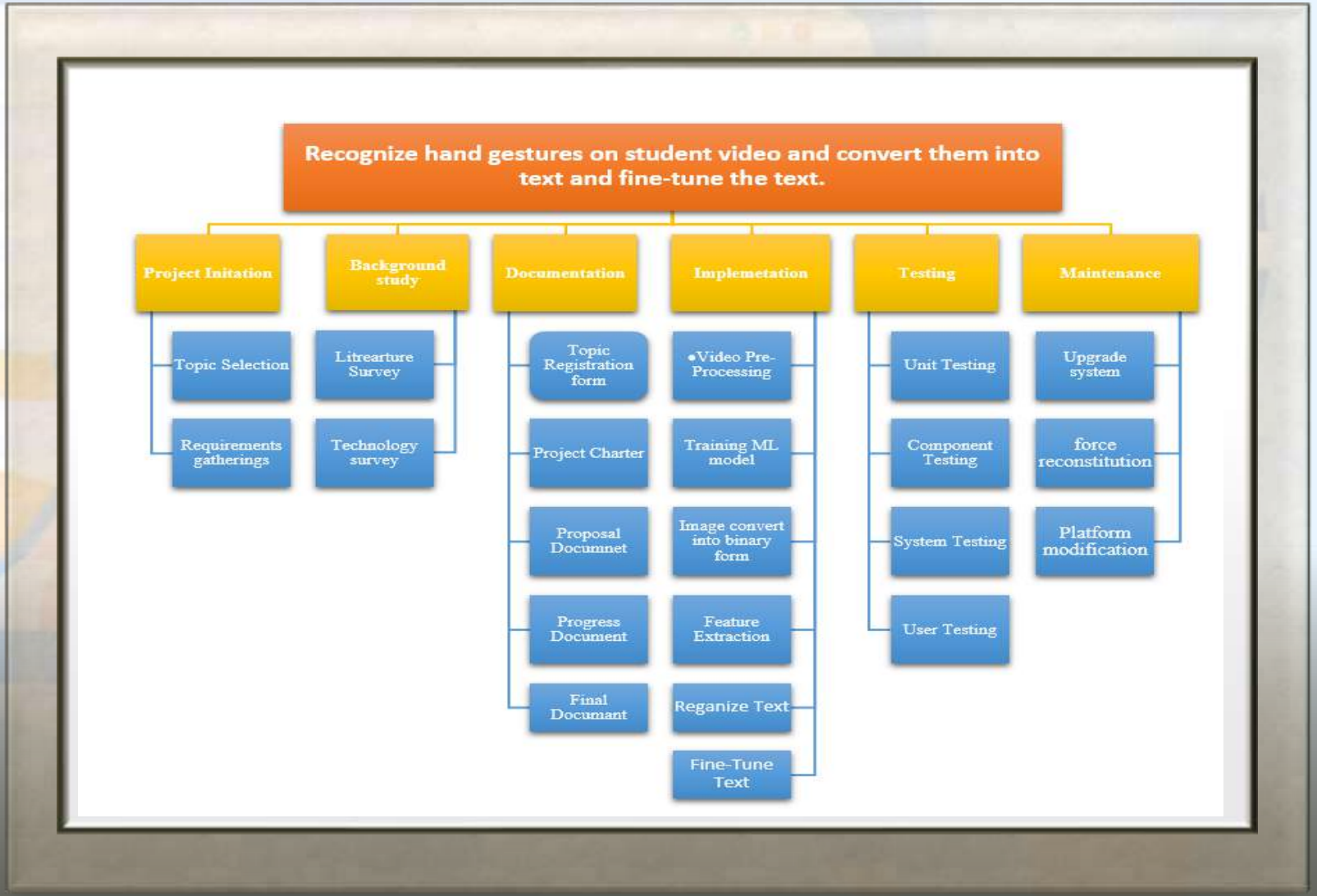
Website using Flask uploading image



Future works

- Preprocessing increase accuracy level
- Flask app upload video
- Fine tune the text
- Complete forum page

FUNCTION WORK BREAK DOWN



Gantt Chart - Function

Task Name	January	February	March	April	May	June	July	August	September	October	November	December
Project Initiation												
Evalation												
Topic Assessment form												
Charter												
Proposal document												
Proposal presentation												
Project planning												
System planning												
Collecting required data												
Selecting Algoritham techonologies and tools												
Implementation												
Video pre-processing												
Skin segmentation												
Feature Extracting												
Classification and Text convert												
Fine tune text NLP												
exprimental analysis												
Testing and finalize												
Research paper												
Testing												
Final report												
Final evaluation												

Reference

[1] https://www.youtube.com/watch?v=iGWbqhdjf2s
[2] https://www.analyticsvidhya.com/blog/2019/09/feature-engineering-images-introduction-hog-feature-desc
[3] https://www.freecodecamp.org/news/how-to-build-a-web-application-using-flask-and-deploy-it-to-the-cloud-3551c985e492/
[4] https://towardsdatascience.com/image-pre-processing-c1aec0be3edf
[5] https://www.mygreatlearning.com/blog/introduction-to-image-pre-processing/
[6] https://flask.palletsprojects.com/en/2.0.x/
[7] https://analyticsindiamag.com/image-feature-extraction-using-scikit-image-a-hands-on-guide/

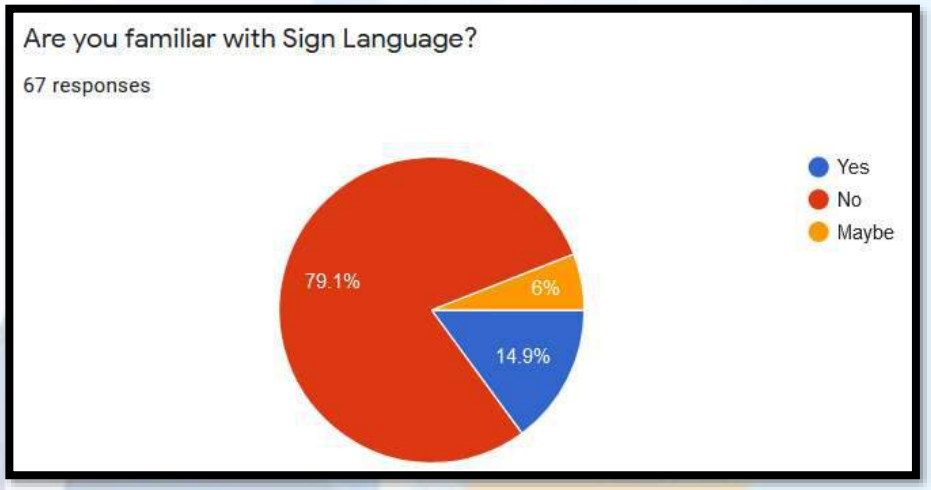
SLIIT
FACULTY OF COMPUTING

IT18152074 – Sangeeth Raj A

Bachelor of Science (Hons) in Information Technology Specializing in Software Engineering

Introduction

- **Are people willing to learn sign language?**
- **Use LMS to teach sign language.**
- **User friendly feature.**
- **Quality video content**
- **Low-resolution laptop webcams**



Research Questions

1. What are the current trending software used for motion detection?
2. What design aspects have been considered when designing LMS for hearing impaired community?
3. What ML technologies to be used?
4. What algorithms to be used to analyze user data?
5. What is the source of dataset?
6. Will dataset be used effectively in training and testing?
7. Will the proposed LMS make an impact in community?

Research Gap & Problem

Research Problem

- There is no LMS on teaching sign language.
- The sign language tutors are lack of knowledge in teaching online platform.
- Lack of dataset for sign language.
- Collecting a considerable amount of dataset takes time.

Research Gap & Problem

Research Gap

- **Algorithms used in hand gesture detection has limitations.**
- **Mostly research are done on hand gesture in image dataset.**
- **Increase the high mean accuracy in detection.**

Features	*Finger-Earth Mover's Distance [2]*	*Superpixel-Based Hand Gesture Recognition [2]*	*Recognizing Chinese Sign Language Based on Deep Neural Network[1]*	*Our Solution*
Achieve accuracy in detection	✔	✔	✔	✔
Fast recognition speed in analyzing	✗	✔	✗	✔
Achieve high mean accuracy in detection	✗	✗	✔	✔

Objective

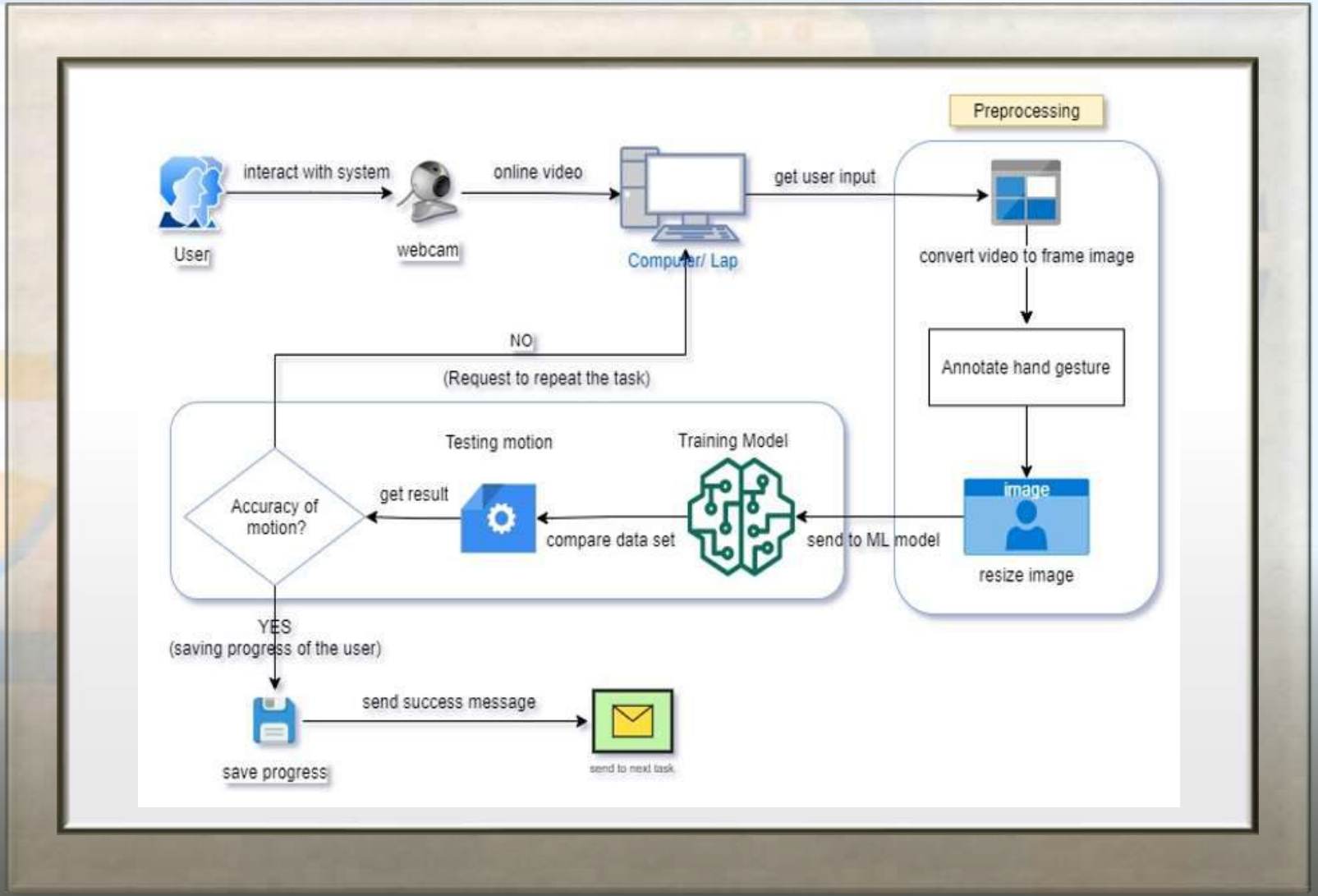
Main Objective

- Detecting the user's motion and analyses motion whether it is similar with the system.

Sub Objective

- Feeding the system with tutorial of the module(dataset).
- Providing correct instruction to user and to follow.
- Getting optimized video from the user(800x600pixel).
- Detect the user's motion using TensorFlow.
- Analyze whether the dataset is accurate by CNN.
- Using algorithm to initialize next stage in module.

Function Overview



Methodology

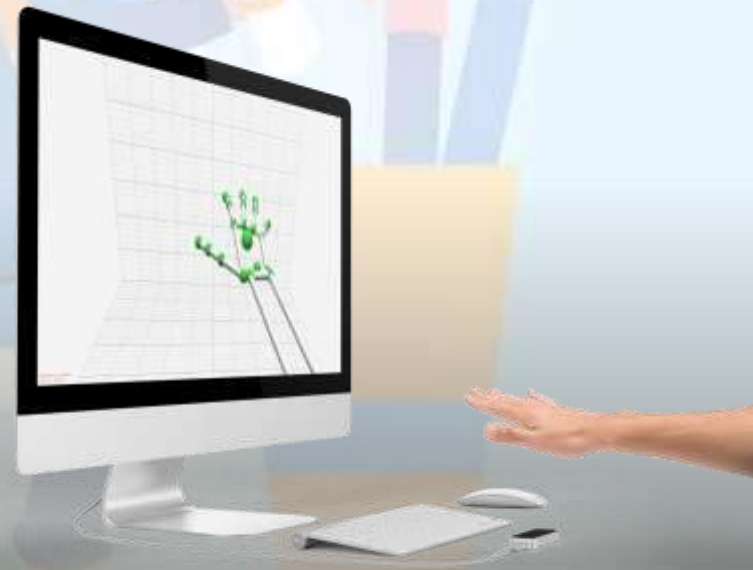
Annotate Hand Gesture

- **TensorFlow model trainer and Faster RCNN configuration.**
- **Using low-resolution images for training ML.**
  1) Faster training
  2) Storage efficiency
  3) Low latency network connections (low internet speed)
- **Images are converted into CSV for train and test image datasets.**
- **Training ML Model.**
- **Test with webcam, detects images using the pre-trained model.**

Methodology

Image Classification

- **Implement ML model with Convolutional Neural Networks(CNN).**
- **Using ‘Keras' library to build a CNN model.**
- **Dataset alphabet of American sign language.**
- **Minimum 500-1000 images per class to train.**
- **Image going through different stage in CNN classifier**
  1) Convolutional Layer
  2) Nonlinearity
  3) Pooling Layer
- **Notifies user's result.**



Project Requirements

Functional Requirements

- Analyze user hand gestures effectively and correctly.
- Analyze user knowledge on learning.

Non-Functional Requirements

- Giving accurate result of user's hand gestures without further ado
- High mean accuracy of detection and analyze motion

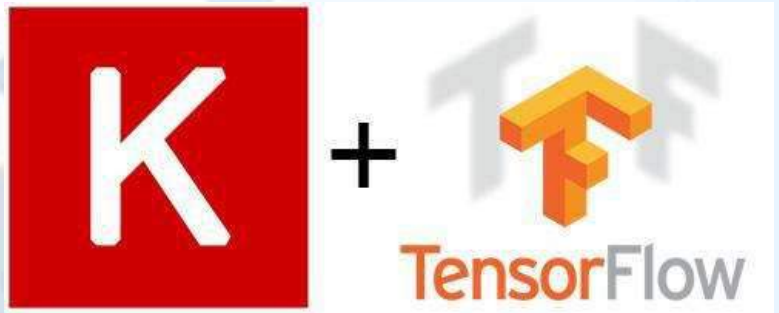
User Requirements

- Personal computer / Laptop
- Webcam
- Internet connection

Technology & Tool Selection

Technologies

- Annotate Hand Gesture
- Image Classifier

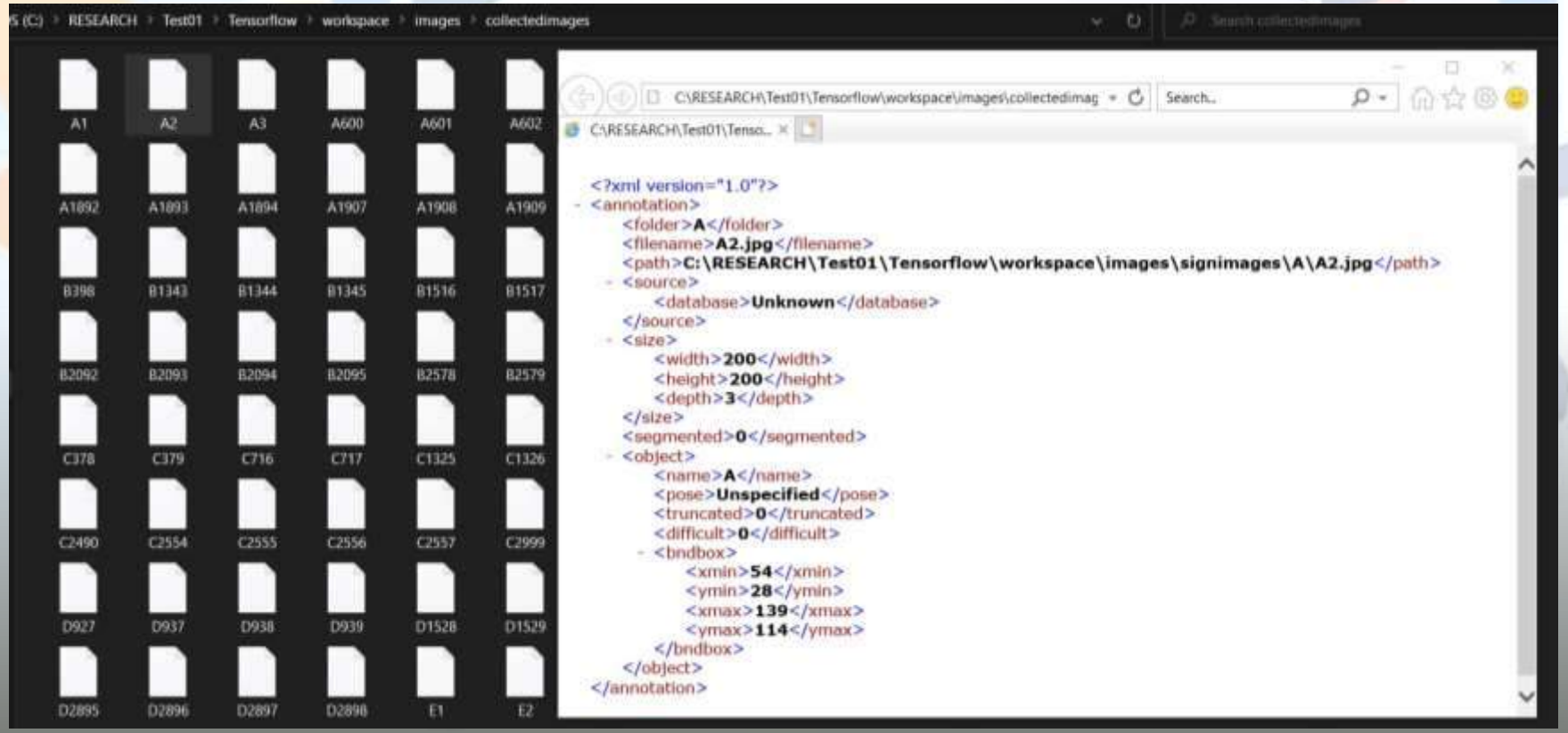


Tools

- Annotate Hand Gesture – TensorFlow, Fast RCNN
- Image Classifier – Python Keras, CNN
- For version controlling – GitLab
- Project Management – MS Planner

Completion of the project

Labeling



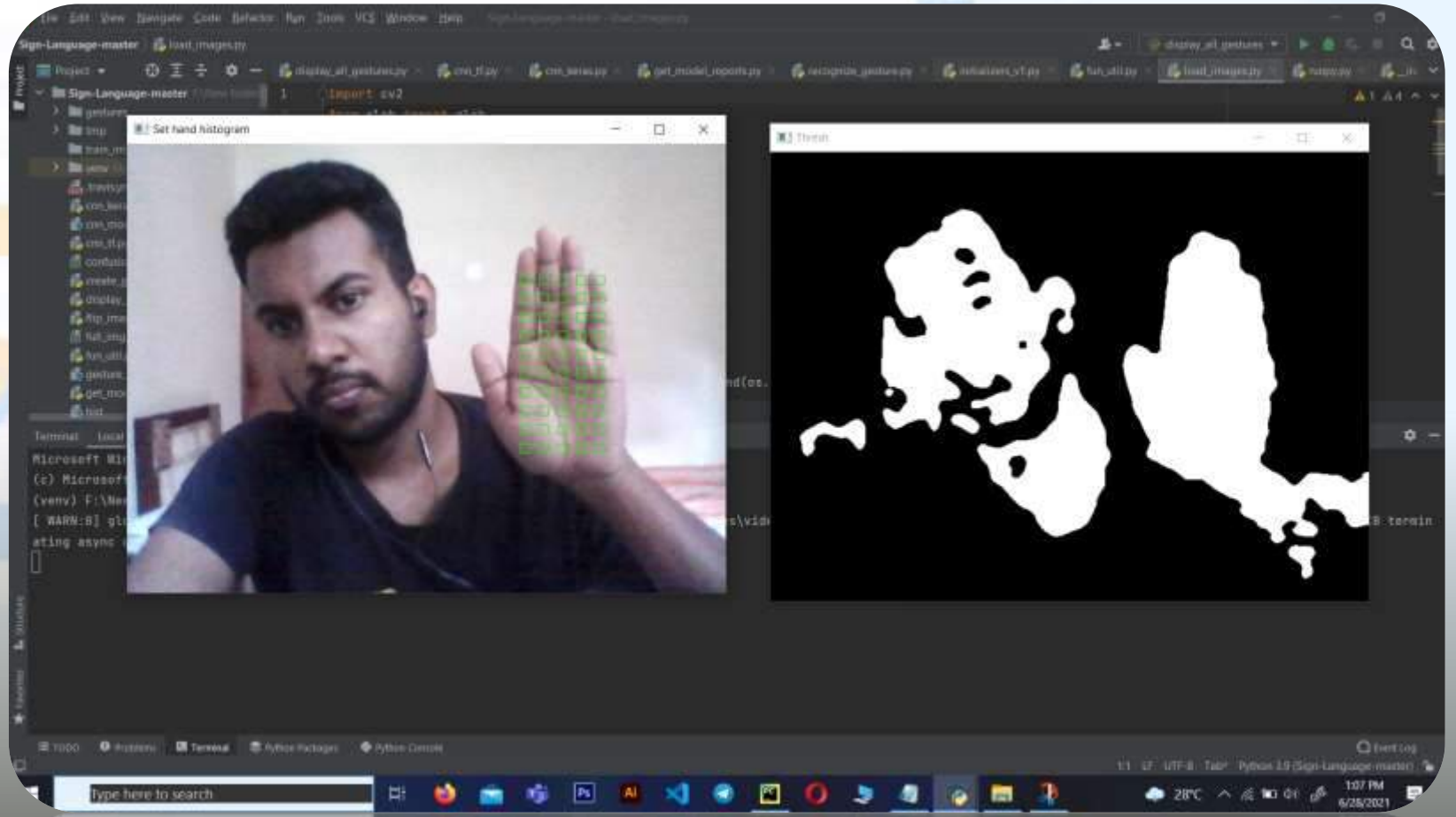
Completion of the project

Identify Skin tone

Completion of the project

Train Model

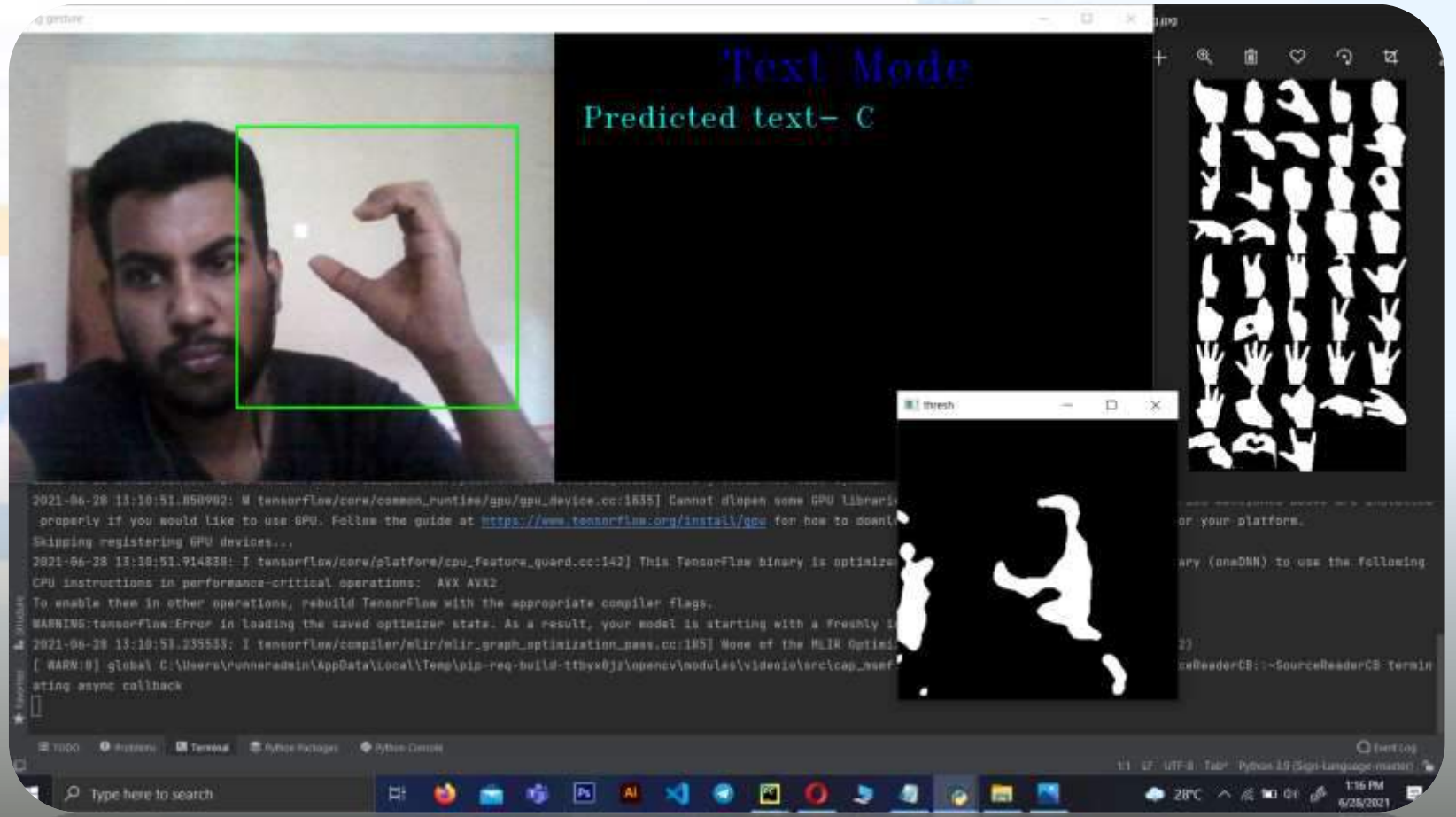
- **Annotate Hand Gesture – TensorFlow, Fast RCNN**
- **Image Classifier – Python Keras, CNN**



Completion of the project

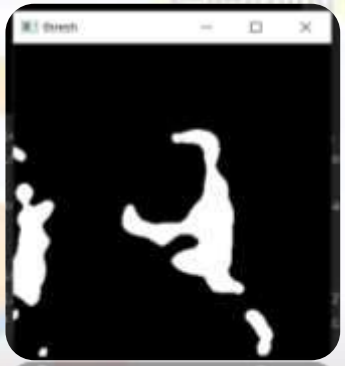
Testing Model

- **Model trained with alphabet and digits**
- **Test with random sign in low light environment with webcam**(USB2.0 VGA UVC)

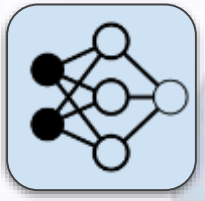


Completion of the project

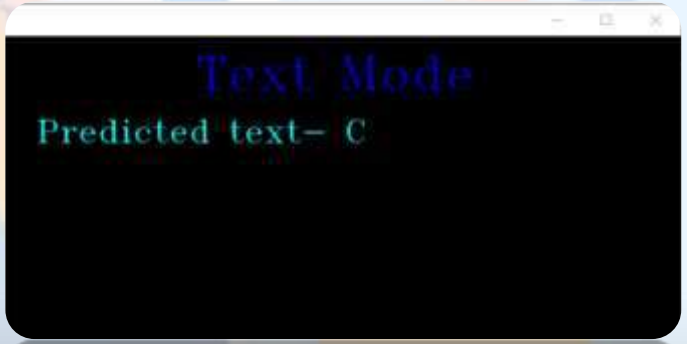
Hand Gesture



Threshold



CNN

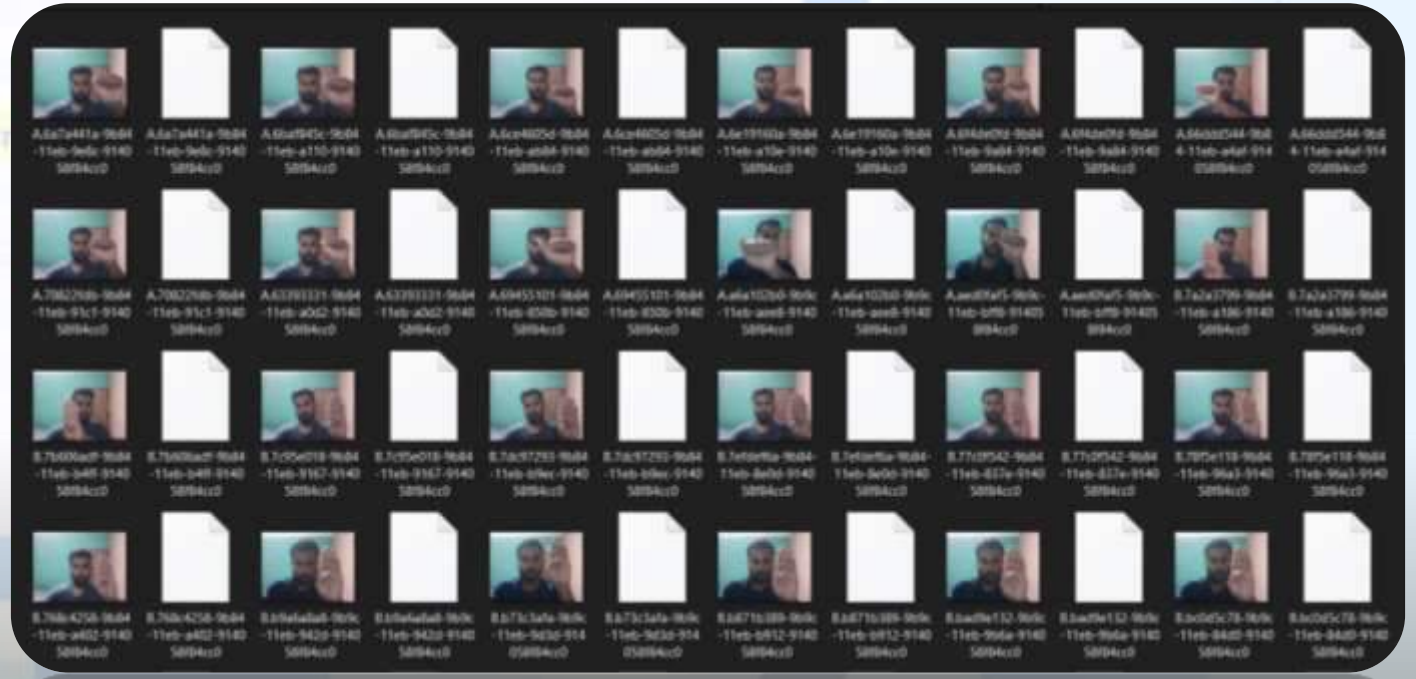


Identified Symbol

Completion of Project

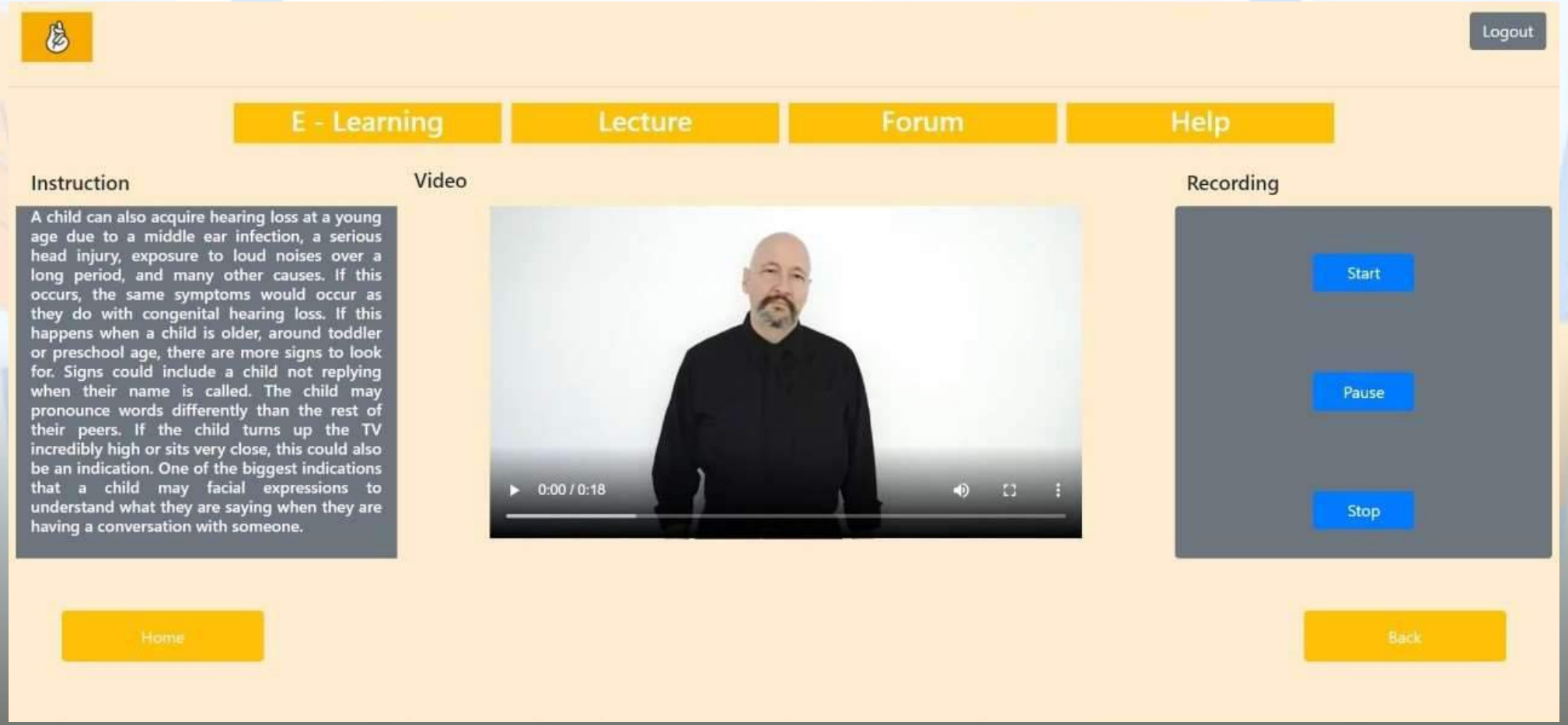
Failed Attempts

- **Build model using TensorFlow tf2_detection_zoo, and train model with own dataset.**



Completion of Project

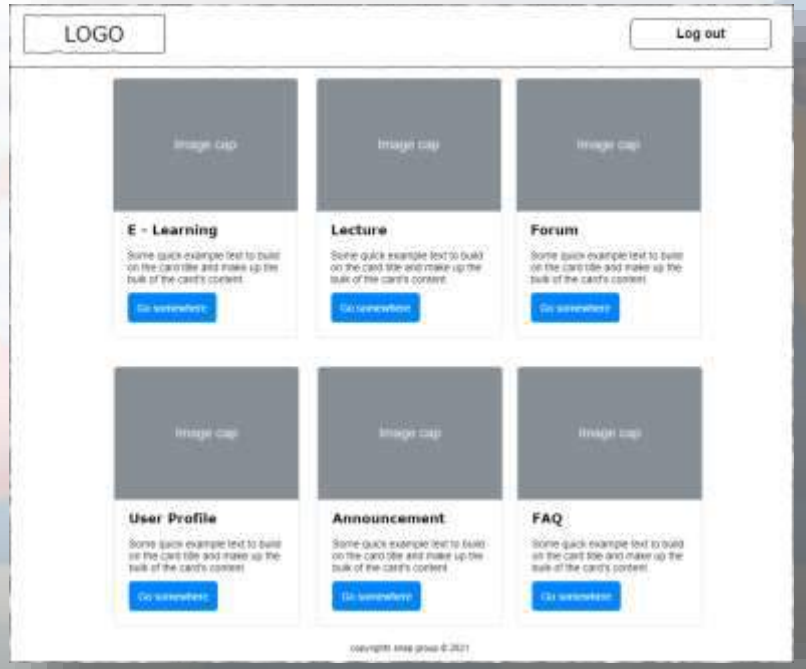
User Interface



SLIIT
FACULTY OF COMPUTING

Completion of Project

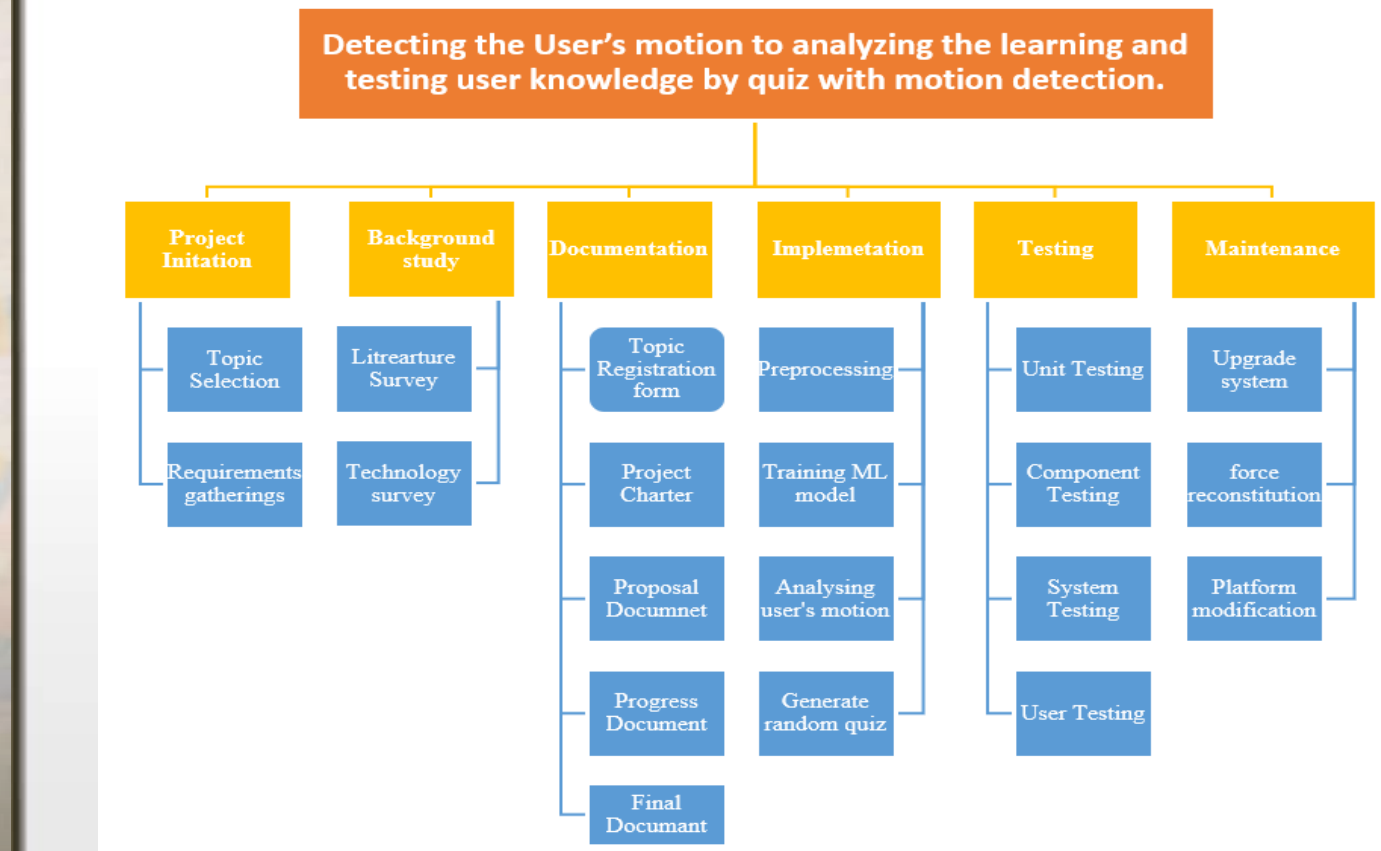
Wireframes



SLIIT
FACULTY OF COMPUTING

Gantt Chart - Function

FUNCTION WORK BREAK DOWN



Reference

1. https://www.python.org
2. https://pysource.com/object-detection-opencv-deep-learning-video-course/
3. https://opencv.org
4. https://tensorflow-object-detection-api-tutorial.readthedocs.io/en/latest/install.html
5. https://keras.io/guides/training_keras_models_on_cloud/
6. https://www.machinecurve.com/index.php/2020/04/13/how-to-use-h5py-and-keras-to-train-with-data-from-hdf5-files/
7. https://blog.roboflow.com/computer-vision-american-sign-language/
8. https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md

THANK YOU

ANY QUESTIONS?