E-LEARNING SYSTEM FOR HEARING IMPAIRED STUDENTS

2021-176

Final Report

BSc. (Hons) in Information Technology
Specializing in Software Engineering

Department of Information Technology

Sri Lanka Institute of Information Technology Sri Lanka

September 2021

E-LEARNING SYSTEM FOR HEARING IMPAIRED STUDENTS

2021-176

Final Report

Pirathikaran V - IT18068610

Niroshan K - IT18144772

Accash R - IT18069600

Sangeeth Raj A - IT18152074

BSc. (Hons) in Information Technology

Specializing in Software Engineering

Department of Information Technology

Sri Lanka Institute of Information Technology Sri Lanka

September 2021

DECLARATION

We declare that this is our own work, and this proposal does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of our knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

	STUDENT NAME	STUDENT NO.	SIGNATURE
1	Sangeeth Raj A	IT18152074	[illegible]
2	Pirathikaran V	IT18068610	[illegible]
3	Niroshan K	IT18144772	[illegible]
4	Accash R	IT18069600	[illegible]

The above candidates are carrying out research for the undergraduate Dissertation under my supervision.

………………………….. Date ……………………

Signature of the supervisor:

(Mrs. Kalpani Manathunga)

………………………….. Date ……………………

Signature of the co-supervisor:

(Mrs. Samanthi Erang Siriwardene)

ACKNOWLEDGEMENT

Under the research module in the final year, we could have gained a great experience and more than that, we could have improved our skills in many areas in the Information Technology field. This is really a fresh situation for us; thus, we need guidance and help from our lecturers. First of all, we would gratefully acknowledge our supervisor Dr. Kalpani Manathunga for her great guidance and encouragement given to our tema. And also, it is a great pleasure to say our thanks to our co-supervisor Ms. Samanthi E.R Siriwardana for her great encouragement and guidance throughout the whole research journey. Even in more difficult time periods during the project completion, they advised our team not to panic and keep working hard for the success of the project. They showed and guided us towards the correct path and how to achieve our own goals. Also, we are really grateful to them for giving their valuable time throughout this year. Also, we are gratefully acknowledged by those who contributed for the success of this project. We would really like to express our thanks to Sri Lanka Institute of Information Technology for providing us with the expertise to finish the mission. They additionally initialized the requirement of the fourth-year project which is essential for finishing the course. We might additionally want to thank Lecture-in-charge, Dr. Janaka Wijekoon for his guidance at some point of the course. He guided us at hard times, motivating and mentioning our mistakes to make the project a success. Nevertheless, We express our sincere thanks to our parents for believing in us and giving mental support and colleagues for their cooperation and continuous encouragement which helped us immensely in the final touch of this project.

2021-176
Department of Information Technology,
Faculty of Computing,
Sri Lanka Institute of Information Technology.

ABSTRACT

With the Spread of global pandemic Covid-19 the learning was transformed to online learning from traditional learning. The use of eLearning platforms was increased. But this had issues with certain communities of people around the world. The hearing-impaired people had many issues with eLearning platforms because of their deficiency in hearing sound. Therefore, through this paper we are introducing a platform through which hearing impaired people can effectively involve in learning. The introducing system will use sign language in addressing the students. We also introduce ways on which hearing impaired students can communicate with the tutors**.** The system has a low light enhancement module to enhance the videos uploaded by the tutor, it has the module to convert the uploaded videos to sign American Sign Language and it can also convert the questions asked via sign language to text. The system also focuses on teaching sign language.

Keywords: *Machine Learning, Video Processing, Low light enhancement, Convolutional Neural Network, OpenPose*

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLE

LIST OF EQUATION

Equation no. **Page no.**

LIST OF ABBREVIATION

Abbreviations	**Description**
ASL	American Sign Language
LMS	Learning Management System
ML	Machine Learning
CNN	Convolutional Neural Network
NLP	Natural Language Process
WHO	World Health Organization
HCI	Human Computer Interaction
XML	Extensible Mark-up Language
ReLu	Rectified Linear Unit
ROI	Region of Interest
HOG	Histogram of Oriented Gradients
SGD	Stochastic Gradient Descent
GCP	Google Cloud Platform
STT	Speech To Text

1. INTRODUCTION

1.1. Background Study

In the year 2020 the world encountered a global pandemic problem with the spread of COVID-19 virus. This pandemic situation transformed many of the industries to online basis with the use of Internet. This new transformation of Industries to online was quickly adapted by the people around the globe. One such sector which transformed to online was the education sector where students started learning through online platforms. Even though this transformation was effective in continuing the learning some group of people encountered lots of difficulties compared to traditional learning. One such group of people is the Hearing-Impaired people. In [1] the author states that the study made by WHO had suggested that approximately 466 million of total population around the world has some sort of hearing deficiency in 2018. This is a total of 6.1% of the world's population. Out of this 432 million are adults and 34 million are children. Initially we conducted a survey with the hearing-impaired community regarding an implementation of an e-learning environment. This survey was conducted at a deaf school in Sri Lanka and we received 75 responses. According to Figure 1 60% of the response preferred to have lectures in sign language on the e-learning platform. This showed us the importance of having an eLearning system using sign language.

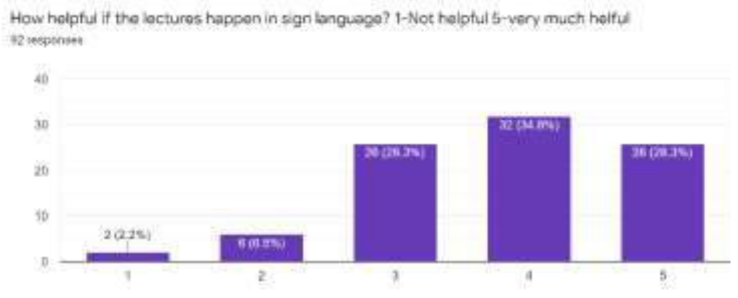


Figure 1 - Survey result

Based on the survey that is conducted we have clearly identified where the hearing-impaired community is facing their learning problems and how we can utilize the technological advancements to provide a better solution to the problems that are faced by the hearing-impaired students.

1.2. Literature Survey

Some Authors have worked on providing solutions for hearing impaired community with various technologies in the past, some of those related work is discussed here.

According to [2] the authors have discussed about using the human computer interaction approach to propose a new eLearning interface with interactional features for the use by students with varying visual and hearing needs. The proposed system is useful for visual/hearing impaired students. The assistive adaptive features used in this technology for hearing impaired students are Providing sign language when the user places the cursor over the text content in the page, Pre-recorded sign language videos, Sign language features to read command in toolbar, Explanation of graphs using sign language. In another work authors have also proposed solutions using modern technologies which can be used in day today tasks. In [3] the authors have proposed an interpreter system that can be used as an Android application. This application can convert the sign language into Normal speaking language. The proposed model was successful for conveying messages from deaf people to others.
We also planned to add low light enhancement for the uploaded videos in our system because it was identified that low light videos can cause some problems for hearing impaired students, this more elaborated in section III A. For identifying low light videos we gathered information from some relevant past works conducted by various authors. Out of them the Authors of [4] have proposed an algorithm for enhancement of low-light videos. The algorithm is first inverting an input low light video and then applying the optimized image de-haze algorithm on the inverted video. Simulations results of this algorithm shows good enhancement results when compared to other frame wise enhancement algorithms. In the work [5], the authors have proposed a real time video enhancement technique for videos with complex conditions like insufficient lighting. This method provided a better approach to enhance the video in low-lighting conditions

without any loss of color. The algorithm is providing effective enhancement using simple computational procedures. The results were analyzed on the videos that were taken on the bad light conditions. But this approach doesn't provide any evidence on the effectiveness of this algorithm in normal light.

When creating a sign language interpreter for our system we identified the importance of captioning techniques in the preprocessing part of the application therefore, we considered some important past research on the captioning techniques used on eLearning systems. According to [6] Authors discuss about two speech recognition techniques. They are using real time captioning using IBM ViaScribe and Post lecture transcription using IBM hosted transcription service. The main processes of these two techniques are the Process of recording instructor's speech, Captioning the methods using a human captioner or providing transcript using the human transcript services and finding errors of the transcripts or captions. These captioning techniques can be now utilized using modern cloud solution.

After Captioning we had to include some text-to sign language conversion techniques. For text to sign language conversion, a few research works performed in the past includes of direct translation and rule-based translation. In the Direct translation approach the English words are directly converted into sign language sequence, regardless of the meaning of the sentence. This is highlighted as a major defect in the research. Further, as in the rule-based approach [7], the user given text input is put under syntactic transformation and it is converted into a median text. Rule based approach can be considered as a successful approach when translating text to sigh language.

When we are creating an eLearning system for hearing impaired students, they should be able to use the system desirably. This brought us the challenge of identify hand gestures using our system. We looked on to some of the systems that was developed for this challenge. The proposed system [8] focuses on recognizing ASL alphabets and double-handed gestures for deaf and dumb people. In their system they have used 4 main components which are real-time hand tracking, hand segmentation, feature extraction and gesture recognition. The camshaft method and Hue-saturation Intensity (HSV) color model was used for hand tracking, gestures detection and segmentation.

In the work [9], a proposal for deaf and dumb people to communicate with ordinary people using a framework that recognizing hand gestures was introduced. Their approach was first to take an image applying skin segmentation using Hue-Saturation-value (HSV) histogram and finding edges and then applying Harris Algorithms for feature extraction. The next steps on their work were feature matching and recognition, here they calculated the minimum value of the matrix. Here they have used skin segmentation techniques to detect hand gestures. The proposed system [10] focuses on making the dual way communication between hearing-impaired and normal people. But they identified some issues on dynamic hand gestures that give the same meaning. Pre-processing, Segmentation, feature extraction, and classification are the techniques they have used. For Preprocessing they have used the Gaussian Blur to reduce the noise in the images. Segmentation was applied to find hand gestures regions from the image area. For feature extraction they have used Eigenvalues and Eigenvectors which can be used to classify hand gestures.

In the year 2016, D . Kelly , C . Markham , and J . Mc Donald in their paper Weakly Supervised Training of a Sign Language Recognition System Using Multiple Instance Learning Density Metrices[17] has proposed an approach using a metric mentioning that not a strong supervised training is need for SL hand gesture recognition system with the help of novel multiple instance learning density technique. Using the metric we can identify isolated SL from a sentence. Our spatiotemporal gesture and hand posture classifiers are then trained using the automatically extracted isolated samples. The research is to evaluate sign language extraction, identify hand gesture and posture classification and spatiotemporal spotting system. This was experimented on a pre-trained model of 48 different vocabulary sign in SL to evaluate overall sign spotting system of 30 sign which resulted in 87% success rate. They suggested a hand posture classification model that can reliably identify hand postures regardless of who is performing the gesture, and a spatiotemporal spotting ML model that can classify gestures on spatiotemporal and detect movement without being specifically trained on a sign. Spatiotemporal ML model and detect posture have be integrated into the framework, using the proposed MIL density algorithm they were able to learn sign language of hand gestures from noisy and weak oversight of translation automatically. The researcher have extending the vocabulary of our automatic training system based on 30 different sign takes only automatic processing of a wider video dataset including accompanying text translations. As a conclusion

Proposed system was able to learn using unsupervised training and dataset were from different source and still was able to identify the object, using this method we can conclude efficiency of the unsupervised model.

In 2020, Tariq Jamil, in his paper “Design and Implementation of an Intelligent System to translate Arabic Text into Arabic Sign Language” [16] has suggested the development of an AI system which transform Arabic written or text to Arabic Sign Language(ArSL). It was created to address the need for the Arabic deaf community to be integrated into society and to be able to communicate with them without difficulty. Text input, parsing, word processing, and ArSL output are the four key steps in the system's implementation. Basically, the sentence that needs to be translated will be passed to the translator user interface and the system will identify each part such as, noun, verb, adjective, and consequently eliminate the words that are meaningless. Then the other meaningful words will be checked with the already updated system’s database full of signs, and hence display the correct output in GIF format. The user interfaces for the translator were created using the ScreenBuilder application and the Java programming language. For quick and accurate text processing, the ‘Farasa' toolkit was used, which also helps to identify the parts of a sentence. Finally, the ArSL signs output was displayed using an animated character supplied by MindRockets, Inc. As a conclusion this research work clearly outlines the implementation process, where the tools and software used for development will be efficient to build an eLearning system and output dataset can be GIF format which mean the GIF can be used as input dataset.

In the year 2020 D. Manoj Kumar, K. Bavanraj, S. Thavananthan, G. M. A. S. Bastiansz, S. M. B. Harshanath and J. Alosious in their paper "EasyTalk, A Translator for Sri-Lankan Sign Language using Machine Learning and Artificial

Intelligence"[15] proposed an approach to convert SL text to auditory format, enabling people to communicate more efficiently. This is divided into four sub parts, Hand Gesture Detector, the first component, uses pre-trained models to capture hand gestures. The Image classification component translates hand gesture signs that have been found. The Voice generator component produces a text output or auditory format for the hand gesture signs that have been recognized. Finally, the Text to Sign Converter turns entered English text into animated images based on sign language. Using these tools, EasyTalk can more accurately identify, translate, and produce relevant outputs. As a conclusion The application translated all NLP connected language and SL using an NLP-based API. Ordinary citizens should use the application's reverse engine to translate voice input into sign language of Sri Lanka. They only need a valid dataset to use this API. The Text and Voice Generator aids in the recognition of word segments from alphabet collections, the correction of spelling, and the conversion of word segments to speech. This reverse translator use Semantic Analysis to get GIF images of relevant SL using text input from the user so that it will be an efficient way to use sign language video or gif as a data using RCNN base ML model.

With All the above-mentioned studies some of the key features and functionalities for a solution on hearing impaired students' inability to learn through eLearning system was proposed in the next section.

1.3. Research Gap

In relation to the literature surveys that were conducted for the module of text-to-sign language conversion, few research gaps were discovered and listed below and in the research proposed in the past is illustrated in a tabular format in Table 1.

- Communication gap between hearing impaired students and tutors
- No low light enhancement and Captioning for eLearning system
- No proper e-learning system for hearing-impaired community
- Existing systems only have recorded videos of sign language representation which takes additional human effort.

- Hearing-impaired community couldn't clear their doubts using any e-learning system.

Table 1 – Illustration of Research Gap

Features	System proposed by			Our solution
	M. S. Nair, N. A. P and S. M. Idicula	T. Jamil	A.S. Drigas, D. Kouremenos, S. Kouremenos and J. Vrettaros	
Sign language tutoring	X	-	-	✓
Availability of study materials like tests, quiz, etc.	-	-	X	✓
Reliable translation of words	-	X	-	✓
Low light enhancement and captioning	X	-	X	✓

1.4. Research Problem

This research area was proposed in behalf the community and things came out of blue, there were so many conflict in the initial stage of the proposal. Here we have listed few problems that we came across in the journey,

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

1.5. Research Question

At the beginning stage of the proposal we had many questions arise from our side those which we uncertain, these were the questions:

1. What are the current trending software in development?

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 main source of dataset?
6. Will dataset be used effectively in training and testing?
7. Will the proposed LMS make an impact in community?

1.6. Research Objective

The proposed LMS is a research study to improve the e-learning method to hearing-impaired students by delivering convert sign language content form lectures video and to increase the level of participation of the hearing-impaired students. The main and sub objectives of the research are as follows.

1.6.1. Main Objective

Solving communication and learning barrier between tutors and hearing-impaired students through learning Management System, these are the main objective of the proposed system.

1. Enhancing the low-light videos and providing transcription in real time.
2. Using the transcription and generating ASL interpretation.
3. Students clear doubts using ASL which can be converted into meaningful sentences.
4. Teaching ASL to both hearing impaired and general users.

1.6.2. Specific Objective

The specific objectives that were focused on this research component are as below.

1. Provide a user-friendly LMS to the students, especially hearing impaired ones.
2. Provide a reliable conversion system that would not collapse the meaning of the original caption.
3. Improve performance rate of the hearing impaired students in their respective study fields.

2. METHODOLOGY

2.1 System Overview Diagram

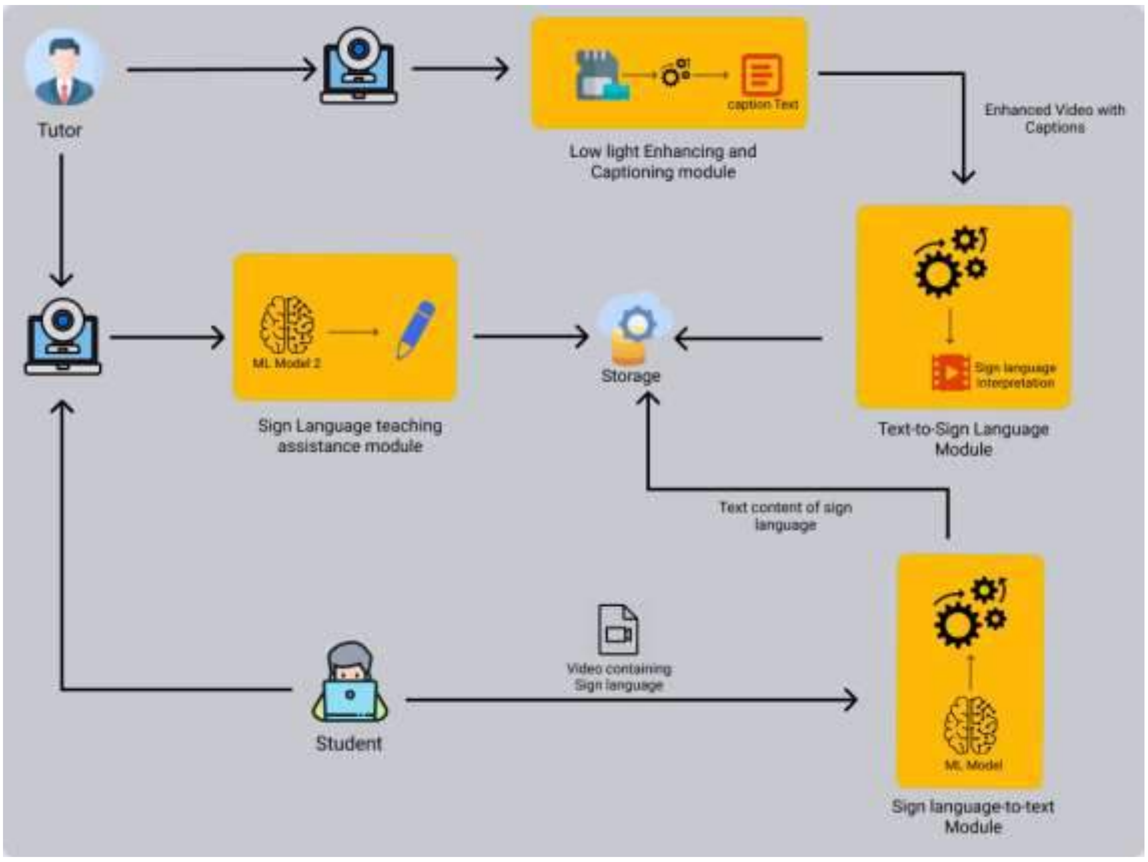

Tutor
caption Text
Low light Enhancing and Captioning module
Enhanced Video with Captions
ML Model 2
Sign Language teaching assistance module
Storage
Sign language interpretation
Text-to-Sign Language Module
Text content of sign language
Video containing Sign language
Student
ML Model
Sign language-to-text Module

Figure 2 - System Overview Diagram

Figure 2 shows the system overview diagram. According to the system diagram the video will be recorded by the lecturer using his/her webcam. Then it will be uploaded to the system. Once the video is uploaded the Low light enhancing and captioning module will enhance the video and produce captions to the enhanced video using proposed algorithms. The output of this module will contain both the enhanced video and the captioned text for the video. The captioned text will be sent to the Text-to-Sign language module and produces the sign language interpretation for the captioned text. This output will be stored in a storage/ database. Sign Language teaching assistance module and the Sign language to text module takes user recorded videos as its input for its modules and produce Sign to text interpretation for sign language teaching functionality and hearing-impaired student question forum functionality respectively.

In the following sections development of each of the module of the system is explained along with their implementation.

2.2 Low Light Enhancement and Captioning

In this section the development strategy of low light enhancement and captioning for uploaded vides is discussed. Further this section is divided into two sub sections, low light enhancement and captioning module.

2.2.1 Low light enhancement module

This section discusses the development of the low light enhancement module. When the video is uploaded to the system the low light frames of the video is identified and first and then they get enhanced. For this purpose, the low light enhancement module uses the automated low light identification algorithm as its preprocessing. For automated low light identification technique, the algorithm creates intensity histograms for each frame Figure 3.

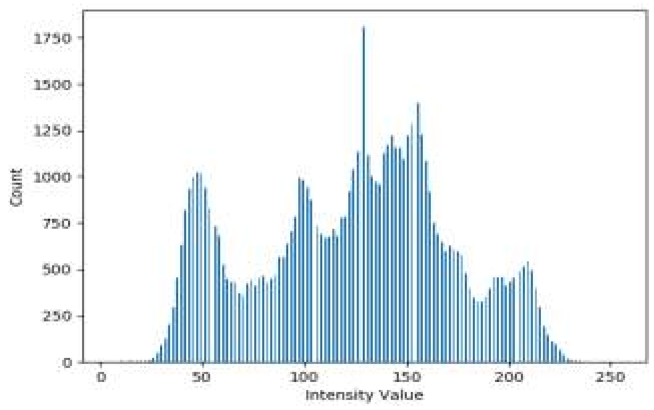


Figure 3 -Intensity histogram

Intensity histograms Figure 3 displays total number of pixels at each intensity level of an image. For an 8-bit image the intensity levels range from 0 – 255. Once the intensity histograms are created the cumulative histogram are created from them. The cumulative histogram (Figure 4) is an intensity representation that counts the cumulative number of pixels in all intensities up to the current intensity.

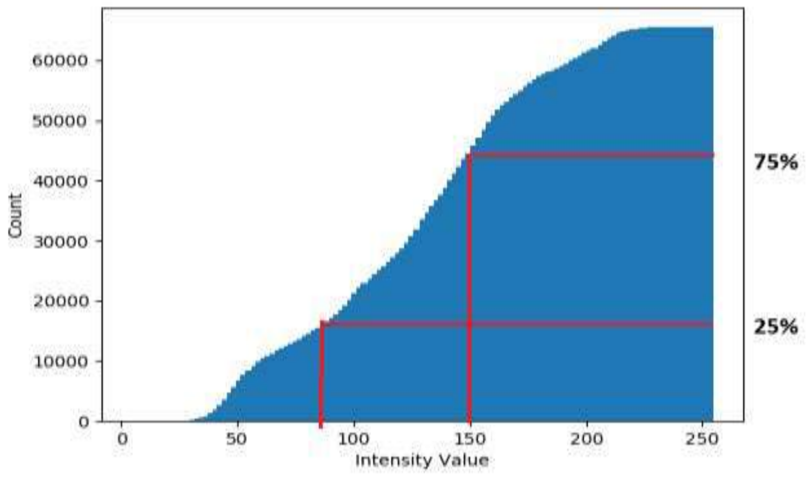


Figure 4 - Cumulative intensity histogram

For the generation of cumulative histogram, the equation (1) was used. This equation uses the intensity distribution of a gray scale intensity histogram to produce the cumulative histogram.

$$\boldsymbol{H(i) = \sum_{j=0}^{i} h(j) \quad for\ 0 \leq i < K} \qquad \textbf{(1)}$$

Equation 1 - Cumulative intensity equation

H in equation (1) represents the number of cumulative pixels at each intensity level (i), h represents the number of pixels in the gray scale intensity histogram at intensity level (i). Intensity level can be in the range from 0 to K which is 0 to 255. To separate the low light frames from the normal light frames the algorithm checks to which intensity values the majority of the pixels direct. According to Fig 3, 25% of the pixels direct to intensity value less than 100 and 75% pixels direct to an intensity value 150. According to the Figure 5, we can identify the nature of the intensity values.

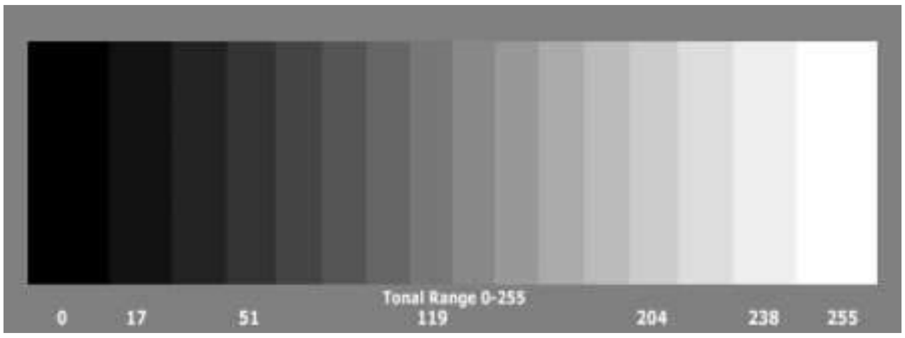


Figure 5 - Gray Scale intensity distribution

When developing the algorithm, an intensity value is assigned as a threshold value. This threshold value is selected such that the intensity of the image can change its nature from dark intensity to light intensity. Therefore, an intensity value >100 and <120 is selected as our threshold value (Figure 5). This threshold value is then mapped with the number of pixels of the cumulative intensity histogram. After selecting an optimum threshold value, both Fig 4 and Fig 5 was compared and identified that the cumulative histogram Figure 4 is of a normal light image because 75% of the total number of pixels falls under a bright region which is intensity level 150. Based on this logic the algorithm was designed to identify low light frames separately from normal light frames. First our image pixel intensities were scaled from the range [0,255] to [0,1]. Then the equation (2) was applied to output the gamma corrected image/ video frame.

$$\boldsymbol{O = C * I^{\gamma}} \tag{2}$$

Equation 2 - gamma corrected image/ video frame

According to equation (2), I in the equation denotes the input image and γ is the gamma value. C is a constant and it is equal to 1. The output O is then scaled back to intensity range [0,255]. For the enhancement algorithm a γ value of 0.5 is selected. After applying gamma correction, a lookup table is created for input pixels and output pixels. The output pixel will be the enhanced version of the input pixel. Using this lookup table each pixel is mapped and enhanced resulting in an overall enhanced frame. When low light frames are enhanced the noise within them also increases. To reduce this median filtering is used. Median filter is a non-linear filter therefore their operation is completely based on pixel values on the neighborhood and they do not use coefficient values as in the linear filter. Median filter computes the median grey level value of the neighborhood which reduces the noise in the image.

2.2.2 Captioning module

On this module the speech from the video is extracted and converted into text. The audio of the video is first extracted separately. The extracted audio is sent as the input to the speech to text model which will output text content. This text is then converted into captions with time stamps using a custom algorithm. For the implementation of this functionality the design decision was to use an industry standard speech to text (STT) model since commercial speech to text models are more accurate and has good results compared to custom speech to text models. As commercial STT models we selected IBM Watson and Google Cloud Platform (GCP) STT model. When comparing these two models it was clear that GCP model has more advance and accurate results. The output from the GCP STT model will be as a continues text. This text wanted to be converted into sentences based on the time stamps. When creating timestamps, the average time taken by the person to read a sentence has to be identified. In our algorithm, the average time taken to read a sentence was selected as 0.5 seconds. Then each word was split and counted. Using the total count of the words, the time taken for reading the caption was identified. Once the total time was identified time taken for each sentence was calculated. To calculate the time taken for each sentence a time frame of three seconds was selected with a word count of six.

2.3 Text to American Sign Language

The procedure in this module consists of five steps as mentioned below.

1. Parsing the English text.
2. Re-arrangement of the sentence based on the ASL grammar rule standard.
3. Eliminating stop words in the sentence.
4. Stemming
5. Conversion of text into video

2.3.1 Parsing the English text

The parsing was conducted as the initial step to prepare the retrieved text to put into the grammar based arrangement. The parser produces outputs in three parts: part-of-speech tagged text, context free grammar representation of phrase structure and type dependency representation.

2.3.2 Re-arrangement of the sentence

Here we rearrange of the sentence based on the ASL grammar rule standard and the grammar standard conversion is a very crucial step in the process, since the language used by the tutors and the American Sign Language have difference in terms of grammar rules. Therefore, it is essential to convert the parsed English text into a format that relates with the sign language grammar rules.

2.3.3 Eliminating stop words in the sentence

In the process of conversion of text into signs, it is clearly seen that some words in a sentence do not contain any meaning or is not applicable for conversion itself. Such words are referred to as stop words and the elimination of those words is done in this step. Stop words include various types, among which a few of them can be determiners (the, a, an, another), coordinating junctions (for, an, nor, but, yet), prepositions (under, from, on, of, towards), plurals ('book' instead of 'books'), interjections, etc.

2.3.4 Stemming

The words containing prefixes or suffixes, tense related suffixes, etc. are reduced into its root form. For example, words like 'adjustment', 'adjustable', 'adjusting' are reduced into its root 'adjust'. Further, if the root word is not found in the sign database, it is matched with its synonym and then the relevant sign will be found.

2.3.5. Conversion of text into video

The string of words that was converted into the ASL format of text, will be available for matching with the sign database available in the system. This will be done accordingly with the name of the sign videos that they are labeled with. Hence, after finding the right matches, the video sequence will be displayed to the user.

2.4 American Sign Language to Text

In our research, my basic functionality is to convert sign language into text and make a meaningful sentence. Deaf and dumb student can ask questions by uploading their questions as a video file. Then the system saves the video and does the relevant steps. The first system does video pre-processing. In the video pre-processing has three steps. The first step is converting the video into a frame by frame image, the second step is to adjust the contrast and the final step is to resize the image. After video pre-processing image background removal, the next step is to convert the image in binary form, the following step is the feature extraction, and the later step is to do the gesture recognition and finally taking output text, fine-tune and save it into the database.

2.4.1 Video pre-processing

There are three subprocesses in this process. The steps are to convert video into frame by frame, adjust contrast, and resize images. First of all, the system taking video from the database and start video processing. In the first step of video pre-processing video input will convert frame by frame and store as sequences of images. The second step is taking images one by one and analyzing the contrast and adjust the contrast according to the requirement. The last step is to resize the image. This step is maintaining a unique size of image and resolution for all images. It will deduce the analysis time for the calculations.

2.4.2 Removal background object

After the video pre-processing image will take it for the next step. In this step image's background and objects will remove from here. It makes the image more definition to identify the hand region.

2.4.3 Feature Extraction

The feature extraction is used to pick up certain features from the unique hand image for each identity. We used HOG transformation to extract feature from the frame. The Histogram of Oriented Gradients (HOG) is a feature descriptor for object recognition in computer vision and image processing. The method counts the number of times a gradient orientation appears in a certain area of a picture.

2.4.4 Classification

Classification is a method after feature extraction that recognize hand gestures with various hand gestures images. We used here SGD classifier. The Stochastic Gradient Descent (SGD) optimization technique is used to determine the values of parameters/coefficients of functions that minimize a cost function. After the classification outputs will save into the specific database location.

2.4.5 Dataset collection

In our research mainly focus on the American sign langue. Therefore we used the American Sign Language data set from Kaggle, Microsoft-ASL and own data set also.

2.5 Teaching American Sign Language

2.5.1 Dataset

ML model implemented using Convolutional Neural Networks. Using CNN classifier we can process different image then we can categorize them accordingly[22]. Keras library and TensorFlow was used to implement CNN ML model this will be discussed in discussion chapter below. Dataset of ASL alphabet is used, each letter will be class. There will be 26 classes and each with minimum 100-150 images per class to train.

2.5.2 Preprocessing

Here we have used vision based approach which eliminate the cost of sensory device, this approach focus on webcam input to gain data from user end.

Gaussian blur filter : is used to extract data from the user which save preprocessing time, there are other method but in extraction of data arithmetic mean and Gaussian mean are most recommend. The Gaussian mean, pixel values farther away from the (x, y)-coordinate center of the region contribute less to the overall calculation of threshold value, here is the general formula to compute threshold value, T.

$$\boldsymbol{T = \ mean(I_L) - C} \qquad \textbf{(3)}$$

Equation 3 - Gaussian mean

where $(\boldsymbol{I_L})$ is the local sub-region of the image, I, and C is some constant which we can use to fine tune the threshold value T.

2.5.3 Image Classification

Once the filter is applied the image going through different stage in CNN classifier layer which are listed below,

1. **Convolutional Layer (1)**

Image from the webcam has low quality which is 128x128 pixels. It will be passing through 3x3 convolution filter which is smaller filter size (3x3 + 3x3) with smaller kernel value we get more layers and less weight that lead to it learn more complex non-linear features. Finally result with 126x126 pixels for each Filter-weights.

2. **Pooling Layer**

   Once the image pass convolution filter the image are down sampled with the help of max pooling which get the maximum value of the feature map and give the summarized value of the feature detected of the image, down sampling of feature map can be seen in Figure 6. We use 2x2 so that the final result will be 63x63 pixels in the end.

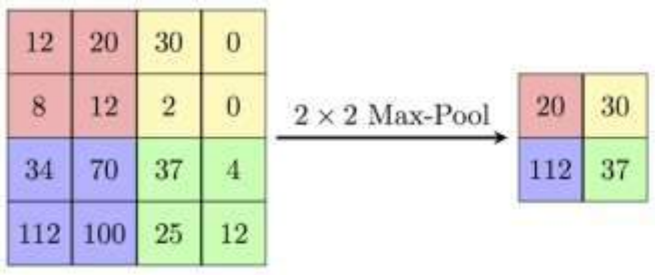


Figure 6 - Maximum Pooling (2x2)

3. **Convolutional Layer (2)**

   Now the image value is 63x63 pixels and this will be filtered again with 3x3 filter and result in 60x60 pixels.

4. **Pooling Layer(2)**

   Here the image with 60x60 pixels is down sampled again with 2x2 so that the final result will be 30x30 pixels.

5. **Densely Layer**

   The layer connected with preceding layer deeply which means it is connects every neurons and here 128 neurons hence it is used for simple neural network. Densely layer perform matrix-vector multiplication where column vectors and row vectors must have equal number of columns where A is a (M x N) matrix and x is a (1 x N) matrix and pooling layer result 30x30 so we take 32 column value and input of the layer is array of 28800 value. To avoid overfitting 0.5 is used as a drop out value.

6. **Densely Layer (2)**

   Result of densely layer (1) used as input to fully connect layer 96 neurons to keep it simple neural network.

Rectified Linear Unit : also known as ReLu is been used in the layers of both convolutional and densely fully connected neurons as Activation Function. Here for each input max(x,0) function is used by ReLu, this helps to learn more features of input and add nonlinearity. So that it results in speeding up training process and reduce computation time.

Max Pooling Layer : there are different types of pooling layers such as Average Pooling Layers, Max Pooling Layers and Global Pooling Layers. We have used Max pooling with pool size of (2, 2) with combination of ReLu activation function. This will reduce computation cost and overfitting.

Dropout Layers : here the layer will drop out activation function which doesn't perform well by setting them to zero, the selection of the activation is one specific but random set is picked. Providing correct output for defined set even if some of the activation function are dropped out, dropout is simply a way to prevent neural network from overfitting in Figure 7 we can see the functionality of applying dropout.

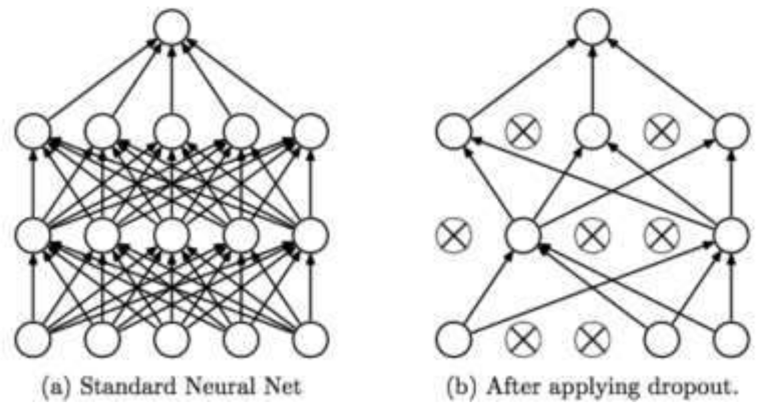


Figure 7 - Dropout Layer process

Adam Optimizer : is used for update ML model in response to the output of the loss function. Adam's optimizer has the advantages of two gradient algorithms which are Root Mean Square Propagation(RMSProp) and Adaptive Gradient Algorithm(ADA GRAD).

2.6 Development Process

2.6.1 Project Management Methodology

At the beginning of a research project requirements may be unclear or undefined due to presence of various sign language and will be subjected to lot of changes throughout the development cycle. The Agile Scrum model Figure 8 represents the scrum process is an ideal methodology for this kind of a project because it adds more flexibility to the software development life cycle and encourages requirement changes throughout the process of development. Agile follows an incremental and iterative development approach, and each iteration will focus on delivering a working product by adding more dataset. As our team consists of four members, having daily scrum meetings will allow each member to have a general understanding of the whole project and be aware of problems faced by other members as all the functionalities depend on one another. Also, this will improve the collaboration between team members encouraging better team work.

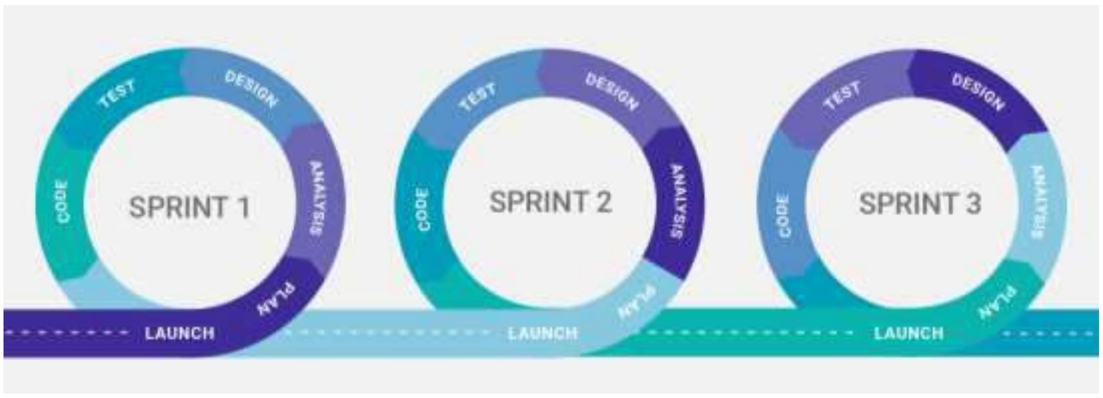


Figure 8 - Agile Scrum Process

2.7 Feasibility Study

The following tools and technologies listed below are used to implement this system,

1. **Designing tools :** Wireframes - Balsamiq
   Diagram - Draw.io

2. **Implementation :** Database - Firebase
   Backend - PyCharm Community Edition(flask)
   Machine Learning – TensorFlow, Keras, OpenCV, Jupyter notebook

User Interface - Visual Studio 2019
Version Control - GitLab
Task Planning - Microsoft Planner

3. **Documentation :** Research Paper/ Final Report - MS Word 2019
Presentation - MS PowerPoint 2019

2.8 Requirements and Commercialization

2.8.1 Requirement Gathering

This phase is one of the important steps that need to be done before implementing any system. It is a must to analyze and read a lot of previous research works related to this project, to know what kind of implementations to be done, what are the technologies used in the previous researches, and what are the research gaps in those existing works.

1. **Literature Review**
   Under the supervision of our former supervisor we discussed the research project idea and she guided us in collection information from online conference paper and article which gave us a knowledge on what we are going to work on and the current trending technologies which supports the project.
2. **Survey Results**
   We prepared a set of questionnaire to hearing impaired community to find out the difficulties they face in learning and if the are willing to switch to e learning platform with the guidance of our supervisor. We visited a special school Senkadagala Deaf & Blind School, Kandy and the survey was given to students in the school and few teacher as well with the help of few voluntaries we were able to complete the survey, you can find the survey in **Error! Reference source not found.**.
3. **Dataset**
   To training the ML model we referred resource from Kaggle, Microsoft Research Open Data.

2.8.2 Commercialization aspect of the product

The web-based nature of the platform we propose has several advantages when considering its potential LMS value. The main reason to implement a learning management system for the deaf community students was to make them feel confident in the education field and to achieve success. It absolutely is very helpful for them as they never got to experience an LMS same as the other students (non-hearing impaired). Therefore, they won't be having the feeling of missing out in a community.

The proposed system is very reliable, highly accurate and rapid enough to follow the tutorials, simple and user-friendly application. These aspects will influence very much in making the application a highly demanded one for the whole deaf community people. Thus it will also increase the user count. Along with that, we can ensure that the students who benefit from this, will come out and shine in the society as they will have no more hurdles in gaining knowledge or skills.

Gradually, there will be expectations for other fields other than education, like IT industries, banking industry, finance industry, advertising industry, etc. to make use of this application, just to convey important information to the hearing impaired users,

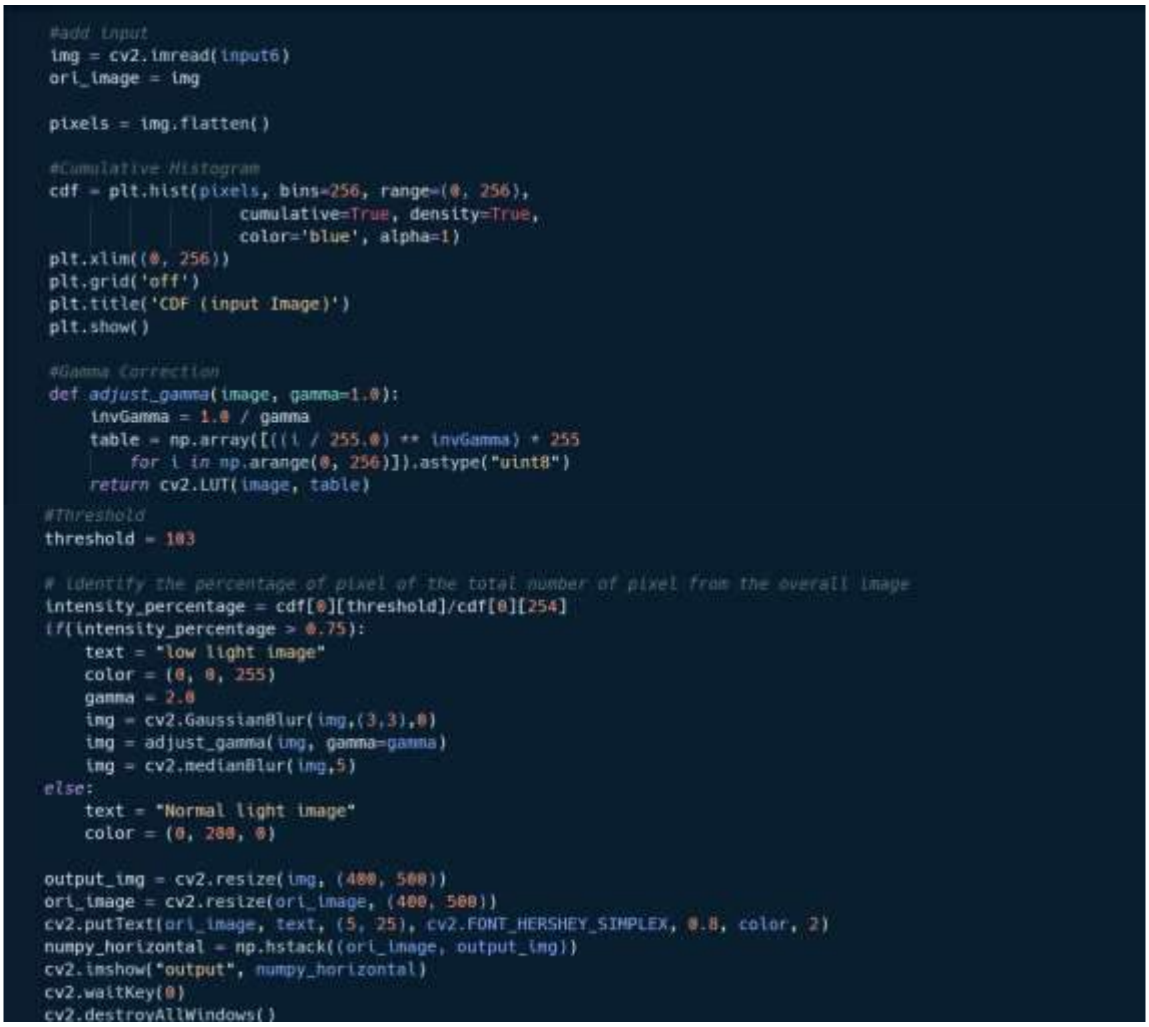
The following feature can be considered as the main commercialization aspects

- It can be hosted on a cloud platform and provided as a *Software as a Service* (SaaS) product, where the customer will pay a one-time fee or a subscription to use it.
- It can be developed as a website and advertisements can be incorporated into the system.
- It can be developed as a Freemium model, where services are provided free of charge and certain premium services can be provided for a fee such downloading feature, certification and etc.

2.9 Implementation

2.9.1 Low light enhancement and captioning module

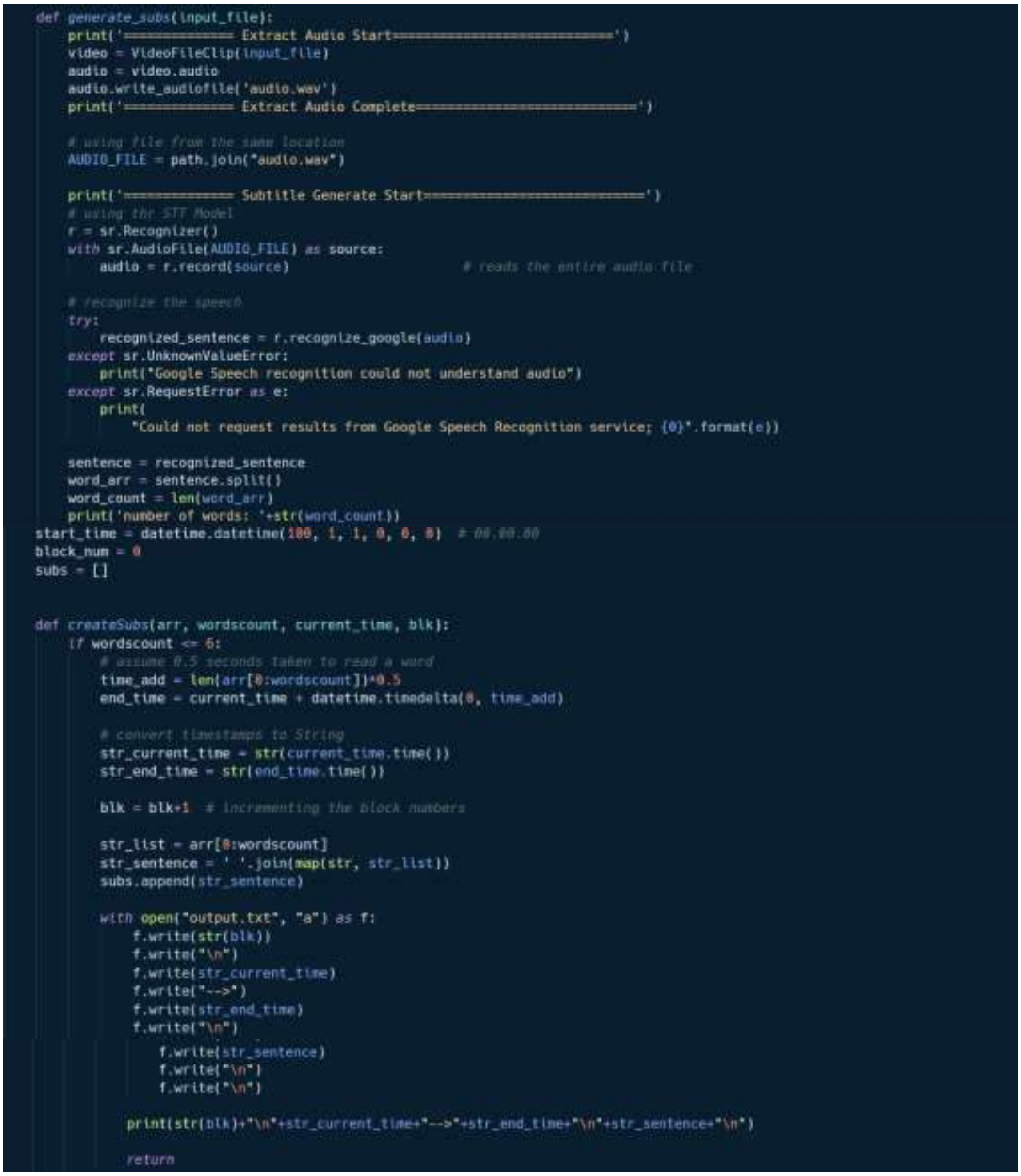
2.9.1.1 Low light enhancement script



```
#add input
img = cv2.imread(input6)
ori_image = img

pixels = img.flatten()

#Cumulative Histogram
cdf = plt.hist(pixels, bins=256, range=(0, 256),
                     cumulative=True, density=True,
                     color='blue', alpha=1)
plt.xlim((0, 256))
plt.grid('off')
plt.title('CDF (input Image)')
plt.show()

#Gamma Correction
def adjust_gamma(image, gamma=1.0):
    invGamma = 1.0 / gamma
    table = np.array([((i / 255.0) ** invGamma) * 255
        for i in np.arange(0, 256)]).astype("uint8")
    return cv2.LUT(image, table)
#Threshold
threshold = 103

# identify the percentage of pixel of the total number of pixel from the overall image
intensity_percentage = cdf[0][threshold]/cdf[0][254]
if(intensity_percentage > 0.75):
    text = "low light image"
    color = (0, 0, 255)
    gamma = 2.0
    img = cv2.GaussianBlur(img,(3,3),0)
    img = adjust_gamma(img, gamma=gamma)
    img = cv2.medianBlur(img,5)
else:
    text = "Normal light image"
    color = (0, 200, 0)

output_img = cv2.resize(img, (400, 500))
ori_image = cv2.resize(ori_image, (400, 500))
cv2.putText(ori_image, text, (5, 25), cv2.FONT_HERSHEY_SIMPLEX, 0.8, color, 2)
numpy_horizontal = np.hstack((ori_image, output_img))
cv2.imshow("output", numpy_horizontal)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

The low light enhancement script is developed using python. The *adjust_gamma* function creates lookup tables for input and output pixels for gamma correction. In this script a threshold value of 103 is selected to differentiate low light videos with normal light videos.

2.9.1.2 Captioning Script



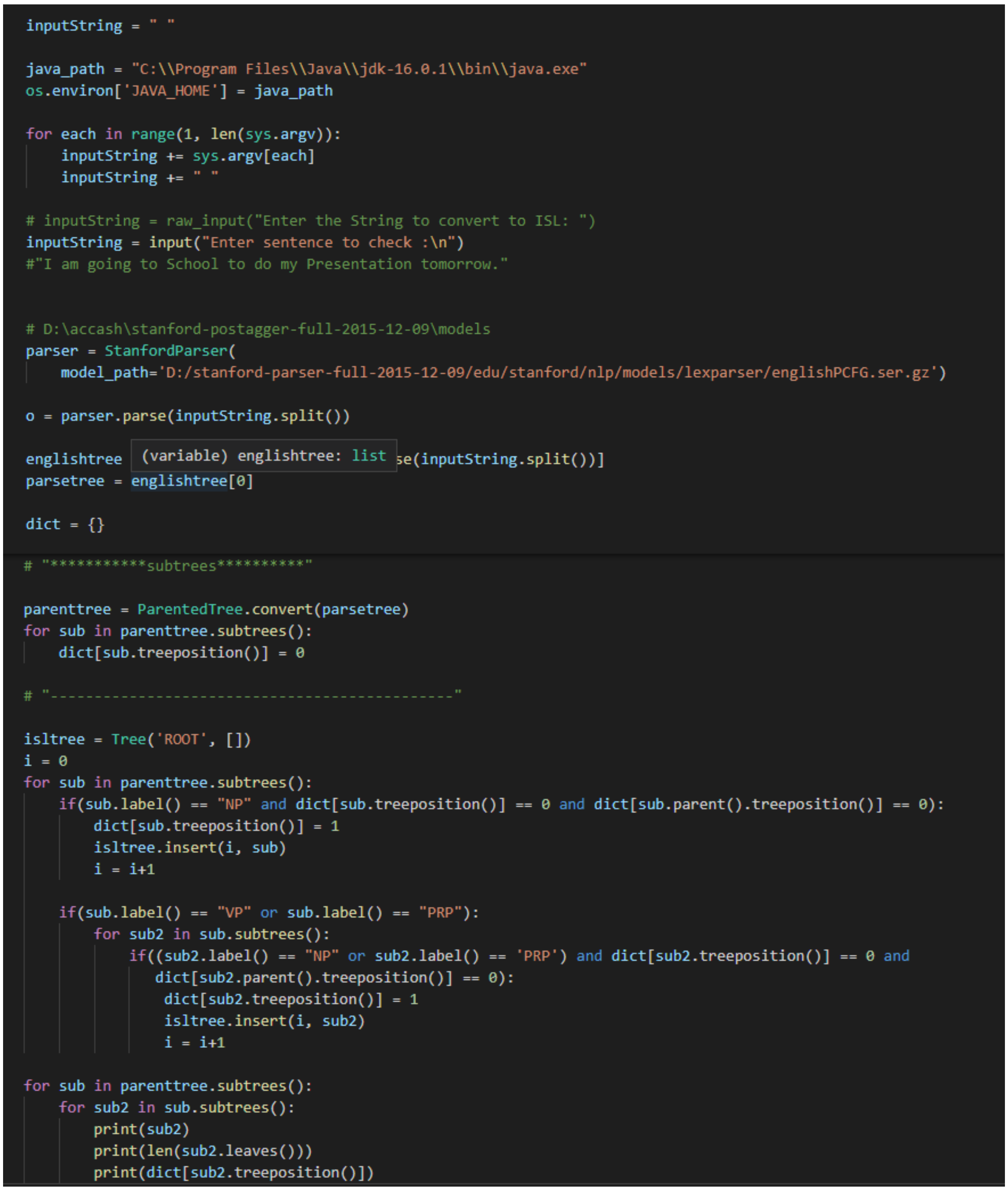
```python
def generate_subs(input_file):
    print('============== Extract Audio Start==============================')
    video = VideoFileClip(input_file)
    audio = video.audio
    audio.write_audiofile('audio.wav')
    print('============== Extract Audio Complete==============================')

    # using file from the same location
    AUDIO_FILE = path.join("audio.wav")

    print('============== Subtitle Generate Start==============================')
    # using thr STT Model
    r = sr.Recognizer()
    with sr.AudioFile(AUDIO_FILE) as source:
        audio = r.record(source)                    # reads the entire audio file

    # recognize the speech
    try:
        recognized_sentence = r.recognize_google(audio)
    except sr.UnknownValueError:
        print("Google Speech recognition could not understand audio")
    except sr.RequestError as e:
        print(
            "Could not request results from Google Speech Recognition service; {0}".format(e))

    sentence = recognized_sentence
    word_arr = sentence.split()
    word_count = len(word_arr)
    print('number of words: '+str(word_count))
start_time = datetime.datetime(100, 1, 1, 0, 0, 0)  # 00.00.00
block_num = 0
subs = []


def createSubs(arr, wordscount, current_time, blk):
    if wordscount <= 6:
        # assume 0.5 seconds taken to read a word
        time_add = len(arr[0:wordscount])*0.5
        end_time = current_time + datetime.timedelta(0, time_add)

        # convert timestamps to String
        str_current_time = str(current_time.time())
        str_end_time = str(end_time.time())

        blk = blk+1  # incrementing the block numbers

        str_list = arr[0:wordscount]
        str_sentence = ' '.join(map(str, str_list))
        subs.append(str_sentence)

        with open("output.txt", "a") as f:
            f.write(str(blk))
            f.write("\n")
            f.write(str_current_time)
            f.write("-->")
            f.write(str_end_time)
            f.write("\n")
            f.write(str_sentence)
            f.write("\n")
            f.write("\n")

        print(str(blk)+"\n"+str_current_time+"-->"+str_end_time+"\n"+str_sentence+"\n")

        return
```

The captioning script is responsible to create captions to the video and save a text file with sentences including timeframes.

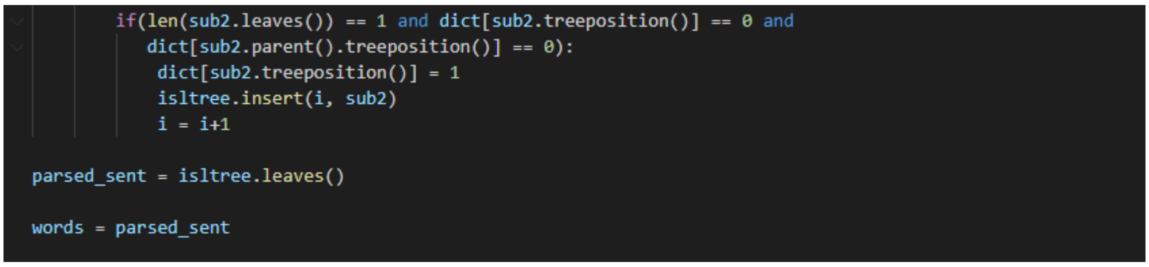
2.9.2 Text to American Sign Language

2.9.2.1 Parsing of the video caption

Rule-based grammar system for translating one language into another must be familiar with both the source and target language. Parsing is the answer to getting this grammatical structure. Stanford Parser is capable of producing three different releases, part of speech Free grammatical representation of the context of the tag text, phrase structure and type pro representation. The parser uses the Penn tree tags to parse the English sentence. The below code shows how it is implemented in python.



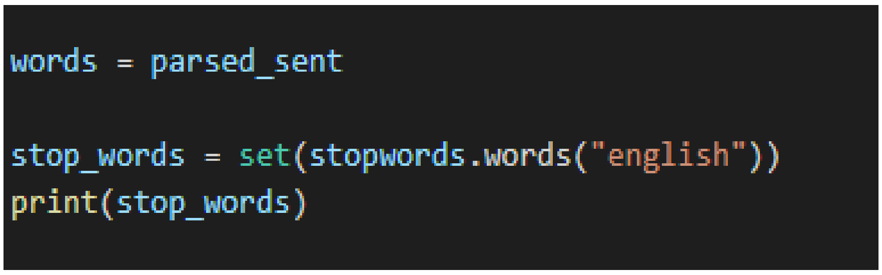
```
inputString = " "

java_path = "C:\\Program Files\\Java\\jdk-16.0.1\\bin\\java.exe"
os.environ['JAVA_HOME'] = 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 = input("Enter sentence to check :\n")
#"I am going to School to do my Presentation tomorrow."


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

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

englishtree  (variable) englishtree: list se(inputString.split())]
parsetree = englishtree[0]

dict = {}

# "***********subtrees**********"

parenttree = ParentedTree.convert(parsetree)
for sub in parenttree.subtrees():
    dict[sub.treeposition()] = 0

# "----------------------------------------------"

isltree = Tree('ROOT', [])
i = 0
for sub in parenttree.subtrees():
    if(sub.label() == "NP" and dict[sub.treeposition()] == 0 and dict[sub.parent().treeposition()] == 0):
        dict[sub.treeposition()] = 1
        isltree.insert(i, sub)
        i = i+1

    if(sub.label() == "VP" or sub.label() == "PRP"):
        for sub2 in sub.subtrees():
            if((sub2.label() == "NP" or sub2.label() == 'PRP') and dict[sub2.treeposition()] == 0 and
               dict[sub2.parent().treeposition()] == 0):
                dict[sub2.treeposition()] = 1
                isltree.insert(i, sub2)
                i = i+1

for sub in parenttree.subtrees():
    for sub2 in sub.subtrees():
        print(sub2)
        print(len(sub2.leaves()))
        print(dict[sub2.treeposition()])
```



```
            if(len(sub2.leaves()) == 1 and dict[sub2.treeposition()] == 0 and
              dict[sub2.parent().treeposition()] == 0):
                dict[sub2.treeposition()] = 1
                isltree.insert(i, sub2)
                i = i+1

parsed_sent = isltree.leaves()

words = parsed_sent
```

2.9.2.2 Stop Words Removal

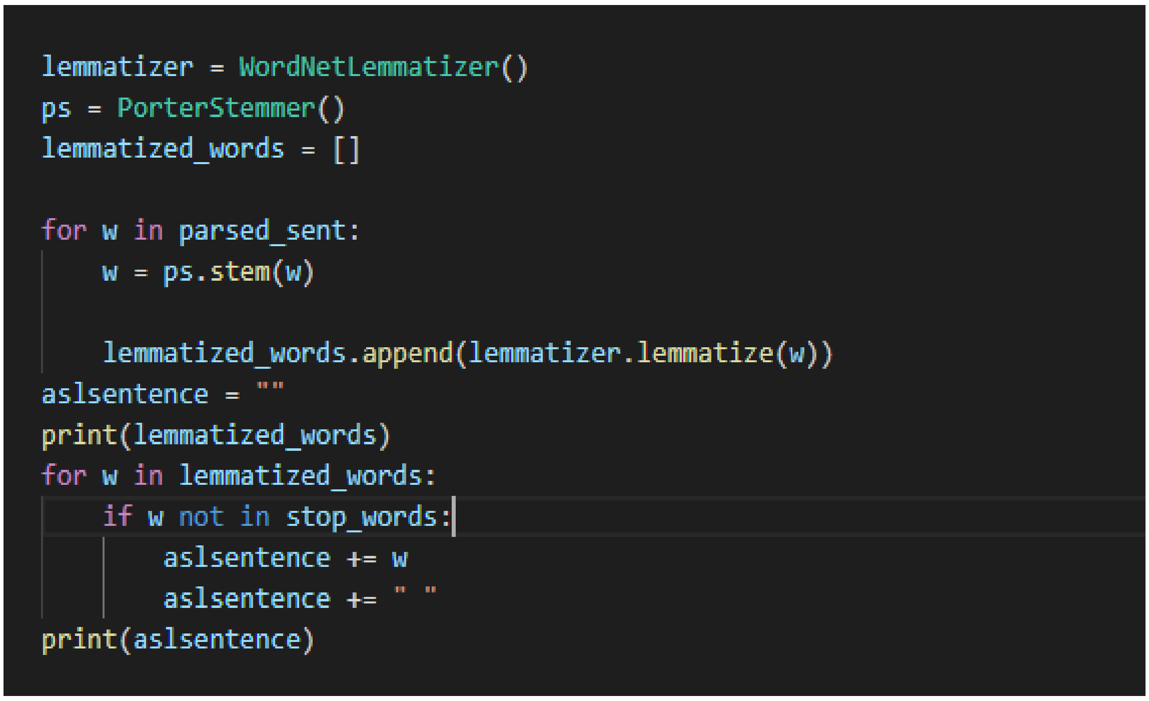
In the process of preprocessing, it is clearly seen that some words in a caption is not applicable for conversion itself. Such words are referred to as stop words and the elimination of those words is done in this step. Stop words include various types, among which a few of them can be determiners (the, a, an, another), coordinating junctions (for, an, nor, but, yet), prepositions (under, from, on, of, towards), plurals (‘book’ instead of ‘books’), interjections, etc. this word will be removed in this phase which will be easier to connect the words with the ASL dataset by eliminating the stop words. The below code represent how stop word will be removed from the input caption.



```
words = parsed_sent

stop_words = set(stopwords.words("english"))
print(stop_words)
```

2.9.2.3 Lemmatization and synonym replacement

The words containing prefixes or suffixes, tense related suffixes, etc. are reduced into its root form. For example, words like ‘adjustment’, ‘adjustable’, ‘adjusting’ are reduced into its root ‘adjust’. Further, if the root word is not found in the sign database, it is matched with its synonym and then the relevant sign will be found. The below code shows how the lemmatization part is done



```
lemmatizer = WordNetLemmatizer()
ps = PorterStemmer()
lemmatized_words = []

for w in parsed_sent:
    w = ps.stem(w)

    lemmatized_words.append(lemmatizer.lemmatize(w))
aslsentence = ""
print(lemmatized_words)
for w in lemmatized_words:
    if w not in stop_words:
        aslsentence += w
        aslsentence += " "
print(aslsentence)
```

2.9.2.4 Video Conversion

After completing the above steps, we get the ASL converted text to find matches from each word database based on basic string matching algorithm between processed input text and labels of videos. Finally a one set of videos can be displayed on the screen one after the other which is the final stage of the process converting text into ASL videos.

This stage can be divided in to two parts:

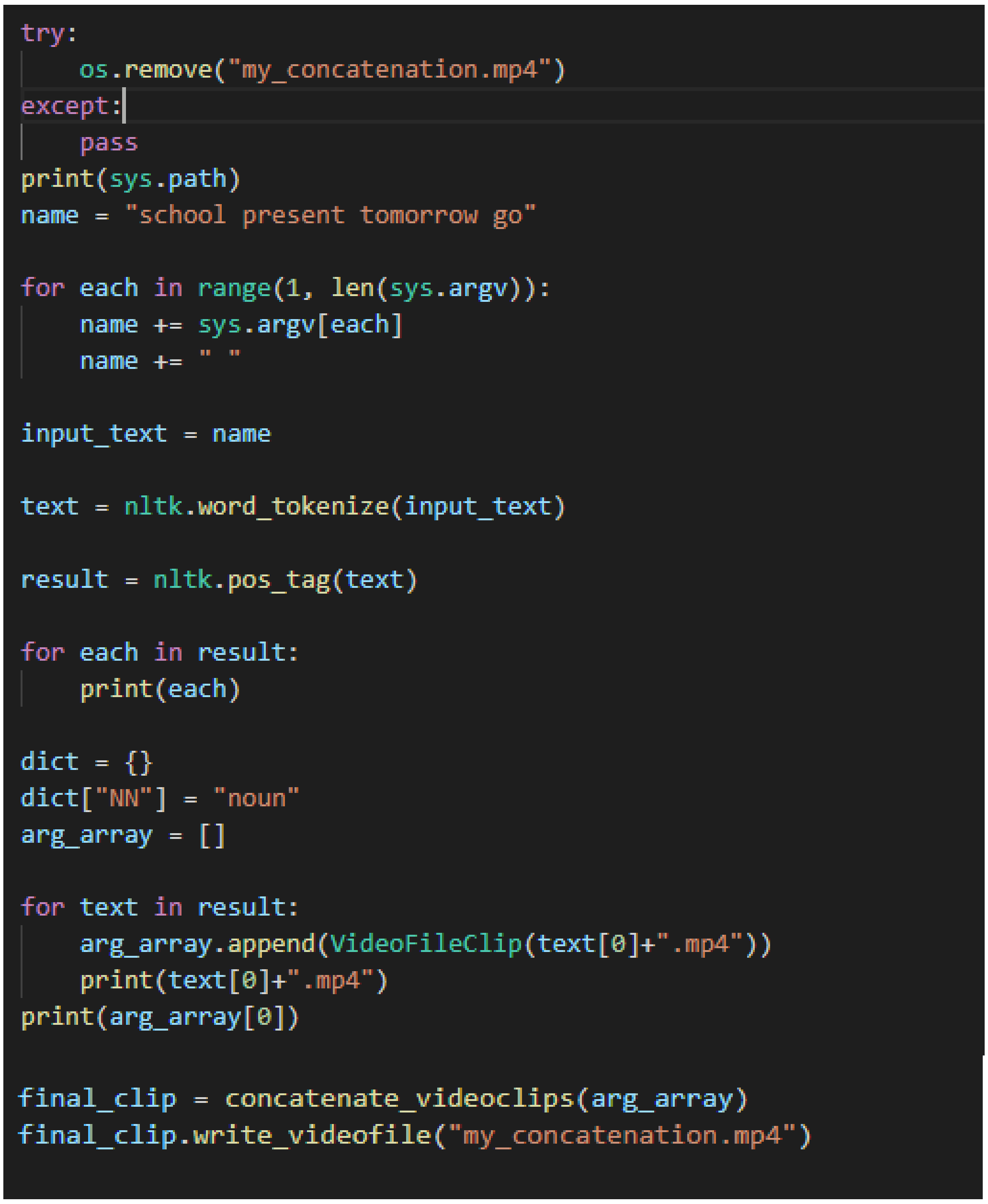
1. **Tokenizing the words**

   In this stage the ASL converted caption will be tokenized into words using NLTK.

2. **Connecting the video with tokenized words**

   Once the tokenization part is completed the system will find matches from each word database based on basic string matching algorithm between processed input text and labels of videos.

The below code shows how it is done:

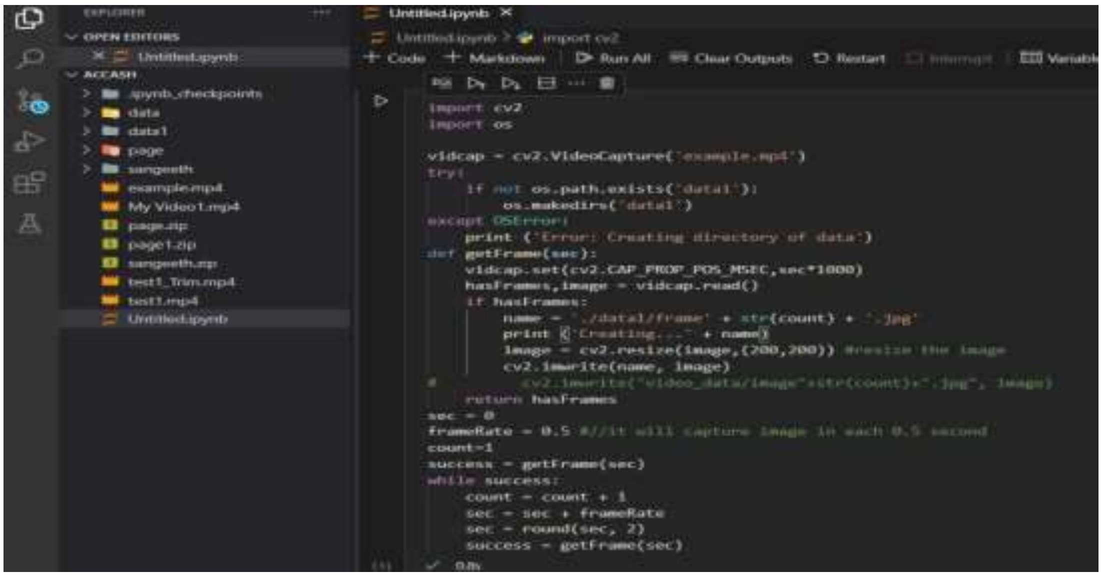


```
try:
    os.remove("my_concatenation.mp4")
except:
    pass
print(sys.path)
name = "school present tomorrow go"

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

input_text = name

text = nltk.word_tokenize(input_text)

result = nltk.pos_tag(text)

for each in result:
    print(each)

dict = {}
dict["NN"] = "noun"
arg_array = []

for text in result:
    arg_array.append(VideoFileClip(text[0]+".mp4"))
    print(text[0]+".mp4")
print(arg_array[0])

final_clip = concatenate_videoclips(arg_array)
final_clip.write_videofile("my_concatenation.mp4")
```

2.9.3 American Sign Language to text

2.9.3.1 Video pre-processing

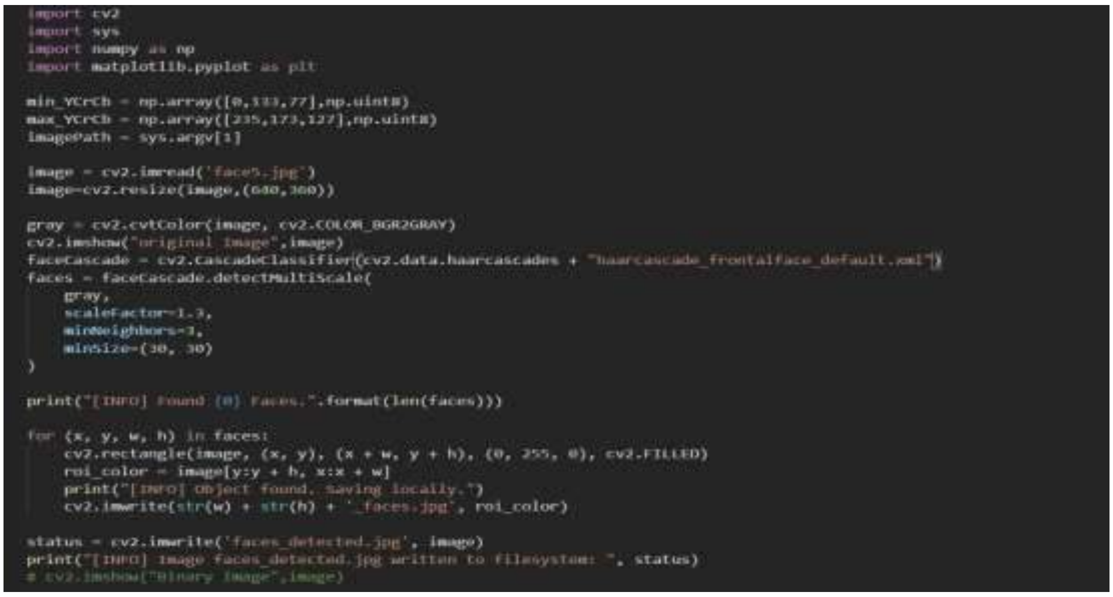
This is contains three sub processing. Converting video to a frame-by-frame format, adjusting contrast, and resizing images are all examples of sub processing. First of all, the system taking video from the database and start video-processing. In the first step of video pre-processing video input will convert frame by frame and store as sequences of images. The second step is taking images one by one and analysis the contrast and adjust the contrast according to the requirement. The last step is to resize the image. This step is maintaining a unique size of image and resolution for all images. It will deduce the analysis time for the calculations.



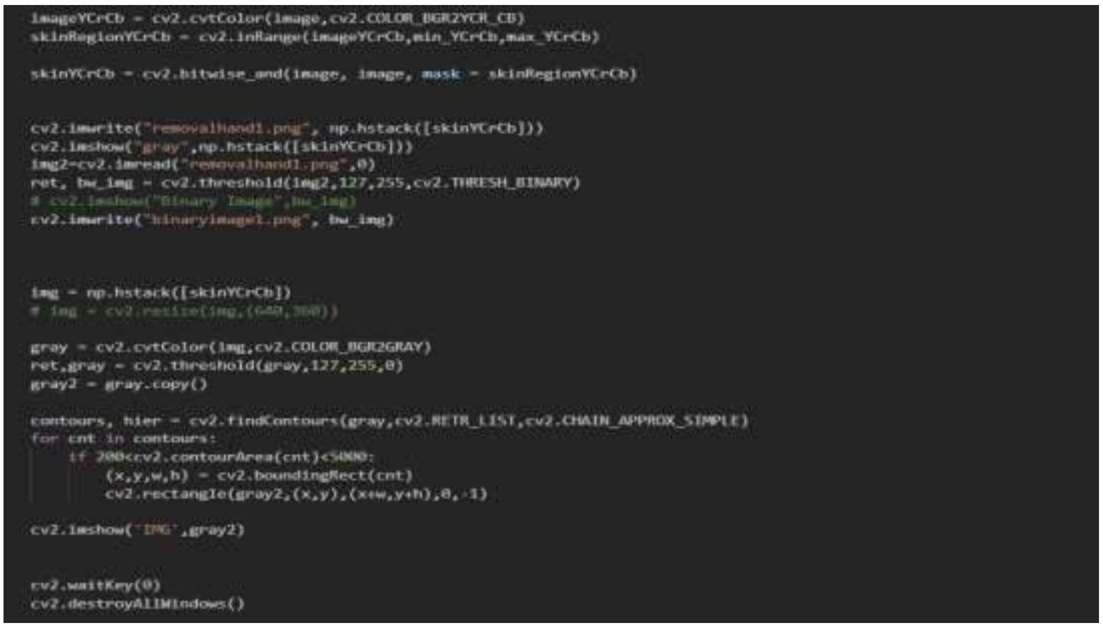
```python
import cv2
import os

vidcap = cv2.VideoCapture('example.mp4')
try:
    if not os.path.exists('data1'):
        os.makedirs('data1')
except OSError:
    print ('Error: Creating directory of data')
def getFrame(sec):
    vidcap.set(cv2.CAP_PROP_POS_MSEC,sec*1000)
    hasFrames,image = vidcap.read()
    if hasFrames:
        name = './data1/frame' + str(count) + '.jpg'
        print ('Creating...' + name)
        image = cv2.resize(image,(200,200)) #resize the image
        cv2.imwrite(name, image)
#           cv2.imwrite("video_data/image"+str(count)+".jpg", image)
    return hasFrames
sec = 0
frameRate = 0.5 #//it will capture image in each 0.5 second
count=1
success = getFrame(sec)
while success:
    count = count + 1
    sec = sec + frameRate
    sec = round(sec, 2)
    success = getFrame(sec)
```

2.9.3.2 Removal background object and convert into binary form

This is stage two of my research project. I need to remove background objects from the images which are stored in the database after the pre-processing. This very important stage because when we recognize hand gestures image contain only the hand gesture region then easy to detect the gestures.



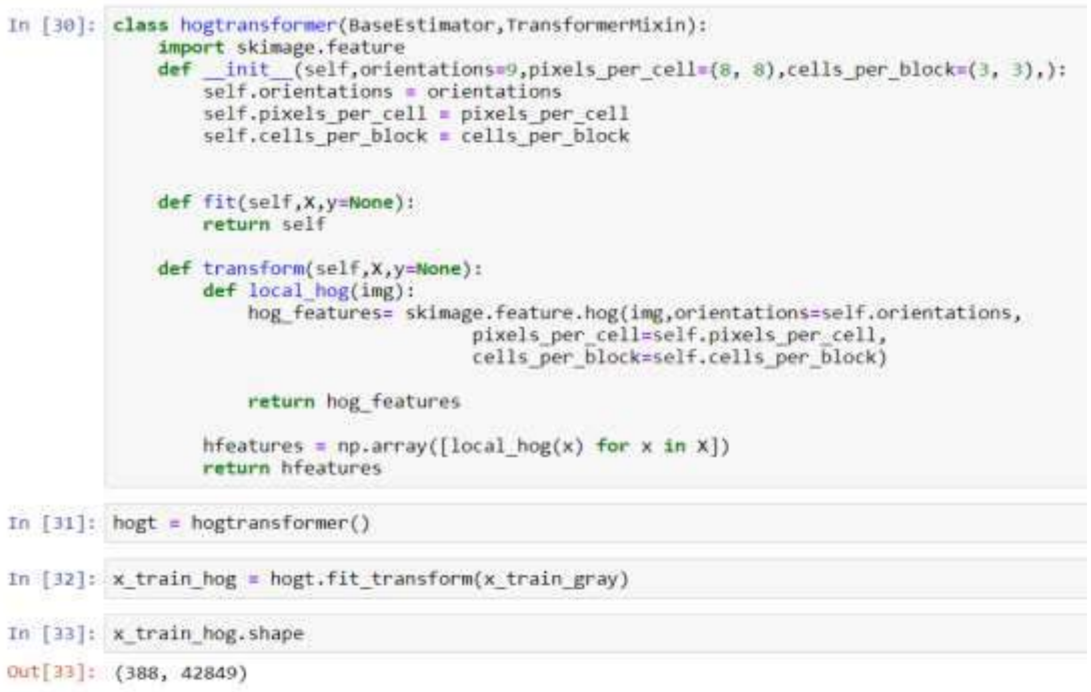
```python
import cv2
import sys
import numpy as np
import matplotlib.pyplot as plt

min_YCrCb = np.array([0,133,77],np.uint8)
max_YCrCb = np.array([235,173,127],np.uint8)
imagePath = sys.argv[1]

image = cv2.imread('face5.jpg')
image=cv2.resize(image,(640,360))

gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
cv2.imshow("original image",image)
faceCascade = cv2.CascadeClassifier(cv2.data.haarcascades + "haarcascade_frontalface_default.xml")
faces = faceCascade.detectMultiScale(
    gray,
    scaleFactor=1.3,
    minNeighbors=3,
    minSize=(30, 30)
)

print("[INFO] Found {0} Faces.".format(len(faces)))

for (x, y, w, h) in faces:
    cv2.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), cv2.FILLED)
    roi_color = image[y:y + h, x:x + w]
    print("[INFO] Object found. Saving locally.")
    cv2.imwrite(str(w) + str(h) + '_faces.jpg', roi_color)

status = cv2.imwrite('faces_detected.jpg', image)
print("[INFO] Image faces_detected.jpg written to filesystem: ", status)
# cv2.imshow("Binary image",image)
```



```
imageYCrCb = cv2.cvtColor(image,cv2.COLOR_BGR2YCR_CB)
skinRegionYCrCb = cv2.inRange(imageYCrCb,min_YCrCb,max_YCrCb)

skinYCrCb = cv2.bitwise_and(image, image, mask = skinRegionYCrCb)


cv2.imwrite("removalhand1.png", np.hstack([skinYCrCb]))
cv2.imshow("gray",np.hstack([skinYCrCb]))
img2=cv2.imread("removalhand1.png",0)
ret, bw_img = cv2.threshold(img2,127,255,cv2.THRESH_BINARY)
# cv2.imshow("Binary Image",bw_img)
cv2.imwrite("binaryimage1.png", bw_img)



img = np.hstack([skinYCrCb])
# img = cv2.resize(img,(640,360))

gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
ret,gray = cv2.threshold(gray,127,255,0)
gray2 = gray.copy()

contours, hier = cv2.findContours(gray,cv2.RETR_LIST,cv2.CHAIN_APPROX_SIMPLE)
for cnt in contours:
    if 200<cv2.contourArea(cnt)<5000:
        (x,y,w,h) = cv2.boundingRect(cnt)
        cv2.rectangle(gray2,(x,y),(x+w,y+h),0,-1)

cv2.imshow('IMG',gray2)


cv2.waitKey(0)
cv2.destroyAllWindows()
```

2.9.3.3 Feature Extraction

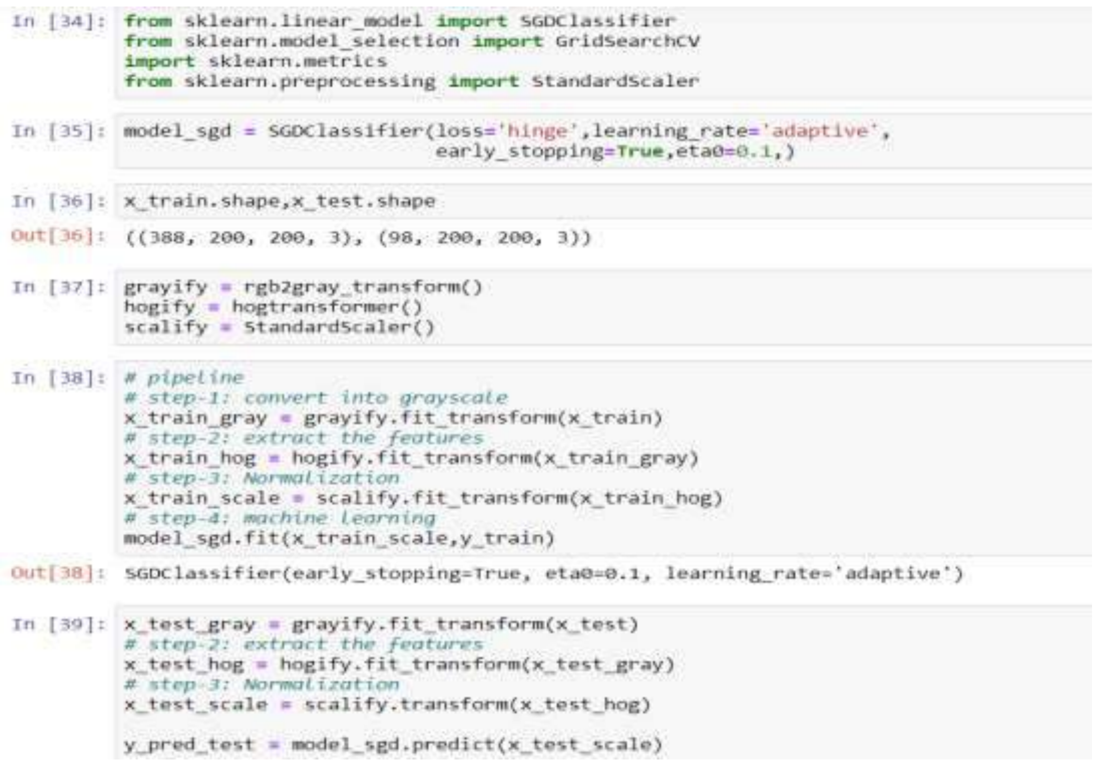
The method of distributing and reducing the initial set of source data into more manageable categories is called feature extraction. Below images show the result of feature extraction.



```
In [30]: class hogtransformer(BaseEstimator,TransformerMixin):
             import skimage.feature
             def __init__(self,orientations=9,pixels_per_cell=(8, 8),cells_per_block=(3, 3),):
                 self.orientations = orientations
                 self.pixels_per_cell = pixels_per_cell
                 self.cells_per_block = cells_per_block


             def fit(self,X,y=None):
                 return self

             def transform(self,X,y=None):
                 def local_hog(img):
                     hog_features= skimage.feature.hog(img,orientations=self.orientations,
                                                  pixels_per_cell=self.pixels_per_cell,
                                                  cells_per_block=self.cells_per_block)

                     return hog_features

                 hfeatures = np.array([local_hog(x) for x in X])
                 return hfeatures

In [31]: hogt = hogtransformer()

In [32]: x_train_hog = hogt.fit_transform(x_train_gray)

In [33]: x_train_hog.shape
Out[33]: (388, 42849)
```

2.9.3.4 Classification

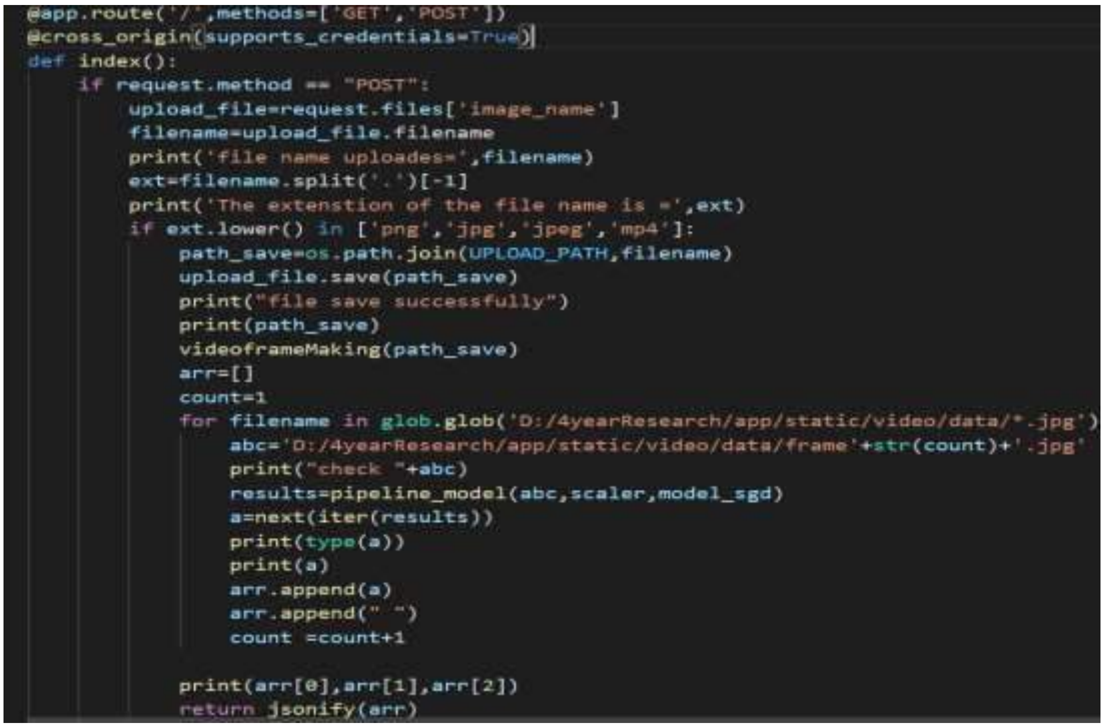
I used here SGD classifier. The Stochastic Gradient Descent (SGD) optimization technique is used to determine the values of parameters/coefficients of functions that minimize a cost function. After the classification outputs will save into the specific database location.



```
In [34]: from sklearn.linear_model import SGDClassifier
         from sklearn.model_selection import GridSearchCV
         import sklearn.metrics
         from sklearn.preprocessing import StandardScaler

In [35]: model_sgd = SGDClassifier(loss='hinge',learning_rate='adaptive',
                                   early_stopping=True,eta0=0.1,)

In [36]: x_train.shape,x_test.shape

Out[36]: ((388, 200, 200, 3), (98, 200, 200, 3))

In [37]: grayify = rgb2gray_transform()
         hogify = hogtransformer()
         scalify = StandardScaler()

In [38]: # pipeline
         # step-1: convert into grayscale
         x_train_gray = grayify.fit_transform(x_train)
         # step-2: extract the features
         x_train_hog = hogify.fit_transform(x_train_gray)
         # step-3: Normalization
         x_train_scale = scalify.fit_transform(x_train_hog)
         # step-4: machine learning
         model_sgd.fit(x_train_scale,y_train)

Out[38]: SGDClassifier(early_stopping=True, eta0=0.1, learning_rate='adaptive')

In [39]: x_test_gray = grayify.fit_transform(x_test)
         # step-2: extract the features
         x_test_hog = hogify.fit_transform(x_test_gray)
         # step-3: Normalization
         x_test_scale = scalify.transform(x_test_hog)

         y_pred_test = model_sgd.predict(x_test_scale)
```

2.9.3.4 Backend flask rest API

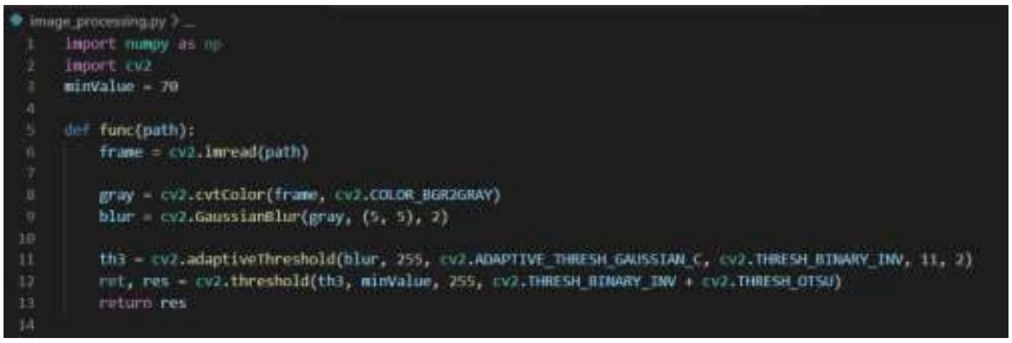
I have used flask for my back API creation. The below code shows how created rest API for the sign language to convert into text.



```python
@app.route('/',methods=['GET','POST'])
@cross_origin(supports_credentials=True)
def index():
    if request.method == "POST":
        upload_file=request.files['image_name']
        filename=upload_file.filename
        print('file name uploades=',filename)
        ext=filename.split('.')[-1]
        print('The extenstion of the file name is =',ext)
        if ext.lower() in ['png','jpg','jpeg','mp4']:
            path_save=os.path.join(UPLOAD_PATH,filename)
            upload_file.save(path_save)
            print("file save successfully")
            print(path_save)
            videoframeMaking(path_save)
            arr=[]
            count=1
            for filename in glob.glob('D:/4yearResearch/app/static/video/data/*.jpg')
                abc='D:/4yearResearch/app/static/video/data/frame'+str(count)+'.jpg'
                print("check "+abc)
                results=pipeline_model(abc,scaler,model_sgd)
                a=next(iter(results))
                print(type(a))
                print(a)
                arr.append(a)
                arr.append(" ")
                count =count+1

            print(arr[0],arr[1],arr[2])
            return jsonify(arr)
```

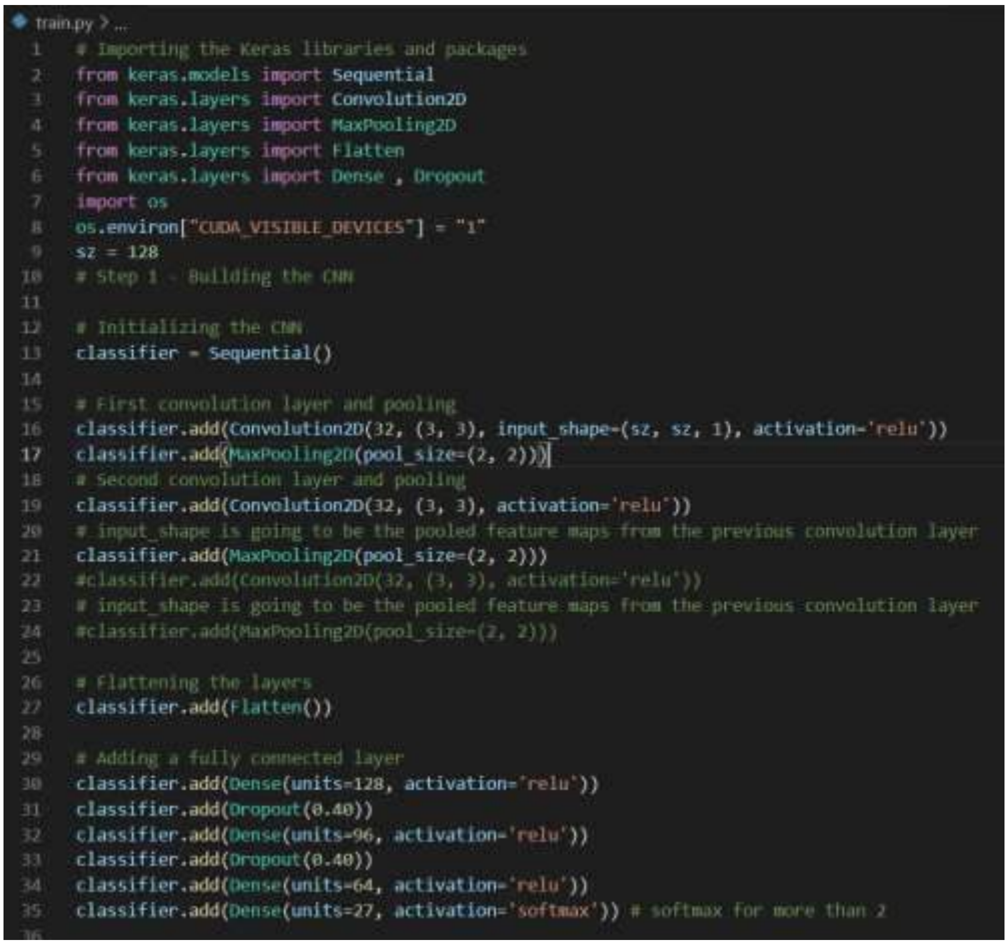
2.9.4 Teaching Assistance

The code implementation of image preprocessing.



image_processing.py > ...
```python
import numpy as np
import cv2
minValue = 70

def func(path):
    frame = cv2.imread(path)

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    blur = cv2.GaussianBlur(gray, (5, 5), 2)

    th3 = cv2.adaptiveThreshold(blur, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY_INV, 11, 2)
    ret, res = cv2.threshold(th3, minValue, 255, cv2.THRESH_BINARY_INV + cv2.THRESH_OTSU)
    return res
```

The code implementation of building CNN



```python
# Importing the Keras libraries and packages
from keras.models import Sequential
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten
from keras.layers import Dense , Dropout
import os
os.environ["CUDA_VISIBLE_DEVICES"] = "1"
sz = 128
# Step 1 - Building the CNN

# Initializing the CNN
classifier = Sequential()

# First convolution layer and pooling
classifier.add(Convolution2D(32, (3, 3), input_shape=(sz, sz, 1), activation='relu'))
classifier.add(MaxPooling2D(pool_size=(2, 2)))
# Second convolution layer and pooling
classifier.add(Convolution2D(32, (3, 3), activation='relu'))
# input_shape is going to be the pooled feature maps from the previous convolution layer
classifier.add(MaxPooling2D(pool_size=(2, 2)))
#classifier.add(Convolution2D(32, (3, 3), activation='relu'))
# input_shape is going to be the pooled feature maps from the previous convolution layer
#classifier.add(MaxPooling2D(pool_size=(2, 2)))

# Flattening the layers
classifier.add(Flatten())

# Adding a fully connected layer
classifier.add(Dense(units=128, activation='relu'))
classifier.add(Dropout(0.40))
classifier.add(Dense(units=96, activation='relu'))
classifier.add(Dropout(0.40))
classifier.add(Dense(units=64, activation='relu'))
classifier.add(Dense(units=27, activation='softmax')) # softmax for more than 2
```

2.10 Testing

For testing our system every all the modules were unit tested individually. Once all the modules are tested, they were integrated together and integration test was done. When doing integration testing the systems performance was evaluated. Section 2.10.1 has some test cases that were used for integration testing.

Table 2 - Illustration Test Case 01

TEST CASE 01	
Pre-requirements	PC or laptop
Description	Testing the automatic low light detection using the proposed algorithm
Test Procedure	Step 1: Insert the video to script Step 2: Run the low light identification script

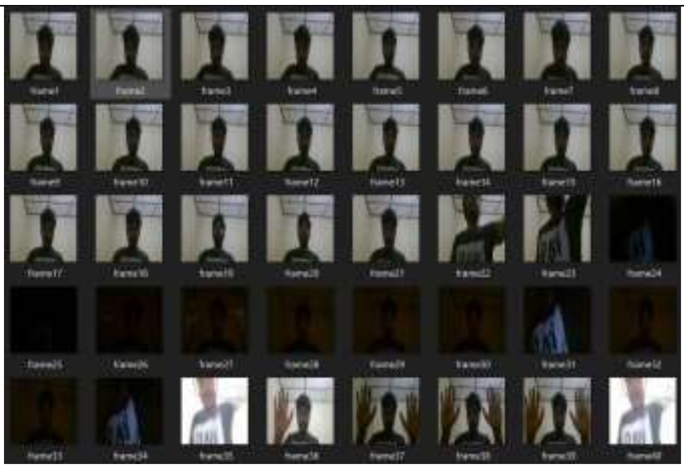
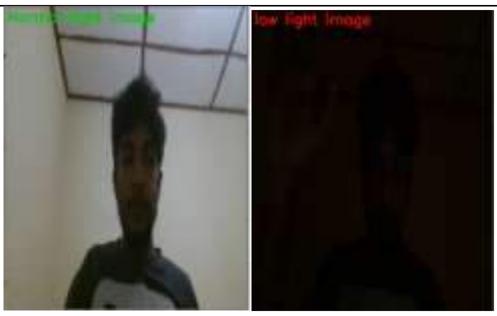
Input	
Expected Output	The output video should contain the low light and normal light labeled frames
Actual results	 All the low light and normal light frames on this video was identified as normal light and low light image.
Result of test case	Pass

Table 3 - Illustration Test Case 02

TEST CASE 02	
Pre-requirements	PC or laptop
Description	Testing the low light enhancement using proposed algorithm.
Test Procedure	Step 1: Insert a low light image. Step 2: Run the low light enhancement algorithm.

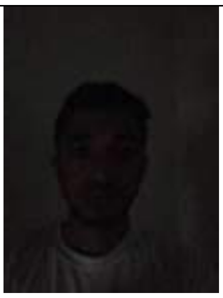
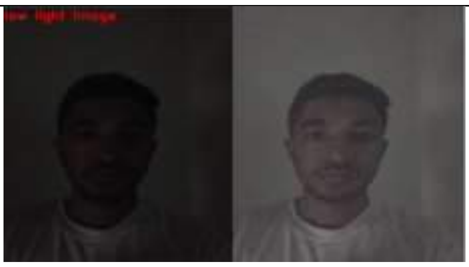
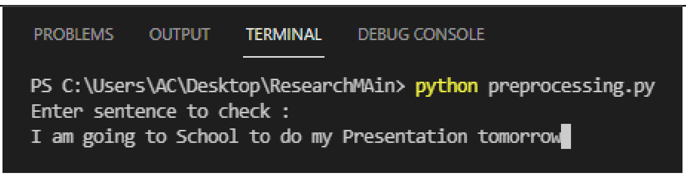
Input	
Expected Output	The low light image has to be enhanced automatically.
Actual results	
Result of test case	Pass

Table 4 - Illustration Test Case 03

TEST CASE 03	
Pre-requirements	PC or laptop
Description	Testing the captioning function with the proposed algorithm.
Test Procedure	Step 1: Insert the video to be captioned Step 2: Run the script.
Input	Speech in the video - This is a low light environment and sound test. As you can see now I will turn off the light. Now the room is a bit dark. Now I will turn on the light now we have normal illumination.
Expected Output	1 00:00:00-->00:00:03 This is a low light environment 2 00:00:03-->00:00:06 and sound test. As you can 3

	00:00:06-->00:00:09 see now I will turn off 4 00:00:09-->00:00:12 the light. Now the room is 5 00:00:12-->00:00:15 a bit dark. Now I will
Actual results	1 00:00:00-->00:00:03 light environment and sound test as 2 00:00:03-->00:00:06 you can see now I will 3 00:00:06-->00:00:09 turn off the light now they 4 00:00:09-->00:00:12 now I will turn on the 5 00:00:12-->00:00:14.500000 light we have normal illumination
Result of test case	Pass

Table 5 - Illustration Test Case 04

TEST CASE 04	
Pre-requirements	PC or laptop
Description	Testing the ASL grammar conversion.
Test Procedure	Step 1: Insert the sentence to convert. Step 2: Run the script.
Input	 `PS C:\Users\AC\Desktop\ResearchMAin> python preprocessing.py` `Enter sentence to check :` `I am going to School to do my Presentation tomorrow`

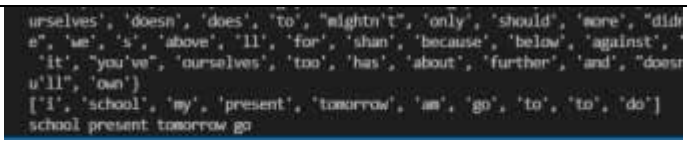
Expected Output	ASL converted grammar - school present tomorrow go
Actual results	 urselves', 'doesn', 'does', 'to', "mightn't", 'only', 'should', 'more', "didn e", 'we', 's', 'above', 'll', 'for', 'shan', 'because', 'below', 'against', ' 'it', "you've", 'ourselves', 'too', 'has', 'about', 'further', 'and', "doesn u'll", 'own'} ['i', 'school', 'my', 'present', 'tomorrow', 'am', 'go', 'to', 'to', 'do'] school present tomorrow go
Result of test case	Pass

Table 6 - Illustration Test Case 05

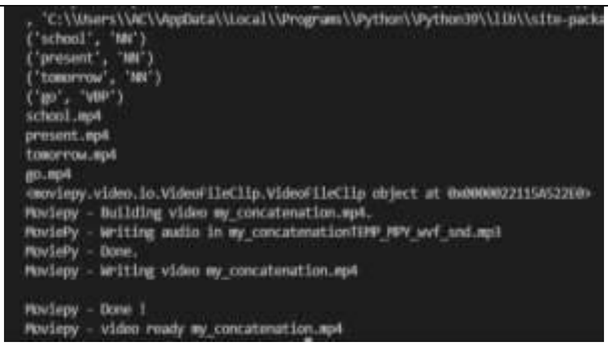
TEST CASE 05	
Pre-requirements	PC or laptop
Description	Testing the video conversion.
Test Procedure	Step 1: Run the script with input(caption)
Input	ASL grammar converted caption
Expected Output	ASL video for each word should be concatenated and displayed as one video.
Actual results	 , 'C:\\Users\\AC\\AppData\\Local\\Programs\\Python\\Python39\\lib\\site-packa ('school', 'NN') ('present', 'NN') ('tomorrow', 'NN') ('go', 'VBP') school.mp4 present.mp4 tomorrow.mp4 go.mp4 <moviepy.video.io.VideoFileClip.VideoFileClip object at 0x0000022115A522E0> Moviepy - Building video my_concatenation.mp4. MoviePy - Writing audio in my_concatenationTEMP_MPY_wvf_snd.mp3 MoviePy - Done. Moviepy - Writing video my_concatenation.mp4 Moviepy - Done ! Moviepy - video ready my_concatenation.mp4
Result of test case	Pass

Table 7 - Illustration Test Case 06

TEST CASE 06	
Pre-requirements	PC or laptop
Description	Testing the video-pre processing
Test Procedure	1.Write pre-processing code 2.Run the Script 3. Save the pre-processing file.

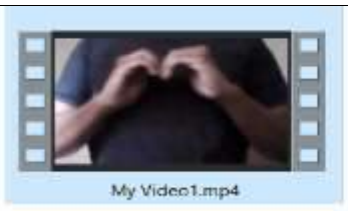
Input	
Expected Output	Same size of image and N number of frames.
Actual results	[illegible]
Result of test case	Pass

Table 8 - Illustration Test Case 07

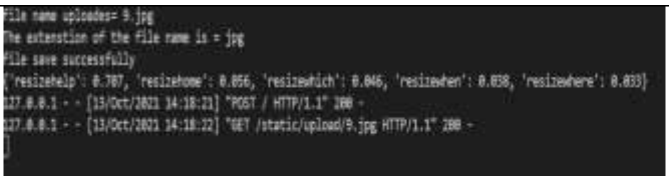
TEST CASE 07	
Pre-requirements	PC or laptop
Description	Testing the model accuracy of hand gesture
Test Procedure	1.Write model accuracy code 2.Run the Script
Input	
Expected Output	Detect the particular hand gesture and display name
Actual results	 file name uplodes= 9.jpg The extenstion of the file name is = jpg file save successfully {'resizehelp': 0.787, 'resizehome': 0.056, 'resizewhich': 0.046, 'resizewhen': 0.038, 'resizewhere': 0.033} 127.0.0.1 - - [13/Oct/2021 14:18:21] "POST / HTTP/1.1" 200 - 127.0.0.1 - - [13/Oct/2021 14:18:22] "GET /static/upload/9.jpg HTTP/1.1" 200 -
Result of test case	Pass

Table 9 - Illustration Test Case 08

TEST CASE 08	
Pre-requirements	PC or laptop and webcam.
Description	Test the trained model by Developer

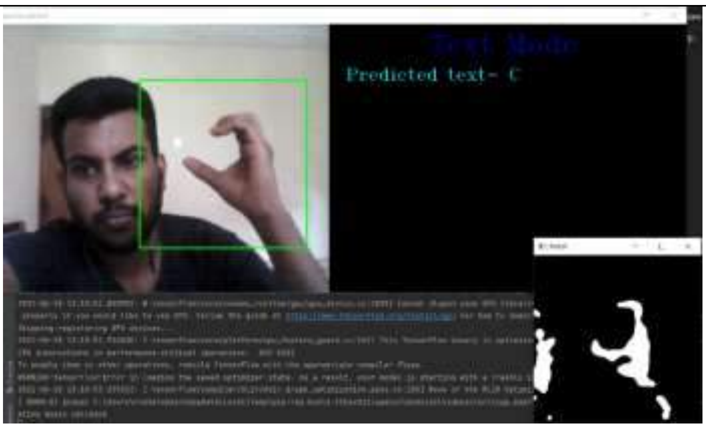
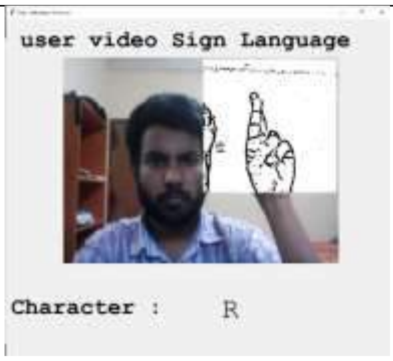
Test Procedure	Step 1: execute function switch on the web cam. Step 2: place the hand in the ROI. Step 3: system automatically detect the ASL. Step 4: quit from the window.
Input & Output	
Expected Output	Detection of letter 'C'
Result of test case	Pass

Table 10 - Illustration Test Case 09

TEST CASE 09	
Pre-requirements	PC or laptop and webcam.
Description	Test the system after UI update – by Developer
Test Procedure	Step 1: execute function switch on the web cam. Step 2: place the hand in the ROI(gaussian filter) Step 3: system automatically detect the ASL. Step 4: quit from the window.
Input	
Expected Output	Detection of letter 'R' and display result next to character

Result of test case	Pass

3. RESULTS & DISCUSSION

3.1 Results

On this section the output of each module of the system is discussed. The results of each modules are explained in the next sections.

3.1.1 Output of low light enhancement and captioning

On this section the low light and normal light images were tested. During testing each of the image was compared with its generated cumulative intensity histogram. As discussed in the implementation section of this functionality, If the percentage of low intensity pixels are greater than the threshold percentage then it was identified as a low light image.

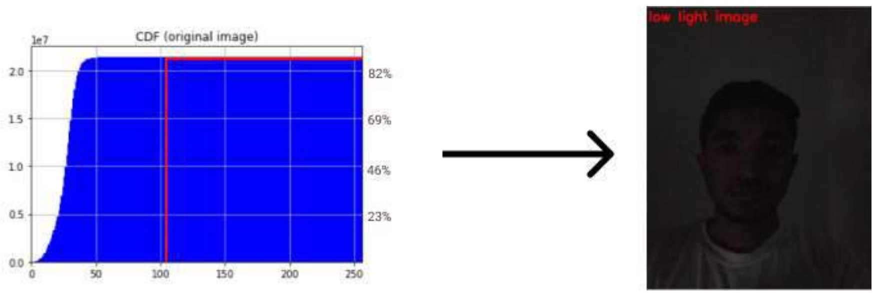


Figure 9 - low light image and cumulative histogram

The Figure 9 shows the low light image and its corresponding cumulative histogram. In the cumulative histogram the intensity value 103 (threshold) maps to a percentage almost equal to 100% which is >75%. Using this data, the algorithm was able to identify the image as a low light image.

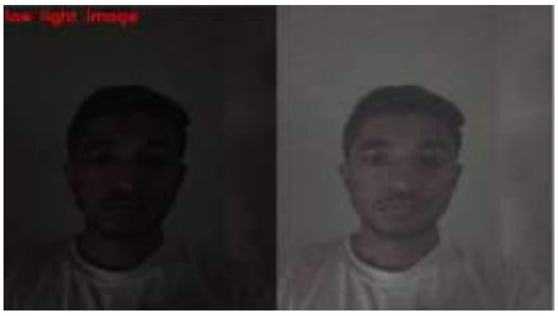
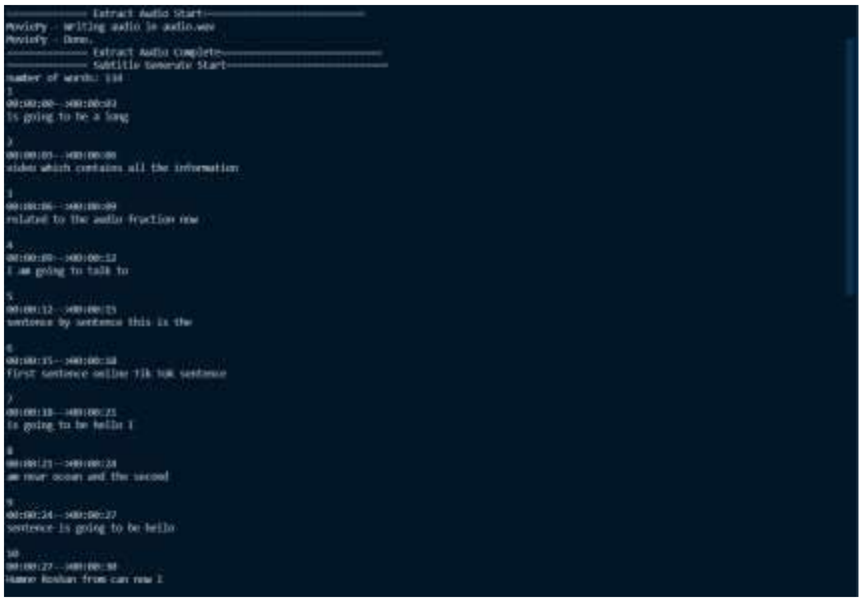


Figure 10 - Enhanced output

Figure 10 shows a low light input and its enhanced output. The enhanced output was obtained through the proposed enhanced algorithm. The enhancement algorithm also has the capability of reducing noise in the image. The enhanced output of Figure 10 has denoised itself using the denoising algorithm.

In the captioning module the output form GCP STT model cannot be directly used as the captioning output. It had to be further processed into individual sentences with timestamps. For this the captioning algorithm uses an additional logic to separate sentences and use their timestamps as discussed earlier. The final output of the captioning module has the sentences along with its timestamps Figure 11.



```
--------------- Extract Audio Start-------------------------------
MoviePy - Writing audio in audio.wav
MoviePy - Done.
--------------- Extract Audio Complete------------------------------
--------------- Subtitle Generate Start-----------------------------
number of words: 134
1
00:00:00 --> 00:00:03
is going to be a long

2
00:00:03 --> 00:00:06
video which contains all the information

3
00:00:06 --> 00:00:09
related to the audio fraction now

4
00:00:09 --> 00:00:12
I am going to talk to

5
00:00:12 --> 00:00:15
sentence by sentence this is the

6
00:00:15 --> 00:00:18
first sentence online tik tok sentence

7
00:00:18 --> 00:00:21
is going to be hello I

8
00:00:21 --> 00:00:24
am near ocean and the second

9
00:00:24 --> 00:00:27
sentence is going to be hello

10
00:00:27 --> 00:00:30
Hamne Roshan from can now I
```

Figure 11 - Caption output

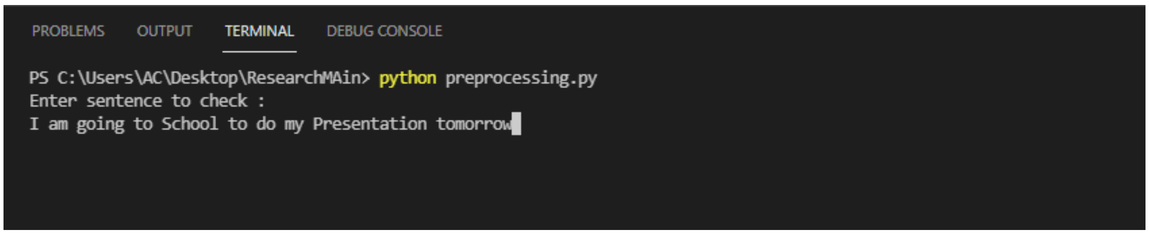
3.1.2 Output of Text to ASL

Output of ASL grammar conversion

When running the preprocessing script it will request for the video caption / text to convert it from English grammar to ASL grammar. Since the language used by the tutors and the American Sign Language have difference in terms of grammar rules. Therefore, it is essential to convert the parsed English text into a format that relates with the sign language grammar rules. This phase includes following steps,

1. Sentences reordering module based on ASL grammar rules.
2. Stop words removal
3. Stemming and lemmatization

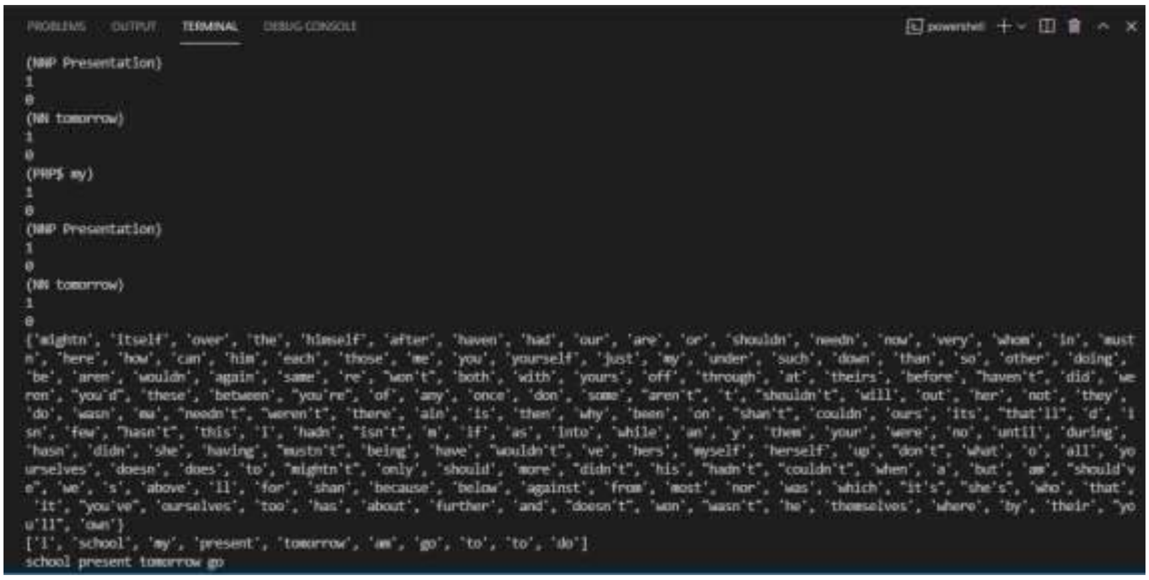
The below picture will show the output after running the script which has been build using NLTK techniques to convert English grammar to ASL grammar
The input caption to convert it into ASL grammar is: “I am going to school to do my presentation tomorrow” which is shown below.



```
PS C:\Users\AC\Desktop\ResearchMAin> python preprocessing.py
Enter sentence to check :
I am going to School to do my Presentation tomorrow
```

Figure 12 - Input Caption

The figure shows the output after converting the caption in to ASL grammar which is “School present tomorrow go”



```
(NNP Presentation)
1
0
(NN tomorrow)
1
0
(PRP$ my)
1
0
(NNP Presentation)
1
0
(NN tomorrow)
1
0
{'mightn', 'itself', 'over', 'the', 'himself', 'after', 'haven', 'had', 'our', 'are', 'or', 'shouldn', 'needn', 'now', 'very', 'whom', 'in', 'must
n', 'here', 'how', 'can', 'him', 'each', 'those', 'me', 'you', 'yourself', 'just', 'my', 'under', 'such', 'down', 'than', 'so', 'other', 'doing',
'be', 'aren', 'wouldn', 'again', 'same', 're', "won't", 'both', 'with', 'yours', 'off', 'through', 'at', 'theirs', 'before', "haven't", 'did', 'we
ren', "you'd", 'these', 'between', "you're", 'of', 'any', 'once', 'don', 'some', "aren't", 't', "shouldn't", 'will', 'out', 'her', 'not', 'they',
'do', 'wasn', 'ma', "needn't", "weren't", 'there', 'ain', 'is', 'then', 'why', 'been', 'on', "shan't", 'couldn', 'ours', 'its', "that'll", 'd', 'i
sn', 'few', "hasn't", 'this', 'i', 'hadn', "isn't", 'm', 'if', 'as', 'into', 'while', 'an', 'y', 'them', 'your', 'were', 'no', 'until', 'during',
'hasn', 'didn', 'she', 'having', "mustn't", 'being', 'have', "wouldn't", 've', 'hers', 'myself', 'herself', 'up', "don't", 'what', 'o', 'all', 'yo
urselves', 'doesn', 'does', 'to', "mightn't", 'only', 'should', 'more', "didn't", 'his', "hadn't", "couldn't", 'when', 'a', 'but', 'am', "should'v
e", 'we', 's', 'above', 'll', 'for', 'shan', 'because', 'below', 'against', 'from', 'most', 'nor', 'was', 'which', "it's", "she's", 'who', 'that',
 'it', "you've", 'ourselves', 'too', 'has', 'about', 'further', 'and', "doesn't", 'won', "wasn't", 'he', 'themselves', 'where', 'by', 'their', "yo
u'll", 'own'}
['I', 'school', 'my', 'present', 'tomorrow', 'am', 'go', 'to', 'to', 'do']
school present tomorrow go
```

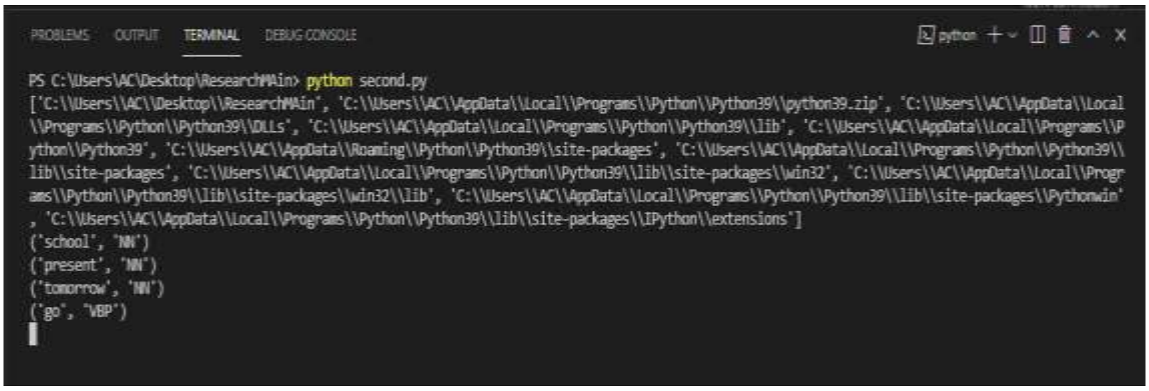
Figure 13 - ASL grammar output

Output of video conversion module

The final stage of converting text into ASL video is video conversion module. This module use the ASL converted caption to find matches for each word from the database based on string matching algorithm between ASL converted input caption and labels of videos. Finally a one set of videos can be displayed on the screen one after the other which is the final stage of the process converting text into ASL videos. This stage can be divided into two phase which is:

1. Tokenizing the ASL grammar converted caption / text
2. Video conversion

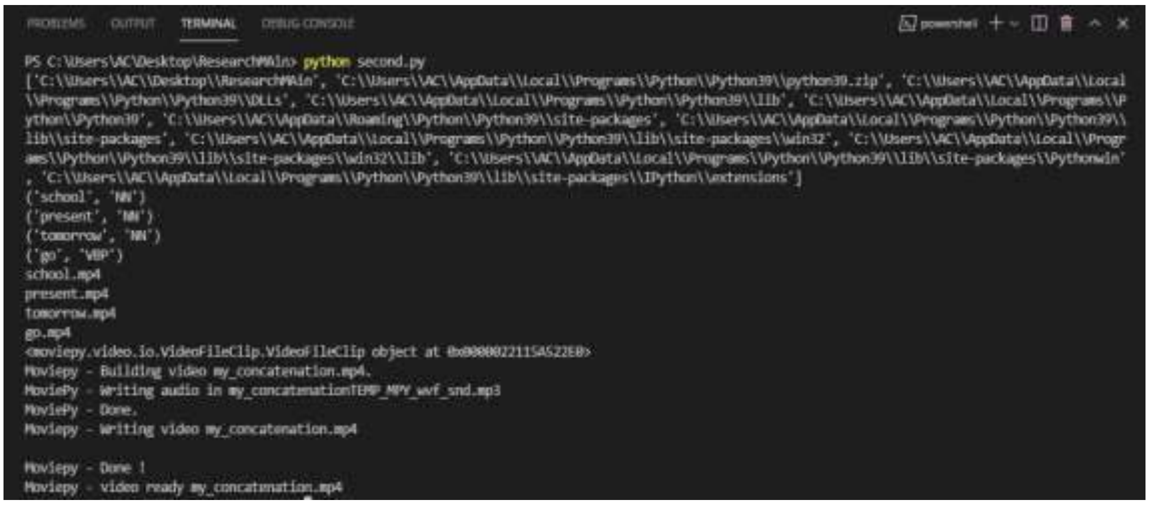
The output after running the video conversion script is shown below:



```
PS C:\Users\AC\Desktop\ResearchMAin> python second.py
['C:\\Users\\AC\\Desktop\\ResearchMAin', 'C:\\Users\\AC\\AppData\\Local\\Programs\\Python\\Python39\\python39.zip', 'C:\\Users\\AC\\AppData\\Local
\\Programs\\Python\\Python39\\DLLs', 'C:\\Users\\AC\\AppData\\Local\\Programs\\Python\\Python39\\lib', 'C:\\Users\\AC\\AppData\\Local\\Programs\\P
ython\\Python39', 'C:\\Users\\AC\\AppData\\Roaming\\Python\\Python39\\site-packages', 'C:\\Users\\AC\\AppData\\Local\\Programs\\Python\\Python39\\
lib\\site-packages', 'C:\\Users\\AC\\AppData\\Local\\Programs\\Python\\Python39\\lib\\site-packages\\win32', 'C:\\Users\\AC\\AppData\\Local\\Progr
ams\\Python\\Python39\\lib\\site-packages\\win32\\lib', 'C:\\Users\\AC\\AppData\\Local\\Programs\\Python\\Python39\\lib\\site-packages\\Pythonwin'
, 'C:\\Users\\AC\\AppData\\Local\\Programs\\Python\\Python39\\lib\\site-packages\\IPython\\extensions']
('school', 'NN')
('present', 'NN')
('tomorrow', 'NN')
('go', 'VBP')
▌
```

Figure 14 - Tokenization

The above figure shows the tokenized part of the caption after running the script and it search for the videos for each word from the database to concatenate.



```
PS C:\Users\AC\Desktop\ResearchMAin> python second.py
['C:\\Users\\AC\\Desktop\\ResearchMAin', 'C:\\Users\\AC\\AppData\\Local\\Programs\\Python\\Python39\\python39.zip', 'C:\\Users\\AC\\AppData\\Local
\\Programs\\Python\\Python39\\DLLs', 'C:\\Users\\AC\\AppData\\Local\\Programs\\Python\\Python39\\lib', 'C:\\Users\\AC\\AppData\\Local\\Programs\\P
ython\\Python39', 'C:\\Users\\AC\\AppData\\Roaming\\Python\\Python39\\site-packages', 'C:\\Users\\AC\\AppData\\Local\\Programs\\Python\\Python39\\
lib\\site-packages', 'C:\\Users\\AC\\AppData\\Local\\Programs\\Python\\Python39\\lib\\site-packages\\win32', 'C:\\Users\\AC\\AppData\\Local\\Progr
ams\\Python\\Python39\\lib\\site-packages\\win32\\lib', 'C:\\Users\\AC\\AppData\\Local\\Programs\\Python\\Python39\\lib\\site-packages\\Pythonwin'
, 'C:\\Users\\AC\\AppData\\Local\\Programs\\Python\\Python39\\lib\\site-packages\\IPython\\extensions']
('school', 'NN')
('present', 'NN')
('tomorrow', 'NN')
('go', 'VBP')
school.mp4
present.mp4
tomorrow.mp4
go.mp4
<moviepy.video.io.VideoFileClip.VideoFileClip object at 0x0000022115A522E0>
Moviepy - Building video my_concatenation.mp4.
MoviePy - Writing audio in my_concatenationTEMP_MPY_wvf_snd.mp3
MoviePy - Done.
Moviepy - Writing video my_concatenation.mp4

Moviepy - Done !
Moviepy - video ready my_concatenation.mp4
```

Figure 15 - Video Conversion

The above figure shows the final results after running the video conversion module. Once tokenization part is completed the system search for the videos for each words in the database to do the string matching algorithm to concatenate videos for each word and present it as a final output of the system.

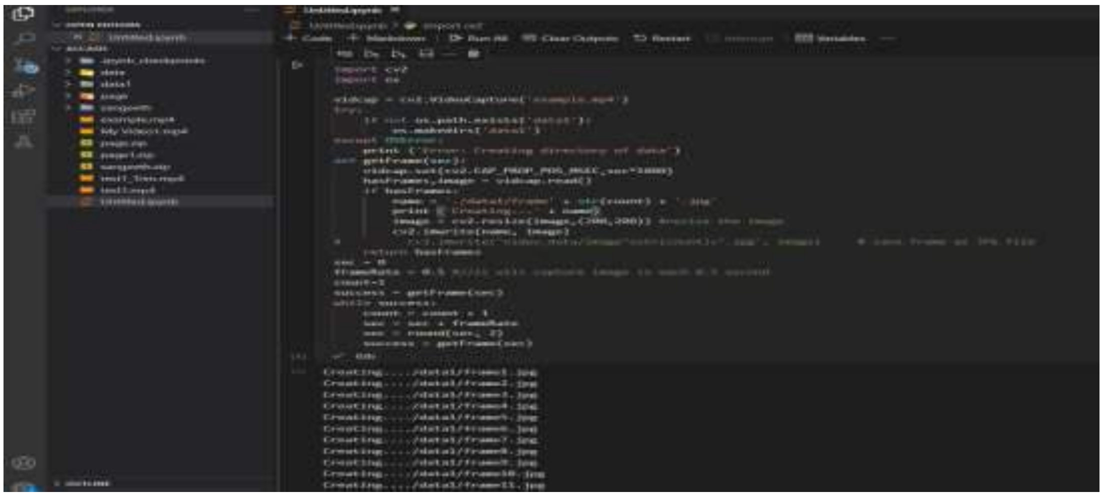
3.1.3 Output of ASL to Text

Here we will discuss the results of our research application. Deaf and a dumb student ask questions and clarify their doubts through the system are very important of this research. It is easy for students are able to discuss with tutors and makes a better understanding of subjects. When comes to e-learning platform especially deaf and dumb students' ability is different from others. Through our system easy way to access the questionaries’ sessions and be able to understand both sides.

The major issue is the accuracy of the outputs. When students asking questions through uploading videos should the system give more accurate results. Because they even cannot understand the English language. Our system is fully dependent on more accuracy. We consider the above requirements and came up with good accuracy of the results of each individual and integrated function. In the below scenario, we will discuss more.

3.1.3.1 Video pre processing

This is stage two of my research project. I need to remove background objects from the images which are stored in the database after the pre-processing. This very important stage because when we recognize hand gestures image contain only the hand gesture region then easy to detect the gestures.

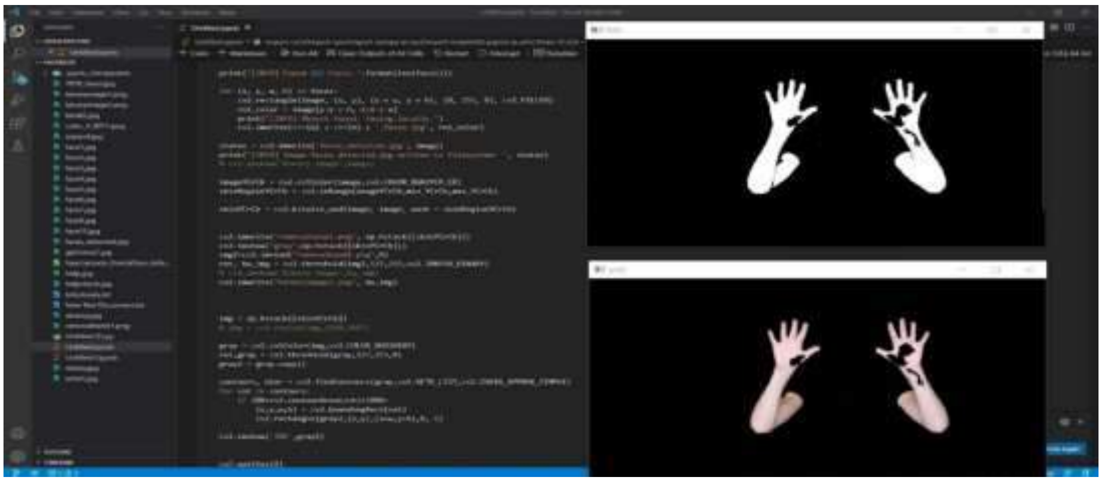


```
import cv2
import os

vidcap = cv2.VideoCapture('example.mp4')
try:
    if not os.path.exists('data1'):
        os.makedirs('data1')
except OSError:
    print ('Error: Creating directory of data')
def getFrame(sec):
    vidcap.set(cv2.CAP_PROP_POS_MSEC,sec*1000)
    hasFrames,image = vidcap.read()
    if hasFrames:
        name = './data1/frame' + str(count) + '.jpg'
        print ('Creating...' + name)
        image = cv2.resize(image,(200,200)) #resize the image
        cv2.imwrite(name, image)
#           cv2.imwrite("video_data/image"+str(count)+".jpg", image)     # save frame as JPG file
    return hasFrames
sec = 0
frameRate = 0.5 #//it will capture image in each 0.5 second
count=1
success = getFrame(sec)
while success:
    count = count + 1
    sec = sec + frameRate
    sec = round(sec, 2)
    success = getFrame(sec)
```

```
Creating..../data1/frame1.jpg
Creating..../data1/frame2.jpg
Creating..../data1/frame3.jpg
Creating..../data1/frame4.jpg
Creating..../data1/frame5.jpg
Creating..../data1/frame6.jpg
Creating..../data1/frame7.jpg
Creating..../data1/frame8.jpg
Creating..../data1/frame9.jpg
Creating..../data1/frame10.jpg
Creating..../data1/frame11.jpg
```

3.1.3.2 Removal background object and convert into binary form

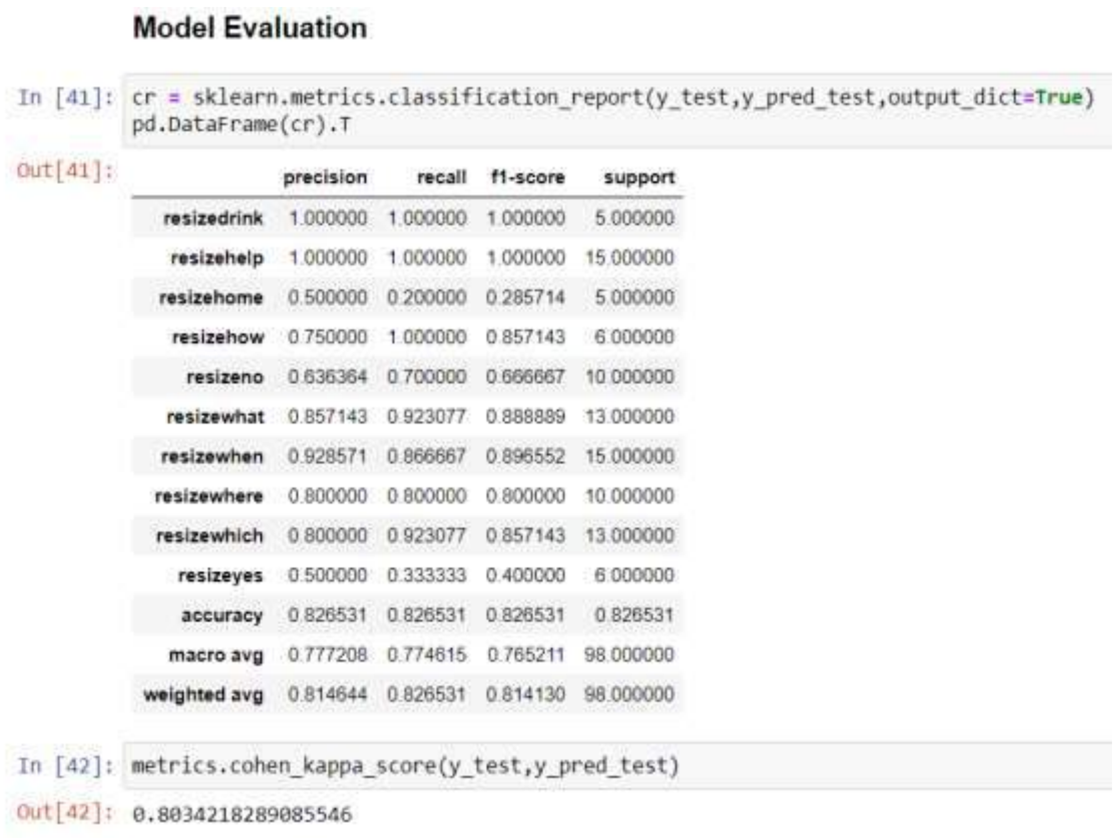
This is stage two of my research project. I need to remove background objects from the images which are stored in the database after the pre-processing. This very important stage because when we recognize hand gestures image contain only the hand gesture region then easy to detect the gestures.



```
[illegible]
```

3.1.3.3 Model Results

Finally I have tested this functionality with a machine learning model and Figure 16 shows the Cohen kappa score accuracy of 0.803 and it predicted 80% of images according to the relevant label and I have used this model for my function on the project.



Model Evaluation

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

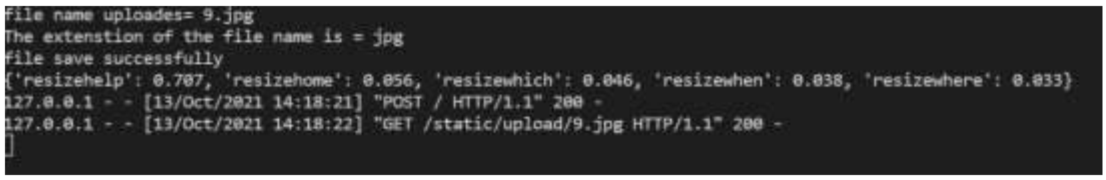
	precision	recall	f1-score	support
resizedrink	1.000000	1.000000	1.000000	5.000000
resizehelp	1.000000	1.000000	1.000000	15.000000
resizehome	0.500000	0.200000	0.285714	5.000000
resizehow	0.750000	1.000000	0.857143	6.000000
resizeno	0.636364	0.700000	0.666667	10.000000
resizewhat	0.857143	0.923077	0.888889	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
```

Figure 16 - Cohen kappa score accuracy

3.1.3.4 Rest API (Flask) Results

I have used flask for my back API creation. The below code shows the results of rest API for the sign language to convert into text.



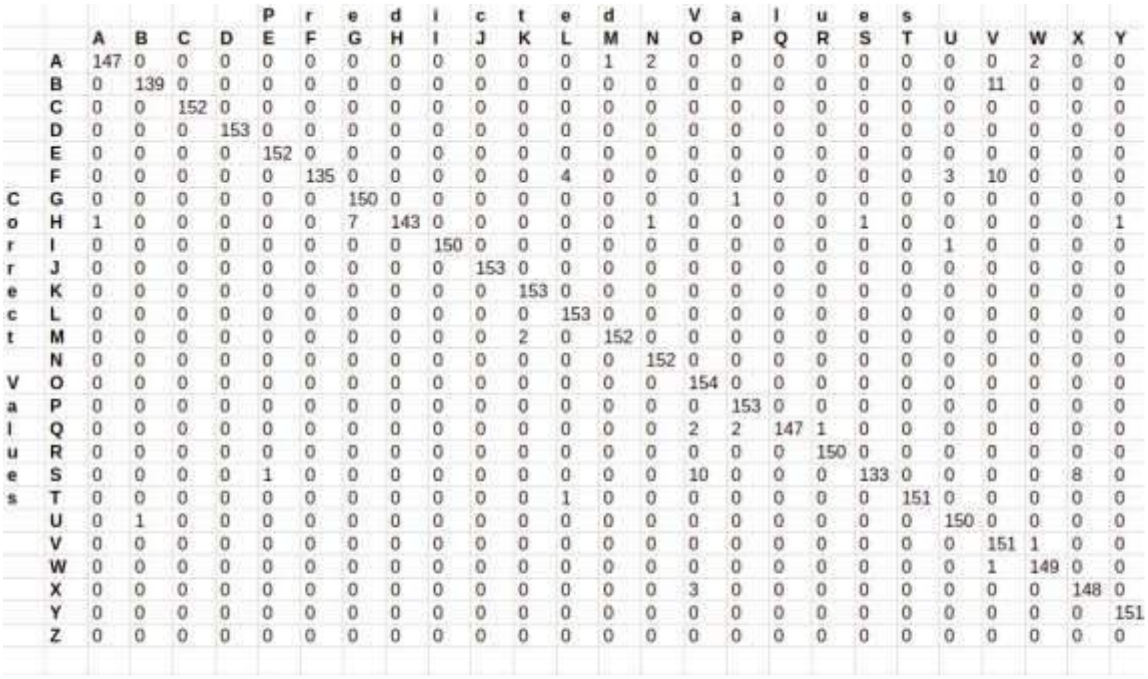
```
file name uploades= 9.jpg
The extenstion of the file name is = jpg
file save successfully
{'resizehelp': 0.707, 'resizehome': 0.056, 'resizewhich': 0.046, 'resizewhen': 0.038, 'resizewhere': 0.033}
127.0.0.1 - - [13/Oct/2021 14:18:21] "POST / HTTP/1.1" 200 -
127.0.0.1 - - [13/Oct/2021 14:18:22] "GET /static/upload/9.jpg HTTP/1.1" 200 -
]
```

3.1.4 Output for Teaching ASL

The motive of this research study was to provide a functionality that will enable the hearing impaired and general students to learn ASL from the online platforms on their own, without any difficulties or others' assistance. The evident from the test result proves that have reached our goal, here are some technical aspect to prove the function is at its acme.

3.1.4.1 Confusion Matrix Results

Below is the confusion matrices of ML model in Figure 17 **Error! Reference source not found.**where the true positives are given in the diagonal and have very few false positive values, this indicates the model less negative value and able to prediction sign without any failure.



						P	r	e	d	i	c	t	e	d		V	a	l	u	e	s					
		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
	A	147	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	2	0	0
	B	0	139	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0
	C	0	0	152	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	D	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	E	0	0	0	0	152	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	F	0	0	0	0	0	135	0	0	0	0	0	4	0	0	0	0	0	0	0	0	3	10	0	0	0
C	G	0	0	0	0	0	0	150	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
o	H	1	0	0	0	0	0	7	143	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1
r	I	0	0	0	0	0	0	0	0	150	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
r	J	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
e	K	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0	0
c	L	0	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0
t	M	0	0	0	0	0	0	0	0	0	0	2	0	152	0	0	0	0	0	0	0	0	0	0	0	0
	N	0	0	0	0	0	0	0	0	0	0	0	0	0	152	0	0	0	0	0	0	0	0	0	0	0
V	O	0	0	0	0	0	0	0	0	0	0	0	0	0	0	154	0	0	0	0	0	0	0	0	0	0
a	P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0
l	Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	147	1	0	0	0	0	0	0	0
u	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	150	0	0	0	0	0	0	0
e	S	0	0	0	0	1	0	0	0	0	0	0	0	0	0	10	0	0	0	133	0	0	0	0	8	0
s	T	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	151	0	0	0	0	0
	U	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	150	0	0	0	0
	V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	151	1	0	0
	W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	149	0	0
	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	148	0
	Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	151
	Z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 17 - Confusion Matrix

3.1.4.2 Model Test Results

The functionality is tested in different testing circumstance such as in complex background, bright light environment and dark light environment, and we got expected out from the system. Table 11 shows the result of test the functionality.

Table 11 - Illustration Test Result of ML Model

Test no.	Test Scenario	No. of test runs	Accuracy (%)
1	Low light background	15	84
2	Dark light environment	15	84
3	Complex environment	15	82

3.2 Research Findings

The ultimate goal of this research study was to provide a system that will enable the hearing impaired students to learn from the online platforms on their own, without any difficulties or others' assistance. It was achieved to almost a hundred percentage, which is evident from the test results. Hence, our developed system is found to consist of the following features.

1. **Reliable and trustworthy**

   As the translations of the captions are tested by many test cases, it was found they were very accurate, that the meaning of the sentences were not collapsed. Therefore, it will be useful for students to learn without any doubts in the conversion system.

2. **User-friendly interface**

   Since the development was done to minimize the difficulties faced during the online learning, it was made sure all the things in the application should be simple enough and must have a user-friendly interface.

3. **High speed**

   The conversions take place very rapidly, since the data has been already put into the system's database. The process is to find the matching words with the same labels of the videos. Although our system has the good features mentioned above, there is a con to it. The output that is generated from the system definitely relies on the vocabulary scale which is fed to the system database. Therefore, some words might be missed, but very rarely.

3.3 Discussion

This chapter discusses the results obtained from the development of the application. We have achieved an accuracy of 90% accuracy for the system. Most of the accuracy level is depends on the conversion between English grammar to ASL grammar since our system accurately converts the caption into ASL with the proof of test cases our system archived 90% of accuracy by developing the system.

When comparing the text to sign language functionality with other existing systems it is proofed that there is no existing LMS system for hearing impaired students is available by doing this research. Since there is no existing LMS for hearing impaired students out system with 90% accuracy is considered as best success rate by developing this system

3.4 Future Works

Even though we find the research completed, there is also expectations for updates in the system by adding more signs to the database with time, and enhancing the synonym finding feature in the system. We expect this would be more helpful and trustworthy to the users of the system. For output generation the data can be created with the same person and also this can be converted to animation output as well which will solve the issue for finding the dataset for some words.

As our system is fully developed based on American Sign Language it is possible to develop the system to support other sign language as well. This can help more target users to get maximum use out of this system. This kind of features can be developed in future to provide more advantage to the hearing impaired students.

4. CONCLUSION

In summary, this research work is about developing a LMS for hearing impaired students. Since the technology is growing very fast in every sector the education sector also rapidly moving into online based education system. When it comes to normal people it is very convenient for them to learn and interact with the tutors through the LMS specially during the pandemic situation but when it comes to hearing impaired students they are facing many difficulties while the education sector is migrating into online platform because of this problem we came up with a solution to solve the communication barrier between the tutor and hearing impaired students and the education barrier. Therefore, in this system as a first phase when the tutor uploads the video the system can do the video enhancement to provide a better experience to the users once that function is completed the caption of the video will be extracted and pass it as input for the function test-to-sign language conversion.

This system also solved the communication barrier while hearing impaired student ask a question from a normal tutor who doesn’t know ASL to answer the questions. This problem is solved by using machine learning technologies by converting the ASL to text format which can be understandable by the normal tutor. In the system the teach ASL part is developed using vision based approach to eliminate the cost of extra gadgets and device, so that the user can directly interact with the system using webcam and learn ASL with help of tutorials. Once he is start the test the system will detect the sign and notify the user. This can benefit both community of hearing impaired and ordinary user to learn ASL from elementary level.

The research has been successfully developed and tested that it accommodates the hearing disabled students to fulfill their requirements and eliminate struggles in shifting to the e-learning platform. The system flawlessly performs all the functionality with good success rate. Therefore, it is hoped that they benefit from this and more users will find this enlightening for their studies.

REFERENCES

[1] A.C. Davis and H. Hoffman, "Hearing loss: rising prevailance and impact," NCBI, 1 October 2019. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6796666/.

[2] 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.

[3] G. Nath and V. Anu, "Embedded sign language interpreter system for deaf and dumb people," in *2017 International Conference on Innovations in Information, Embedded and Communication Systems*, 2017.

[4] Jiangto, Wen and W. Li, "An Efficient and Integrated Algorithm for Video Enhancement in Challenging Lighting Conditions," *Computer Vision and Pattern Recognition,* 2011.

[5] G. Mittal, S. Locharam and S. sasi, "An Efficient Video Enhancement Method Using LA*B* Analysis," in *IEEE International Conference on Video and Signal Based Surveillance*, 2006.

[6] R. Ranchel, T.-D. Teresa, Y. Guo, K. Bein, H. Martin, J. P. Robinson and B. S. Duerstock, "Using speech recognition for real-time captioning and lecture transcription in the classroom," *IEEE Transactions on Learning Technologies,* vol. 6, no. 4, pp. 299-311, 2013.

[7] Z. Tmar, A. Othman and M. Jemni, "A rule-based approach for building an artificial English-ASL corpus," in *2013 International Conference on Electrical Engineering and Software Applications*, Hammamet, 2013.

[8] A. S. Ghotkar, K. Rucha , K. Sanjana , A. Surbhi and H. Mithila , "Hand gesture recognition for Indian Sign Language," in *International Conference on Computer Communication and Informatics*, Coimbatore, 2012.

[9] A. Sood and M. Anju , "AAWAAZ: A communication system for deaf and dumb," in *2016 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO),* Noida, 2016.

[10] S. Rathi and G. Ujwalla , "Development of full duplex intelligent communication system for deaf and dumb people," in *2017 7th International Conference on Cloud Computing, Data Science & Engineering - Confluence,* Noida, 2017.

[11] M. Kim, D. Park, D. K. Han and H. Ko, "A novel framework for extremely low-light video enhancement," in *IEEE International Conference on Consumer Electronics (ICCE)*, 2014.

[12] I. Maglogiannis, "A Benchmarking of IBM, Google and Wit Automatic Speech Recognition Systems," in *Artificial Intelligence Applications and Innovations.*, 2020.

[13] X. Hu, L. Tan, J. Zhou, S. Ali, Z. Yong, J. Liao and L. Liu, "Recognizing Chinese Sign Language Based on Deep Neural Network," in *IEEE*

International Conference on Systems, Man, and Cybernetics (SMC), Toronto, Canada, 2020.

[14] M.G.Grif, "Approach to the Sign Language Gesture Recognition Framework Based on HamNoSys Analysis," in *XIV International Scientific-Technical Conference on Actual Problems of Electronics Instrument Engineering (APEIE)*, Novosibirsk, Russia, 2018.

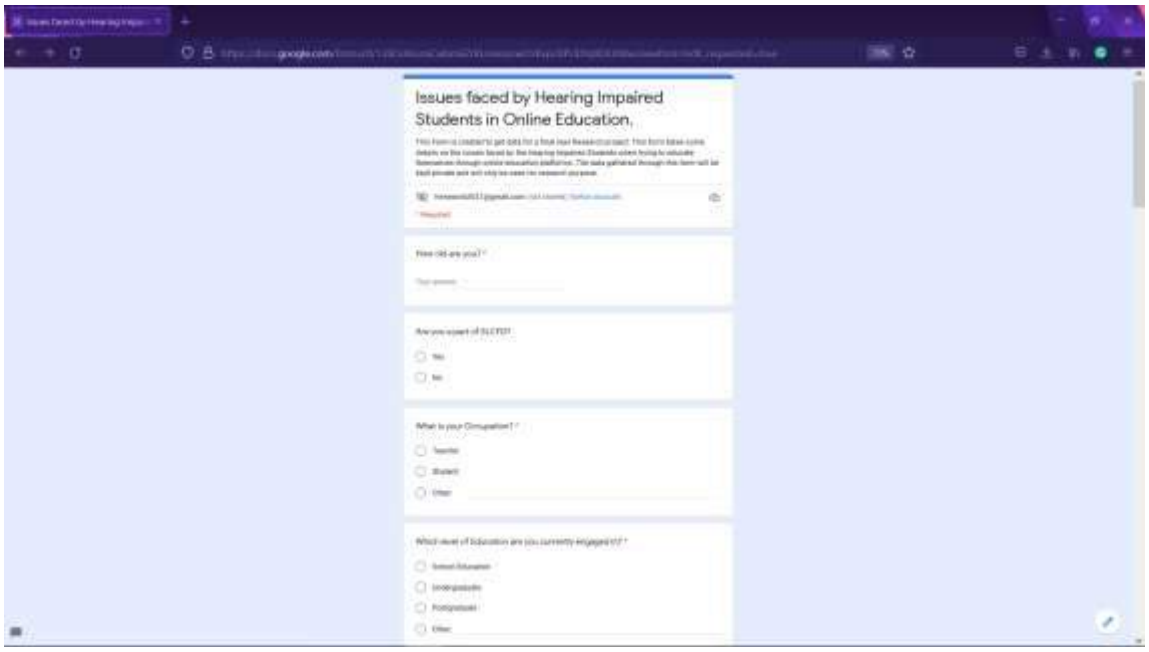
[15] D. M. Kumar, K. Bavanraj, S. Thavananthan, G. M. A. S. Bastiansz, S. M. B. Harshanath and J. Alosious, "EasyTalk: A Translator for Sri Lankan Sign Language using Machine Learning and Artificial Intelligence," in 2nd International Conference on Advancements in Computing (ICAC), Malabe, Sri Lanka, 2020.

[16] T. Jamil, "Design and Implementation of an Intelligent System to translate Arabic Text into Arabic Sign Language," September 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9255774.

[17] D. Kelly, J. Mc Donald and C. Markham, "Weakly Supervised Training of a Sign Language Recognition System Using Multiple Instance Learning Density Matrices," in IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 41, no. 2, pp. 526-541, April 2011, doi: 10.1109/TSMCB.2010.2065802.

APPENDICES

Survey form



Issues faced by Hearing Impaired Students in Online Education.

This form is created to get data for a final year Research project. This form takes some details on the issues faced by the Hearing Impaired Students when trying to educate themselves through online education platforms. The data gathered through this form will be kept private and will only be used for research purpose.

[illegible]@gmail.com (not shared) Switch account

* Required

How old are you? *

Your answer

Are you a part of SLCFD?

○ Yes

○ No

What is your Occupation? *

○ Teacher

○ Student

○ Other:

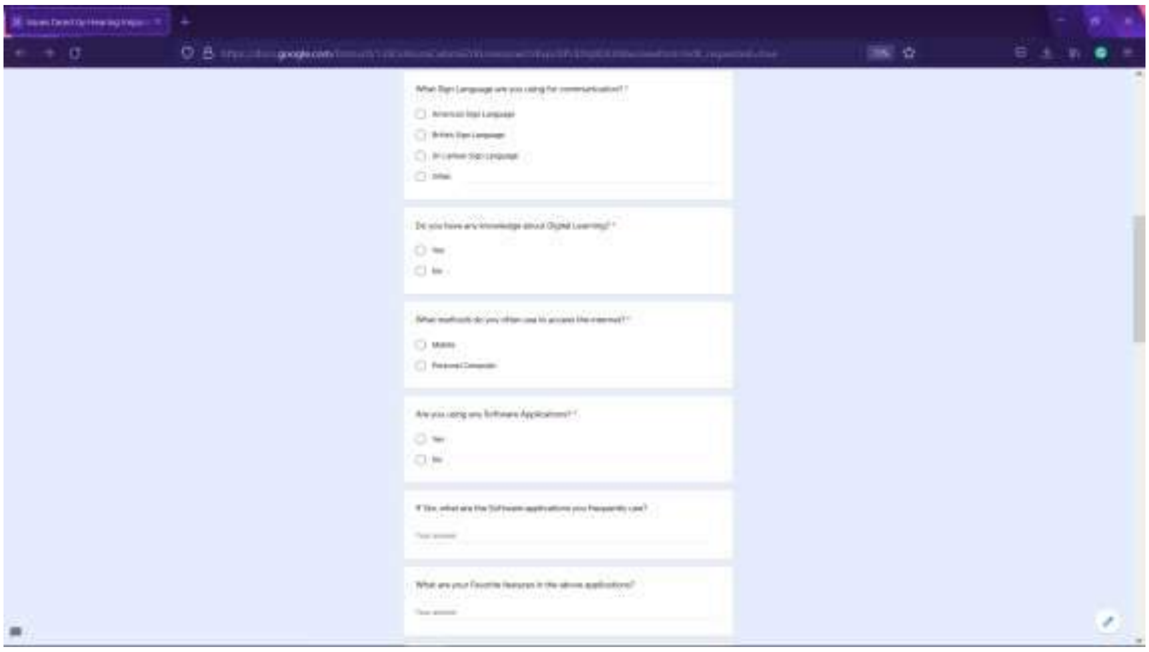
Which level of Education are you currently engaged in? *

○ School Education

○ Undergraduate

○ Postgraduate

○ Other:



What Sign Language are you using for communication? *

☐ American Sign Language

☐ British Sign Language

☐ Sri Lankan Sign Language

☐ Other:

Do you have any knowledge about Digital Learning? *

○ Yes

○ No

What methods do you often use to access the internet? *

○ Mobile

○ Personal Computer

Are you using any Software Applications? *

○ Yes

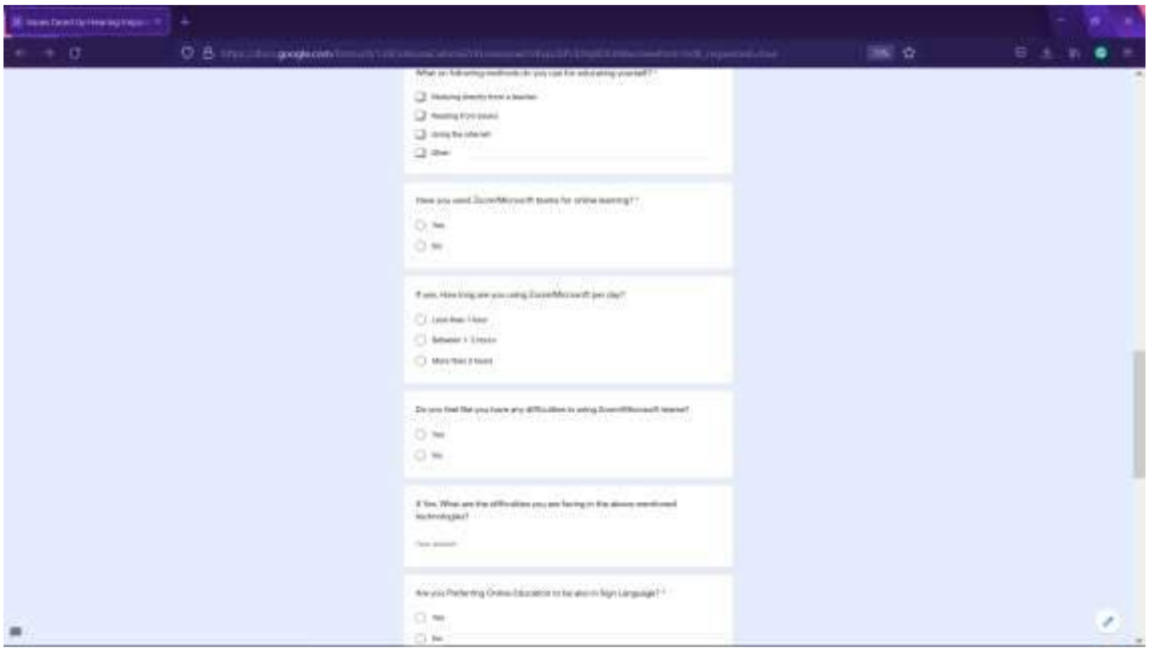
○ No

If Yes, what are the Software applications you frequently use?

Your answer

What are your Favorite features in the above applications?

Your answer



What are following methods do you use for educating yourself? *

- [ ] Studying directly from a teacher
- [ ] Reading from books
- [ ] Using the internet
- [ ] Other

Have you used Zoom/Microsoft teams for online learning? *

○ Yes
○ No

If yes, How long are you using Zoom/Microsoft per day?

○ Less than 1 hour
○ Between 1- 3 hours
○ More than 3 hours

Do you feel that you have any difficulties in using Zoom/Microsoft teams?

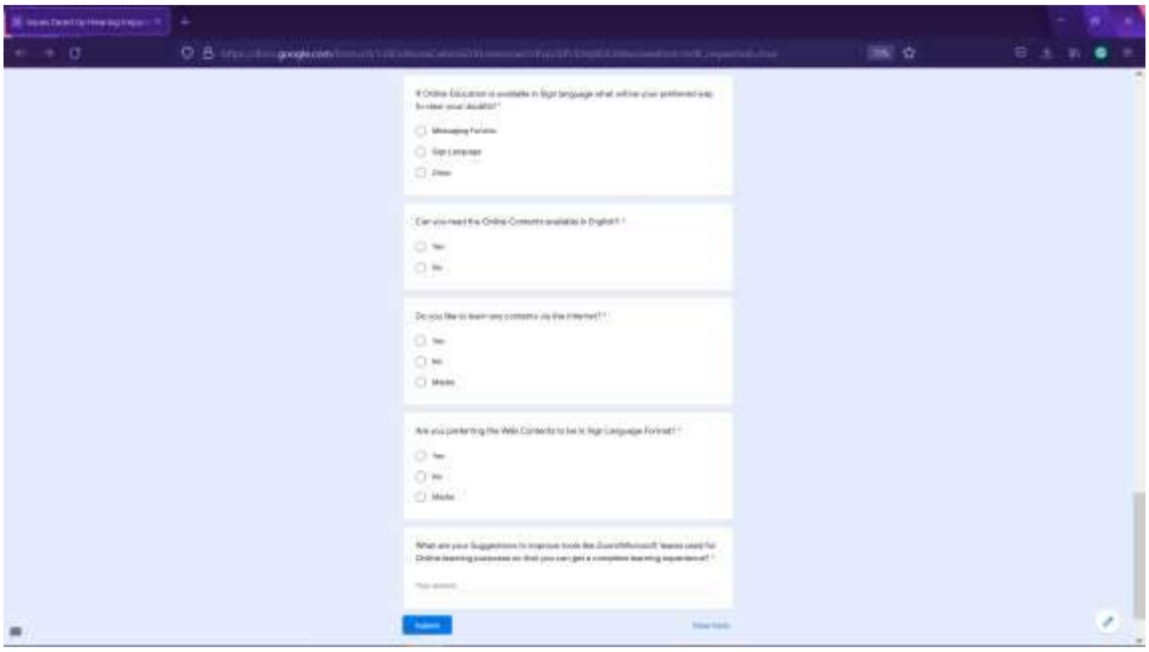
○ Yes
○ No

If Yes, What are the difficulties you are facing in the above mentioned technologies?

Your answer

Are you Preferring Online Education to be also in Sign Language? *

○ Yes
○ No



If Online Education is available in Sign language what will be your preferred way to clear your doubts? *

○ Messaging Forums
○ Sign Language
○ Other

Can you read the Online Contents available in English? *

○ Yes
○ No

Do you like to learn any contents via the internet? *

○ Yes
○ No
○ Maybe

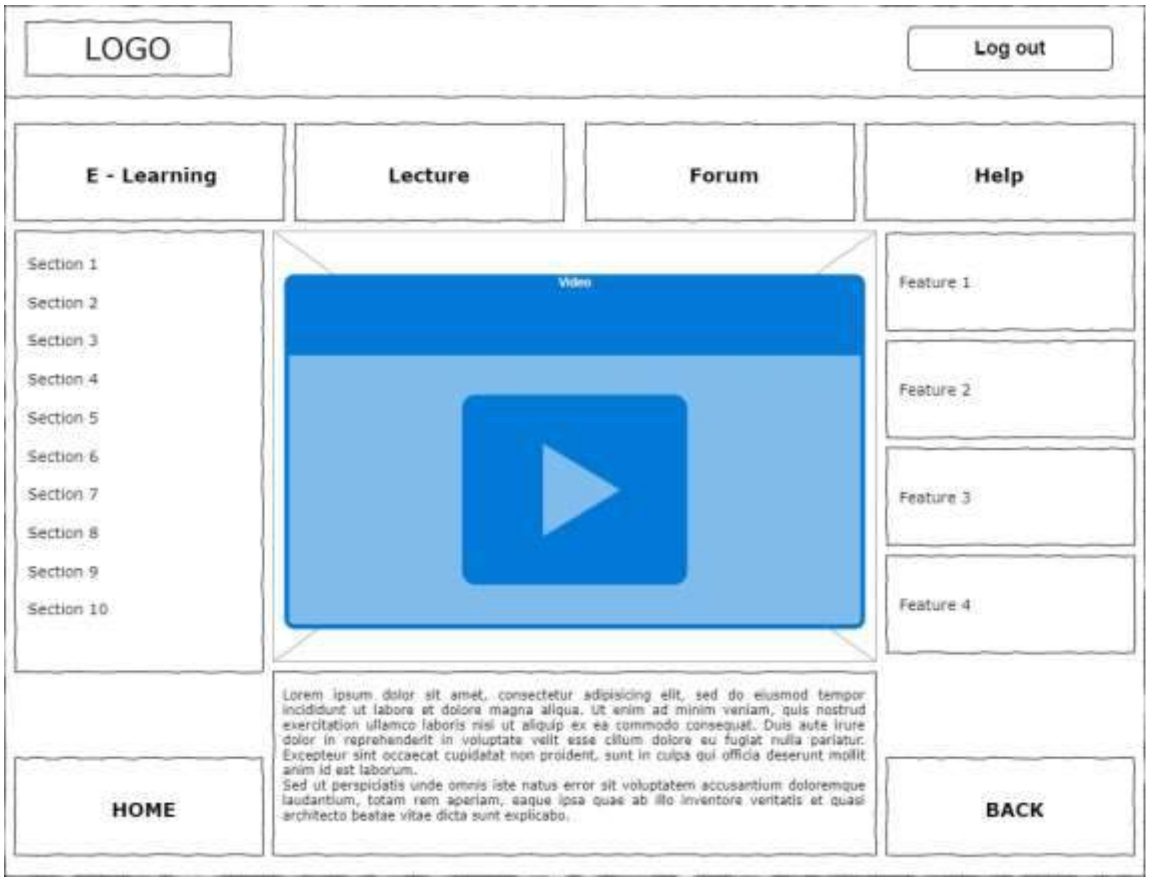
Are you preferring the Web Contents to be in Sign Language Format? *

○ Yes
○ No
○ Maybe

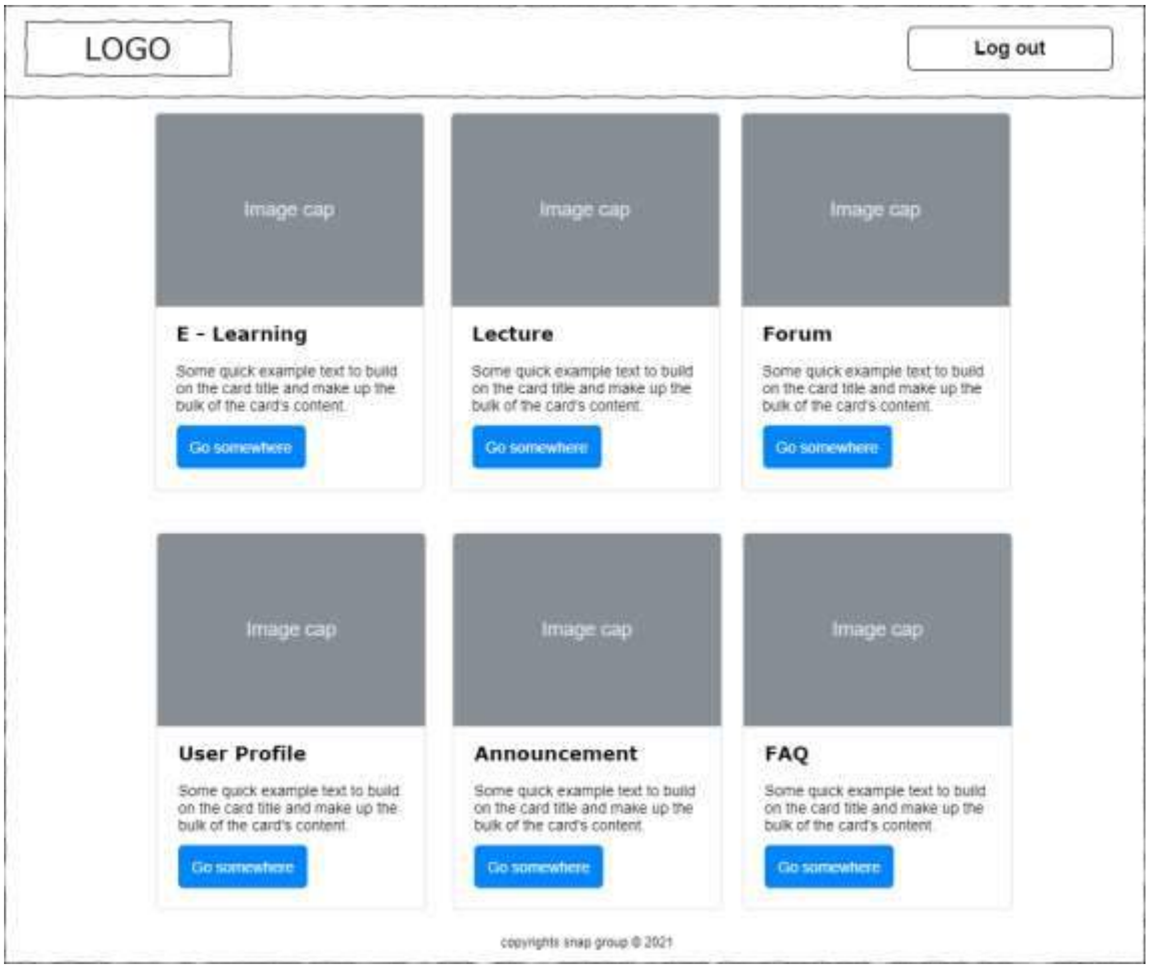
What are your Suggestions to improve tools like Zoom/Microsoft teams used for Online learning purposes so that you can get a complete learning experience? *

Your answer

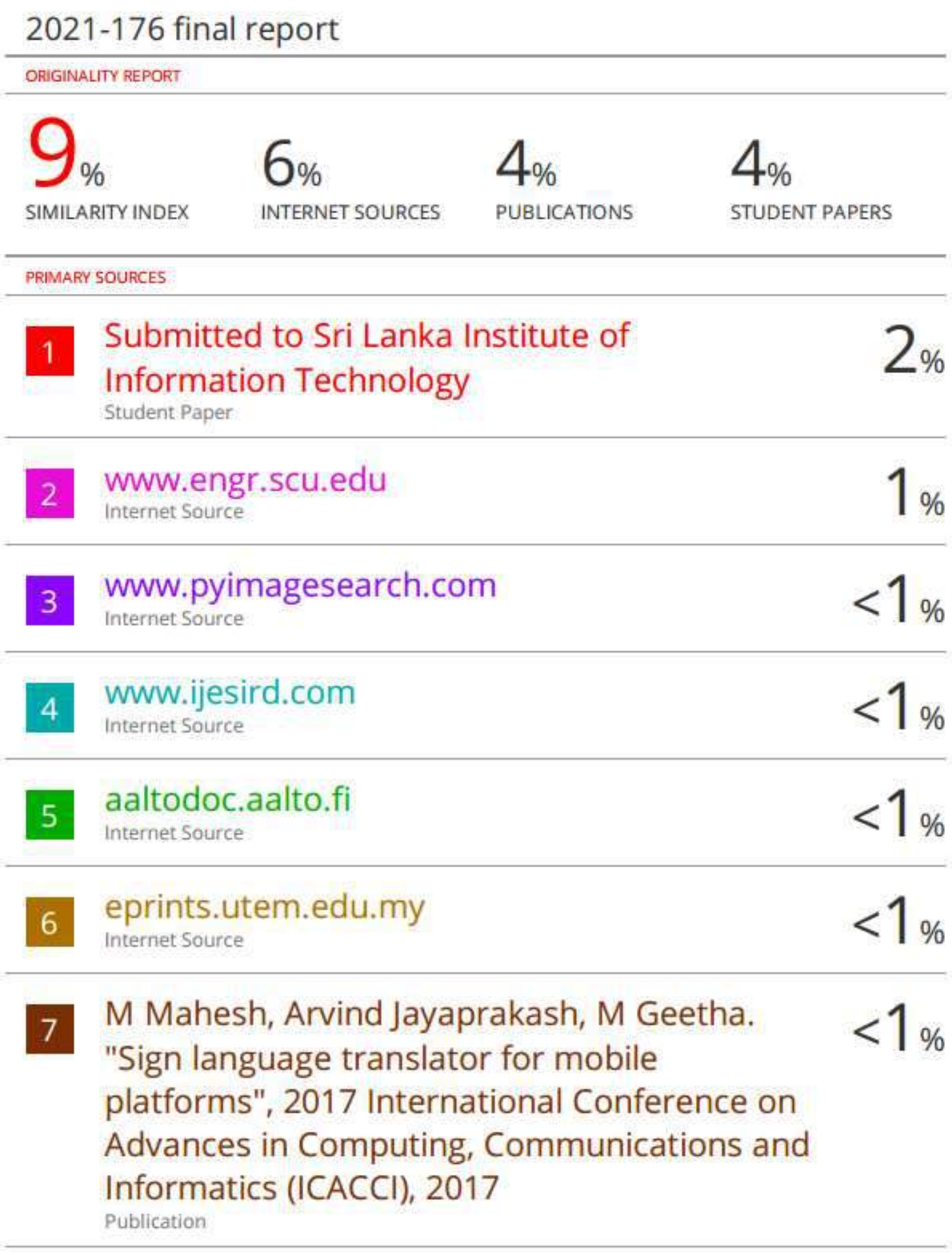
Wireframe



Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.


LOGO
Log out
Image cap
E - Learning
Some quick example text to build on the card title and make up the bulk of the card's content.
Go somewhere
Image cap
Lecture
Some quick example text to build on the card title and make up the bulk of the card's content.
Go somewhere
Image cap
Forum
Some quick example text to build on the card title and make up the bulk of the card's content.
Go somewhere
Image cap
User Profile
Some quick example text to build on the card title and make up the bulk of the card's content.
Go somewhere
Image cap
Announcement
Some quick example text to build on the card title and make up the bulk of the card's content.
Go somewhere
Image cap
FAQ
Some quick example text to build on the card title and make up the bulk of the card's content.
Go somewhere
copyrights snap group © 2021

Plagiarism Report



2021-176 final report

ORIGINALITY REPORT

9%	6%	4%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Sri Lanka Institute of Information Technology Student Paper	2%
2	www.engr.scu.edu Internet Source	1%
3	www.pyimagesearch.com Internet Source	<1%
4	www.ijesird.com Internet Source	<1%
5	aaltodoc.aalto.fi Internet Source	<1%
6	eprints.utem.edu.my Internet Source	<1%
7	M Mahesh, Arvind Jayaprakash, M Geetha. "Sign language translator for mobile platforms", 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2017 Publication	<1%