E-LEARNING SYSTEM FOR HEARING IMPAIRED STUDENTS

Niroshan Krishnamoorthy

(IT18144772)

Final (Draft) Report

B.Sc. (Hons) Degree in Information Technology Specializing in Software Engineering

Department of Computer Science and Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

September 2021

E-LEARNING SYSTEM FOR HEARING IMPAIRED STUDENTS

Niroshan Krishnamoorthy

(IT18144772)

Dissertation submitted in partial fulfillment of the requirements for the Bachelor of Science (Honors) in Information Technology

Department of Computer Science and Software Engineering

Sri Lanka Institute of Information Technology

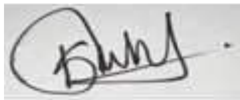
Sri Lanka

September 2021

Declaration

I declare that this is my own work, and this dissertation 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 my 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.

Also, I hereby grant to Sri Lanka Institute of Information Technology, the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or another medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Name	**Registration No.**	**Signature**
Niroshan Krishnamoorthy	IT18144772	

The above candidate is carrying out research for the undergraduate dissertation under my supervision.

------------------------------------- ------------------------

Signature of the supervisor: Date:

------------------------------------- ------------------------

Signature of the co-supervisor: Date:

Abstract

With the Spread of Global Pandemic Covid-19 the Learning was transformed to Online learning from traditional learning, Therefore the use of eLearning platforms was increased. But this had issues with certain communities like hearing impaired community around the world. The hearing-impaired people had many issues with eLearning platforms because of their deficiency in hearing sound. Therefore, we are proposing a platform through which hearing impaired people effectively involve in learning. The proposing system will use sign language in addressing the students. The proposing eLearning platform mainly uses recorded videos to be uploaded to the system by the tutor. If the video is not with visually clear content, then the hearing-impaired student may face difficulties when following the video therefore, we have proposed a low light detection technique which can identify low light frames of a video and enhance them. When converting a speech to sign language we need to first convert it to text. For this purpose, we have proposed an algorithm which uses the robust google speech recognition engine to produce text from speech. This text will then reproduce into captions using our proposed algorithm. The proposed low light enhancement section of the system was tested with some test cases to generate the required output with a considerable enhancement by identifying majority of the low light parts of the videos and images. Also, caption section of the system was tested with some test cases for the captioning module in this study.

Keywords: Video Processing, Image Processing, Speech Recognition, Low light enhancement

Acknowledgement

I would like to extend my sincere gratitude to my project supervisor Dr. Kalpani Manathunga, for her vital guidance and the invaluable knowledge she shared throughout the year to make the research successful. The motivation provided by her was crucial for me to complete the different phases of the study successfully. I also would like to thank my co-supervisor Ms. Samanthi E.R Siriwardana, for her support to complete the research. During the most intense and difficult periods of the project, they advised me and my team to stay motivated and complete the project successfully. I would also like to express my gratitude to Sri Lanka Institute of Information Technology for providing us with the expertise to finish the project despite the global pandemic. I would also like to thank the Lecturers-in-charge, Dr. Janaka Wijekoon and Mr. Jagath Wikramarathne for their guidance at some phases of the project. I also sincerely thank all the project group members, who worked with me and helped me to complete this research successfully.

Niroshan Krishnamoorthy

Department of Computer Science and Software Engineering,

Sri Lanka Institute of Information Technology.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

WHO	World Health Organization
RGB	Red, Green and Blue
GCP	Google Cloud Platform
STT	Speech to Text
FPS	Frames Per Second
UI	User Interface
HCI	Human Computer Interaction

LIST OF EQUATIONS

1. INTRODUCTION

1.1 Background

In the year 2020 the World encountered a Global Pandemic with the spread COVID-19 virus. This pandemic situation transformed many of the industries to Online with the use of Internet. This new transform 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 groups of people encountered lots of difficulties compared to the traditional in-class learning. One such group of people was 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.

According to the statistics proposed by J.Elflein in [2] South Asian region had the maximum number of hearing impaired population in 2018 which is 131 million. It is expected to increase to 267 at the end of the year 2050. And also, the hearing-impaired population on different regions on the globe is statistically displayed in the Fig 1.

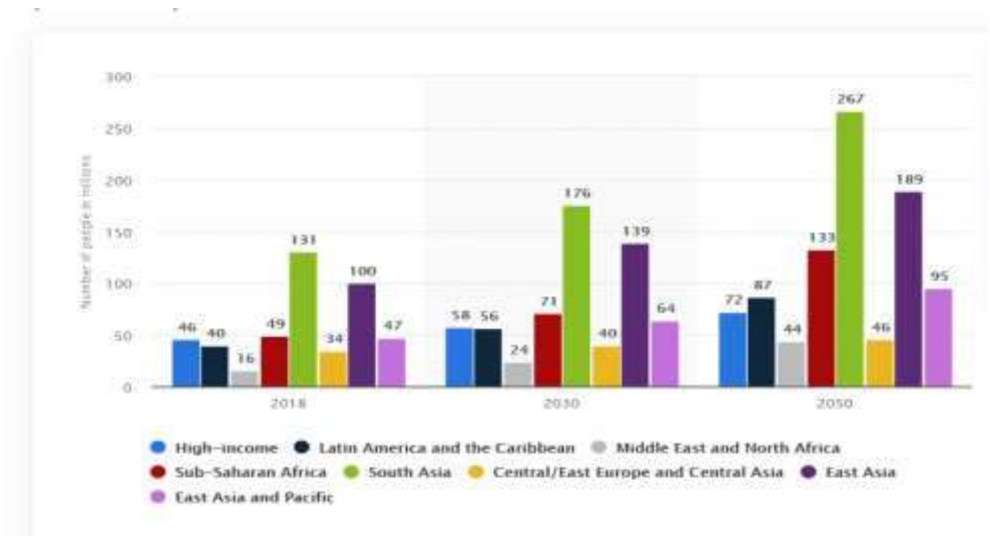


Figure 1: Number of people with hearing loss world wide

According to the Figure 1 we can clearly say that the population of hearing-impaired people is a significant amount when compared to the global population. Therefore, it is very important to find solutions for the problems faced by this community.

The shift towards eLearning has increased quite significantly than the traditional learning methods among the students. According to I. Moriera in [3] he has showed that there is a general shift towards online learning among many students. The Figure 2 shows some reasons why students find eLearning more effective than the traditional learning.

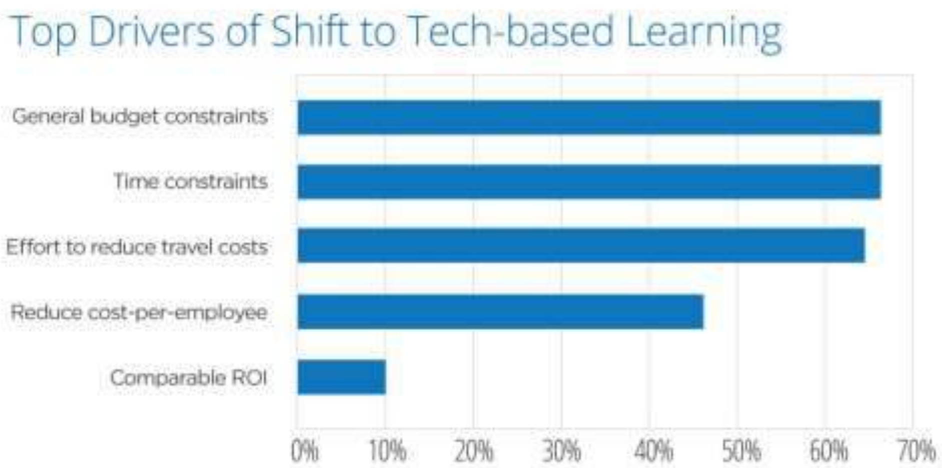


Figure 2: Top Drivers for the shift towards eLearning

Source: Novations Group

Fig 2. shows us the main Drivers to shift towards eLearning systems and the percentage rates for why students have chosen the eLearning. We can see that more than 60% of students have chosen the General Budget constraints, Time constraints and Effort to reduce travel cost as their key Drivers to shift towards eLearning. These outcomes show us that the eLearning systems have solved many problems that was available in the traditional base learning. Along with these drivers the global pandemic of 2020 was also a major reason for the whole world to shift towards online base learning.

According to [4], The Authors have conducted a study on the Impact of eLearning within the students of two universities in Saudi Arabia. The Authors have used Success/Impact measurement techniques for measuring the results from the study. The analysis of the results shows eLearning systems have increased the ability to interpret information accurately. And also, eLearning systems have increased the students understanding of the area of study. Therefore, the overall study indicates that the use of eLearning has a positive impact on students.

With all these advantages eLearning systems also comes with some major drawbacks which have been addressed by the people around the world. According to the article [5] the study has found out some challenges related to eLearning system as show in Figure 3.

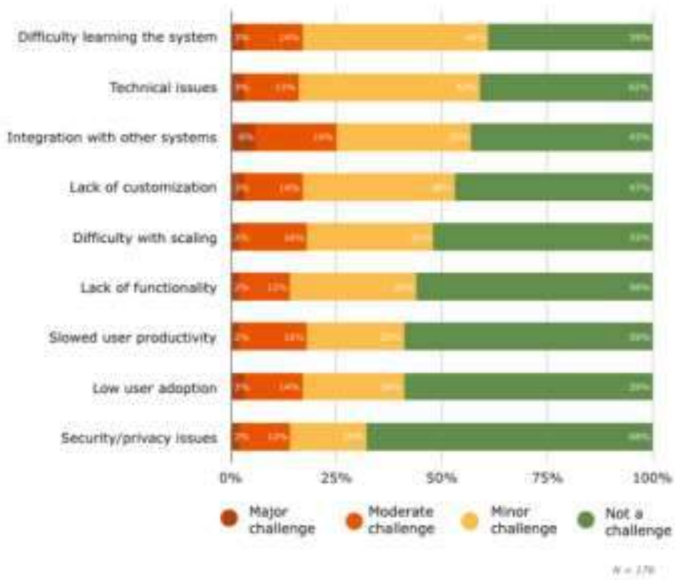


Figure 3: Biggest eLearning challenges

According to the Fig 3. the study shows us that slowed user productivity, Low user adoption and Difficult to learning the system are some common challenges faced by many groups of students around the world.

With all these disadvantages the hearing-impaired students face a major issue with accessibility of these type of eLearning systems [6]. Authors also state that even though MOOC platforms contain audio and visual content in the eLearning environment these platforms need additional features in terms of accessibility requirements for visual and hearing impairment students.

Scholars suggest that eLearning stakeholders should support efforts to develop interactive learning systems through instructional interfaces for users who experience the limitation of hearing that prevents them from "receiving sounds through the ear" [7].

With all these research that has been done in the past it is vital to understand that the use of a proper eLearning system that supports hearing-impaired students to continue and improve their learning skills.

Based on the above findings it is clear that there is a need for an eLearning system that can solve the problems faced by the hearing-impaired community. To Add on to this statement we conducted a survey which gathered some information on how a proper eLearning system can be helpful for the hearing-impaired community.

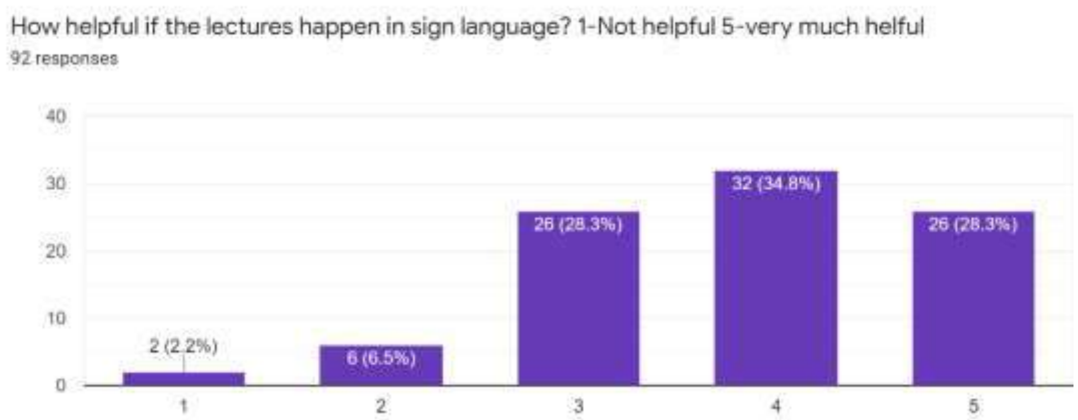


Figure 4: Lecture in Sign Language Response

The Fig 4. shows the results of how helpful it will be if the lectures happen in sign language. According to the results most of the responses depict that it's better when lectures happen in sign language. The reason for getting such a result is because many people or students had faced problems because they couldn't properly understand what is being taught or when tutors can't convey what they are trying to teach due to the conversational gap that is available between the hearing-impaired community and the tutors. Therefore, it is clear that we need a proper eLearning system that can fill the gap between hearing-impaired community and others.

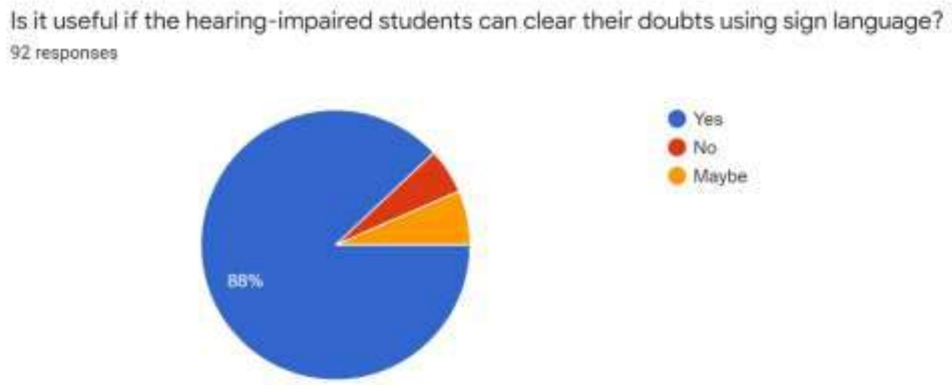


Figure 5: Clearing Doubts through sign language response

In our survey we also got the results on how hearing-impaired students can be affected when they are trying to clear their doubts. According to the Fig 5. More than 85% of the responses state that it will be helpful if we can provide a way for hearing impaired students to clear their doubts using sign language. This was because the existing eLearning systems do not have a way for hearing-impaired students to communicate with the tutors.

Based on the survey that is conducted we have clearly identified where the hearing-impaired community faces their 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 when learning.

1.2 Literature Review

For problems faced by hearing impaired people researches have proposed many solutions. But most of them are to reduce the communication gap between them. In [8] Nath, et al 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 very successful for conveying messages from deaf people to others. The Authors have also stated that usage of these solution in various application where the hearing-impaired people can get maximum productivity throughout the day. According to the authors they are planning to use this solution in application like gesture-controlled robot, gesture-controlled doors and vehicles, gesture-controlled keyboard and mouse to interact with the computer and gesture-controlled appliances. Many solutions like this have been introduced to the hearing-impaired community but most of them are to act as an interpreter of sign-language to speaking languages.

Researchers have also proposed some eLearning solutions for students with different visual and hearing needs. According to [9] Farhan, et al have discussed about using the human computer interaction (HCI) 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 as well as for the students without any defects. 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. Both students and teachers recognized that the proposed system interface had valuable features of interactional communication to support all students, regardless of visual or hearing ability. The authors also have suggested on making this system more robust in their future work.

Since there was a requirement to add low light enhancement techniques to the uploaded videos, we needed a mechanism to fulfill this functionality. For this several enhancement techniques were handled by many researches.

According to [10] Jiangto, et al 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. To increase the speed of the algorithm temporal correlations between the subsequent frames are utilized. Simulations results of this algorithm shows good enhancement results when compared to other frame wise enhancement algorithms. The Authors also state that the algorithm can be improved for better pre-processing of the filters in the future.

In [11] Lee, et al have proposed a video contrast adjusting technique using Spatio-Temporal histogram specification method. This algorithm uses depth guidance to identify the background and foreground of an image. The experimental results have showed that the algorithm has enhanced the salient foreground objects efficiently and preserving background details faithfully. The Authors state that the algorithms can efficiently enhance depth segments of color images. Since there is a lack of depth data the authors have planned to conduct future research on adaptively enhancing the contrast of visually important objects without requiring depth data.

According to [12] Mittal, et al 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 approach was based on histogram manipulation La*b* color model of video frame. 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. The Authors have not mentioned anything about the effectiveness of the algorithm in the normal light when compared to the low light.

Kim, et al of [13] have found a new approach for noise reduction and enhancement of extremely low-light video. They have used motion adaptive temporal filtering based Kalman structure updating. They have used Spatio-temporal filtering for the noise reduction of extremely low light videos. After filtering out, some noise will be remaining those noise level will be raised by tone mapping and at final stage Spatial Noise reduction will be used to remove uplifted noise leaving us with a quality image.

This algorithm was very effective in moving parts of the image under low light. The Authors have expected to use this algorithm in various software systems in the future. The Systems that are planning to use this system in the future are consumer CCTV cameras, black box camera for vehicles and video signal-based surveillance systems. The proposed system is said to be highly promising in the low light conditions.

In [14] Kanchana, et al have proposed a video enhancement framework consisting of bilateral tone adjustment and Saliency-Weighted Contrast enhancement. This algorithm will enhance the parts of an image where humans give more attention to. This algorithm has given efficient results in removing noise from an image. This algorithm performs efficiently in regions where humans give more attention. As a future work this algorithm can be used in enhancing the quality of the low-grade video surveillance cameras. The author is expecting to use this algorithm for license plate detection and surveillance cameras.

Since there was a requirement to extract speech from text, we looked into some captioning techniques used in the eLearning systems. According to [15] Ranchel, et al discusses about two scientific speech recognition technique experiments. They are using real time captioning using IBM Via Scribe 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. The study tells us that the Real-time captioning and Post lecture transcriptions has improved the students note taking ability. The future improvements of this study are to check how students with disabilities utilize these technologies. Most of these systems use manual captioning techniques which require additional effort.

1.3 Research Gap

There are many researchers working in making solutions for hearing-impaired community. Multiple researches have been conducted on providing eLearning solutions for people with different disabilities. Most of the systems developed had additional overheads when using the system.

When looking into the past research works, we identified some factors which were not considered by the scholars who did research on eLearning systems. These factors can benefit the hearing-impaired students as well as other students and tutors who use the eLearning system. One such factor is, there was no video enhancement capabilities introduced to the system when a tutor uploads a video. The system proposed by [9] Farham, et al allows the tutor to upload the video. This video was then directly viewed by the hearing-impaired students. If the video had some missing details, then the hearing-impaired student will find it difficult to follow along with the video because the only way a hearing-impaired student communicate is with visual communication. Therefore, if the lecturer had to prepare the video to provide visually clear content, then he has to edit the video with a 3rd party editing software. This approach needs additional effort by the lecturer. If the lecturer didn’t have prior knowledge in video editing, he may need to hire a professional to do the job. Through are system we are proposing a technique where the system will automatically enhance low light hidden sections of the uploaded video.

The next factor considered was based on the automatic low light enhancement. If we are to automate the low light enhancement the system should be capable of identifying the low light videos separately with the normal light videos. The algorithm [10] proposed by Jiangto, et al was effective in enhancing videos that are recorded in low light videos. But if this algorithm was applied to the normal light videos, then it would produce bright outputs which can be unpleasant. Fig 6 shows the effect of enhancement algorithm on normal light videos. Therefore, we wanted to propose a technique which can identify the low light videos separately with the normal light videos and enhance them.

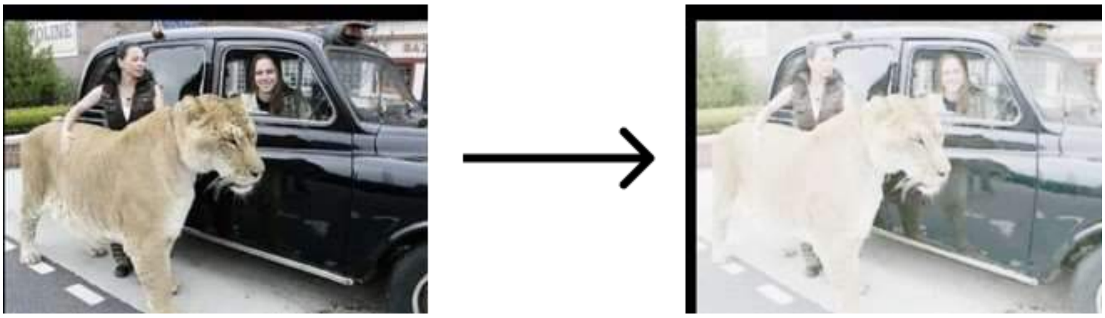


Figure 6: Applying enhancement algorithm on normal light video

The algorithms [10], [14] proposed by the authors are effective in enhancing low light videos. But they are mainly designed for platform specific tasks. Kanchana, et al have mentioned that bilateral tone adjustment algorithm [14], can be used effectively in security cameras and other specific devices. Therefore, when developing a low light enhancement algorithm, We wanted to develop an algorithm which can run in any web server since the eLearning application was going to be a web application.

When developing an eLearning system for hearing impaired students we wanted to extract speech from the uploaded lecture videos so that it can be used in sign language interpretation functions. Captioning lessons have been one important functionality in eLearning system. Ranchel, et al in their work [15] have proposed some techniques which requires additional captioning service for the videos. This again was an additional effort for the tutor. Tutor will need to write captions for the videos and attach them along with the videos so that the system can use them for captioning services. The proposal [16] of Julius T. for Disability aware eLearning system states the use of captioning lessons using a captioner for eLearning systems. This proposal mainly talks about a human captioner. Since there were no efforts taken to develop automated captioning system in eLearning systems, we wanted use automated captioning techniques for the uploaded videos so that the tutor will not have to make additional efforts to create captions for the videos.

So, I am introducing a system using special algorithms to enhance the video quality of the uploaded lecture videos and providing system generated captions for the uploaded videos.

Comparison between the existing solutions and the proposed solution

Similarities:

- Provide video lecture features
- Provide Sign Language Interpretation

Differences:

Table 1: Functionality Comparison with available systems

System	Automated low light enhancement for the uploaded tutor videos	System Generated Captions for the tutor videos
System proposed by W. Farhan and J.Razmak [9]	X	X
System proposed by R.Ranchel, Teresa, Y. Guo and K. Bain [6]	X	X
Our System	✓	✓

According to the literature review above, the following research gaps have been identified.

- No eLearning System has automated video enhancement ability in them. All the available systems need a human effort to edit and enhance the video using a 3rd party video editing software.
- When low-light enhancement algorithms are used in brighter images the contrast of the image increases and provides a very bright image which disturbs the eye. Therefore, we need to identify at which part of the video we need to use low-light enhancement techniques.
- All the available systems are producing captions or transcriptions that is written by a human and that has to be separately updated to the eLearning system for a lecture video.

1.4 Research Problem

Hearing-Impaired students face difficulties when learning through an eLearning system. Understanding their main problems and designing a system as a solution for their problems is vital. To understand their problems, we conducted survey. Based on the survey that is conducted by our team I identified two important features that is missing and not addressed in the existing eLearning systems.

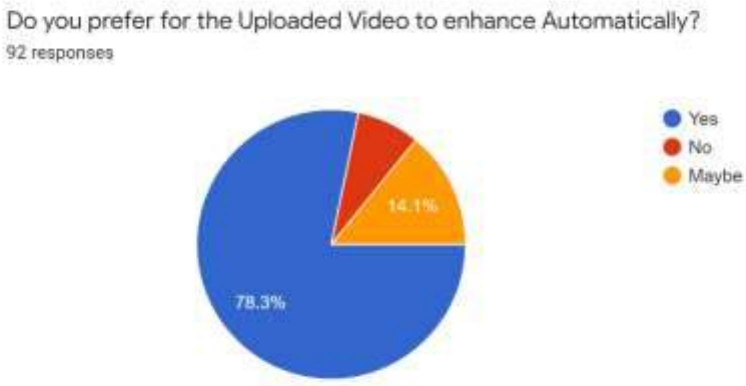


Figure 7: Video Enhancement Response

According to the Fig 7 we identified 78% of the responses preferred to have automated video enhancement feature in a system. This is mainly because they have faced a problem of low-quality video in eLearning systems or rather, they need to use additional tools to enhance the video if they are uploading a video to the online learning system. Therefore, this survey result suggests that video enhancement feature will be helpful for students when watching the lectures or it will be helpful for the tutors when uploading the video to the system since it is being enhanced automatically. Enhancing the videos also helps for hearing impaired students actively participate in the learning process without any issues.

As mentioned in the Research gap (section 1.3) eLearning systems doesn't have a video enhancement features within them. This causes the tutors to have an additional skill to edit the videos that are uploaded to the system. This was time consuming and requires additional work. To solve this, I wanted to introduced an algorithm which automates the low light video enhancement.

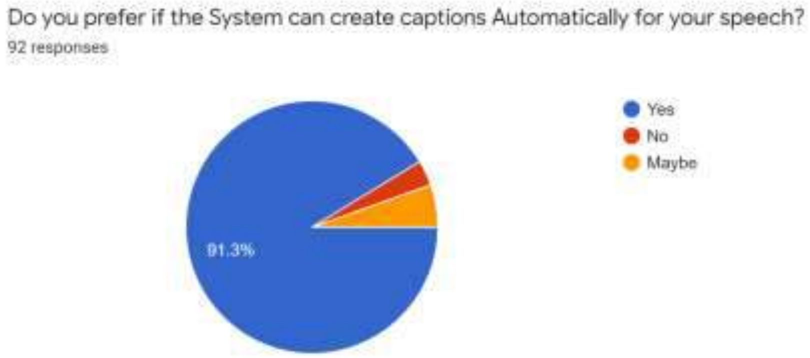


Figure 8: Caption generation Response

Based on the results on Fig 8, 91% of the responses preferred to have an automated captioning feature, this showed us people preferred automated captioning because it required additional human effort to create captions for a content. As discussed earlier, when developing an eLearning system for hearing-impaired students captioning is important. When we are trying to convert a speech to sign language the machine needs to understand the speech first and relate the speech to sign language. For this purpose, the text act as the intermediate medium. It is much easier to interpret sign language from text rather than interpreting sign language from voice [8]. Therefore, captioning section in this system is a crucial function.

Once the videos are uploaded to the systems the tutors need to write captions or transcripts on the system by their own or use another human to do the job. Therefore, I wanted introduced an algorithm that can achieve this process without any human efforts and provide captions in real time.

"How to Automate the Low light enhancement and Captioning processes in real time once the video is uploaded to the system?"

2. OBJECTIVES

The LMS systems we proposed is trying to reduce the communication gap between hearing-impaired students and tutors in eLearning system. We proposed a solution in which the uploaded lecture videos can be converted to sign language. We have also implemented a feature on the system on which hearing-impaired students can clear their doubts through sign language, this increases the engagement of the hearing-impaired students in learning through online.

2.1 Main Objective

The main objective of this component is to automate the process of identifying low light parts on a video and enhancing it. And also, once the video is enhanced some of the noise in this video will be removed using some noise filtering techniques. This component also carries the functionality of automating the caption generation for the video.

2.2 Specific Objectives

<u>Video Enhancement</u>

- Using a proposed algorithm to identify low light videos.
- Enhance the low light videos and reduce the noise in them.

<u>Automated Captioning</u>

- Extracting Audio from the video content.
- Identify the voice in the audio content.
- Convert the Speech to Text using Speech-to-Text Model.
- Adding time stamps to the output using a proposed algorithm.

3. METHODOLOGY

The main functionality of this component is to Enhance the video that is uploaded by the lecturer. Once the video is uploaded the low light parts of the video will be identified automatically and enhanced using low light enhancement and noise filtering algorithm. Then the enhanced video is used extract speech out of it. The extracted speech is then generated into text and captions will be produced.

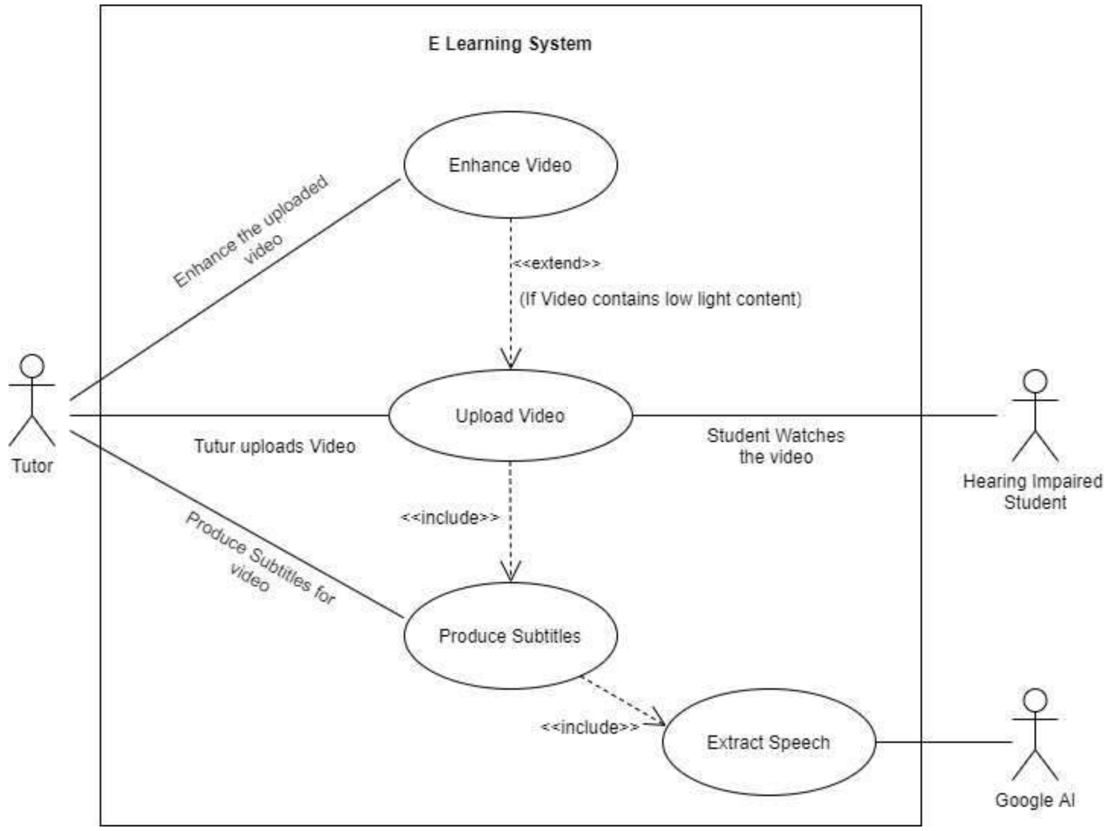


Figure 9: Use case diagram

3.1 System Overview Diagram

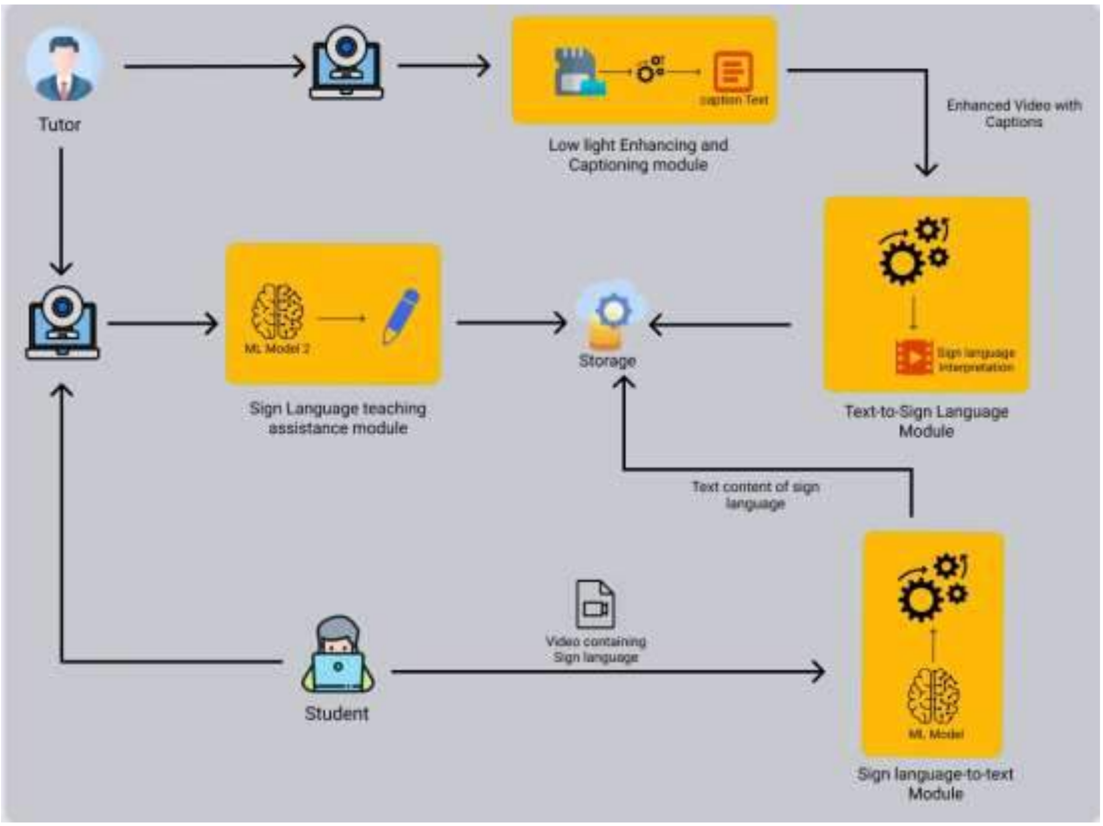


Figure 10: System Overview diagram

Fig 10 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.

3.2 Function Overview Diagram

The following diagram shows the function overview of low light enhancing and captioning module of Fig 10.

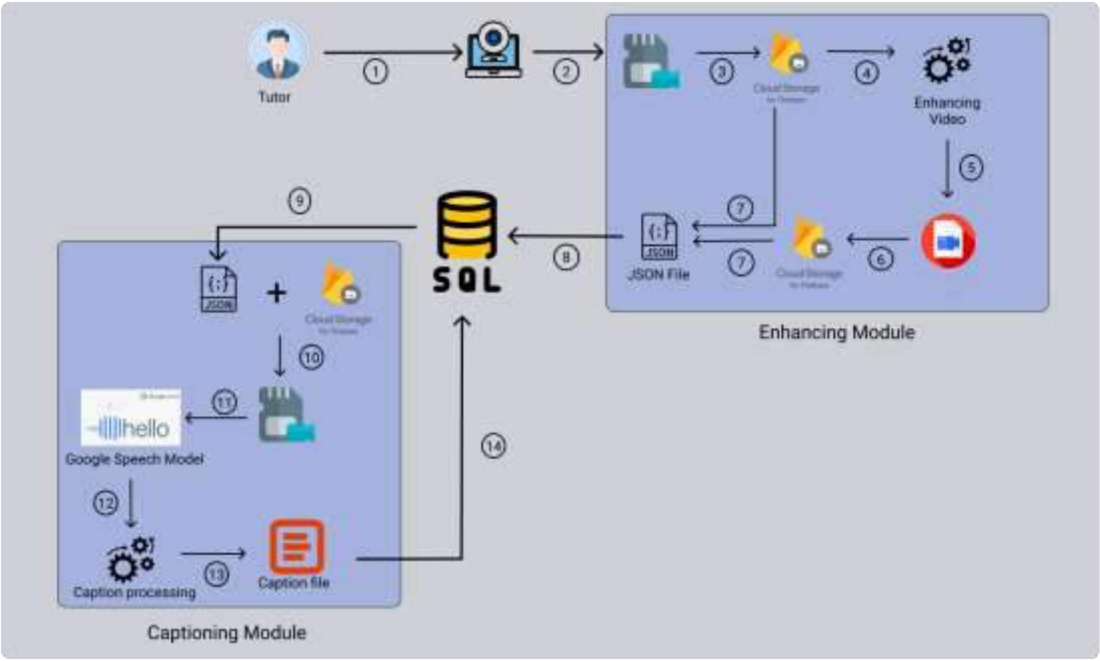


Figure 11: Low light enhancement and captioning function overview

According to Fig 11. The low light enhancing and captioning function is divided into two modules Enhancing module and Captioning Module. For easy understanding of the flow of functionality each step is numbered.

Step 1: Tutor Uploads the Recorded (Webcam) video to the system.

Step 2: The Uploaded video is sent to the Enhancement Module in the backend server.

Step 3: The Backend server sends the module to Firebase Storage to store it.

Step 4: The Video is taken back from the Storage and enhanced using the proposed low light enhancement technique.

Step 5 & Step 6: The enhanced video is then stored back in the firebase. (Now the firebase has both the enhanced and the original video.)

Step 7: The original and the enhanced video URL is retrieved from the firebase and stored in a JSON file.

Step 8: This JSON file is converted into a record and stored in a SQL database.

Step 9: The record from the SQL is retrieved upon the user request and the firebase URL of the original video of that particular record is obtained which is then sent to the captioning module.

Step 10: Using the obtained firebase URL the video is retrieved.

Step 11: Retrieved video is then sent into Google Speech to text module and text output is produced for the speech in the video.

Step 12: Text output is sent to proposed captioning module to produce captions with timestamps.

Step 13: Using the captions a caption file is generated which used in the text to sign language module.

Step 14: The captioned text is then updated to the database.

The next sections of this report will explain how the automated low light identification, low light enhancement and Captioning functionalities are developed.

3.3 Enhancement Module.

Once the tutor uploads the video to the system the video enhancement module does the preprocessing to identify low light parts separately from normal light parts in the uploaded video. Once the low light parts are identified then it is enhanced using low light enhancement techniques. To identify the low light frames in videos, the enhancement module uses intensity histograms in the preprocessing.

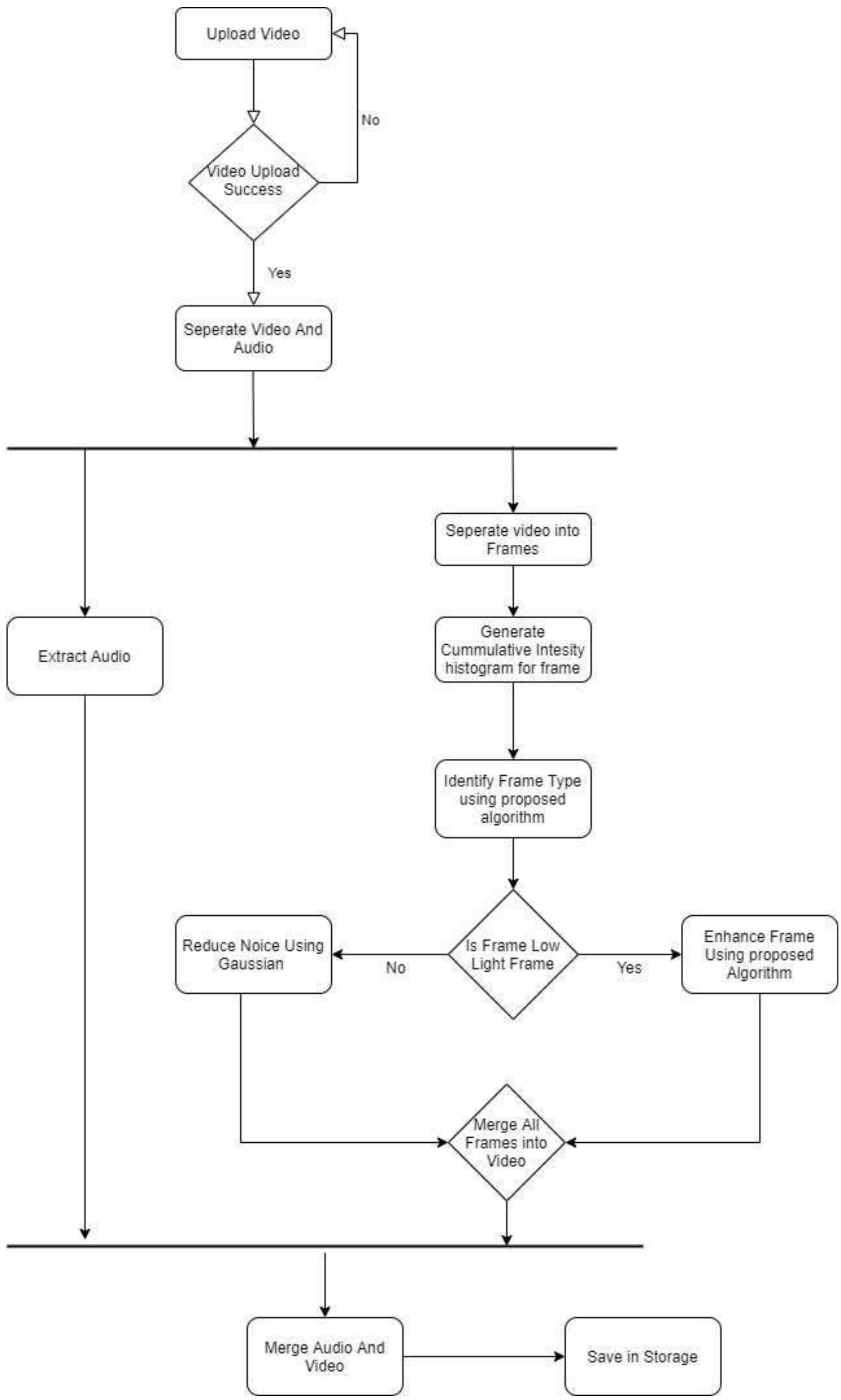


Figure 12: Enhancement flow chart

3.3.1 Intensity histograms and cumulative intensity histograms

In image or video processing the histogram of image or frame of a video is referenced as the histogram of pixel intensity values. This histogram shows the number of pixels in a frame or image at each intensity values [17]. If the image is an 8-bit gray scale image or video frame then the histogram will have 256 different intensity level (Fig 13). Each of these intensity levels are mapped to the total number of pixels in that intensity level. Therefore, the x-axis of the intensity histogram will be the intensity levels and the y-axis of intensity histogram will be the total number of pixels in each intensity level.

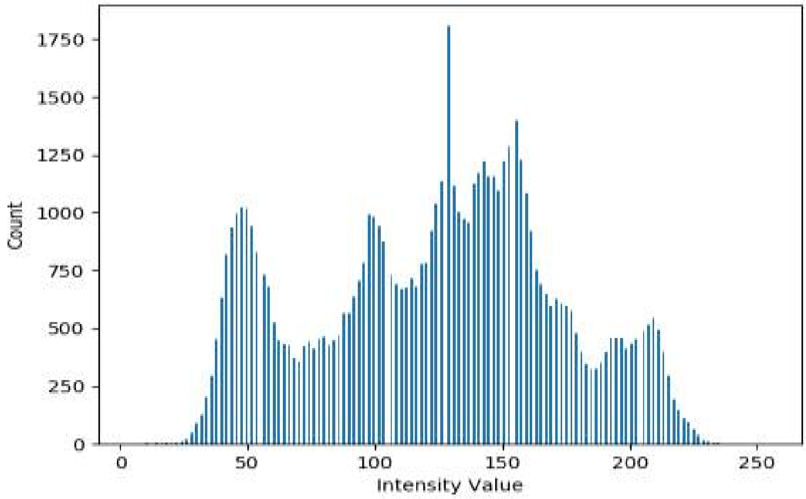


Figure 13: Gray intensity histogram

Histograms can be developed for color images and frames as well. When this type of histogram is developed, they mainly use three channels. These three channels are red, green and blue (RGB) channels. On color intensity histograms each of the channel data represents the total number of pixels at each intensity level of a particular channel [17]. Fig 14 shows a RGB channel color histogram for a color image.

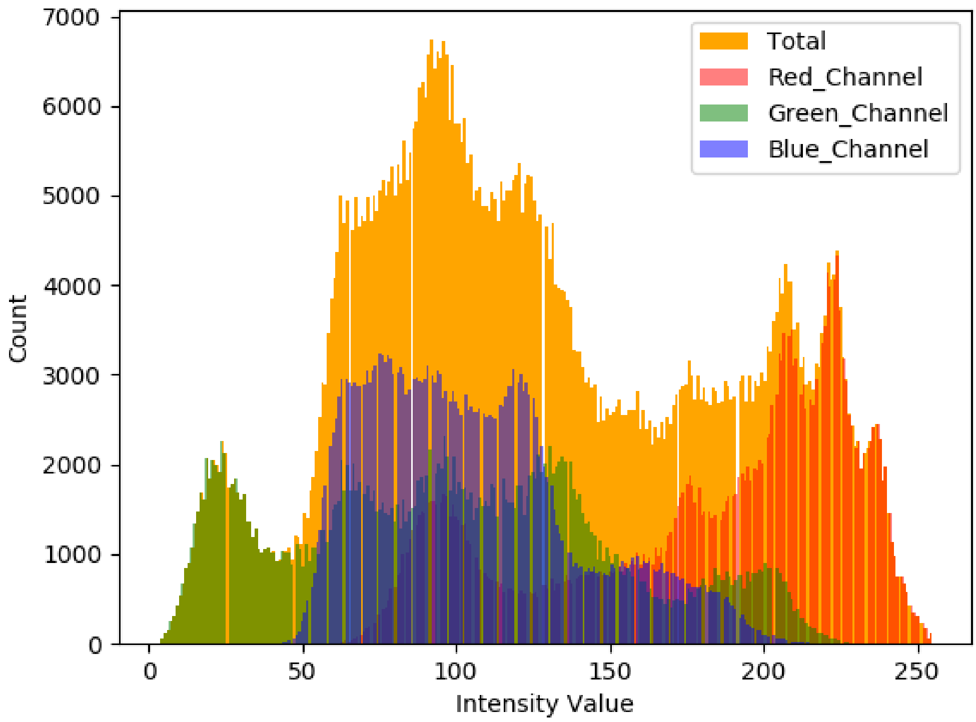


Figure 14: Color intensity histogram

For the development of our proposed algorithm, we used 8-bit gray scale histograms because it doesn't require heavy resources to process them and since our algorithm is only differentiating dark frames from light frames then the grey scale intensity histogram is highly effective.

From the intensity histogram a cumulative intensity histogram was developed. A cumulative histogram is an intensity representation that counts the cumulative number of pixels in all intensities up to the current intensity [18]. Fig 15 shows the representation of a cumulative histogram.

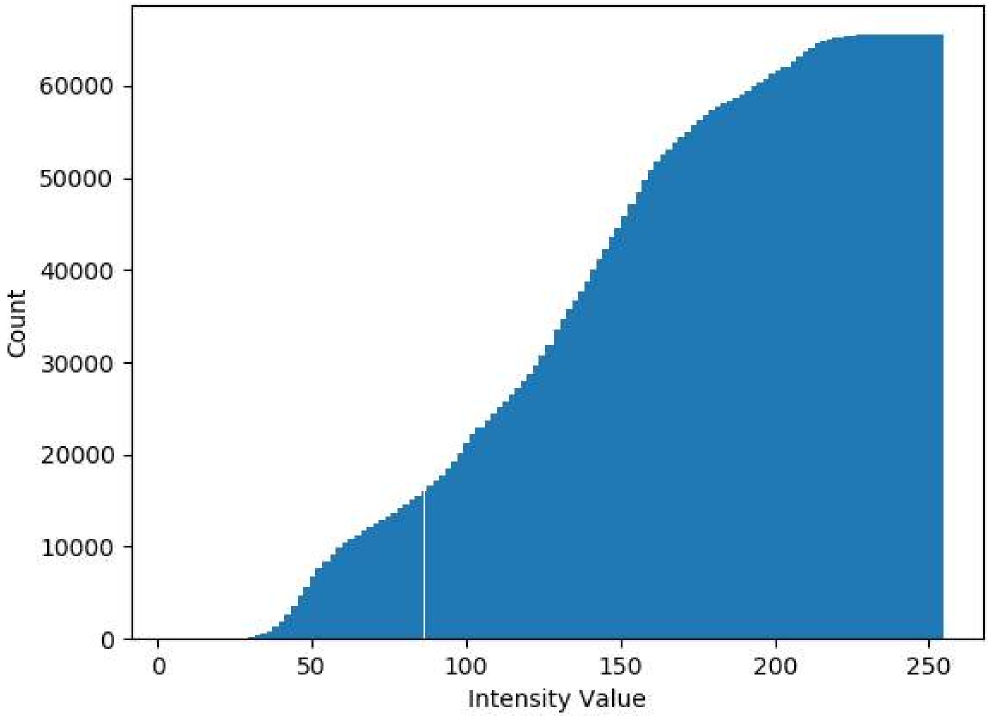


Figure 15: Cumulative intensity histogram

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.

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

Equation 1: Cumulative intensity equation

Equation (1) was used from [19]. 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.

3.3.2 Low light identifying technique

Once the tutor uploads a video the video was broken down into separate frames (Fig 16) and cumulative histogram was generated for each frame in the grey-scale according section 3.3.1.

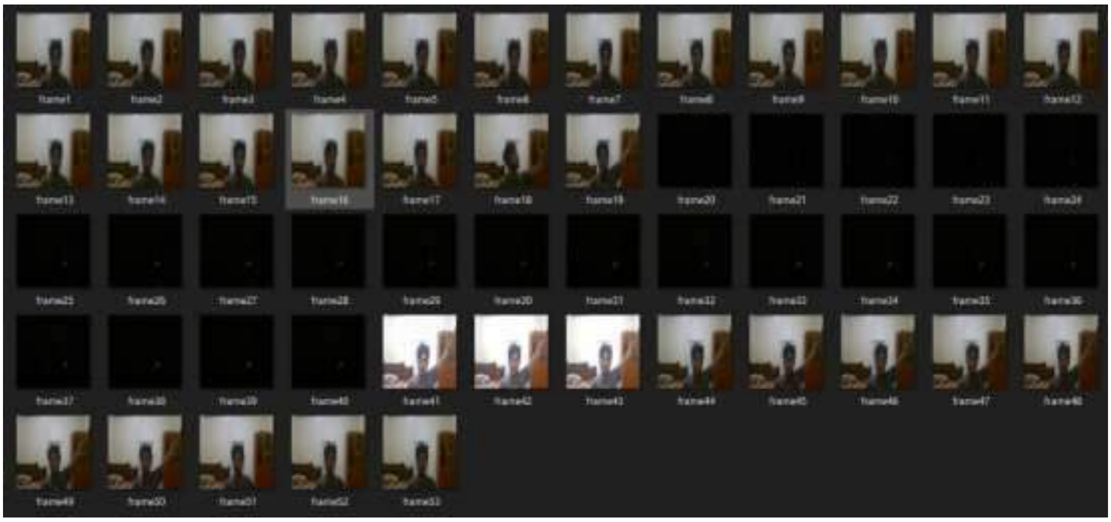


Figure 16: Frames of video

Krutch, et al have identified that intensity distribution of images captured under low light conditions show more than 50% of total number of pixels fall under low intensity values and for day light condition more than 50% of total number pixels fall under high intensity values [20]. Based on this an algorithm was developed which can identify the low intensity frames and high intensity frames.

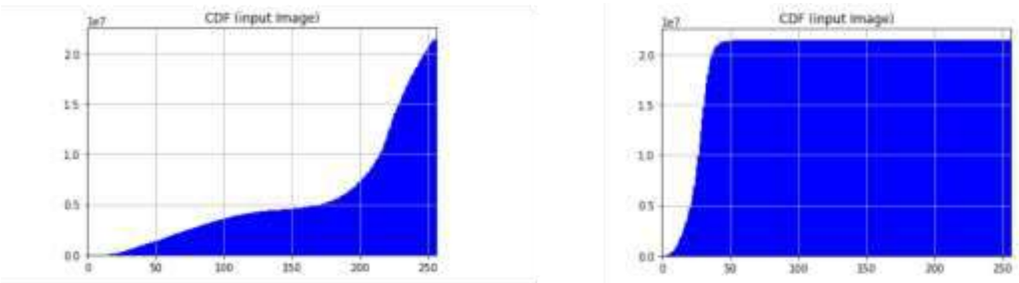


Figure 17: Cumulative histogram of low and normal light image

The histogram on the left of the Fig 17 shows the cumulative histogram of a normal light frame and the histogram on the right of the Fig 17 shows the cumulative histogram of a low light frame. After comparing these two histograms It was found out that we can differentiate both these histograms significantly and therefore, we can easily separate normal light and low light frames.

To separate the low light frames with normal light frames, it was identified to which intensity the majority of the pixels will be mapped to. Fig 18 shows the mapping of the intensity level with its corresponding pixel count and the overall percentage to which this pixel count is calculated for a cumulative intensity histogram.

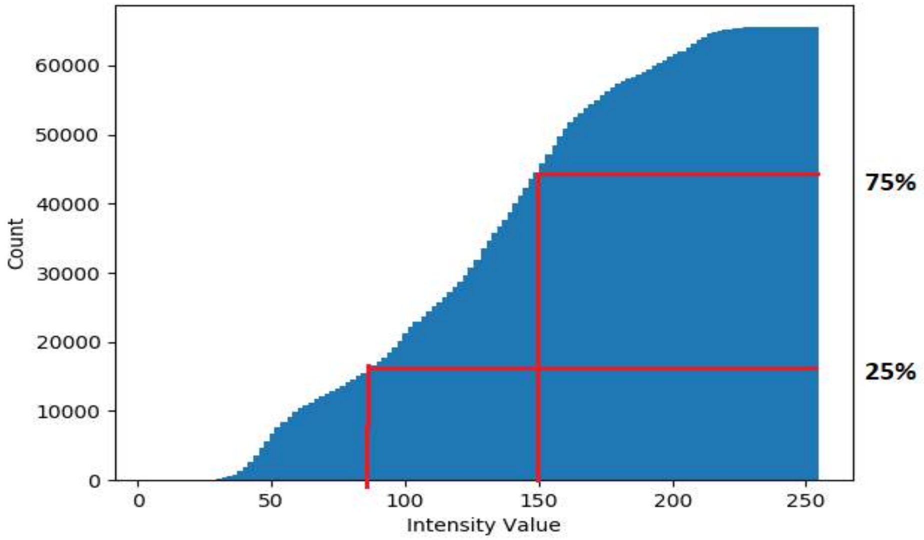


Figure 18: Mapping of cumulative intensity histogram

When comparing Fig 18 we can state that 25% of the total number of pixels corresponds to an intensity level less than 100 and 75% of the total number of the pixel corresponds to an intensity of 150. Therefore, using a cumulative graph we identified to which intensity value the majority of the pixel value will map. According to Fig 18, majority (75%) of the pixel values will corresponds to an intensity level of 150 which means that most of the pixels of this image are in the intensity level 150. This will be the threshold value which decides whether the frame shifts to a low light frame or day light frame.

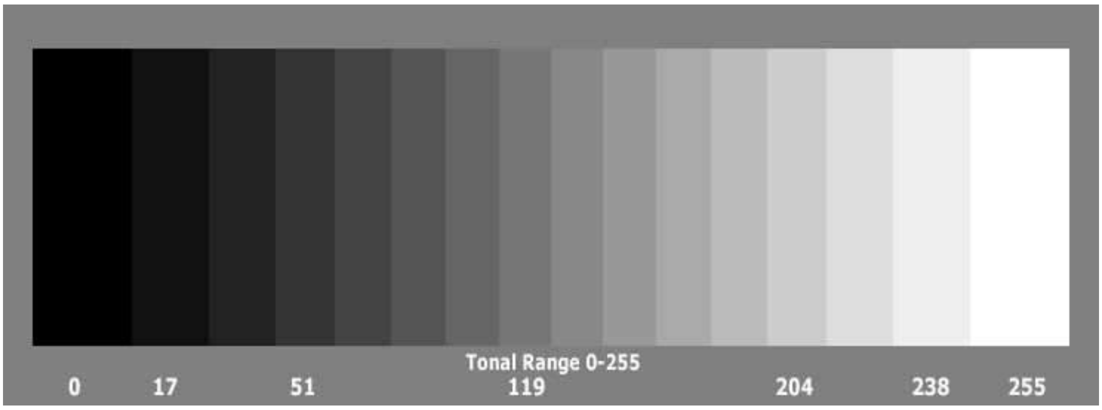


Figure 19: Gray scale intensity distribution

The Fig 19 shows the intensity distribution of an image in grey scale. It is clear that intensity values less than 119 are shifting to a darker region. As a result of this I selected an intensity value which is >100 and <120 as our threshold value to detect the low light frames. After selecting an optimum threshold value, both Fig 18 and Fig 19 was compared and identified that the cumulative histogram (Fig 18) 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.

3.3.3 Enhancement technique

Once the low light frames of a video are identified it was enhanced using a low light enhancement algorithm. For this process gamma correction algorithm was used. Since gamma correction operates in the linear intensity space it is more intuitive [21]. 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.

$$O = C * I^{\gamma} \qquad (2)$$

Equation 2: Gamma correction equation

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].

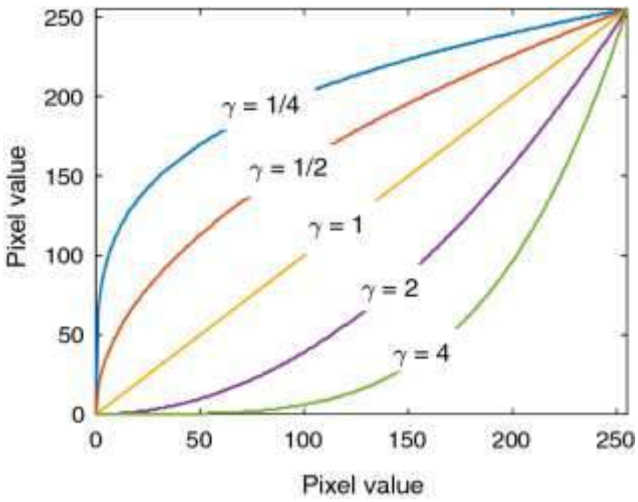


Figure 20: Gamma correction graph

According Fig 20 the x-axis shows the input pixel value and the y-axis shows the corresponding output pixel value after applying gamma correction. A $\gamma < 1$ was selected to enhance the low light frames because it increases the intensities of the pixel after gamma correction.

Figure 21: Gamma corrected images

Fig 21 shows the original and gamma corrected low light images. A gamma value is considered such that $\gamma = 1/G$. G is the gamma values that is available on the Fig 21. A gamma value(G) of 2.0 is selected for our algorithm because it showed the most optimum results for low light images. This value of G was then used for the algorithm to enhance the low light videos in our system.

3.3.4 Noise filtering technique

After enhancing was applied to the low light images it additionally enhanced to noise in the low light images. To reduce the noise in the image median filtering technique was 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 [22]. Median filter computes the median grey level value of the neighborhood which reduces the noise in the image.

For the noise reduction in our algorithm, we have selected a median filter with a kernel size of five. Therefore, the neighbor with a surrounding of five pixels will be used.

Table 2: Non-Linear kernel with size 5

120	155	168	178	168
45	78	100	126	101
128	101	90	178	89
90	104	198	79	103
96	145	152	62	103

Table 2 shows a non-linear kernel with size 5. The kernel is part of the image and the content inside the kernel is the pixel values of the of that part. When we are performing median filtering on a kernel, we first organize the neighboring pixel values in the ascending order. Therefore, the pixels surrounding 90(center pixel) along with the center pixel will be organized in ascending order. Then we calculate the median from the sequence of numbers from the ascending order. Using this technique, we get a median filtered pixel value. This pixel value then will be set as the center pixel. Once the median filter is applied to the above kernel (Table 2) the pixel value 90 was changed with 103. Likewise, every pixel value will be changed with the median filtered pixel for each neighborhood of the pixel.

3.3.5 Code implementation for enhancement module

Input files

+ Code + Markdown

```python
filename = 'video.mp4'
frames_per_second = 24.0
res = '720p'

inputFile1 = 'D:/BSc/4th year/Research/Development/dataset/LL_new.mp4'
inputFile2 = 'D:/BSc/4th year/Research/Development/dataset/LL_video_2.mp4'
inputFile3 = 'D:/BSc/4th year/Research/Development/dataset/LL_env.mp4'
```

Python

function to change resolution

```python
def change_res(cap, width, height):
    cap.set(3, width)
    cap.set(4, height)
```

Python

video Dimensions

```python
STD_DIMENSIONS = {
    "480p": (640, 480),
    "720p": (1280, 720),
    "1080p": (1920, 1080),
}
```

Python

function to get the dimensions

```python
def get_dims(cap, res='1080p'):
    width, height = STD_DIMENSIONS["480p"]
    if res in STD_DIMENSIONS:
        width, height = STD_DIMENSIONS[res]
    change_res(cap, width, height)
    return width, height
```

Python

Video types

```python
VIDEO_TYPE = {
    '.avi': cv2.VideoWriter_fourcc(*'XVID'),
    # '.mp4': cv2.VideoWriter_fourcc(*'H264'),
    '.mp4': cv2.VideoWriter_fourcc(*'XVID'),
}
```

Python

function to get video types

```python
def get_video_type(filename):
    filename, ext = os.path.splitext(filename)
    if ext in VIDEO_TYPE:
        return VIDEO_TYPE[ext]
    return VIDEO_TYPE['avi']
```

Python

Gamma correction function

```python
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)
```

Python

Video Enhancement Function

```python
def enhanceVideo(file):
    print('enhancing video')
    cap = cv2.VideoCapture(file)
    fps = cap.get(cv2.CAP_PROP_FPS)
    print('Frames per second: '+"%.2f" % fps+'%')
    out = cv2.VideoWriter(filename, get_video_type(
        filename), fps, get_dims(cap, res))
    allframes = 0
    lowContrastframe = 0

    #Threshold
    threshold = 103

    while(cap.isOpened()):
        ret, frame = cap.read()
        allframes = allframes + 1
        if frame is None:
            break

        pixels = frame.flatten()

        #Cumulative Histogram
        cdf = plt.hist(pixels, bins=256, range=(0, 256),
                       cumulative=True, density=True,
                       color='blue', alpha=1)
```

```python
        plt.xlim((0, 256))
        plt.grid('off')
        plt.title('CDF (input Image)')
        plt.show()

        # 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
            frame = cv2.GaussianBlur(frame,(3,3),0)
            frame = adjust_gamma(frame, gamma=gamma)
            frame = cv2.medianBlur(frame,5)
        else:
            text = "Normal light image"
            color = (0, 200, 0)

        cv2.putText(frame, text, (5, 25), cv2.FONT_HERSHEY_SIMPLEX, 0.8,
                    color, 2)

        out.write(frame)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

    cap.release()
    out.release()
    print('video saved')
    cv2.destroyAllWindows()
    return (lowContrastframe/allframes)*100
```

Extract Sound function

```python
def extractSound(file):
    print('extracting sound')
    video = VideoFileClip(file)
    audio = video.audio  # 3.
    audio.write_audiofile('audio.mp3')
```

Python

Merge Function

```python
import os
import datetime

def mergeFiles(video,audio):
    print('start merging')
    name = datetime.datetime.now().strftime('%Y-%m-%d_%H-%M-%S')  # Current time
    os.system("ffmpeg -i "+ video +" -i "+ audio +" -strict -2 -f mp4 output/" +
              name + ".mp4")  # Use ffmpe to merge
    os.remove(video)  # Delete the intermediate video file
    os.remove(audio)
    print('merging complete')
```

Python

Main Function

```python
if __name__ == '__main__':
    extractSound(inputFile1)
    percent = enhanceVideo(inputFile1)
    print('enhaced frame percentage: '+"%.2f" % percent+'%')
    mergeFiles("video.mp4","audio.mp3")
```

Python

Main function runs three separate functions. Each function has its own unique functionality hence there are no side effects from these funtions which improves the overall performance of the functionality.

The *extractSound* function is responsible for seperating the sound from the video. After seperating the audio we can enhance the video using *enhanceVideo* function. Once the video has been enhanced we merge both the audio and the video using the *mergeFiles* function.

The *adjust_gamma* function is responsible in enhancing the video frame by creating a lookup table for input pixels and output pixels after gamma correction. The *get_video_type* function is responsible in using the selected codec standards when saving videos. The *get_dims* function is responsible for selecting the dimensions of videos which are enhanced.

3.4 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.

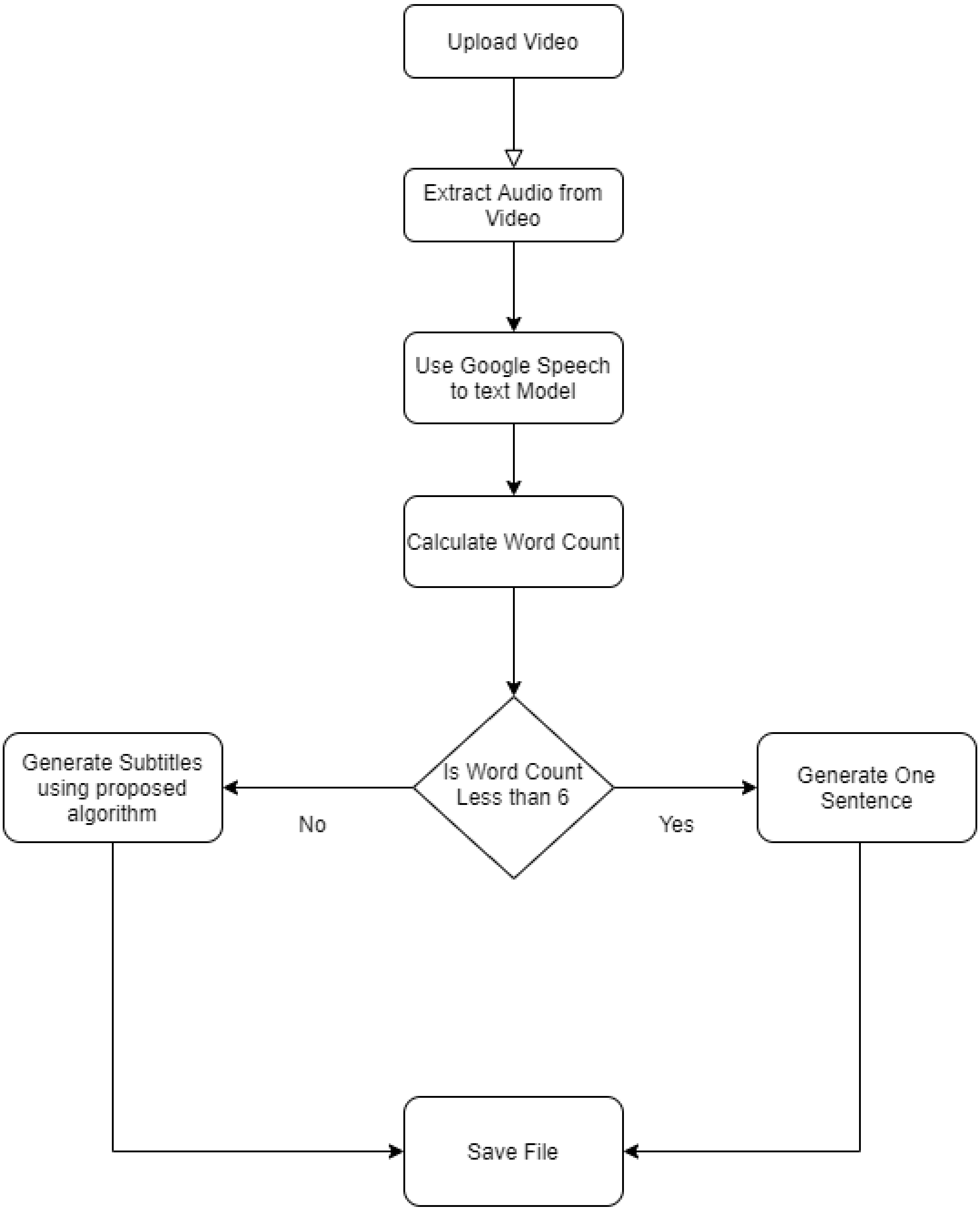


Figure 22: Caption flow chart

3.4.1 Speech to text model

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 [23].

Table 3: Percentage (%) accuracy of predicted sentences

IBM	GOOGLE
26.67	43.33

According to Foteini, et al they have identified that the GCP STT model has the lowest mean error compared to other STT models. According to the results of Table 3 they have identified that the GCP STT model responds better regardless of the speaker [23]. Based on their findings we concluded that the usage of GCP STT model will be more flexible and advantages.

3.4.2 Captioning algorithm

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 [24]. 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 [24].

3.4.3 Code implementation of captioning module

```python
from os import path
import datetime
import speech_recognition as sr

# Extract Audio from Video
from moviepy.editor import *

videofile1 = 'D:/BSc/4th year/Research/Development/dataset/LL_env.mp4'
videofile2 = 'D:/BSc/4th year/Research/Development/dataset/sr_1.mp4'
videofile3 = 'D:/BSc/4th year/Research/Development/dataset/1min.mp4'
# output_file.title

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 the STT Model
    r = sr.Recognizer()
    with sr.AudioFile(AUDIO_FILE) as source:
        audio = r.record(source)                    # reads the entire audio file
```

```python
    # 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())
```

```python
            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
        else:
            start = 0
            end = start + 6

            # assume 0.5 seconds taken to read a word
            time_add = len(arr[start:end])*0.5
            end_time = current_time + datetime.timedelta(0, time_add)
```

```python
        # 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[start:end]
        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")

        new_arr = arr[end:wordscount]
        newcount = len(new_arr)
        new_currentTime = end_time
        # calling the function recursively
        createSubs(new_arr, newcount, new_currentTime, blk)
```

```python
    createSubs(word_arr, word_count, start_time, block_num)

    os.remove('audio.wav')

    print('============= Subtitle Generate Complete===========================')
#       return subs

if __name__ == '__main__':
    generate_subs(videofile)
```

The main function has a function called *generate_subs* function which is responsible to generate subtitles for the video. First the audio is extracted from the video and then sent to the GCP STT model. The output will then be split and the word count is identified. Then the timestamps are calculated with the proposed algorithm.

The final output of this function will be a separate text file with the timestamps for each sentence.

3.5 Development Process

For the development of this system our team decided to select agile methodology because of the nature of the project. Since this was a research project, we decided that there can be changing requirements in the middle of the project and we did not want it to affect the system. We selected Scrum as our Agile methodology where we had Sprints which had time frames. During each sprint a phase of the project was completed. We also had repetitive sprints to manage the changing requirements and to adopt with new research findings.

3.6 Feasibility Study

For the development of low light enhancement and captioning technique we planned to use python as our main language. Since there were also machine learning and other processing scripts to be used python was the best option which can utilize the entire CPU with its multiprocessing capabilities. Following are some specific tools used for the development on this study.

Image and Video Processing –	OpenCV
Scripting -	Jupyter notebook
Database -	SQLite
Storage -	Firebase Firestore
Backend Server –	Flask (Python library)
Frontend -	ReactJS (JavaScript library)
Version Control -	GitLab
Task Manager -	Microsoft Teams Planner

3.7 Tool Selection

3.7.1 OpenCV

OpenCV was selected as the main image processing and video processing library for this study. OpenCV is highly effective in manipulating images and video frames with efficient inbuilt methods. Python was the main developing language used to develop OpenCV algorithms in this study.

3.7.2 Jupyter notebook

Jupyter notebook was used to develop the scripts for low light enhancement and captioning modules. Jupyter notebook is an open-source publication which allows us to write scripts in python for machine learning and various other implementations.

3.7.3 SQLite

To store the enhanced video details and captioning details the SQLite database was used. Since this database has good performance with flask server the SQLite database was used as the relational database for this study.

3.7.4 Firebase firestore

Firebase was selected as the main storage unit to store the enhanced videos and captioned files. Since firebase has better performance in low bandwidth and has high availability this was selected as the main storage unit for this study.

3.7.5 Flask and reactJS

Flask was used as our main backend server library. Since flask is a python library it was easy for the integration of low light enhancement and captioning python scripts to the backend server. For creating UI elements, we used ReactJS as our frontend JavaScript library because it has code reusability and high performance. The backend and frontend servers were communicating through REST API.

3.7.6 GitLab and Microsoft teams planner

Gitlab was our main version control tool and Teams Planner was the tool used for task tracking.

4. REQUIREMENTS AND COMMERCIALIZATION

4.1 Requirements

The main functional requirements of the system proposed in this report are listed below,

- Automated low light identification of uploaded videos.
- Automated enhancement of low light frames.
- Generation of automated captions for uploaded videos.

The non-functional requirements of the system proposed are,

- User-friendliness of the application.
- Performance of the application.

4.2 Commercialization

The solution proposed by this report solves a problem that rarely addressed in the education sector. Although, there have been some efforts to create systems for hearing impaired students, this product provides a more advance solution for the problem. Since the domain of this problem involves the education sector, this solution can be further improved and standardized to improve the quality of education and the efficiency of teaching. Since our system is directly using hearing-impaired students as its end user this system can be used in the hearing-impaired schools or tutor centers. This system can also be used as a web application for the use of general public for learning sign languages.

The low light enhancement module introduced in this solution can be used as an enhancement technique for the use in low light video recording software. The captioning module is also capable of integrating itself with other content delivery software. This brings more automated features like automated enhancement and automated captioning into certain software. This reduces the additional human efforts when using the software.

5. TESTING

5.1 Unit Testing

The unit testing was done to validate the functionality of each of the units of the system. The major requirement of unit testing for this system was done to check the automated low light detection and enhancement technique with automated captioning. Photos taken in different lighting conditions and videos recorded in different lighting conditions were used for testing.

Low light enhancement and automated captioning units are tested as separated entities when unit testing. We compared the enhanced output images and video frames with its inputs. When testing the automated captioning unit, we tested how the captions are separated with the given time frame of the video along with its accuracy. Each unit was tested with three or more test cases.

5.2 Integration Testing

Low light enhancement module and the automated captioning module was integrated and tested together to get the final output of the functionality. During the integration testing the system was checked with several test cases to see if there were any unexpected outputs when the system was integrated. Each of the test cases had videos recorded using web cams.

When writing test cases for videos, the videos were separated into several frames and frames were used as the input for the test cases. This testing method reduced complexity of writing test cases for the videos.

5.3 System Testing

The solution was integrated with the other components of the complete research and the performance was evaluated using practical data.

5.4 Test Cases

Table 4: Test case 1 - low light identification

Test case	Test case 01
Description	Testing the automatic low light detection using the proposed algorithm
Test Procedure	1. Run the system and input a normal light image to the input field. 2. Run the low light identification script. 3. Output is displayed.
Input	
Expected Output	The output image should have a label as normal light image.
Actual results	
Pass/ Fail	PASS

Table 5: Test case 2- low light identification

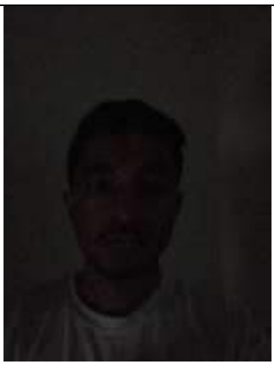
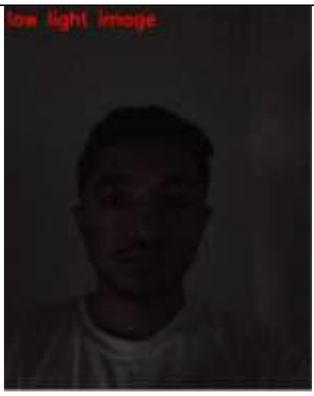
Test case	Test case 02
Description	Testing the automatic low light detection using the proposed algorithm
Test Procedure	1. Run the system and input a low light image to the input field. 2. Run the low light identification script. 3. Out is displayed.
Input	
Expected Output	The output image should have a label as low light image.
Actual results	
Pass/ Fail	PASS

Table 6: Test case 3- low light identification

Test case	Test case 03
Description	Testing the automatic low light detection using the proposed algorithm
Test Procedure	1. Input a video with low light frames to the input field. (Algorithm is capable of separating the video into separate frames) 2. Run the low light identification script. 3. Output is displayed.
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.
Pass/ Fail	PASS

Table 7: Test case 4 - low light enhancement

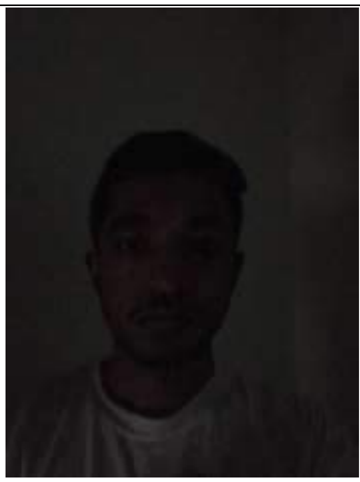
Test case	Test case 04
Description	Testing the low light enhancement using proposed algorithm
Test Procedure	1. Input the low light image in the input field. 2. Run the low light enhancement script. 3. Output is displayed.
Input	
Expected Output	The low light image has to be enhanced automatically.
Actual results	
Pass/ Fail	PASS

Table 8: Test case 5 - low light enhancement

Test case	Test case 05
Description	Testing the low light enhancement using proposed algorithm
Test Procedure	1. Input the low light image in the input field. 2. Run the low light enhancement script. 3. Output is displayed.
Input	
Expected Output	The low light image has to be enhanced automatically.
Actual results	
Pass/ Fail	PASS

Table 9: Test case 6 - low light enhancement

Test case	Test case 06
Description	Testing the automatic low light enhancement using the proposed algorithm
Test Procedure	1. Input the video with low light frames in the input field. 2. Run the low light enhancement script. 3. The enhanced video will be saved.
Input	
Expected Output	The low light video frames have to be enhanced.
Actual results	  Low light frames of the video have been enhanced.
Pass/ Fail	PASS

Table 10: Test case 7 - Captioning

Test case	Test case 07
Description	Testing the captioning function with the proposed algorithm.
Test Procedure	1. Input the video to be captioned in the input field. 2. Run the captioning script. 3. Captions will be saved as a word file.
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 6 00:00:15-->00:00:18 turn on the light now we 7 00:00:18-->00:00:21 have normal illumination.
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
Pass/ Fail	PASS

The computer used for testing had the following specifications.

- Operating System – Windows 10 Home (64 bit)
- CPU - Intel Core i5 H-Series (2.5GHz)
- RAM – 8GB
- GPU - Nvidia GEFORCE MX150 (2GB)
- Video Codecs installed -
    - bdmjpeg
    - xvidvfw
    - msyuv
    - tsbyuv
    - msvidc

6. RESULTS AND DISCUSSION

6.1 Results

This section explains about the results that were obtained during the testing of the low light enhancement and captioning functions. The low Light enhancement module was mainly divided into sub sections on this section. The low light identification and low light enhancement are the main sub module that will be discussed along with the captioning module.

6.1.1 Output of low light identification

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.

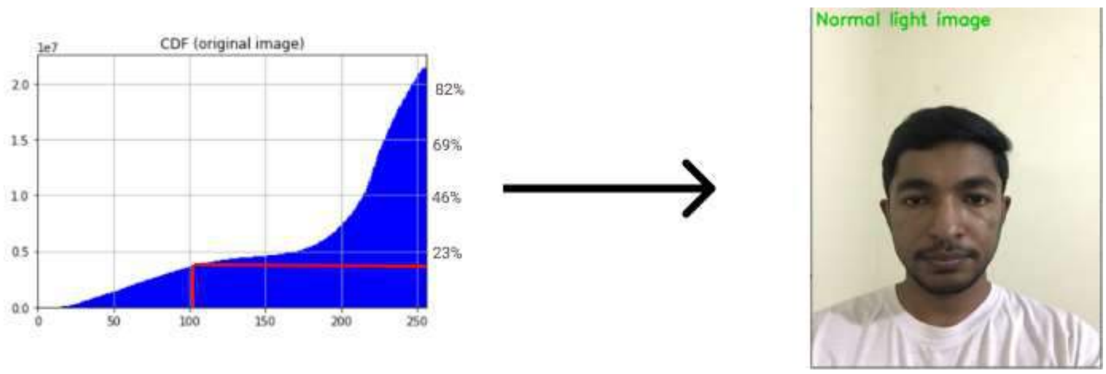


Figure 23: Normal light image with cumulative histogram

Fig 23 shows a normal light image and its corresponding cumulative histogram. For testing purposes, we selected our threshold intensity value as 103. This threshold value is selected such that the intensities below the threshold represents low light(dark) intensity values and the intensities above the threshold represents normal light (bright) intensity values. At the intensity of value 103 the cumulative histogram was mapping to a percentage value <25% which is <75% of the total intensity of the image (Fig 23). Therefore, based on the proposed algorithm on section 3.3.2 the image was identified as a normal light image.

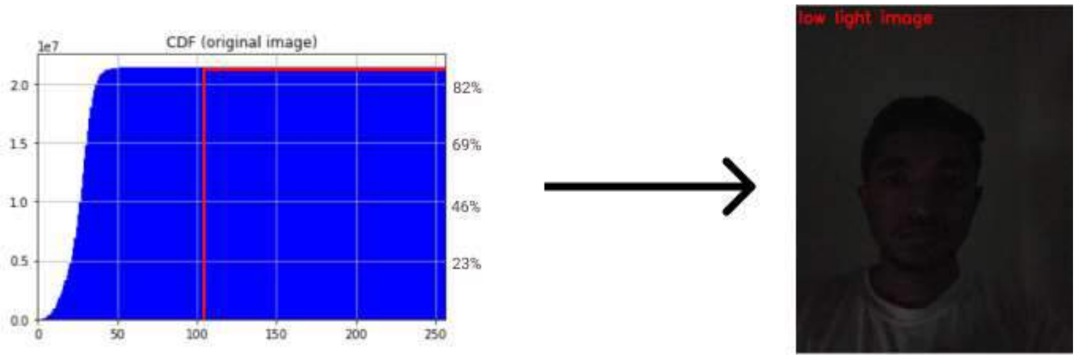


Figure 24: Low light image with cumulative histogram

The Fig 24 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.

Similarly, the videos were also tested. When testing videos, the algorithm is capable of separating the videos into different frames and identifying low light and normal light frames in the video (Table 6).

6.1.2 Output of low light enhancement

On this section the low light enhancement results are discussed. During the testing phase of this functionality images and videos were tested to check the enhancement capability of the algorithm.

Before enhancing the images, they were tested to with low light identification algorithm to check the nature (low light or normal light) of the image. If the image is identified as a low light image, then it will be enhanced using the enhancement algorithm proposed in section 3.3.3.

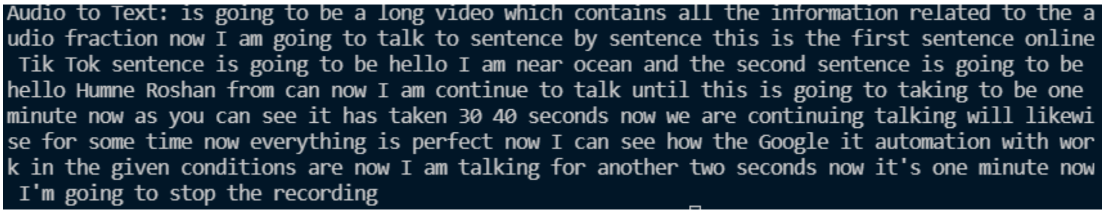
Figure 25: Low light enhanced output

Fig 25 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 Fig 25 has denoised itself using the denoising algorithm (section 3.3.4).

When testing videos using this module the videos were divided into separate frames and each frame was tested for being a low light frame and if it identifies itself as a low light frame then the frame will be enhanced using the low light enhancement algorithm (Table 9).

6.1.3 Output of captioning algorithm

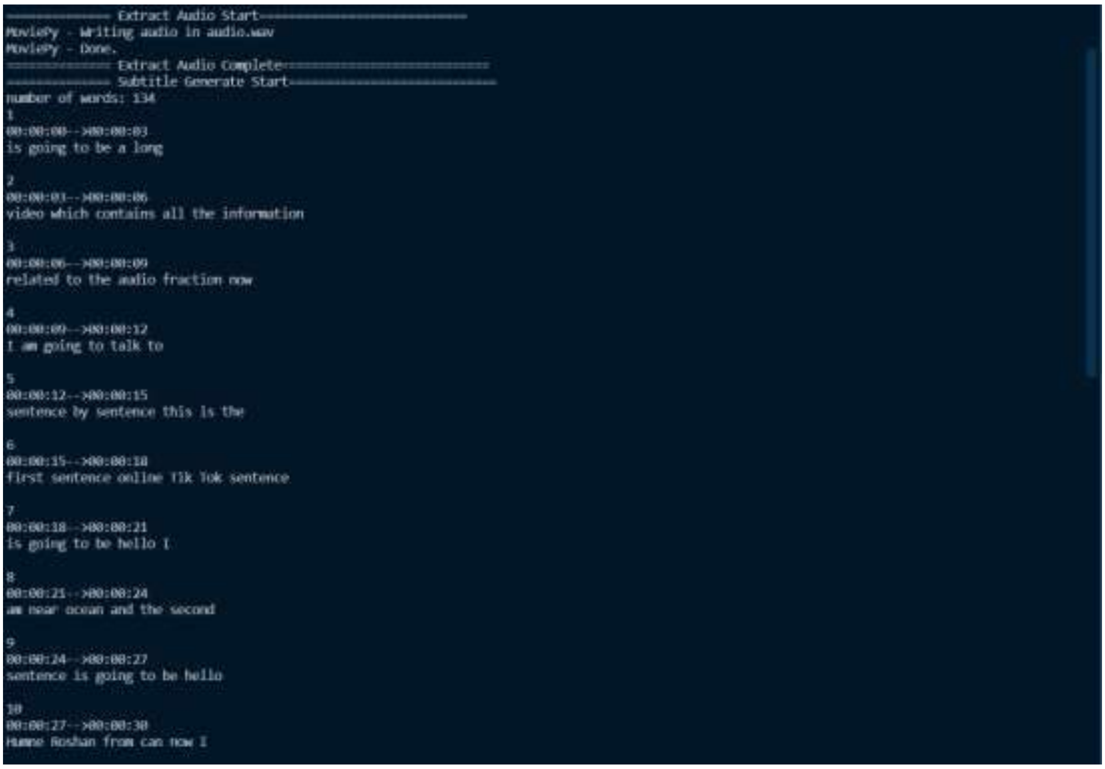
For this algorithm the video was sent as the input and the audio in the video is first extracted. And audio is then sent to the google speech to text model to extract the speech in the audio. The output of the GCP STT model was a continuous sentence (Fig 26).



```
Audio to Text: is going to be a long video which contains all the information related to the a
udio fraction now I am going to talk to sentence by sentence this is the first sentence online
 Tik Tok sentence is going to be hello I am near ocean and the second sentence is going to be
hello Humne Roshan from can now I am continue to talk until this is going to taking to be one
minute now as you can see it has taken 30 40 seconds now we are continuing talking will likewi
se for some time now everything is perfect now I can see how the Google it automation with wor
k in the given conditions are now I am talking for another two seconds now it's one minute now
 I'm going to stop the recording
```

Figure 26: Direct output of GCP STT model

Since the output form GCP STT model has a continues sentence it 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 in the section 3.4.2. The final output of the captioning module has the sentences along with its timestamps (Fig 27).



```
=============== 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
Humne Roshan from can now I
```

Figure 27: Final output of captioning module

6.2 Research Findings

The study was focused on finding a solution for hearing impaired students. An eLearning system was developed for the hearing-impaired students as there was no proper eLearning environment for hearing impaired students. The main research findings recognized from this study was to automate the low light enhancement of videos captured in low light environment and providing the automated captioning for the speech in the videos. The section 6.1 shows that most of the output obtained from low light enhancement and captioning are efficient and useful in the eLearning system. The low light enhancement is useful in providing visually clear content to hearing impaired students. The captions produced through the captioning algorithm is useful in the conversion of speech to sign language.

The low light enhancement algorithm was effective as it used simple computational process for identifying low light videos and enhancing low light frames. This gives the system a higher advantage in uploading and processing videos. The captioning algorithm was highly effective and accurate when the speech in the audio is sharp and clear. Therefore, when the speech is clear the output gives exact text and its timeframes.

6.3 Discussion

6.3.1 On low light enhancement

In this study the automated low light enhancement was used for the uploaded videos in the eLearning system. When uploading videos to the eLearning environment which is created for hearing-impaired students the uploaded video has to be visually clear. Providing a clear content is subjective. A visual clear content can be with less noise or can be with good contrast content etc. For this study we mainly focused on automating the low light enhancement process of the uploaded video. The uploaded video will be checked to have low light frames and if the low light frames are present then the algorithm will enhance them to provide visually clear content. When the videos are enhanced, the algorithm output the percentage to which the videos have been enhanced (Fig 28).

Figure 28: Enhancement algorithm output

According to Fig 28 the video has been read at 20 frames per seconds (FPS) and the enhanced frames percentage is 15.56%. This value can differ based on one video to another depending on the low light frames available on the video. Similarly, some videos recorded through web cameras were enhanced using this algorithm and their respective enhanced percentages were given on table 11.

Table 11: Enhanced percentage of videos

Video	Enhanced Percentage
Video 1	15.56%
Video 2	20.04%
Video 3	45.56%
Video 4	12.76%
Video 5	32.31%

6.3.2 On captioning

When converting speech to sign language it cannot be directly converted it has to be converted to an intermediate medium which can be converted into sign language. For this text was the most prominent solution. Therefore, to convert speech to text we have used the GCP STT which has the highest accuracy rates [23]. The text output of this model is then converted into sentences with timeframes using our proposed algorithm. This gave us the timeframes to which the captions can be used to converted into sign language. This method was effective in converting real time audio into sign language and also highly effective as there is no additional human effort is required to convert the speech to text. But to have a successful output of this captioning unit we need clear audio without any background noise and the speech on the audio must be sharp and loud.

6.3.3 User interface design

The domain of this study was maintained under Human Computer Interaction (HCI). Therefore, the human perspective of the system was highly considered when creating the UI. When creating the UI for the enhancement and captioning module first Mockups were created and the mockups were then re-engineered to create UI for the system.

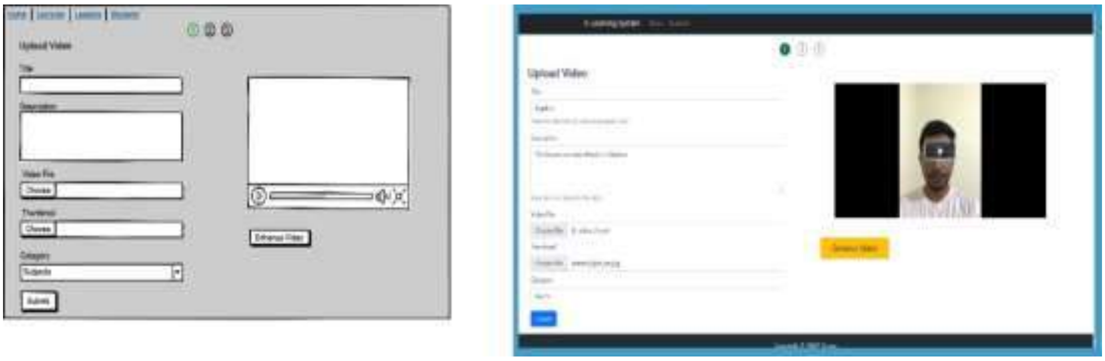


Figure 29: Upload form mockup and UI

Figure 30: Enhanced Mockup and UI

Figure 31: Captions Mockup and UI

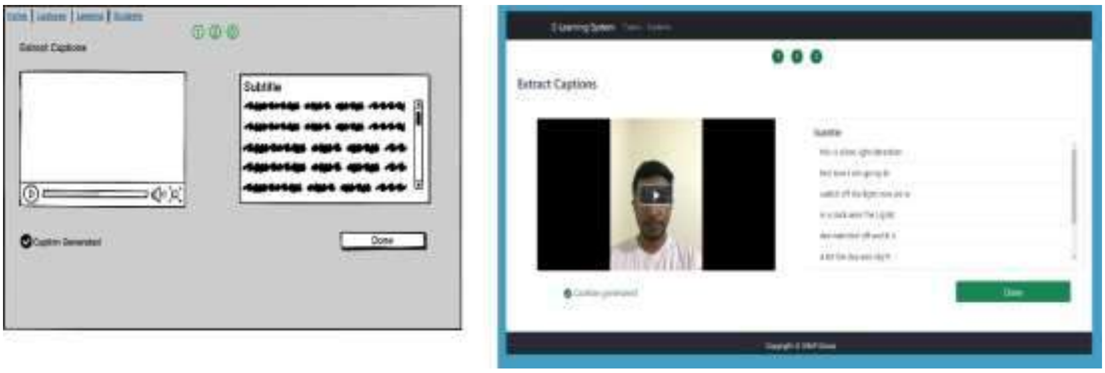


Figure 32: Caption generated mockup and UI

7. CONCLUSION

There was a need for a proper eLearning system for the hearing-impaired students to improve their learning during the crisis like global pandemic where most of the sectors shifted to online. When hearing impaired students tried to use the traditional eLearning systems, they encountered some problems. The lessons were mainly conveyed through videos and if the videos are visually unclear then the hearing-impaired students faced difficulties. As a solution for this issue, we introduced automated low light enhancement for videos through this study. The low light enhancement algorithm is capable of automatically identifying low light frames of a video and enhancing and denoising it. These enhanced videos proved to have clear content when compared to its original video in the testing phase of this study. For the low light identification and enhancement algorithm we have used a simple computational process which can be used in computers with less processing powers and servers. This allows the algorithm to be used in web applications and small-scale software. Along with the low light algorithm we have also introduced a technique to create automated captions for the video. This captioning algorithm helps converting the speech into text which will then be used to convert into sign language. To convert speech to text we have used the GCP STT model since it proved to be highly efficient compared to other commercial STT models. The final output of the captioning module will have sentences with its corresponding time stamps which can be used to easily converted into sign language.

This system also has other features like sign to text and sign language teaching assistant. With all these functionalities integrated the system proved to be effective among the hearing-impaired students. This system can also be optimized with its own functionalities. The low light enhancement functionality can be more optimized to work effectively in low network bandwidth and the captioning module can be more optimized to identify the speech with more accuracy and low quality of sound in the future works.

REFERENCE

[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] J. Elfein, "Statista," 11 June 2020. [Online]. Available: https://www.statista.com/statistics/888654/number-of-people-with-hearing-loss-worldwide-projections-by-region/.

[3] Moriera and Ines, "The Corporate Shift towards eLearning," docebo, 2011. [Online]. Available: https://www.docebo.com/blog/corporate-shift-to-elearning/.

[4] S. Alkhalaf, S. Drew and T. Alhussain, "Assessing the Impact of e-Learning Systems on Learners: A Survey Study in the KSA," in *Cyprus International Conference on Educational Research*, 2012.

[5] Anonymous, "Biggest LMS Software challenges," eLogic Learning, 9 November 2016. [Online]. Available: https://elogiclearning.com/top-3-lms-software-challenges-and-how-to-avoid-them/.

[6] R.Ranchel, Teresa, Y. Guo and K. Bain, "Using speech recognition for real-time captioning and lecture transcription in the classroom," in *IEEE Transactions of Learning Technologies*, 2013.

[7] P. A. Ahmed and M. Ganaie, "Neuroticism of Physically Challenged Viz. Visually Impaired, Hearing Impaired and Orthopedically Impaired Secondary School Students of Kashmir Division," *Journal of Pharmacy and Alternative medicine,* vol. 5, 2015.

[8] 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.

[9] 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.

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

[11] J.-T. Lee, C. Lee, J.-Y. Sim and C.-S. Kim, "Depth-guided adaptive contrast enhancement using 2D histograms," in *IEEE International Conference on Image Processing*, 2014.

[12] 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.

[13] M. Kim, D. Park, D. K. Han and H. Ko, "A novel approach for denoising and enhancement of extremely low-light video," *IEEE Transactions on Consumer Electronics,* vol. 61, no. 1, 2015.

[14] D. BakkiyaLakshmi, R. Kanchana and V. Nagarajan, "Video enhancement using tone adjustment," in *International Conference on Communication and Signal Processing*, 2012.

[15] 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.

[16] J. T. Nganji, "Designing Disability-Aware E-Learning Systems: Disabled Students' Recommendations," *International Journal of Advanced Science and Technology,* vol. 48, pp. 61-70, 2012.

[17] K. Singh and R. Kapoor, "Image enhancement using Exposure based Sub Image Histogram Equalization," *Pattern Recognition Letters,* vol. 36, pp. 10-14, 2014.

[18] P. Garg and T. Jain, "A Comparative Study on Histogram Equalization and Cumulative Histogram Equalization," *International Journal of New Technology and Research (IJNTR),* vol. 3, no. 9, pp. 41 - 43, 2017.

[19] W. Burger and M. J. Burge, Digital Image Processing, London: Springer-Verlag , 2008.

[20] R. Krutsch and D. Tenario, "nxp," June 2011. [Online]. Available: https://www.nxp.com/docs/en/application-note/AN4318.pdf. [Accessed 29 09 2021].

[21] S. Rahman, R. M. Md, A. Al-Wadud, G. Dastegir and M. Shoyaib, "An adaptive gamma correction for image enhancement," *EURASIP Journal on Image and Video Processing.,* p. 35, 2016.

[22] G. George, R. M. Oommen, S. Shelly, S. S. Philipose and A. M. Varghese, "A Survey on Various Median Filtering Techniques For Removal of Impulse Noise

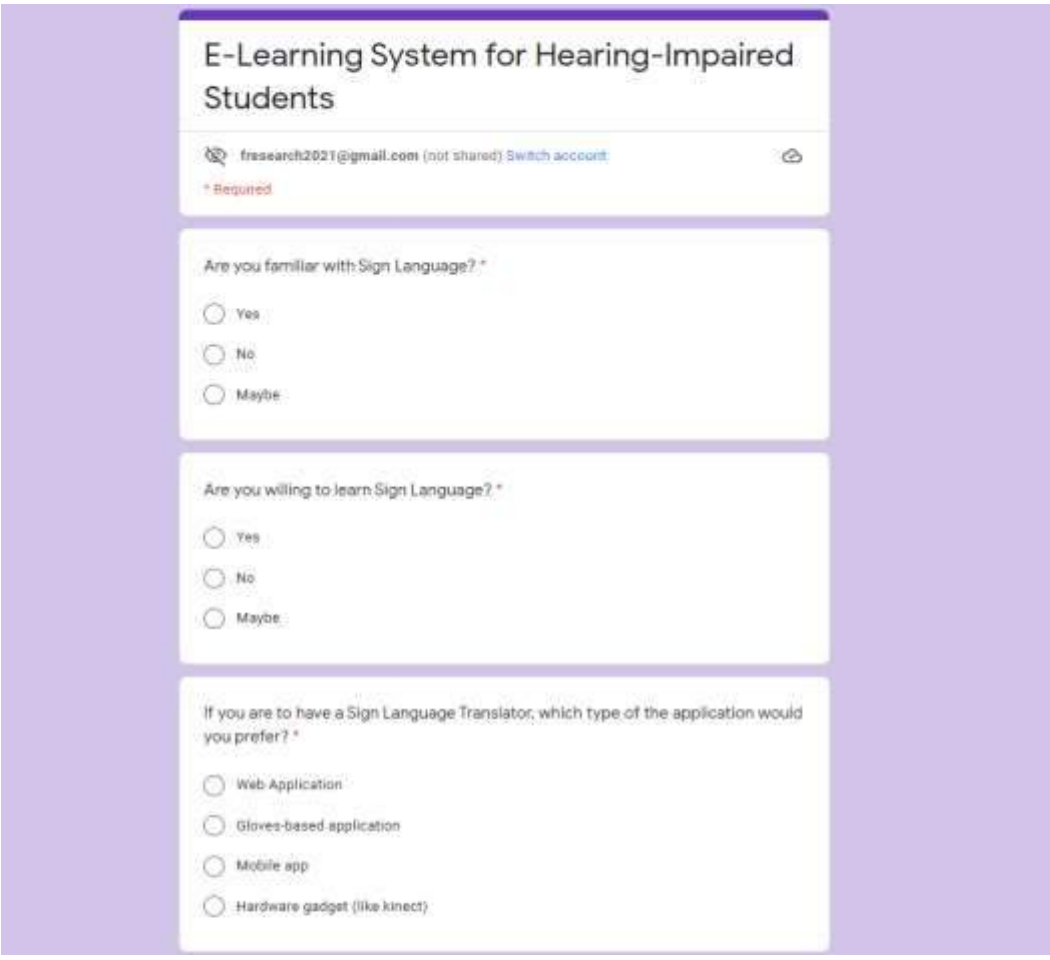
From Digital Image," in *2018 Conference on Emerging Devices and Smart Systems (ICEDSS)*, Tiruchengode, India, 2018.

[23] F. Filippidou and L. Moussiades, "A Benchmarking of IBM, Google and Wit Automatic Speech Recognition Systems," in *IFIP International Conference on Artificial Intelligence Applications and Innovations*, Greece, 2020.

[24] Anonymous, "Reading Time by Word Counts," Capitalize my title, 2015. [Online]. Available: https://capitalizemytitle.com/reading-time/10-words/. [Accessed 4 October 2021].

APPENDICES

Survey



E-Learning System for Hearing-Impaired Students

fresearch2021@gmail.com (not shared) Switch account

* Required

Are you familiar with Sign Language? *

- Yes
- No
- Maybe

Are you willing to learn Sign Language? *

- Yes
- No
- Maybe

If you are to have a Sign Language Translator, which type of the application would you prefer? *

- Web Application
- Gloves-based application
- Mobile app
- Hardware gadget (like kinect)

Do you prefer for the Uploaded Video to enhance Automatically? *

- ◯ Yes
- ◯ No
- ◯ Maybe

Do you prefer if the System can create captions Automatically for your speech? *

- ◯ Yes
- ◯ No
- ◯ Maybe

How helpful if the lectures happen in sign language? 1-Not helpful 5-very much helful *

1	2	3	4	5
◯	◯	◯	◯	◯

Will it be helpful if your interpreter can identify the face expression of the user? *

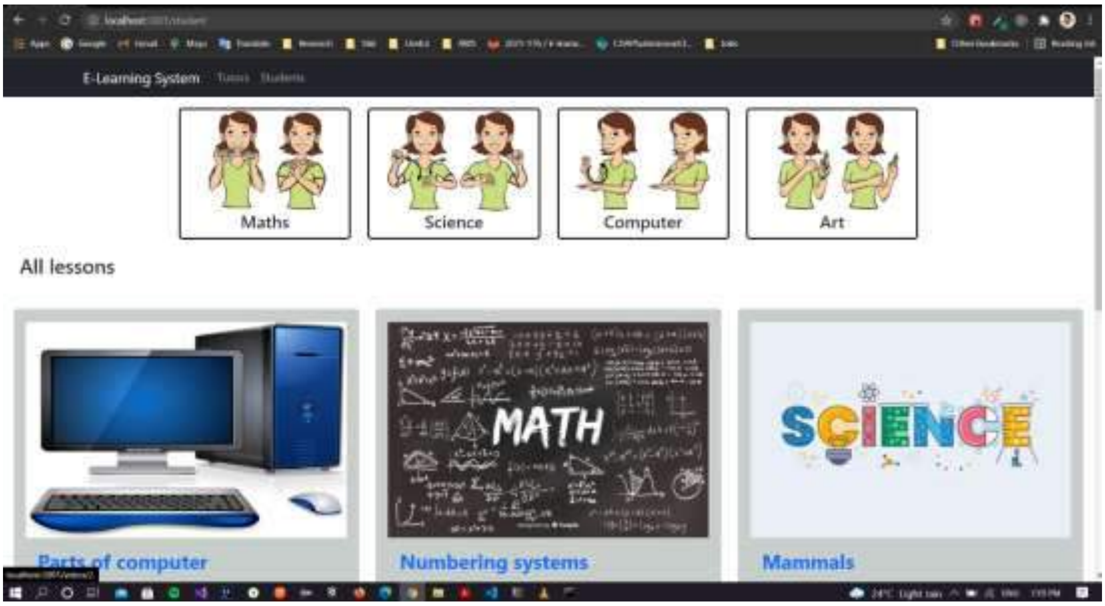
- ◯ Yes
- ◯ No
- ◯ Maybe

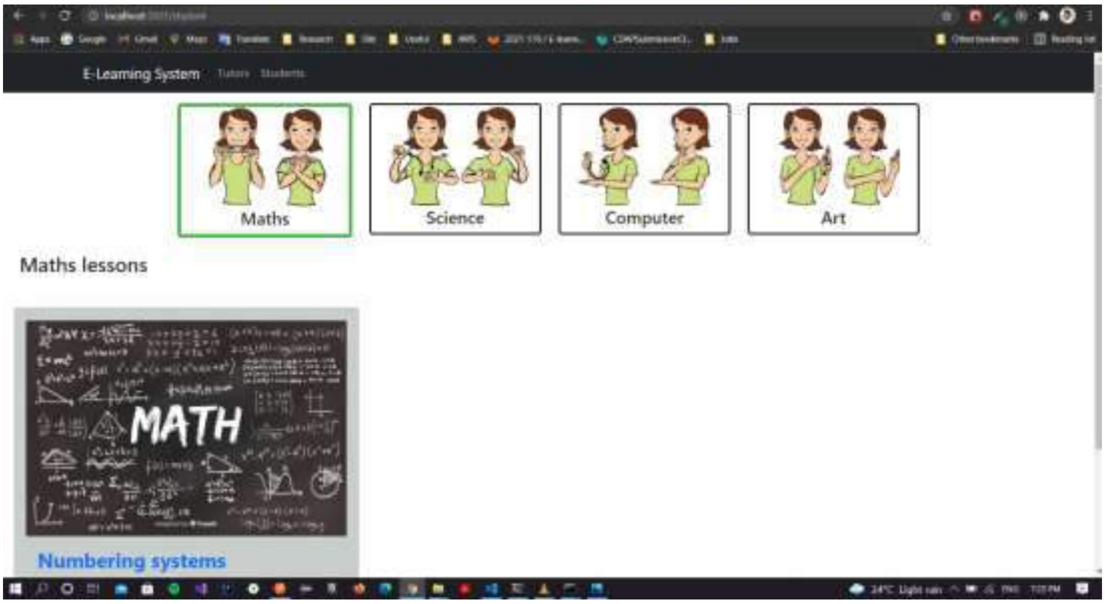
Is it useful if the hearing-impaired students can clear their doubts using sign language? *

- ◯ Yes
- ◯ No
- ◯ Maybe

Submit Clear form

Other UIs of the system





Plagiarism Report