Search Technologies Report on Video Summarization

Student Names:	Vincent Achukwu Alan Wu Niall Bermingham
Student IDs:	17393546 18332283 18392656
Student Emails:	vincent.achukwu2@mail.dcu.ie alan.wu2@mail.dcu.ie niall.bermingham4@mail.dcu.ie
Programme:	BSc in Computer Applications
Module Code:	CA4009
Date of Submission:	26/11/2021

Table of Contents

Table of Contents 2

Declaration 3

Abstract 4

1. Introduction 4

2. User Analysis 4

2.1 The User 4

2.2 Scenarios 5

2.2.1 Searching for a video 5

2.2.2 Uploading a video 5

2.2.3 Searching for keyframes in a video 6

3. Scientific Functional Description 7

3.1 Overall Architecture 7

3.2 Algorithms 7

3.2.1 Keyframes 8

3.2.2 Speech recognition 8

3.2.3 Storing and Retrieval of Video 9

3.2.4 Video Chapters 9

4. Evaluation 11

4.1 Evaluation Strategy 11

4.1.1 Overall objective 11

4.1.2 Testing 11

4.2 Data used 12

4.3 Search Queries 12

4.4 Data Collection 12

4.5 Evaluation Metrics 12

5. Conclusion 13

6. References 13

6.1 Figures 13

Declaration

A report submitted to Dublin City University, School of Computing for module CA4009: Search Technologies, 2021/2022.

I understand that the University regards breaches of academic integrity and plagiarism as grave and serious. I have read and understood the DCU Academic Integrity and Plagiarism Policy. I accept the penalties that may be imposed should I engage in practice or practices that breach this policy. I have identified and included the source of all facts, ideas, opinions, viewpoints of others in the assignment references. Direct quotations, paraphrasing, discussion of ideas from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged and the sources cited are identified in the assignment references. I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work. By signing this form or by submitting this material online I confirm that this assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study. By signing this form or by submitting material for assessment online I confirm that I have read and understood DCU Academic Integrity and Plagiarism Policy (available at: http://www.dcu.ie/registry/examinations/index.shtml)

Name(s): Vincent Achukwu, Alan Wu, Niall Bermingham
Date: 26/11/2021

Abstract

This report details the implementation of a machine learning search system for videos. It details the use of video summarization. This process involves keyframe analysis, speech recognition and video chaptering as methods to break down footage and create a searchable database that can retrieve a user's query.

The system will be run as a web application and the user will either input a query and retrieve a relevant video or upload a video and be given a descriptive video summarization. The system is assuming that the user can input a query that relates to what they want to search for and that they will provide a video of high quality to be processed.

The backend of the system will use a server to retrieve a user query request to the database or to process the video into a format that can be stored in the database. This processed data will be sent back to the frontend web application so the user can view the results of their query. The data shown will be a list of relevant videos if the user was searching using a query. If the user had uploaded a video the end result will be a text video summarization of the video they had just uploaded.

1. Introduction

Our Search technologies project is based on video search and summarization. We will be incorporating various methods to create this application such as using video keyframes, image recognition, speech recognition and video chapters to summarize the video so that it can be identified using the methods mentioned above.
Our application will be client-server based and will provide the user with the ability to search through our database and find relevant videos based on a provided search query, they will also be able to search for frames within the video they have chosen.
To achieve this we will use a machine learning model to summarise the video and use the methods mentioned above to create a database of relevant information for each video which will become the backbone of our project.

2. User Analysis

2.1 The User

Upon using the application we are expecting that the user would first be seeking a video or is uploading a video to the service. They are required to have minimum technical knowledge about how to type in a search query or to upload the video however the user would not need any specialised knowledge on how the videos are searched or how the videos are sorted to use this application. We aim to make this application accessible to any private individual who is searching or uploading a video online.

2.2 Scenarios

When using the system we expect the user to enter a search query, the more specific the query the more specific the video will be provided to the user. For less specific queries entered into our system, the user will be provided with a mix of videos based on their popularity, viewer retention and basic search term match to try and retrieve the most relevant videos for the user. The user will also be able to upload a video that will be sorted and categorised using a machine learning model, summarised and the user will be able to include basic search terms which will help the system to search and find the video if such terms are sent as a query to search for the video. For more information about how these methods will impact the system, they will be highlighted more in the Scientific Functional Description section of this report.

2.2.1 Searching for a video

A user is searching for a video with “cute cats”. The search query is entered into the application and is sent to the search engine. The search engine finds relevant videos and requests those videos from the database and displays those videos to the user.

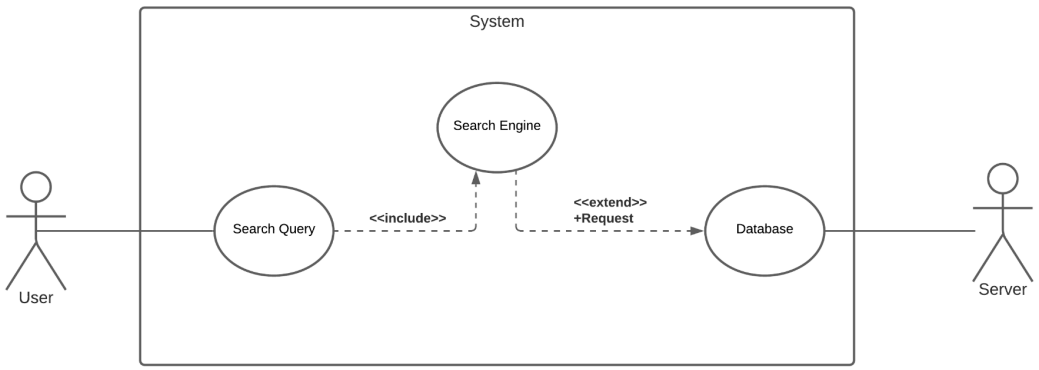

Fig 1

2.2.2 Uploading a video

A user is uploading a video with “cute dogs” as the title. The video is sent to be processed by our system to retrieve a summarization of the video using keyframes, speech recognition etc. When the video is processed by the application it is uploaded to the database and the user is given a generated summarization of the video.

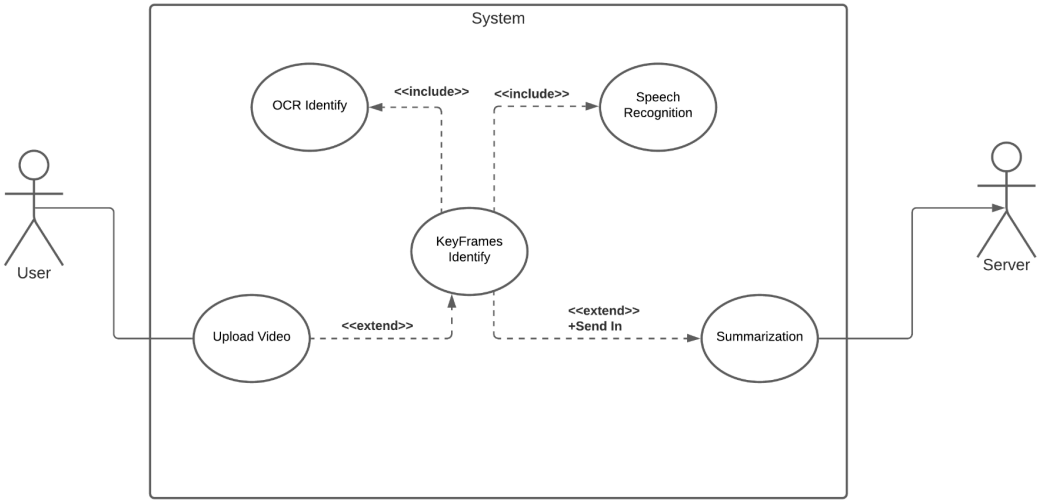

Fig 2

2.2.3 Searching for keyframes in a video

A user is looking for a short clip that contains relevant information about “TikTok memes”. The search query is entered into the application and is sent to the search engine. The search engine finds relevant videos and requests those videos from the database and displays those videos to the user. When a user has selected a video, the web application will search through the video for the most relevant keyframe relating to the search topic using the video’s summarization retrieved previously.

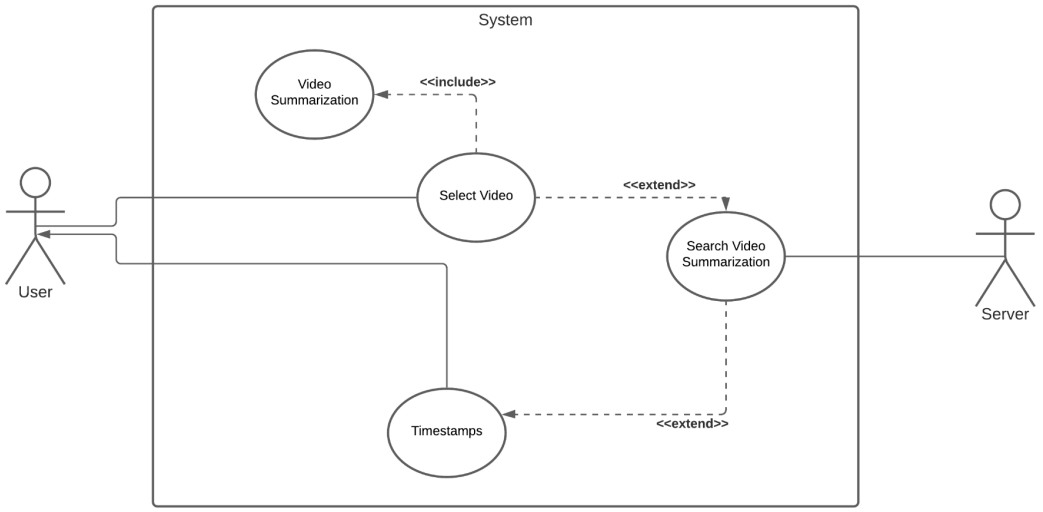

Fig 3

3. Scientific Functional Description

3.1 Overall Architecture

The plan for our video summarization system is for it to be implemented as a web application. The design will comprise a front-end and back-end system. Video summarization will be done with a machine learning algorithm via an LSTM (Long Short Term Memory) model, and this will be done by keyframes involving image recognition and OCR (Optical Character Recognition), and summarization through timestamps. The user can either search for a video and be provided with relevant videos with a descriptive summary (e.g timestamps related to the search query) or submit a video and view the video summarization for it. The machine learning algorithm and methods would be invoked on the videos after which the result is displayed to the front-end of the application. Upon uploading a video, we will use keyframes of that video which will be processed through machine learning to categorize the video that will be uploaded to the server. The retrieved data from the categorisation process will enter a database which will be used as the basis of our video search engine.

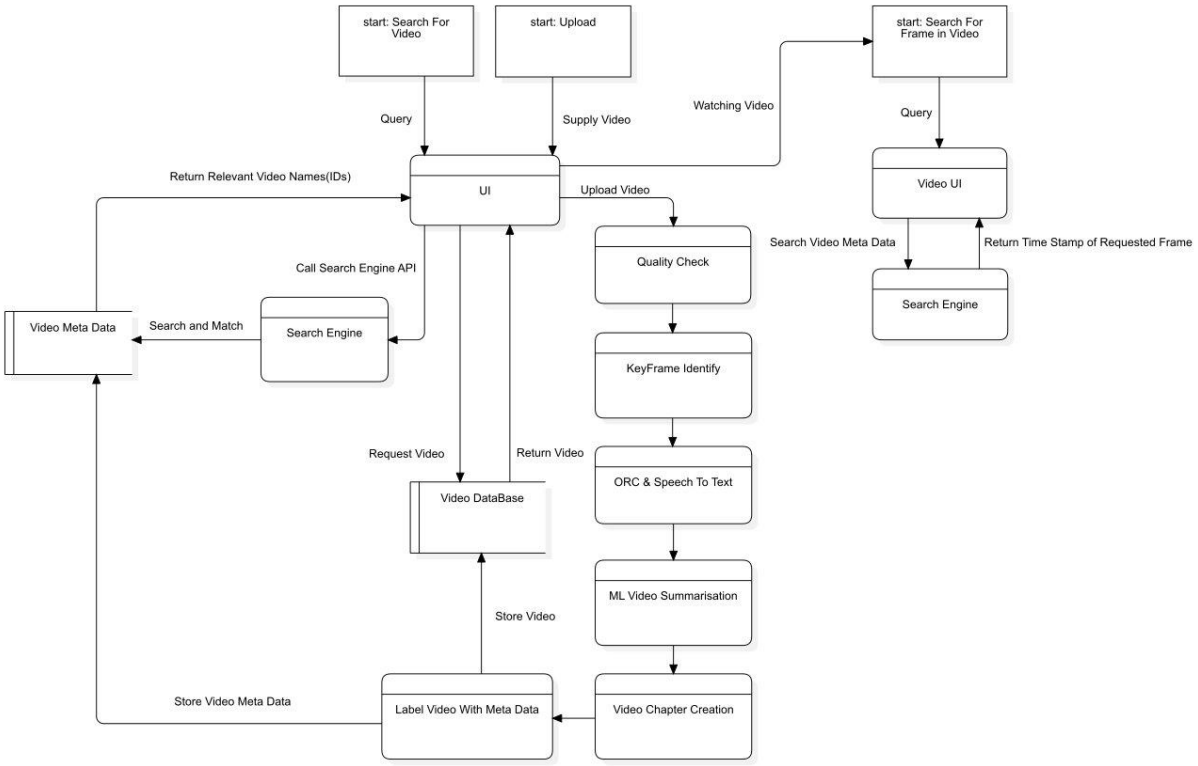

Fig 4

3.2 Algorithms

With regards to the algorithms to be used, we will mainly focus on the LSTM model for the video summarization through keyframes (which involves image recognition per frame and OCR), speech recognition (involving speech-to-text and captions), and video chapters through timestamps. Along with that, raw videos will be stored in servers, and video indexes and metadata will be stored in MySQL. We will retrieve videos using a video retrieval algorithm [1]. Overall, the video summarization process will be a computationally expensive

process as we would have to consider various attributes of videos such as size, quality, extensions, etc.

3.2.1 Keyframes

With the use of OCR and image recognition per frame, users will be able to view the summarization of videos via keyframes. When searching for a video, the LSTM algorithm will try matching the search queries with the frames within videos that are related to the search queries. The OCR process will try to determine what is present within a frame, allowing for this matching process to produce the summary [2]. Essentially, is it similar to image searching, except whatever the user is searching for will be informed of what part of the video their search query is referring to. This also relates to the idea behind how our timestamps will be produced. The use of keyframes can be done by chopping up videos by a factor of 10, with the minimum requirement of the videos on our system being 24 frames, and a maximum of 120 frames.

As for the pros and cons of this approach, the pros are that this will reduce the need of a user having to watch through the entirety of a video in order to find a specific clip within that video. For example, a 1-hour video tutorial about a math concept can be summarised by having the user search a specific topic about that concept and they'll be directed to the part of the video which discusses that topic. Another advantage of this approach is the accuracy of our Machine Learning algorithm. The cons are that the presence of video effects can cause issues when working with keyframes. This can distort the raw video quality which would impact the image recognition and OCR process. Another limitation to this approach is the efficiency. This may take a long time to process for the Machine Learning algorithm especially considering that videos can be of different quality based on framerate, file extensions, as well as video length.

3.2.2 Speech recognition

This process will make use of the speech-to-text and captions, where the audio of the videos are converted into text and the user can retrieve a textual transcript of the video. This also improves the summarization of videos by making the search queries match with what is in the captions of the video. Essentially, it would almost be like searching via video captions (on the back-end side of things), but on the user end, it's still the same process where they can either upload a video to summarise or search a video that will produce video summarizations based on the search query. This can be advantageous to those with special accessibility needs that would require watching or listening to videos with captions, hence, being able to retrieve the audio or captions transcripts of the video. Upon doing research on this methodology, this functionality will be powered by an open-source speech-to-text engine called DeepSpeech [3].

A downside to this approach is that the audio may not be in sync with the video itself, hence it can lead to inaccurate summarizations. Though this may not be that big of a problem for those only looking for a text transcript of the video transcripts (e.g music video lyrics), it can still be misleading depending on the videos being watched (as previously mentioned, tutorial videos). This point relates to the video chapters feature of our system.

3.2.3 Storing and Retrieval of Video

As for the storage of the videos, we will have raw videos stored in servers, and video indexes and metadata to be stored in the MySQL database. Our video retrieval algorithm is inspired by the paper [1]. The plan is to have the videos processed and retrieved to the user, who is searching/uploading a video, the summarizations of the videos in an effective and efficient manner. The process will involve tagging the videos and arranging them in an effective manner for fast retrieval and access [1]. Videos are composed of multimodal data such as colour, textual, motion, audio, and other visual streams. We know that the more of these forms of data exist to represent a video, the more accurate the video retrieval. However, the more of these features there are, the more complex it can get and there will be a tradeoff between video retrieval and complexity [1].

3.2.4 Video Chapters

We will also implement a video chapter mechanism with the use of video timestamps. By using keyframes and speech recognition, we'll split the video into chapters for the user to browse specific parts of the video which would relate to their query. If the user searches for a video with search queries, our system (which would have already processed the videos) will produce the relevant search results with relevant timestamps in relation to the search queries. This will also make use of the video retrieval algorithm [1]. The advantage of this approach is that it minimises the time spent browsing through videos to get to a certain part [4]. Our algorithm will process the videos with keyframes and audio and will determine where the search query matches the portion of a video. The con of this approach is that if the audio and video are not synchronized, this can lead to inaccurate results for the timestamps and will not match the search query of the user. YouTube makes use of video chapters where the uploader of the video specifies with timestamps what a specific part of the video is discussing. Our system is aiming to almost automate that by basing it on users' search queries. Since videos may already have been processed, this can speed up the video retrieval process since it won't have to be processed again given it's stored in our application system. When it comes to uploading videos, those videos will have to go through the machine learning algorithm in order to achieve video summarization.

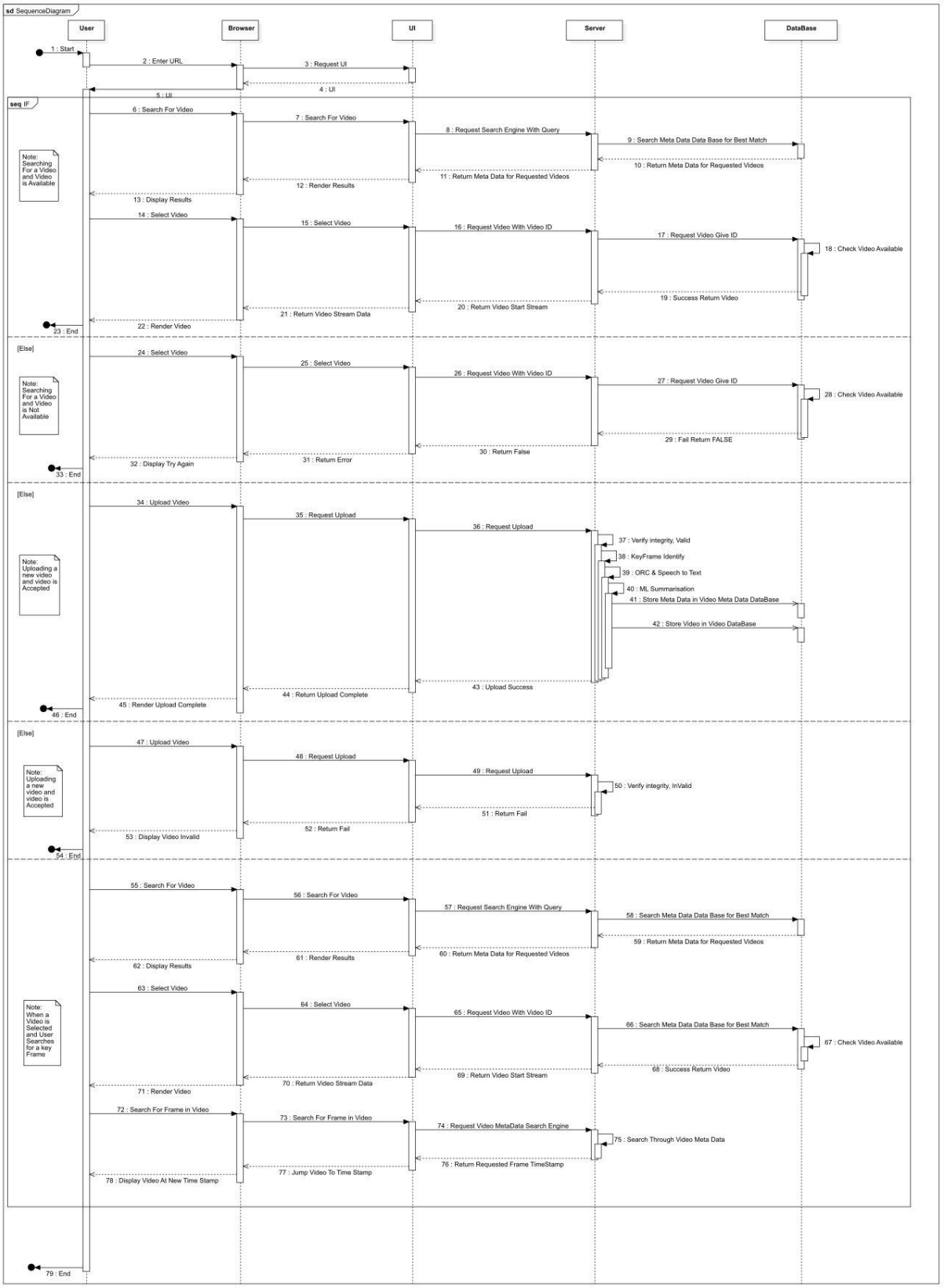
sd SequenceDiagram
User
Browser
UI
Server
DataBase
1 : Start
2 : Enter URL
3 : Request UI
4 : UI
5 : UI
seq IF
6 : Search For Video
7 : Search For Video
8 : Request Search Engine With Query
9 : Search Meta Data Data Base for Best Match
10 : Return Meta Data for Requested Videos
11 : Return Meta Data for Requested Videos
12 : Render Results
13 : Display Results
Note: Searching For a Video and Video is Available
14 : Select Video
15 : Select Video
16 : Request Video With Video ID
17 : Request Video Give ID
18 : Check Video Available
19 : Success Return Video
20 : Return Video Start Stream
21 : Return Video Stream Data
22 : Render Video
23 : End
[Else]
24 : Select Video
25 : Select Video
26 : Request Video With Video ID
27 : Request Video Give ID
28 : Check Video Available
29 : Fail Return FALSE
30 : Return False
31 : Return Error
32 : Display Try Again
Note: Searching For a Video and Video is Not Available
33 : End
[Else]
34 : Upload Video
35 : Request Upload
36 : Request Upload
37 : Verify integrity, Valid
38 : KeyFrame Identify
39 : ORC & Speech to Text
40 : ML Summarisation
41 : Store Meta Data in Video Meta Data DataBase
42 : Store Video in Video DataBase
43 : Upload Success
44 : Return Upload Complete
45 : Render Upload Complete
Note: Uploading a new video and video is Accepted
46 : End
[Else]
47 : Upload Video
48 : Request Upload
49 : Request Upload
50 : Verify integrity, InValid
51 : Return Fail
52 : Return Fail
53 : Display Video Invalid
Note: Uploading a new video and video is Accepted
54 : End
55 : Search For Video
56 : Search For Video
57 : Request Search Engine With Query
58 : Search Meta Data Data Base for Best Match
59 : Return Meta Data for Requested Videos
60 : Return Meta Data for Requested Videos
61 : Render Results
62 : Display Results
63 : Select Video
64 : Select Video
65 : Request Video With Video ID
66 : Search Meta Data Data Base for Best Match
67 : Check Video Available
68 : Success Return Video
69 : Return Video Start Stream
70 : Return Video Stream Data
71 : Render Video
Note: When a Video is Selected and User Searches for a key Frame
72 : Search For Frame in Video
73 : Search For Frame in Video
74 : Request Video MetaData Search Engine
75 : Search Through Video Meta Data
76 : Return Requested Frame TimeStamp
77 : Jump Video To Time Stamp
78 : Display Video At New Time Stamp
79 : End

Fig 5

4. Evaluation

4.1 Evaluation Strategy

There are four main areas that we can evaluate to ensure the usability of our system. These areas will be evaluated and tested manually by the team.

4.1.1 Overall objective

The overall objective of the evaluation is to test our system and find out how relevant our returned results are, and to also find and fix as many problems as possible.

4.1.2 Testing

There are 4 main parts of our system

1. Initial keyframe identifying, speech recognition and OCR
2. Summarization of videos using long short term memory ML model
3. Storing and retrieval of the video
4. Searching for a frame/scene in a video

Therefore we can broadly divide the work into four parts and we can test the system in these smaller batches.

In order to test the keyframes, we need to have a human randomly search for frames in the video and see if that frame is recorded (e.g Test if the Car Crash scene is recorded). For speech recognition, we use the same strategy and test if a certain sentence is recorded correctly. OCR testing will use the same method as the others but we do realise that OCR will require more effort to test and will be less reliable than the other two technologies.

To test the summary of a video, we will need to compare the video summary to what a human summary will be and then determine the accuracy of the summary engine.

To test our storing and retrieval system, we will need to have a list of videos already prepared in our database and try different queries to determine if the returned result is desirable.

To test our frame search, we will require someone to search up a scene in or not in a video and determine the accuracy of our search engine.

Unfortunately, there will be a substantial requirement of manual human evaluation in our system due to the nature of how it works, but we would automate any part of this process where possible.

4.2 Data used

The data that will be collected and used for our system will mainly consist of publicly available videos. Due to the nature of these types of videos, there are some things to consider when evaluating them. These videos will be completely inconsistent, this means that all our videos will be considered unstructured.

4.3 Search Queries

For this, we will start by using queries we generated on our own. By submitting these queries to our search engine, we can examine how relevant the returned results are, then we can utilise queries generated by users. By examining the returned results of these and also viewing the popularity of the videos using metrics like audience retention and click rate, we can measure how relevant these results are to the end-user.

4.4 Data Collection

We will obtain a large amount of initial video data from publically available video hosting platforms, and also with our system's inherent feature that can take user-uploaded videos.

We will keep track of audience retention, video click rate, video view time to further help us determine the success of our system.

We will collect a large number of search queries made by users of our system to help with testing.

All the videos uploaded to our application will go through an initial round of algorithms that will identify keyframes, transcribe the audio, and then summarise. These sets of data will be stored as video metadata in our metadata database.

4.5 Evaluation Metrics

There are a few metrics we need to measure to ensure the usability of our system

- Response Time
    - Testing the server and client
- Audience retention
    - Determine how long a viewer stay focused on a video
- Click rate
    - Determine how popular a video is
- Video rank relevance
    - Use audience retention and click rate
- Accuracy
    - Determine how accurate our system is by dividing how much success by the total
    - How accurate is our keyframe, speech to text, summary, returned videos

5. Conclusion

In conclusion, our system will enable users to search for a video or in a video by submitting a query. This is achieved by using technologies such as keyframe identification, speech recognition, video summarization, and storing and retrieval of videos. The user requires minimum technical knowledge to use the system. The system uses a client-server based architecture and it will be evaluated with users queries that should match related sections of video returned by the system.

6. References

[1] Patel, B. and Meshram, B., 2012. [online] *Arxiv.org*. Available at: <https://arxiv.org/ftp/arxiv/papers/1205/1205.1641.pdf> [Accessed 18 November 2021].

[2] Zhang, K., Chao, W. and Grauman, K., 2016. [online] *Link.springer.com*. Available at: <https://link.springer.com/content/pdf/10.1007%2F978-3-319-46478-7_47.pdf> [Accessed 19 November 2021].

[3] Deepspeech.readthedocs.io. 2020. *Welcome to DeepSpeech's documentation! - Mozilla DeepSpeech 0.9.3 documentation*. [online] Available at: <https://deepspeech.readthedocs.io/en/r0.9/?badge=latest> [Accessed 19 November 2021].

[4] Emad, A., Bassel, F., Refaat, M., Abdelhamed, M., Shorim, N. and AbdelRaouf, A., 2021. *Automatic Video summarization with Timestamps using natural language processing text fusion*. [online] Ieeexplore.ieee.org. Available at: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9376115> [Accessed 20 November 2021].

6.1 Figures

Fig 1 - Basic Use Case Diagram of search query scenario
Fig 2 - Basic Use Case Diagram of upload video scenario
Fig 3 - Basic Use Case Diagram of keyframe search scenario
Fig 4 - Data Flow Diagram of system
Fig 5 - Sequence Diagram of 3 use cases