B. Tech Final Year Project Report

Content Based Lecture Video Retrieval Using Video Text Information

Group Members: Arjun Gupta
Enrollment Number: 14103144
Batch: B4

Allocated Supervisor : Vikas Saxena
Panel Member 1: Shardha Porwal
Panel Member 2: Suma Dawn

1. Introduction:

General Introduction

Lecture videos are becoming ubiquitous medium for e-learning process. E-lecturing has evolved more competent popular lectures. The extent of lecture video data on the World Wide Web is increasing fastly. Therefore, a most appropriate method for retrieving video within huge lecture video library is required. The text displayed in a lecture video is closely related to the lecture content. Therefore, it provides a valuable source for indexing and retrieving lecture video contents.

Problem Statement

In the last decade e-lecturing has become more and more popular. E-lecturing has evolved more competent popular lectures. These videos consist of textual information on slides as well as in presenter's speech. The amount of lecture video data on the World Wide Web (WWW) is growing rapidly. Therefore, a most appropriate method for retrieving video within huge lecture video library is required, a more efficient method for video retrieval in WWW or within large lecture video archives is urgently needed. The objective of the system is to retrieve a video on the basis of its contents rather than retrieving video according to its title and metadata description in order to provide an accurate result for the search query.

Solution Approach

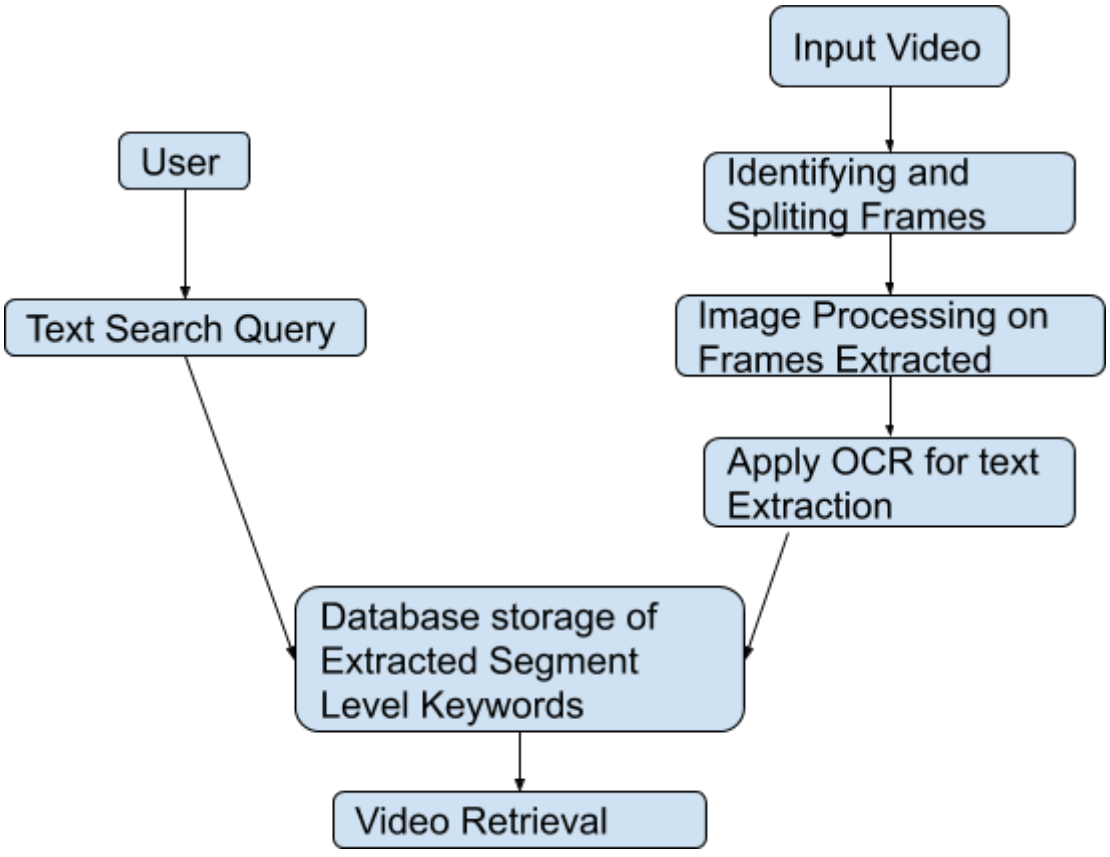
For this purpose, we have to implement a model which captures the various frames from a video lecture. Resulted captured frames are then distinguished according to the duplication property.

To implement this system, firstly we have to separate out contents on presentation slides. For mining textual information written on slides we apply OCR (optical character recognition) algorithm.Finally, we will store extracted textual results into database against particular timestamp and unique id by performing automatic video

indexing. When user will put a search query, then results will be displayed according to video contents. This technique will be beneficial for the user to search a suitable video within a short period of time.

We also propose to develop and implement a system for searching the desired video from large video database.

- Development of fast responsive system for searching related videos.
- Obtain efficient video retrieval technique.
- Reduce time required for searching desired videos.
- Creating comparatively effective search engine as compared to traditional search engines.
- Get specific videos ranked according to desired contents.



2. Literature Survey

In almost every research paper, they presented an approach for content-based lecture video indexing and retrieval in large lecture video archives. In order to verify the research hypothesis they apply visual as well as audio resource of lecture videos for extracting content-based metadata automatically. Several novel indexing features have been developed in a large lecture video portal by using those metadata and a user study has been conducted.

They extract metadata from visual as well as audio resources of lecture videos automatically by applying appropriate analysis techniques. For evaluation purposes they developed several automatic indexing functionalities in a large lecture video portal, which can guide both visually- and text-oriented users to navigate within lecture video.

For visual analysis, they propose a new method for slide video segmentation and apply video OCR to gather text metadata. Furthermore, lecture outline is extracted from OCR transcripts by using stroke width and geometric information. A more flexible search function has been developed based on the structured video text.

In order to overcome the solidity and consistency problems of a content-based video search system, they propose a keyword ranking method for multi-modal information resources. Techniques like TF-IDF have been used.

Techniques/ Tools:

- OCR(optical character recognition)
- ASR(automatic speech recognition)
- TFIDF(term frequency–inverse document frequency)
- SIFT (Scale Invariant Feature Transform)
- OPENCV,
- Python-Tesseract
- FFMPEG.

3. Analysis, Design and Modeling

3.1 Overall description of the project

The text displayed in a lecture video is closely related to the lecture content. Therefore, it provides a valuable source for indexing and retrieving lecture video contents. Textual content can be detected, extracted and analyzed automatically by video OCR (Optical Character Recognition) techniques. We present an approach for automated lecture video indexing based on video OCR technology: Firstly, we developed a novel video segmenter for an automated slide video structure analysis. Having adopted a localization and verification scheme, we perform text detection secondly. We employ SWT (stroke width transform) not only to remove false alarms from the text detection, but also to analyze the slide structure further. To recognize texts, a multi-hypotheses framework is adopted, that consists of multiple text segments.

OCR, spell checking and result merging processes. Finally, we implemented a novel algorithm for slide structure analysis and extraction by using the geometrical information of detected text lines.

3.2 Functional requirements:

Online available lecture videos involve diverse domain and different structures. The content in the lecture slides may include variety of diagrams, tables or graphical representation with various font styles.

The main functional requirements are:

- To correctly extract the textual contents present in the lecture slides.
- As video quality of online webcasts is poor and noisy, the proposed system must be capable of handling such quality issues.

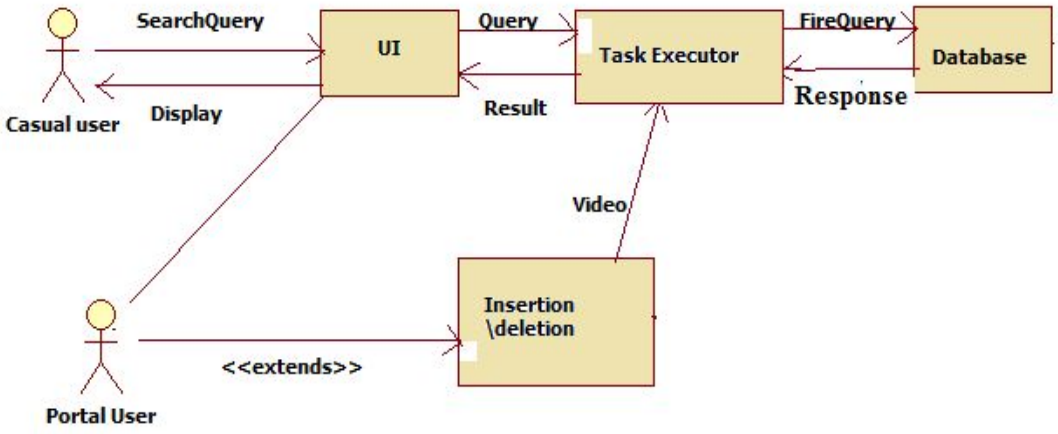
- A modular programming approach must be followed as it is suitable for adding new functional requirement and future modifications.
- Application should be robust enough to handle different video formats that are commonly available on web.
- Programming language and platform should also be chosen with keeping in mind of those mentioned facts.

Nonfunctional requirements

Some non functional requirements involve:

- Low processing time.
- Easily understandable user interface.
- Good user friendly representation.
- The proposed application should be portable to commonly available platforms and operating systems.
- As the application will deal with the long duration lecture videos, the memory requirement will not be less, but this issue should be taken cared properly so the memory usage does not cross the limit.

3.3 Overall architecture



3.4 Test Plan

For test data preparation lecture videos of different presentation styles are downloaded from YouTube for testing the performance of the developed algorithms. Ground Truth for representative key frames are prepared manually for each video and compared with experimental results. Testing of the developed algorithm for Keyframe Identification is conducted as follows:

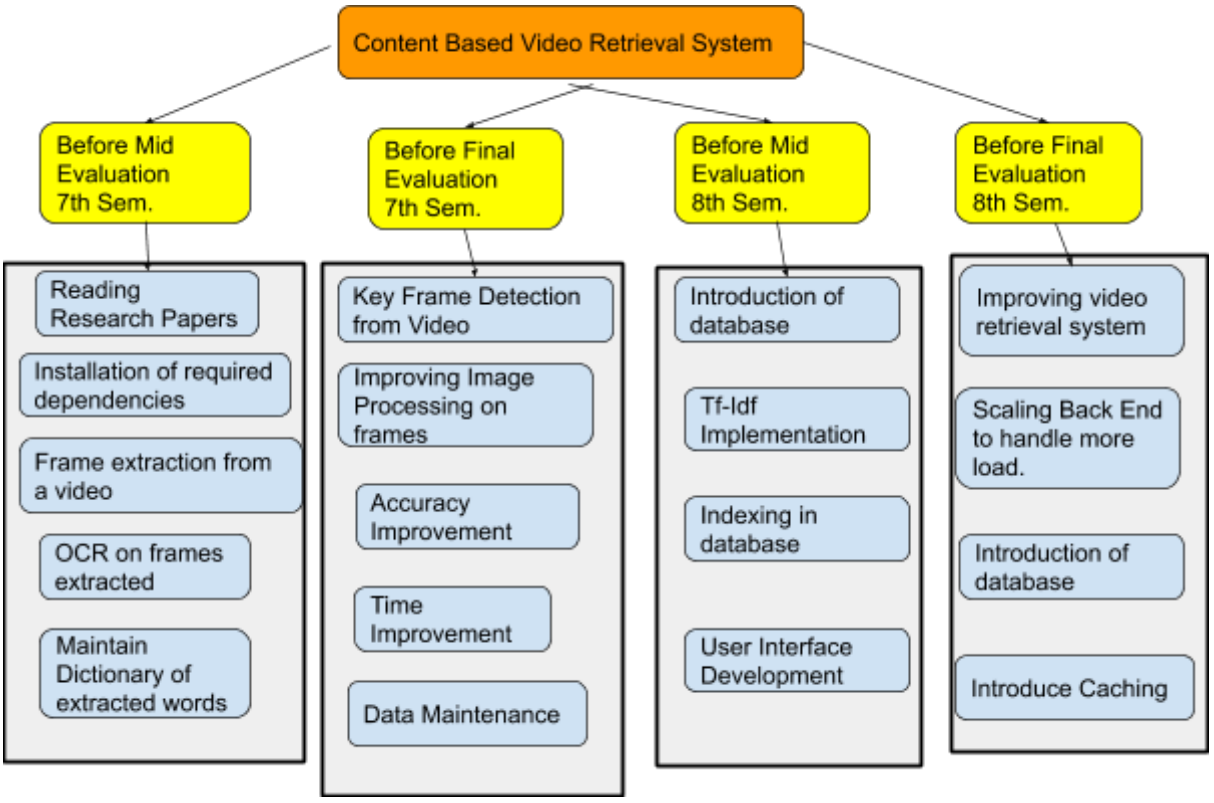
$$\textbf{Accuracy} = \frac{\text{Total number of correctly identified key frames (Tpfr)}}{\text{Total number of Key Frames from Ground Truth}}$$

$$\textbf{Precision} = \frac{Tpfr}{Tpfr + Fp} \qquad \textbf{Recall} = \frac{Tpfr}{Tpfr+Fn} * F1 \qquad \textbf{Score} = \frac{2*Precision*Recall}{Precision + Recall}$$

Fp: Number of wrongly identified frames **Fn:** number of missed key frames

Appendix

A. Work Breakdown Structure



B. Details of Tool/Technology Used

1) FFmpeg

FFmpeg is a free software project that produces libraries and programs for handling multimedia data. FFmpeg includes libavcodec, an audio/video codec library used by several other projects, libavformat (Lavf),an audio/video container mux and demux library, and the ffmpeg command line program for transcoding multimedia files. FFmpeg is published under the GNU Lesser General Public License 2.1+ or GNU General Public License 2+ (depending on which options are enabled).

The name of the project is inspired by the MPEG video standards group, together with "FF" for "fast forward".The logo uses a zigzag pattern that shows how MPEG video codecs handle entropy encoding.

Command line tools

- ffmpeg is a command-line tool that converts audio or video formats. It can also capture and encode in real-time from various hardware and software sources such as a TV capture card.
- ffserver is an HTTP and RTSP multimedia streaming server for live and recorded broadcasts. It can also be used to time shift live broadcasts.
- ffplay is a simple media player utilizing SDL and the FFmpeg libraries.
- ffprobe is a command-line tool to display media information (text, CSV, XML, JSON), see also Mediainfo.

2) Tesseract

Tesseract is an optical character recognition engine for various operating systems. It is free software, released under the Apache License, Version 2.0,and development has been sponsored by Google since 2006.In 2006 Tesseract was considered one of the most accurate open-source OCR engines then available.

Tesseract was in the top three OCR engines in terms of character accuracy in 1995. It is available for Linux, Windows and Mac OS X. However, due to limited resources it is only rigorously tested by developers under Windows and Ubuntu.

Tesseract up to and including version 2 could only accept TIFF images of simple one-column text as inputs. These early versions did not include layout analysis, and so inputting multi-columned text, images, or equations produced garbled output. Since version 3.00 Tesseract has supported output text formatting, hOCR positional information and page-layout analysis. Support for a number of new image formats was

added using the Leptonica library. Tesseract can detect whether text is monospaced or proportionally spaced.

Installing Tesseract

You can either Install Tesseract via pre-built binary package or build it from source.

Supported Compilers are:

- GCC 4.8 and above
- Clang 3.4 and above
- MSVC 2015, 2017

3) OpenCV

OpenCV (Open Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision.Originally developed by Intel, it was later supported by Willow Garage and is now maintained by Itseez. The library is cross-platform and free for use under the open-source BSD license.OpenCV supports the Deep Learning frameworks TensorFlow, Torch/PyTorch and Caffe.OpenCV is written in C++ and its primary interface is in C++, but it still retains a less comprehensive though extensive older C interface. There are bindings in Python, Java and MATLAB/OCTAVE. The API for these interfaces can be found in the online documentation. Wrappers in other languages such as C#, Perl,Ch, Haskell and Ruby have been developed to encourage adoption by a wider audience.All of the new developments and algorithms in OpenCV are now developed in the C++ interface.
OpenCV runs on a variety of platforms. Desktop: Windows, Linux, macOS, FreeBSD, NetBSD, OpenBSD; Mobile: Android, iOS, Maemo, BlackBerry 10.The user can get official releases from SourceForge or take the latest sources from GitHub.OpenCV uses CMake.

w

OpenCV (Open Source Computer Vision Library) is released under a BSD license and hence it's free for both academic and commercial use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can take advantage of multi-core processing. Enabled with OpenCL, it can take advantage of the hardware acceleration of the underlying heterogeneous compute platform.

Adopted all around the world, OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 14 million. Usage ranges from interactive art, to mines inspection, stitching maps on the web or through advanced robotics.

C. REFERENCES

[1].Haojin Yang, Christoph Meinel "Content Based Lecture Video Retrieval Using Speech and Video Text Information" IEEE, 27 February 2014
http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6750040

[2].Mr. Pradeep Chivadshetti, Kishor Sadafale , Kalpana Thakare "Content Based Video Retrieval Using Integrated Feature Extraction and Personalization of Results" IEEE
13 June 2016
http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7489372

[3]. Asha S , Sreeraj M, "Content Based Video Retrieval using SURF Descriptor" Advances in Computing and Communications (ICACC), IEEE, 19 December 2013
http://ieeexplore.ieee.org/document/6686373/

[4]. Aditi P. Sangale, Santosh R. Durugka, "Content Dependent Video Retrieval System" International Journal Of Engineering And Computer Science(IJECS) 5 May 2015
http://www.ijecs.in/issue/v4-i5/87%20ijecs.pdf

[5]. Vigneshwari.G, "A SURVEY ON CONTENT BASED LECTURING VIDEO RETRIEVAL" International Journal of Computer Science and Mobile Computing(IJCSMC), 11, November 2014
http://www.ijcsmc.com/docs/papers/November2014/V3I11201471.pdf