Video Lecture Summarization Using Machine Learning

Prajakta M. Gaikwad[1], Mansi D. Kulkarni[2], Bhagyashri R. Turankar[3], Sonam B. Borhade[4]
[1,2,3,4] *Computer Engineering, Sinhgad Institute of Technology & Science, Narhe, Savitribai Phule Pune University, Pune*

Abstract **- In today's fast-growing world of internet, we see a good number of educational institutions and professors helping the student community by uploading video lectures online. These video lectures, although extremely effective in teaching students topics from scratch, are not very helpful when it comes to quick revisions. In our attempt to video summarization, we look at summarizing the videos to help students revise these lectures quickly. We also attempt to build an in-video search, to help students in topic-wise preparation. The search will help students pick out parts of video lectures specific to the topic of focus, saving them the extra manual effort of going through the lectures in search of their topic of interest.**

Index Terms **- Video lectures, Quick revision, Video Summarization, Topic-wise preparation, Lecture transcripts.**

I.INTRODUCTION

In our attempt of video summarization, in an input video, to capture the important information of the input video we need to select a subset of the video frames or shots to generate a summary of that video. The tool to assist video search, retrieval, browsing, etc., video summarization is a useful tool in the fast & growing world and the large amount of video lectures available online.

Video summarization model consists of automatically generating video frame summaries, which are of two types- static summaries or dynamic summaries. Static summaries are series of key frames, while dynamic summaries are series of shots.

There are extensive availability and use of the digital video because of recent multimedia technology. These large collections of videos are used by various technical applications, due to which technology needs the tools that can index efficiently, search or browse and retrieve the relevant data. Video shot boundary detection is the initial step for automatic annotation of the digital video sequences. It divides the video stream into shots, which are a set of meaningful and manageable frames or segments, which are the basic elements for indexing. A shot is an uninterrupted and continuous segment in a video sequence that defines the building blocks of video content. It is comprised of a number of consecutive frames filmed with a single camera with variable durations.

Videos can be represented by a hierarchical structure consisting of several levels (frame, shot, and scene). Among the various structural levels, shot level organization has been considered appropriate for browsing and content-based retrieval.

The methodology of shot detection works on two principles.[11]

1. Scoring: The similarity/dissimilarity between the two video frames is represented by the respective pair of consecutive frames of a video.
2. Decision: All scores are calculated and evaluated then if a shot is detected if the score is considered high.

As a solution to our problem of lecture summarization, we use the combination of Video Processing, Speech Processing, Image Processing and Machine Learning. During the lecture, professors generally design their slides in such a way, that each point in the slide and appears one by one as they speak. From each split video, we extract the last image frame of that particular shot. Then we will extract audios from each video split. It is important to convert the audio to text, so that we can extract the important keywords spoken for each of the sub-topics. The slides for the lecture will be an ordered sequence of images extracted from each of the video splits along with the key phrases extracted from the text corresponding to the audio of each of the video splits.

II. MOTIVATION

Students cannot use video lectures for quick revisions. Summarizing these lectures in the form of a slideshow,

comprising of text and images will help students in the quick revisions.

We often see that professors tend to cover multiple sub-topics in a single video lecture. Consider a student, who wants to focus only on a specific sub-topic for preparation. If we can find all the video clips corresponding to that particular sub-topic and offer it to the student, nothing like it.

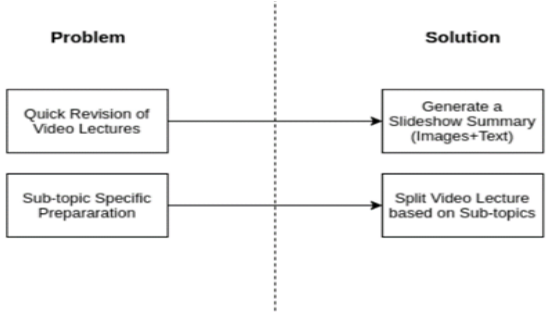

Fig 1. The problems and how to solve them

III. METHODOLOGY

Let us now discuss, overview of the theory related to the approach towards building a basic version of the intended features for video summarization.

a. Shot Detection: This method is all about looking for the positions in a video where one position is replaced by other with a discrete visual content. Shot transition detection splits up a film into basic temporary units which are called as shots.
b. Speech Recognition: It is the method where spoken language is converted to text by computers. This system requires training, where a speaker reads text or vocabulary in the system. The speakers voice is analyzed by system and the persons speech is recognized.
c. Ontology: It is process of naming and defining of the types, properties and inter relationship between the entities. It establishes relationships between the variables in the system. A good way to represents an ontology is a Graph, which we will be using for our use-case.

IV. APPROACH

We aim to give an overview of the approach towards building the intended features for video summarization, search.

1. Splitting the Video: We wish to split the video in a manner so as to separate the sub-topics taught in the Lecture. If we can capture the moment in the video where the professor moves on to the next slide, we can say that the professor has moved to the next sub-topic. It is this change in slide that we wish to detect. Hence, we split the video wherever a shot transition takes place.

An additional benefit of detecting shots is that we will also be finding out when the camera focus changes from the professor (and the teaching board) to the slides. This helps us capture content written on the black board as well. For detecting shot transitions in the video, we use the OpenCV library.

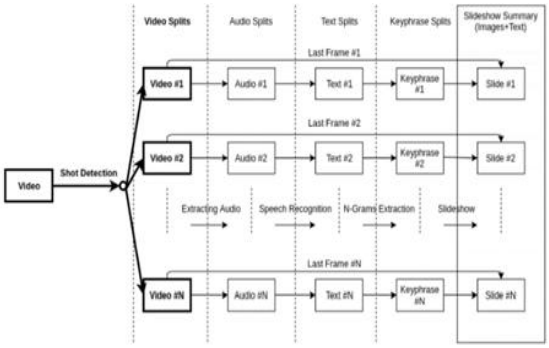

Fig 2. The First step - Splitting the video using shot detection

2. Extracting the Images: During the lecture, professors generally design their slides in such a way, that each point in the slide appears one by one as they speak. We are not really interested in these intermediate, incomplete slides. From each split video, we extract the last image frame of that particular shot. We can safely say that the last frame will not contain any of the intermediate incomplete slides, but only the complete one.

For extracting the last frame in the video, we count the total number of frames in the video, loop over all the frames until we reach the required frame using OpenCV.

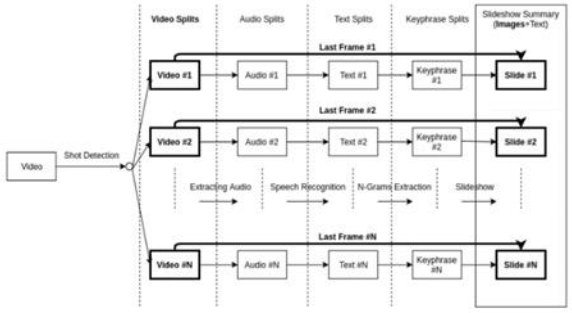

Fig 3. The Second step - Extracting the last frame of each shot

3. Extracting the Audio: We now extract audios from each video split. This is necessary to find out what the teacher is saying during the video lecture.
For separating audio from the video, we make use of online tools.

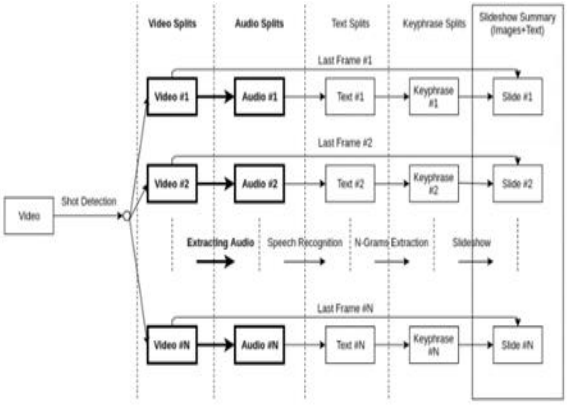

Fig 4. The Third step - Extracting the audio from each video split

4. Speech to Text: It is important to convert the audio to text, so that we can extract the important keywords Spoken for each of the sub-topics.
We use the Google Speech API for the speech to text conversion.

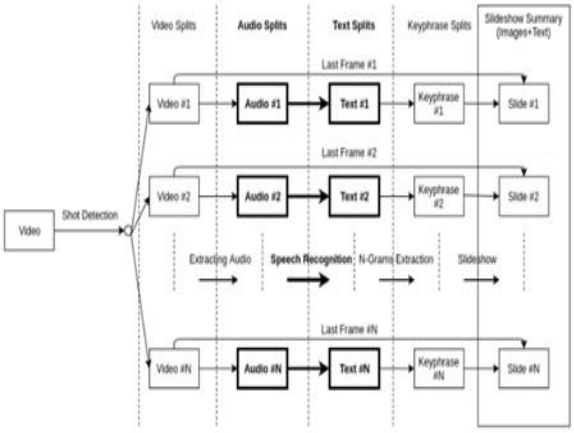

Fig 5. The Fourth step - Perform Speech Recognition on each audio split

5. Text to Key phrases: To extract the key phrases of each sub-topic, we need to do so for each text split Individually. For that, we follow these steps:

1. Convert JSON output to raw text.
2. Remove special characters, punctuations etc.
3. Remove stop words.
4. Generate all possible 1-Grams and 2-Grams,
5. And record the frequency of each of them.

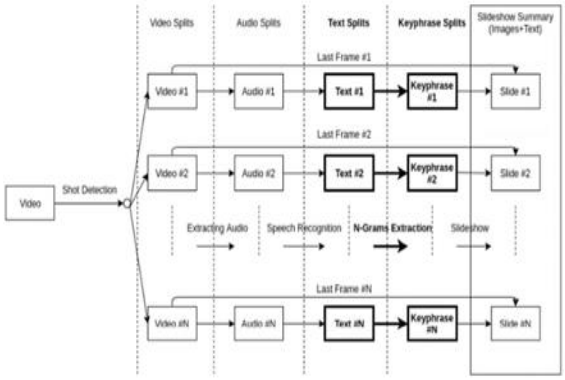

Fig 6. The Fifth step - Extracting keyphrases from each text split

6. Slideshow Summary: The slides for the lecture will be an ordered sequence of images extracted from each of the videos splits along with the key phrases extracted from the text corresponding to the audio of each of the video splits.

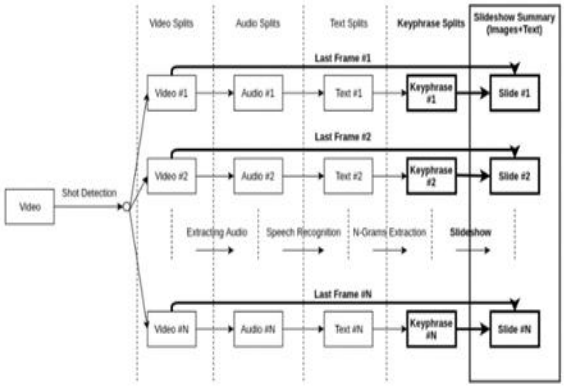

Fig 7. Approach - Slideshow summary obtained by combining image and text splits

7. Video Clip Search: For the search, we expect the user to enter a key phrase. Based on the key phrase entered, we search our repository of computed key phrases for each of the video splits. We match the key phrases and then return the search results, results solely corresponding to the user's topic of interest. On clicking on any one of the results, the particular video split will start playing. If the user feels that he needs to study the same topic from another lecture, he can simply click on the other search results to do so.
The relevance of the results and the order in which to show them is very important. For this purpose, we construct an ontology tree of all computer science topics. If the keyphrase entered is not any one of the nodes of the tree, we do a trivial search on the

keyphrases of all the video splits. However, if it is one of the topics in computer science (and hence is one of the nodes of our ontology tree), we do a specialized search. Informally, we first return the results corresponding to the node itself, followed by the results corresponding to its children, followed by the results corresponding to its immediate siblings, and finally followed by results corresponding to its parent. In each of these four phases, for each node, we calculate the frequency of its occurrence in the video split and sort the results based on the frequency.

We now depict our algorithm more formally:
Let Node K denote the node corresponding to the user's search query.

a. Search for K in the entire keyphrase splits repository. For each match found, we compute the frequency of K in each split, sort them by descending order of frequency and return the first set of results.

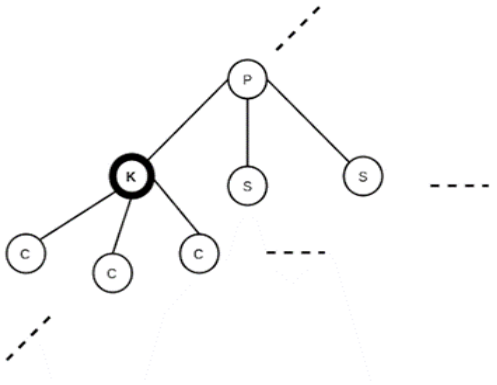

Fig 8. Search results corresponding to search query K

b. Let C be a child of K. For every such C, we search for C in the entire keyphrase splits repository. For each match found, we compute the frequency of C in each split, sort them by descending order of frequency and return the second set of results.
[Ref Fig 9]

c. Let S be a sibling of K. For every such S, we search for S in the entire keyphrase splits repository. For each match found, we compute the frequency of S in each split, sort them by descending order of frequency and return the third set of results.
[Ref Fig 10]

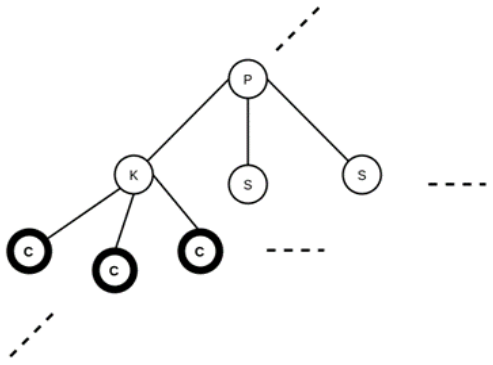

Fig 9. Search results corresponding to children C of search query K

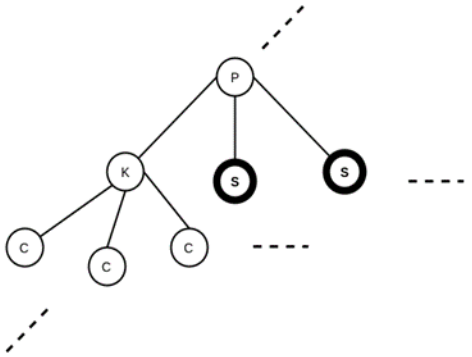

Fig 10. Search results corresponding to siblings S of search query K

d. Let P be the parent of K, We search for P in the entire keyphrase splits repository. For each match found, we compute the frequency of P in each split, sort them by descending order of frequency and return the fourth set of results.

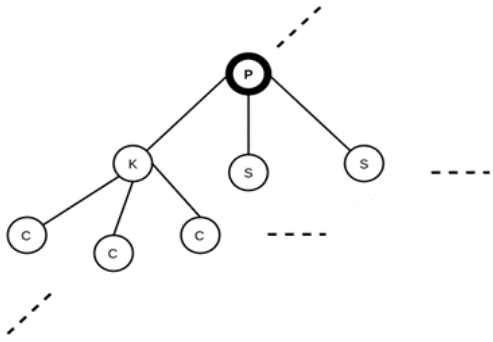

Fig 11. Search results corresponding to parent P of search query K

V. RESULTS

To calculate the accuracy of implementation we need to calculate shot detection and speech recognition accuracy.

We calculate the quality of the summarization results by assigning one of the following two labels to each obtained image slide:

1. Correct Hit: This is a valid slide.
2. False Hit: This is an invalid slide that is, slide need not be a part of the slideshow.

We also manually calculate the number of "Missed Hits" - slides which should have been a part of the slideshow but did not make it in by our method.

Shot Detection Accuracy: As discussed in the theory, we obtain the following values for C, M and F from the test video lecture using content-aware detection.

Threshold (T)	Correct Hits (C)	Missed Hits (M)	False Hits (F)	Recall (R) $C/(C+M)$	Precision (P) $C/(C+F)$	Accuracy (Q) $2PR/(P+R)$	Accuracy (%)
10	58	2	12	0.96	0.82	0.88	88%
20	58	2	8	0.96	0.87	0.91	91%
30	53	7	3	0.88	0.95	0.91	91%
40	52	8	1	0.86	0.98	0.91	91%
50	33	27	0	0.55	1.00	0.35	35%

Fig 12. Computing all the values C, M, and followed by precision and recall to finally determine the accuracy of our method.

Using the above table, we can plot a graph to show Shot Detection Accuracy graphically which is found to be 85% in average.

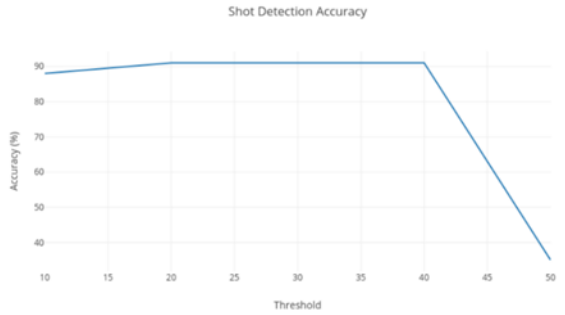

Fig 13. Variation of shot detection accuracy with different values of threshold

Speech Recognition Accuracy: The error rate for Google Speech API is 4.9%. Hence, we can say that the accuracy of our speech to text conversion is 95.1%.

Overall Accuracy: Since we run the components sequentially one after the other, we multiply the accuracy of Shot Detection and Speech Recognition to estimate the accuracy of our summary and search.

Overall Accuracy = Shot Detection Accuracy * Speech Recognition Accuracy.

Threshold (T)	Shot Detection Accuracy (Q)	Overall Accuracy (Shot Detection Accuracy * 0.95)	Overall Accuracy (%)
10	0.88	0.83	83%
20	0.91	0.87	87%
30	0.91	0.87	87%
40	0.91	0.87	87%
50	0.35	0.33	33%

Fig 14. Computing overall accuracy and comparing shot detection accuracy with overall accuracy

Using the above table, we can plot a graph to show Overall Accuracy graphically which is found to be 87% in average.

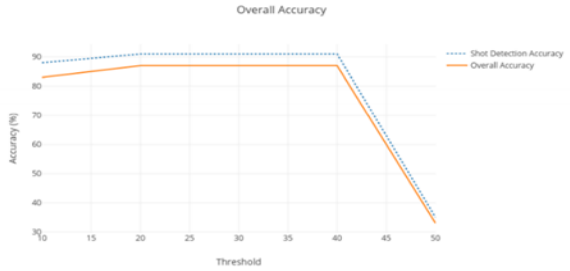

Fig 15. Variation of overall accuracy with different values of threshold and comparison with the shot detection accuracy

VI. IMPLEMENTATION

Following are some snapshots of the demo website displaying the deliverables of a summarized video.

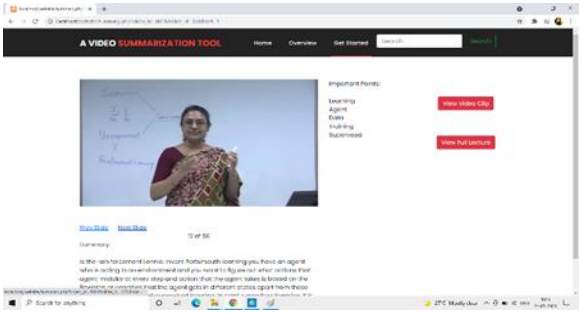

Fig 16. A captured slide may contain the contents of the lecturer's slides. One can also see that the important points are perfectly in sync with the corresponding captured slide with summary.

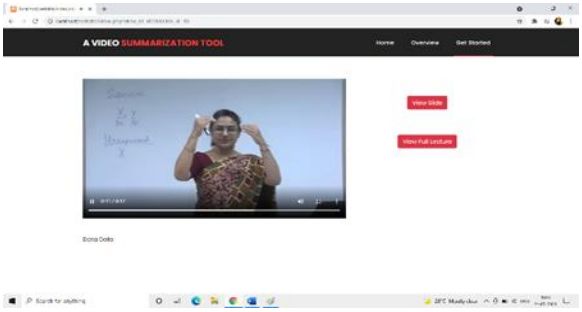

Fig 17. The video clip which opened after clicking on one of the search results from Fig 7

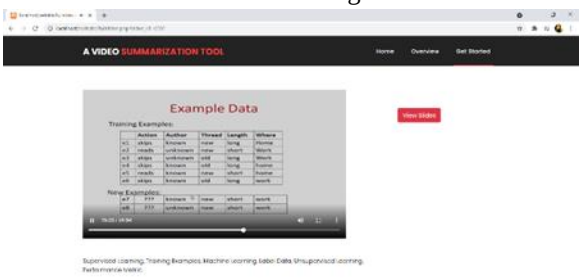

Fig 18. Full Lecture Video Clip

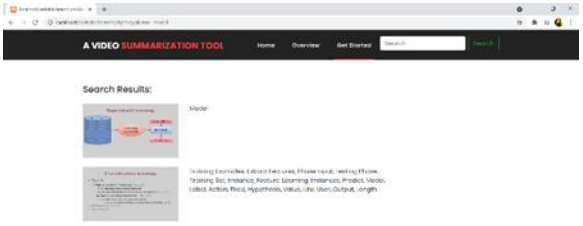

Fig 19. Searching for a topic displays the related slides and important points.

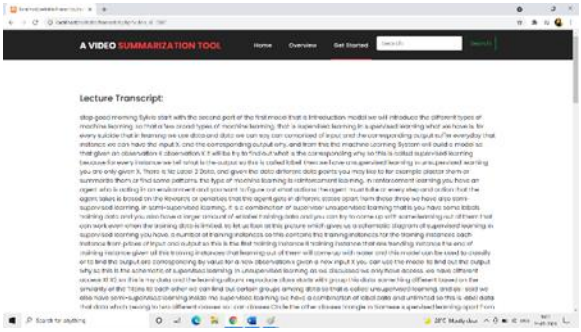

Fig 20. Video transcript generated from video lecture

VII. CONCLUSION

The Video Summarization is a laborious task which was automated with the help of Machine Learning. We studied shot detection approach which is suitable model for shot detection. The objectives to study summarization approach is fulfilled and the results are satisfactory.

The obtained accuracy is 87% using a content-aware detection. It was tested on various cases, and the result obtained was adequate and acceptable.

Looking at the tables and the graphs, we can say that the optimal threshold value using content-aware detection for our use-case lies close to 30.

For values above 30, the accuracy decreases steeply. For values less than 30, the accuracy decreases gradually.

We can conclude that the accuracy of the individual components of our design is good, however, with a huge scope for improvement.

Moreover, since we execute these components one after the other sequentially, the overall accuracy falls down, as the errors of each component get carry forwarded and affect the further components.

VIII. FUTURE WORK

To address the issues discussed in the conclusion, we aim to increase the overall accuracy of the summarization and search by:

1. Increasing the accuracy of each component.
2. Developing ways to filter and cover up for the errors getting carry forwarded from the previous components.

REFERENCE

[1] Mahesh Kini M Karthik Pai Department of Computer Science and Engineering Department of Computer Science and Engineering India. “A Survey on Video Summarization Techniques” In 2019

[2] Sandra E. F. de Avila†, Antonio da Luz Jr.†‡, Arnaldo de A. Araujo´†, and Matthieu Cord§†Computer Science Department — Federal University of Minas Gerais. “VSUMM: An Approach for Automatic Video Summarization and Quantitative Evaluation” In IEEE Conference, 2016

[3] Mrs.Poonam, S.Jadhava , Prof. Dipti S. Jadhav* a Department of Information Technology, Ramrao Adik Institute of Technology, Navi Mumbai, 400706, India. “Video Summarization using Higher Order Color Moments (VSUHCM)”

[4] Ting Yao, Tao Mei, and Yong Rui Microsoft Research, Beijing, China. “Highlight Detection with Pairwise Deep Ranking for First-Person Video Summarization”.

[5] Wei Zhang Faculty of Computer Guangdong University of Technology Guangzhou City, Guangdong Province, China “State Transition-Based for Cooperative Shot Boundary Detection”

[6] Zuzana Cernekov´a, Ioannis Pitas, Senior Member, IEEE, and Christophoros Nikou, Member, IEEE"] Information Theory-Based Shot Cut/Fade Detection and Video Summarization"
[7] Muhammad Bagus Andra Department of Computer Science and Kumamoto University Kumamoto, Japan "Automatic Lecture Video Content Summarization with Attention-based Recurrent Neural Network"
[8] Chong-Wah Ngo, Yu-Gang Jiang, Xiaoyong Wei Feng Wang, Wanlei Zhao, Hung- Khoon Tan and Xiao Wu Department of Computer Science City University of Hong Kong. "Experimenting VIREO-374: Bag-of-Visual-Words and Visual-Based Ontology for Semantic Video Indexing and Search"
[9] Qixiang Ye, Member, IEEE and David Doermann, Fellow, IEEE "Text Detection and Recognition in Imagery: A Survey"
[10] Purnendu Banerjee Society for Natural Language Technology Research Module 130, SDF Building Kolkata-700091, India. "Automatic Detection of Handwritten Texts from Video Frames of Lectures"
[11] P. Balasubramaniam; R Uthayakumar (2 March 2012). Mathematical Modelling and Scientific Computation: International Conference, ICMMSC 2012, Gandhigram, Tamil Nadu, India, March 16-18, 2012. Springer. pp. 421–. ISBN 978-3-642-28926-2.
[12] Weiming Shen; Jianming Yong; Yun Yang (18 December 2008). Computer Supported Cooperative Work in Design IV: 11th International Conference, CSCWD 2007, Melbourne, Australia, April 26-28, 2007. Revised Selected Papers. Springer Science & Business Media. pp. 100–. ISBN 978-3-540-92718-1.
[13] Breakthrough/PyScene Detect. A Python /OpenCV-based scene detection program, using threshold/content analysis on a given video. Github:Breakthrough/PySceneDetect.
[14] Online Audio Converter. Convert audio files to MP3, WAV, MP4, M4A, OGG or iPhone Ringtones. Convert MP4 to FLAC.