Content Searching In Videos

Prafulla Kasar, Ketan Gaikwad, Ganesh Khaire, Sachin Jaware, Jyoti Thakur

Department of Computer Technology Shatabdi Institute College of Engineering (Pune University) Nasik, India

ABSTRACT

The large number of video lectures within digital archives is making critical the indexing and retrieval process. Indeed, most of the systems base the indexing process on few generic text information (e.g., course title and teachers name) and this creates problems to students who are looking for very speci_c topics and hence want to browse video in details. Moreover, additional metadata could provide useful information to those users who access the educational materials by means of assistive technologies. In this paper, we propose an approach that allows students to take on-the-y notes while watching a video lecture and uses these notes to enrich video lectures with metadata that will be helpful to the indexing and re-trieval process. To extract the visual information, we apply video content analysis to detect slides and (OCR)Optical Character Recognition to obtain their text and (ASR) Automatic Speech Recognition is used to extract spoken text from the recorded audio. In this paper we present an approach for automated video indexing and video search in large lecture video archives. Firstly, we apply automatic video segmentation and key- frame detection. Then; we apply video Optical Character Recognition (OCR) technology on key-frames to extract textual metadata and Automatic Speech Recognition (ASR) on lecture audio tracks.

I. INTRODUCTION

1.1 Introduction to System

Text is a high-level semantic feature used for the content-based information retrieval. In lecture videos, texts from lecture slides serve as an outline for the lecture and are very important for understanding. So after segmenting a video _le into a set of key frames (all the unique slides with complete contents), the text detection procedure will be executed on each key frame and the extracted text objects will be further used in text recognition and slide structure analysis processes. In In the following, we present, a workow for gathering video textual information, including video segmentation/lecture slide extraction, video OCR, ASR, and keyword extraction from OCR and ASR results. We can detect the unique lecture slides by using a Connected Component (CC)-based segmentation method, The detected slide keyframes are further utilized by a video OCR engine.

II. LITERATURE SURVEY

2.1 Related work done

In the literature, di_erent methods have been proposed to index video lectures. Briey, they can be summarized into four main categories: i) image analysis, ii) speech recognition, iii) teachers annotation, and iv) students annotation. The common idea of these approaches is to add textual metadata to the video lectures and to use these metadata to describe the video contents so that it is possible to identify video results matching a given set of keywords.

2.1.1 Using image analysis

Since video lectures typically contain presentation slides or teachers notes, image analysis techniques are used to extract meta-data from appropriate portions of the video and to create an index that can be used for searching within the video. Some studies in this _eld are based on automated video segmentation and OCR analysis in lecture video indexing. Another interesting approach is based on the segmentation of lecture videos, which can be combined with features devoted to measure slides with similar content, or with the application of an OCR software for gathering textual metadata from slide streams, or for extracting lecture slide images and then applying text detection processes. All these techniques are typically di_cult to automate and time-consuming to do manually.

2.1.2 Using speech recognition

Some studies consider the audio track as a source of useful information to index a video lecture. For instance, Repp et al. proposed an indexing method for computer science courses based on their existing recorded videos. The transcriptions from a speech-recognition engine are su_cient to create a chain index for detailed browsing inside a lecture video. The index structure and the evaluation of the supplied keywords are presented. Kamabathula and Iyer proposed a system that automatically generates tags for video lectures. In particular, by using the produced audio transcription, the system takes keywords from users as a query and returns a list of videos as the results.

2.1.3 Using teachers' annotation

Teachers annotations are considered as metadata and/or tags from experts. A teacher is a designer of the educational experience, a facilitator towards active and successful learning, and as a subject expert may sca_old the learning experience . Metadata provided from teachers can provide great bene_ts to learners, since they can support them in learning content better and they can help in adequately retrieving learning content . Some projects have been done with the aim of indexing educational materials for communities of teachers, such as the ASK - LOST 2.0 system and the Cloudworks Web site , which has been created to let lecturers discuss their practices and ideas of educational design. Experts tags and metadata can be totally di_erent than tags used by students. In particular, Bateman et al. highlighted that teachers are not merely students peers; hence, their annotations may be more relevant and useful to students. However, it is worth noting that metadata and tags require a critical mass before they become useful to a community. Thus, in this context, collaborative and social tagging can provide great bene_ts.

2.1.4 Using students' annotation

Among many other retrieval techniques, users are most commonly accustomed to employ text-based queries. These textual queries must be matched somehow to the non-textual lecture videos for retrieval purpose. Therefore, it is necessary to annotate the videos with textual keywords which are highly relevant to the video content . This means that the creators of metadata need no longer be metadata experts or authors of the educational resources. Instead, the generation of metadata is done by the users of the educational resources, who can describe educational resources with tags that are meaningful to them and that can facilitate users searching and retrieval of previously used and already known educational resources. The collection of tags independently created by di_erent users can be de_ned as Folksonomy and it can constitute an alternative of the corresponding taxonomy used by the metadata experts. Social tagging of educational resources can be positively exploited in this sense, because they are typically thought to be re-used in di_erent contexts and with di_erent goals. In fact, educational resources and didactical materials are not intended and designed to be enjoyed only by their creators and by the teachers. Re-usability of didactical materials is one of the most important characteristics of learning objects. Thus, mechanisms to capture the re-contextualisation process are important so that eventually educational resources could carry users re-contextualization together with their creators ones. This can enhance both educational resources searchability (for various contexts of use) and their metadata descriptions . In fact, several studies investigate advantages coming from social tagging activities, when applied to educational resources and didactical materials. In particular, the most interesting issues investigated include the following ones:

1. Searchability: social tagging can signi_cantly improve the searchability of educational resources .
2. Metadata: social tagging can enlarge the educational resources metadata (added by creators or by the teachers), thanks to additional information provided by the social tags (added by the learners) .
3. Vocabularies: social tagging can enlarge the formal vocabulary terms (de_ned and used by the creators or by the teachers) with additional terms that can derived by the folksonomy resulting from the social tagging activities.

2.2 Existing System

The advances in networking and multimedia technologies have led to the widespread use and availability of digital video lectures. Indeed, many educational institutes use video lectures to improve the e_ectiveness of teaching in and out of classrooms and to support distance- learning students . If on the one side, this process increases the availability of educational material, on the other side, the large number of video lectures within digital archives is making critical the indexing and retrieval process . The main challenge that video lecture providers are facing is how to retrieve the appropriate video. Currently, most of the systems index video material according to few textual information like course title, teachers name and keywords, and the retrieval process provides students with a list of full- length (e.g., 60 or 90 minutes) video lectures. Since the requested information is often covered by only a few minutes of the video lecture, it is often cumbersome for students to search through an entire video, or across many videos, in order to _nd what they are looking for; the problem is exacerbated if students are not familiar with the area, or if the topic is very speci_c , or if they access by means of assistive technologies, in particular screen readers and screen magni_ers or if they need captions . Therefore, a mechanism able to _nd the proper lecture and the proper position inside a video lecture would be highly appreciated by both students and providers. The most critical challenge to build this system is the recognition of teaching topics: without these information it is not possible to browse videos is details. In literature, there are two di_erent approaches to understand the contents of a video: manual or automatic. The manual approach can provide a very accurate and detailed description, but it is time-consuming and expensive ; the automatic approach is usually based on low-level audio/video analysis, but, unfortunately, traditional image/audio analysis techniques may fail when applied to video lectures due to the heterogeneity of the material (e.g., slide-based video vs. teacher-based video, high de_nition vs. low de_nition cameras) and due the little correlation of lowlevel feature with educational videos .

III. PROBLEM STATEMENT

3.1 Problem Statement

To build this system is the recognition of teaching topics: without these information it is not possible to browse videos is details. In literature, there are two di_erent approaches to understand the contents of a video: manual or automatic.

3.2 Objectives

1. To develop an algorithm to retrieve the appropriate information in a large video archive more e_ciently
2. To the non-textual lecture videos for retrieval purpose
3. To o_er a visual guideline for the video content navigation.

IV. PROJECT REQUIREMENTS

4.1 Proposed System

In the _rst step, the entire slide video is analyzed. We try to capture every knowledge change between adjacent frames, for which we established an analysis interval of three seconds by taking both accuracy and e_ciency into account. This means that segments with a duration smaller than three seconds may be discarded in our system. Since there are very few topic segments shorter than three seconds, this setting is therefore not critical. Then we create canny edge maps for adjacent frames and build the pixel di_erential image from the edge maps. The CC analysis is subsequently performed on this di_erential image and the number of CCs is then used as a threshold for the segmentation. A new segment is captured if the number exceeds Ts1. Here we establish a relatively small Ts1 to ensure that each newly emerging knowledge point will be captured. Obviously, the _rst segmentation step is sensitive to animations and progressive build-ups. The result is thus too redundant for video indexing. Hence, the process continues with the second segmentation step based on the frames in the _rst step. In the second segmentation step the real slide transitions will be captured. The title and content region of a slide frame is _rst de_ned. We established the content distribution of commonly used slide styles by analyzing a large amount of lecture videos in our database. Let Rt and Rc denote the title and content region which account for 23 and 70 percent of the entire frame height respectively. In Rt we apply CCbased di_erencing as described above with a small threshold value of 1 for capturing the slide transitions. Here any small changes within the title region may cause a slide transition. For instance, two slides often di_er from each other in a single chapter number. If there is no di_erence found in Rt, then we try to detect the _rst and the last bounding box object in Rc vertically and perform the CC-based di_erencing within the object regions of two adjacent frames.

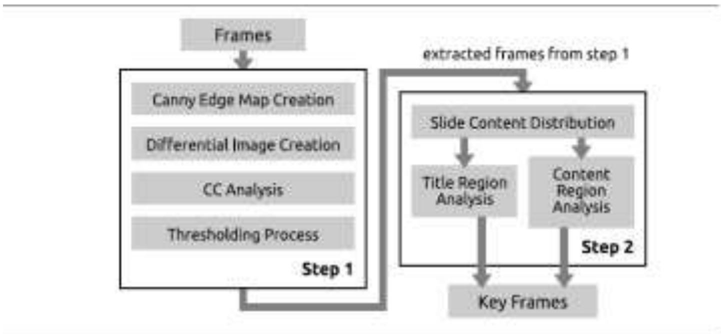

Figure 1: Block Diagram

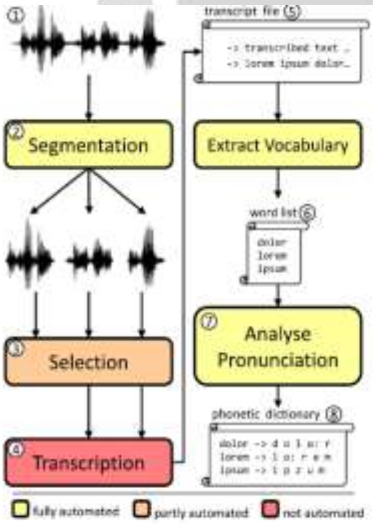

Figure 2: Working

V. RESULT

This project gives the testing results of the system. system is tested against all the test cases. Unit testing, functional testing and acceptance testing are performed under testing analysis.

5.1 Test Cases And Test Result
5.1.1 Unit Testing
In computer programming, unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures are tested to determine if they are _t for use. Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming, a unit could be an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are short code fragments created by programmers or occasionally by white box testers during the development process. Ideally, each test case is independent from the others. Substitutes such as method stubs, mock objects, fakes, and test harnesses can be used to assist testing a module in isolation. Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended. We have tested all modules individually Modules such as :

Video Segmentation
_ To test whether Segmentation is been done or not.
_ To test whether Data is been extracted or not.

Indexing
_ To test weather indexing of videos are been done properly or not.

5.1.2 Integration testing

Integration testing is a logical extension of unit testing. In its simplest form, two units that have already been tested are combined into a component and the interface between them is tested. A component, in this sense, refers to an integrated aggregate of more than one unit. In a realistic scenario, many units are combined into components, which are in turn aggregated into even larger parts of the program. The idea is to test combinations of pieces and eventually expand the process to test your modules with those of other groups. Eventually all the modules making up a process are tested together. Beyond that, if the program is composed of more than one process, they should be tested in pairs rather than all at once. Integration testing identi_es problems that occur when units are combined.
By using a test plan that requires you to test each unit and ensure the viability of each before combining units, you know that any errors discovered when combining units are likely related to the interface between units. This method reduces the number of possibilities to a far simpler level of analysis. Over all application testing is been done after integrating of modules.

5.1.3 Validation testing

Veri_cation and validation testing are two important tests which are carried out on a software before it has been handed over to the customer. The aim of both veri_cation and validation is to ensure that the product is made according to the requirements of the client, and does indeed ful_ll the intended purpose. While veri_cation is a quality control process, the quality assurance process carried out before the software is ready for release is known as validation testing. Its goal is to validate and be con_dent about the product or system, and that it ful_lls the requirements given by the customer. The acceptance of the software from the end customer is also its part. Often, testing activities are introduced early in the software development life cycle. In validation testing we had tested weather proper input video query is been entered or not. Proper video data is been extracted or not or weather improper speech is been applied to improper words.

5.1.4 GUI Testing

GUI testing is a process to test application's user interface and to detect if application is functionally correct. GUI testing involves carrying set of tasks and comparing the result of same with the expected output and ability to repeat same set of tasks multiple times with di_erent data input and same level of accuracy. GUI Testing includes how the application handles keyboard and mouse events, how di_erent GUI components like menu bars, toolbars, dialogs, buttons, edit _elds, list controls, images etc. reacts to user input and whether or not it performs in the desired manner. Implementing GUI testing for your application early in the software development cycle speeds up development improves quality and reduces risks towards the end of the cycle. GUI Testing can be performed both manually with a human tester or could be performed automatically with use of a software program. TO test whether .net and java GUI is properly managed as per ow in use case diagram. To test all controls of In GUI testing check weather .Net module GUI is been Working properly.

5.1.5 Outcome's

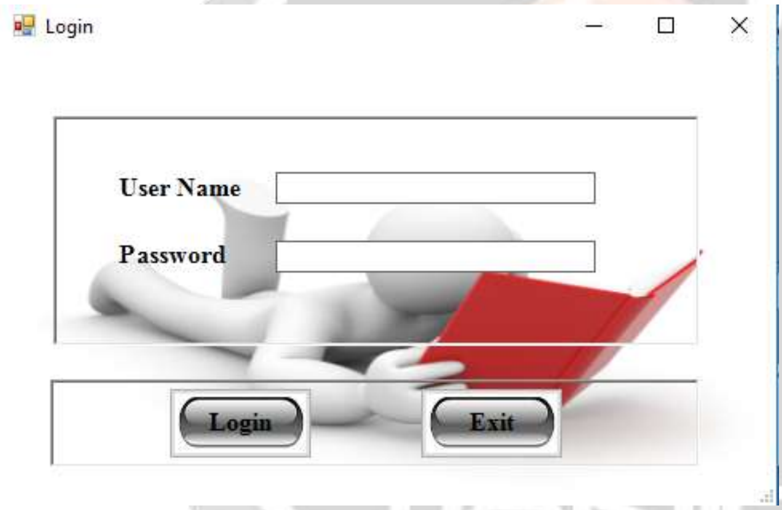

Figure: - Login in the application

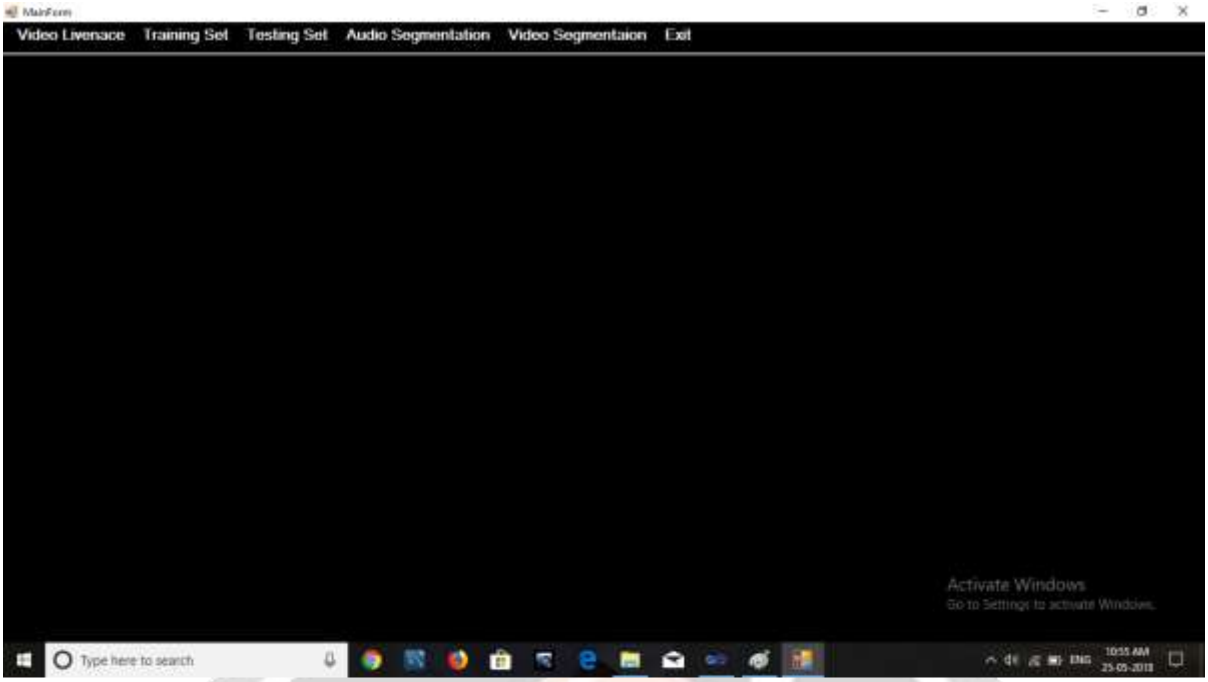

Figure: - Working in the query form

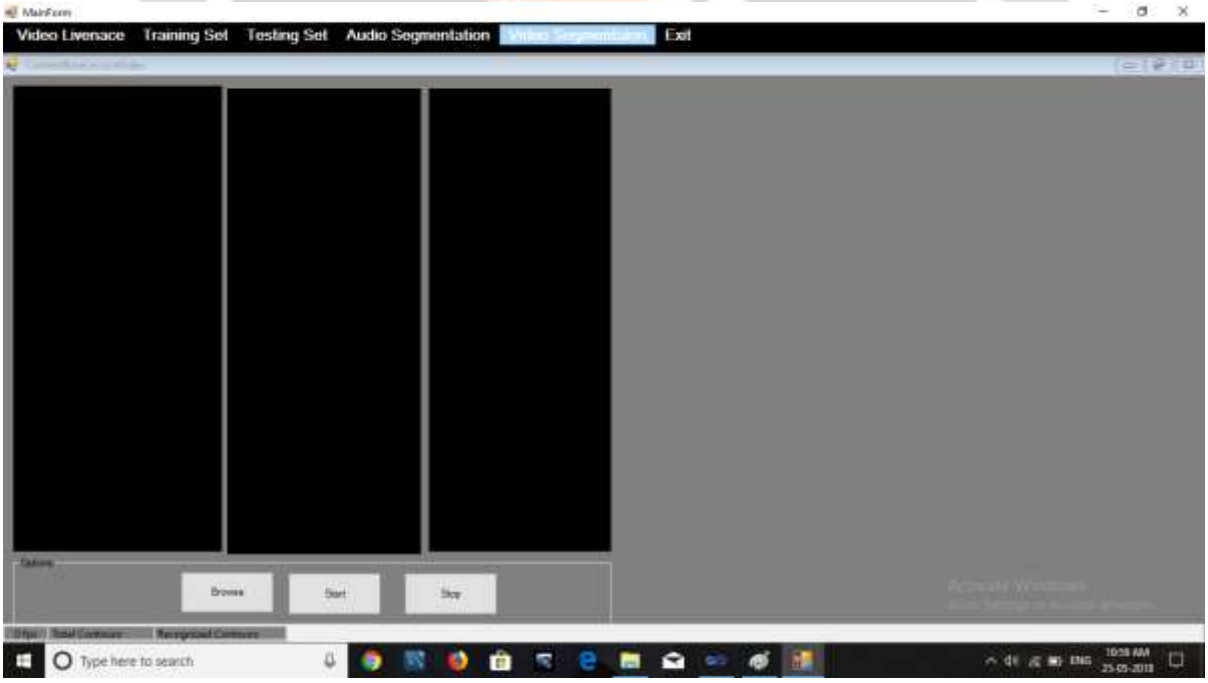

Figure: -Video gets divided in segments

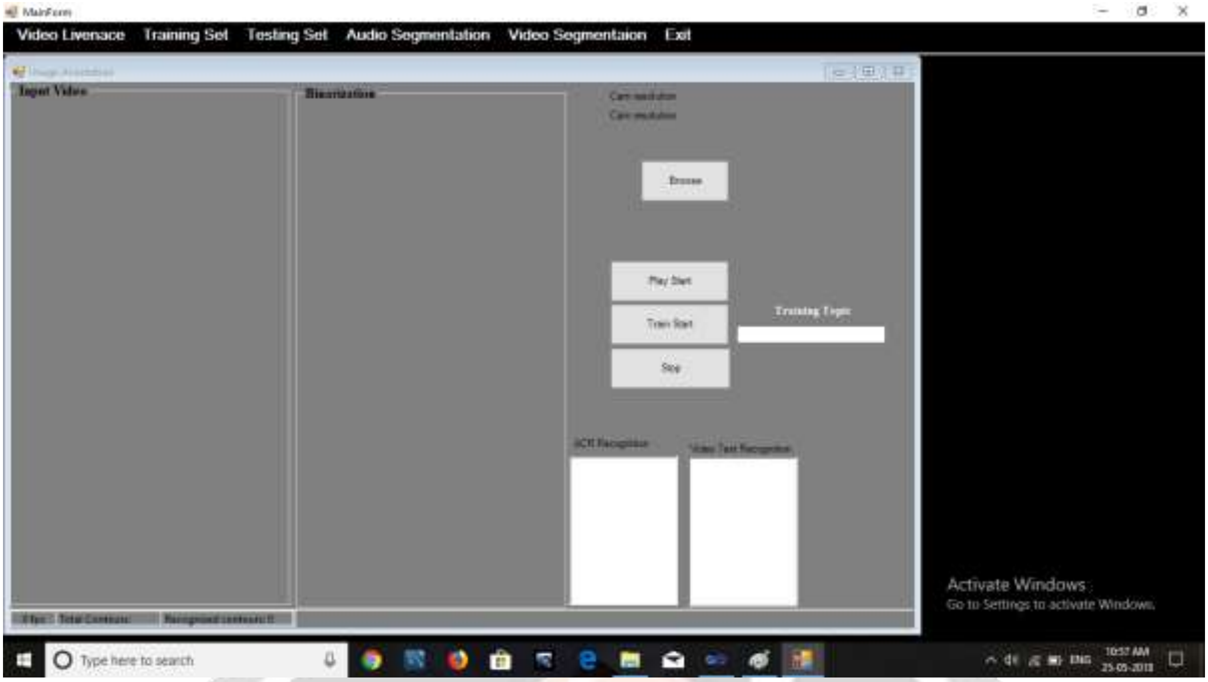

Figure: -Training of Data

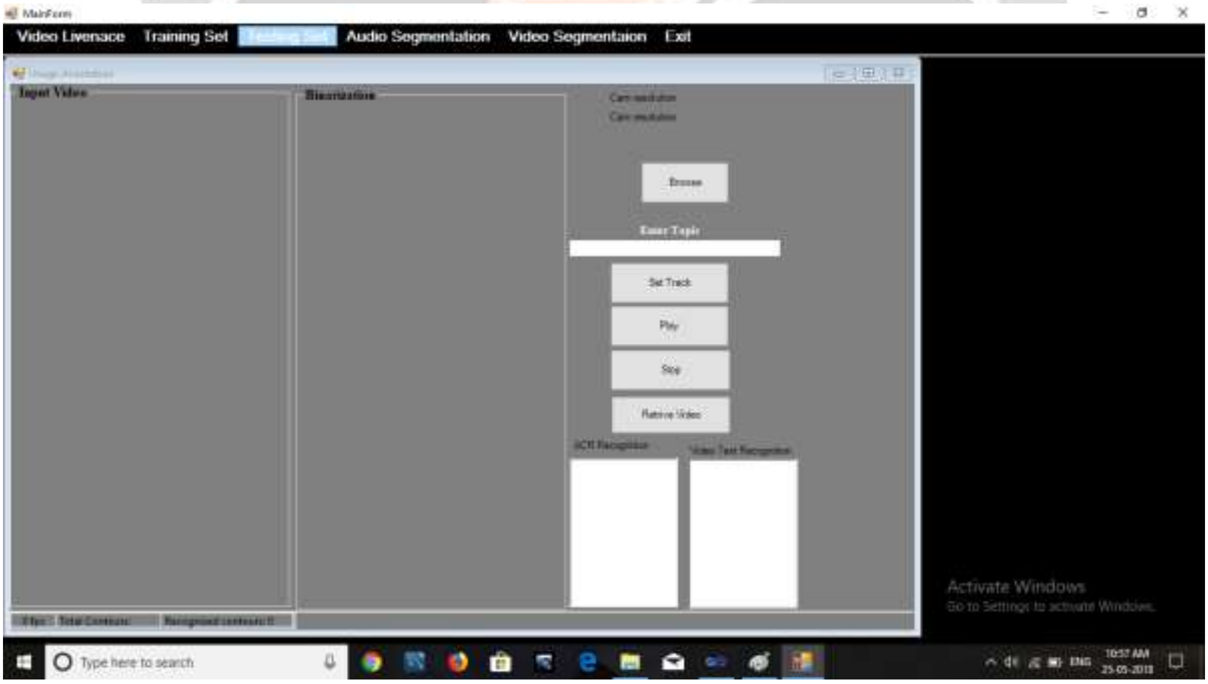

Figure: - Testing of data

VI. CONCLUSION

Motivated by the limitations of current video lecture systems, in this paper, we proposed SOLE a novel approach to index video lectures. SOLE fosters social tagging, uses a discrete notes space to categorize students notes and allows students to take their notes on-the-y while watching a video lecture. To evaluate SOLE at this early stage, we developed a prototype version and we asked for volunteer participants to test the proposal.

According to the results obtained in the MOS evaluation, students appreciated both the SOLE prototype and the idea of using notes to index and to browse video lectures.

VII. REFERENCES

1. P.Geetha, Vasumathi Narayanan. An Effective Video Search Re-Ranking for Content Based Video Retrieval. IEEE 2011; 55-60.
2. Takahiro Yoshida, Kazuki tada, Seiichiro Hagai. A Keyword Accessible Lecture Video Player and Its Evaluation. IEEE 2003; 610-614.
3. Wen-Hsuan Chang, Jie-Chi Yang, Yu-Chieh Wu. A Keyword-based Video Summarization Learning Platform with Multimodal Surrogates.
IEEE 2011; 37-41.
4. Microsoft Project Tuva, http://research.microsoft.com/apps/tools/tuva/.
5. Word stream http://www.wordstream.com/blog/ws/2012/12/10/three-types-of-search-queries.
6. Marco Furini, "On gamifying the transcription of digital video lectures", Entertainment Computing, 2015.
7. A.S. Imran, L. Rahadianti, F.A. Cheikh, S.Y. Yayilgan, "Semantic tags for lecture videos", Semantic Computing (ICSC) 2012 IEEE Sixth In- ternational Conference on, pp. 117-120, Sept 2012.
8. V.K. Kamabathula, S. Iyer, "Automated tagging to enable _ne-grained browsing of lecture videos", Technology for Education (T4E) 2011 IEEE international Conference on, pp. 96-102, July 2011.
6. Stefano Ferretti, Silvia Mirri, Marco Roccetti, Paola Salomoni, "Notes for a collaboration: On the design of a wiki-type educational video lec-ture annotation system", International Conference on Semantic Com- puting (ICSC 2007), pp. 651-656, 2007.