SlideThruLectures : Generating Transcript Slides from Online Lecture Videos using Knowledge based AI and Computer Vision

Imroze Aslam Malik
Georgia Institute of Technology
Atlanta, USA
imalik8@gatech.edu

ABSTRACT
This paper describes a software that takes video lectures and transcript text from online courses as input and uses a novel algorithm, based on image processing and knowledge based AI, to generate transcript slides with images and text extracted from the input. It performs automated note-taking to facilitate easy and efficient learning, revision, scanning and analysis of course content by providing a way to quickly skim through important information in the videos. The algorithm uses a cognitive architecture with knowledge representations and production rules based on tracking temporal behavior of edges of visual content in a grid to deal with the problem of finding all important video frames to extract, while dealing with challenges like occlusions, redundancy, dynamic content and continuous visual information build-ups etc.

Author Keywords
Online Courses; MOOCs; Video Summarization; Key-Frame Detection; Visual Tracking; Finite State Machine; Scripts; Rule-based Reasoning.

INTRODUCTION
Online courses and MOOCs have become a popular and useful source of learning by offering the best quality learning material with flexibility of watching it anywhere and anytime. The content of these online courses is mostly, just in the form of videos and hence requires playing the videos in order to find specific information or analyze the course content. There is important and note-worthy visual information in particular frames of videos as well as important statements in the audio. Retention of knowledge after a single lecture is mostly low [1] and technical details are likely to be forgotten after a while, so students often take notes while watching, to save the content in a well summarized, compressed and easily accessible format for revising the concepts later when they are required for projects, application and exams etc.

Manual note-taking requires significant amount of time which leads to lagging behind in courses and stress in case of deadlines based on content of those lectures. The note-taking is often fast and doesn't involve much active cognition and it also disturbs the focus and attention on lecture due to cognitive workload [1].

It is often hard to effectively note-down visual information in videos as there are images, figures, complex equations and graphs etc. which are hard to manually reproduce. The legibility of notes is another issue that makes it harder to take advantage of shared notes.

Apart from the ineffectiveness of videos in quickly search and retrieval of summarized and important information, permanently storing all the videos consumes significant space while viewing them online costs bandwidth.

Slides are also common and useful tool delivering tutorials and lectures so that can also be a use for slides extracted from online instructional videos.

Having a well summarized visual representation in the form of a well illustrated document is not only useful for revision after watching videos, but also for quicker learning of material before or without watching them, or to analyze and compare the key aspects of content of online courses before taking them and watching the videos.

This paper presents an open-source software tool named SlideThruLectures which is intended to solve these problem and help students with their online learning and retention of knowledge. It uses image processing on video lecture files and text parsing on transcript files to extract important frames in the video that should be captured as a whole, and time-stamped pieces of caption text. That information is used to generate slides with images captured from video. The caption text can also be broken down and embedded as text slides between image slides at the right places, to store the spoken information as well. This can help in detailed reading as well as facilitating indexing and content search within the document.

The algorithm used by the system was designed in an attempt to come up with a simplistic yet robust and efficient method that could utilize knowledge and rules about the characteristics and behavior of visual content that could help in identification of the most optimal frames to capture while dealing with some challenges and constraints. The algorithm is based on edge similarity based object tracking, rule based reasoning and a knowledge representation similar to scripts and finite state machine.

PROBLEM STATEMENT

SlideThruLectures is fairly robust for different types of instructional videos but it focuses on the modern and common digitally produced instructional videos on MOOC platforms like Udacity, Coursera and edX etc.

Those videos have digital visual content being constructed sequentially on a background while there can be overlayed instructors on that background delivering the lecture and digitally writing down content in some cases too. There may not be useful information and just instructors in some frame sequences. Embedded dynamic content and videos are less common but present. The camera is fixed and content visuals are cleaner.

The problem under consideration is different as compared to general purpose video summarization or simple key-frame extraction where the methods are based on big global changes in rich and colorful random scenes.

Following are the requirements, constraints and challenges for capturing the frames:

- Content build-ups should be handled properly. There are often multiple points, sentences, equations or visual components in a slide or build-up and they appear sequentially instead of appearing at once. They should be captured only when all of them are present.
- There can be written content that is written continuously by instructor.
- Instructor's body can cause occlusion and hide the content. It is common for instructor's hand that appears to write on background.
- Instructor's body is not static and changes the overall content of screen continuously.
- There can be dynamic content and embedded videos which change the overall visual content and not all frames in them are important.
- Visual content may fade and appear slowly instead of appearing and disappearing at once.
- There are tiny deformities in edges, even in digital content due to shading and imperfect edge detection so object outlines don't stay fully identical.
- Decisions about capturing current or past frames can depend on future frames.
- Assumptions like having a fixed background, board, paper, slide region and slideshow etc. won't hold.
- It is very important, not to miss capturing frames with important information while adding frames with same, redundant or incomplete incomplete information is bad too. There should a proper way to keep balance and avoid both of these cases.

RELATED WORK

There is apparently no commercial or open-source software available to do the task under consideration. The problem to be solved is related to video summarization and key-frame detection but it is different and more challenging as compared to simple video summarization as discussed above. There are several products [2,3,4,5] for video summary and video synopsis but they are concerned with high level distinct scene detection for applications like surveillance.

There are many research papers on video summarization and key-frame extraction in general and several paper dealing with information extraction from instructional videos as well. They have their own assumptions, limitations, trade-offs and input/output formats.

The paper [6] deals only with black-board style lectures where all text and visual content is drawn. It uses stroke tracking, dynamic programming, line breaking algorithm variant and temporal difference from transcript and makes document with drawn visual content outline between the transcript text.

The approach in [7] uses SIFT features difference between frames to detect slide transition. Threshold is based on mean and standard deviation and recursive interval pruning is used to decrease the high computation cost. It assumes that a bright, rectangular slides region/projection will be present and SIFT is not free to use.

The method in [8] uses EMD and Canny Edge difference to get shot boundaries, then classifies shot as slide or non-slide based on color descriptors, text, face and LLC and then edge difference thresholding to merge frames that build up to form a slide. It focuses on slide frame and region classification and assumes that there is a slideshow being presented and video shifts from presenter lecture to complete frames as slide images.

The paper [9] uses DCT and morphology based OCR and text detection to extract only lines of text from slides to generate transcript and ignores other visual information.

The approach in [10] deals specifically with chalk board lectures where board is erased after filling up, and uses mid-level feature of text and figures on board and extracts the written content at appropriate time using statistical modelling, clustering, content fluctuation curve, shifting window and Hausdorff-distance.

The paper [11] is specifically for handwritten slide scene type in which content is written and pages are changed and uses rule-based reasoning based on ink pixel count.

The method in [12] assumes a fixed background and slides being flipped and replaced. It uses background template, DCT based caption detection and background energy and caption energy based frame similarity to detect slide transition.

The approach in [13] uses statistical modelling of distribution of content pixels to extract written content

while dealing with instructor occlusions but it assumes a fixed color board and instructor writing on it.

SlideThruVideos aimed to make an easy to use open-source software with a useful output format while using an algorithm that has a more robust, simple and generic approach to deal with limitations and assumptions of existing methods, specially for the modern digitally produced MOOC videos.

METHODOLOGY

Main Idea

The algorithm in SlideThruLectures is based on creating a cognitive agent for tracking visual components in lecture videos while exploiting procedural knowledge about some generally true assumptions about their behavior to deal with the challenges and observing a sequence of expected events to detect optimal video frame to capture.

The algorithm utilizes the following assumptions:

- Useful content appearing on screen stays there for a time long enough to note or at least read it.
- Edges of Instructors and moving things are not fully stable and identical over time intervals in seconds.
- Disappearance of existing useful content on screen is a useful and better indicator of slide transition as compared to new part of data being added.
- Majority of content is digital data, which is easier to detect and has consistent outlines with no displacement.
- It takes around 3 sec of screen time including reappearances to establish the fact that a useful visual content has appeared and it is staying there unlike fast moving object or short lived noise.
- It takes around 4 sec of screen time to establish the fact that a useful visual content has faded away or disappeared and it was not just temporarily occluded.
- Good time to capture a frame is around 1 sec before important consistent content starts to vanish permanently.

The basic idea behind the approach used in SlideThruLectures is: “save video frame just before something that stayed almost identical and stable long enough on screen is about to permanently disappear”.

Algorithm

The algorithm has following high level steps:

- Video frames are sampled at 1 fps to decrease computation time and detect changes in visual content over time intervals.
- The images are divided into a grid of R*C cells based on given resize width and number of columns C. A bin is created for each cell to facilitate localized similarity comparison while reducing computational complexity.
- Canny Edge Detection [14] is used on the sampled images and region of interest (ROIs) of edge image are extracted corresponding to cell boundaries.
- Object outline regions in video frames are represented as concept of Frames [15] from Knowledge based AI (KBAI), which is a knowledge representation structure with slots and fillers similar to fields in classes. These frames contain pixel locations of object edges, flag for presence of object in current frame, information about appearance, disappearance time and two timers representing the current state. These objects actually represent an instance of edge ROI points at a particular time. Theses structural frames would be referred to as ROI Frames to avoid confusing them with video frames.

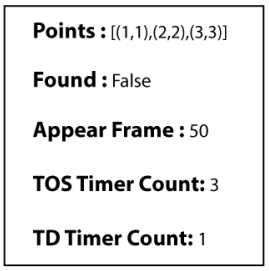

Figure 1. Example of Edge ROI Frame

- The algorithm is based on tracking the ROI Frames over time. They are matched against same cell’s edge ROIs in current video frame by using a threshold on percentage of Frame ROI points overlapping with current video frames Edge ROI. This type of similarity measure represents checking whether the outline represented by edges in one ROI is present in edges of another ROI, making it suitable for our task’s tracking requirement as deformities over time, in edges of digital content are not in the form of displacements. We define an operator F(A,B) to check whether outline in A, the set of edge points in an edge ROI, is found in another set B.

$$F(A,B) = \frac{|W(A) \cap W(B)|}{|W(A)|} * 100$$

where W(X) is set of white pixels in ROI edge points set X.

- There are two timers representing the state of ROI Frames named Timer On-Screen (TOS) and Timer Decay (TD) representing the number of video frames passed in which an Edge ROI was on screen and number of video frames passed since which the Edge ROI has been vanishing/decaying respectively.

- The sequence of events to detect optimal frame to capture is based on the concept of Scripts [15] in KBAI, which are sequence of frames to facilitate forming and detecting sequence of events as stories understanding. They allow instantiation of stories and generation of expectations based on them which can help in detecting if our hypothesis about events pattern is true by checking if the expectation breaks. In our algorithm we have scripts about appearance and disappearance of visual content to check if something that stayed long enough has actually disappeared and not just occluded, and the new objects are not just temporary noise. These scripts expect sequence of continuous appearance and disappearance and if that expectation is not met then hypothesis break and state of ROI Frames is reset.
- The sequence of events and behaviors of Edge ROI Frames, in our key-frame detection can also be viewed as a Finite State Machine (FSM) model that transitions between a set of finite states based on current state and events. This FSM and underlying Scripts are controlled by a set of rules similar to rule based reasoning based production system rules [15] in KBAI which map percepts to actions. The percepts are ROI Frame timer states and actions are ROI Frame/FSM state updates, adding and removing new ROI Frames and detection of frame to capture.
- New Edge ROI Frames are generated and added to corresponding bin by using a threshold on F(C,A) where C is edge ROI of a cell in current video frame and A is set resulting from union of all edge ROIs in that cell. New ROI Frame is added to a cell bin only if there is no established (TOS=4) non-decaying (TD>0) ROI Frame in that cell bin in order to avoid useless new Frames which decrease speed. Establishment of one ROI Frame is enough to consider cell to have useful visual content. When adding new Frames it also ensured that the number of points in Edge ROI are above a threshold, in order to avoid noise and use only rich visual content areas for decision making.

$$F(C, \bigcup_{s \in S} s) > T1$$
$$|\nexists (TOS(N) = 3 \wedge TD(N) = 0 \; \forall \; N \in S) \wedge |C| > T2$$

  where T1 and T2 are thresholds for match finding and minimum number of points in ROI respectively and S is the set of all ROI Frames in the cell bin.
- Checking for presence of an existing ROI Frame in new video frame is done by using a threshold on F(A,C) where A is set of points in an ROI Frame to find and C is the set of edge ROI points in same cell in current video frame. This check assigns value to found slot in the ROI Frame. If it is found in new video frame, then its TOS timer is incremented and TD timer is reset. Otherwise its TD timer is incremented if TOS is above a threshold of 3 to consider it "established" and the Frame ROI is prematurely removed if that's not the case. These prematurely removed Frame ROI are highly dynamic visual content like instructor bodies or temporary noise and fadings.
- If any existing Frame ROI's TD timer goes above a threshold and it was established by staying on screen for enough time, then it is detected as disappearance of note-worthy content build-up and the optimal frame to capture is sampled 1 sec before the time of start of vanishing of that Frame ROI. Another additional layer of filtering is applied to avoid capturing redundant and almost identical frames by getting the Mean Squared Error (MSE) of edge image of video frame under consideration for extraction and the edge image of last optimal key-frame that was detected and extracted.
- All the FSMs/ROI Frames with time of birth before the disappearing build-up's ROI Frames and ROI Frames that are established and decaying are removed from bins, as their information was present in build-up and captured. This serves as a reset for new scene.
- The .srt transcript file is parsed and caption text is grouped based on time-stamps and added at right place between the captured images in slides, if the option for using transcript is selected.

Following diagrams shows the production rules for rule based reasoning and their alternate representation in the form of an FSM. The Edge Frame ROIs are represented as randomly placed circles in corresponding cells. The boundary color of circles represents color coding for TOS while inner color represents TD. This helps in visualization and debugging.

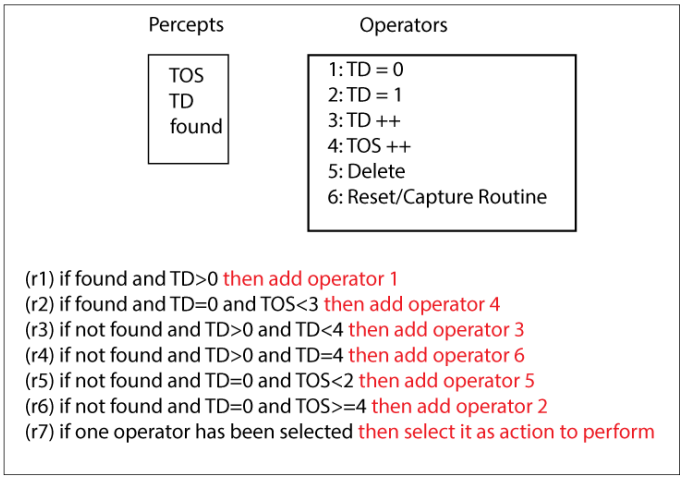

Figure 2. Production Rules for ROI Frame State Update

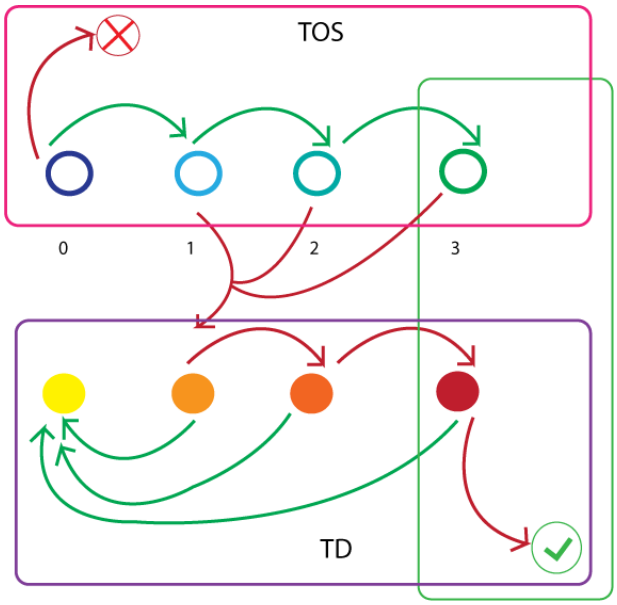

Figure 3. Finite State Machine Representation of the Timers of Edge Frame ROIs

The rules are based on sequence of Frame states that form the Script of a visual content outline being dynamic or noise, and Script of note-worthy established visual content permanently disappearing from screen.

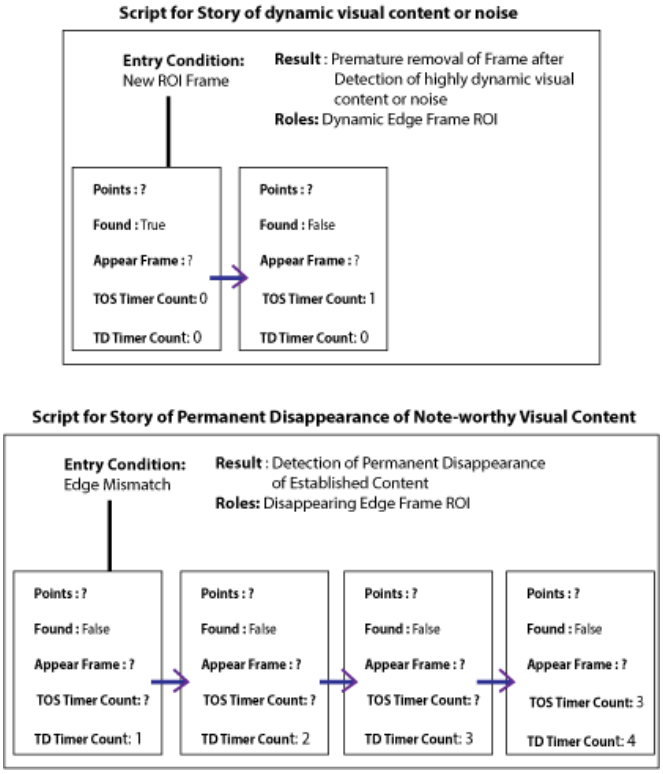

Figure 4. Scripts

System Architecture and Features

SlideThruLectures is an open-source software developed using Python 3. It uses OpenCV 3, NumPy, PIL and imutils to help in implementation of the algorithms and PyPPTX to generate slides from lectures. The GUI was developed using tkinter and consists of a single screen.

Following are three features supported by SlideThruLectures:

1. Generating PPT Slides from a single lecture video.
2. Generating PPT Slides from a lesson/course folder containing multiple lectures videos by concatenating the slides generated by each video.
3. Parsing .srt Transcript files and adding caption text between images in slides.

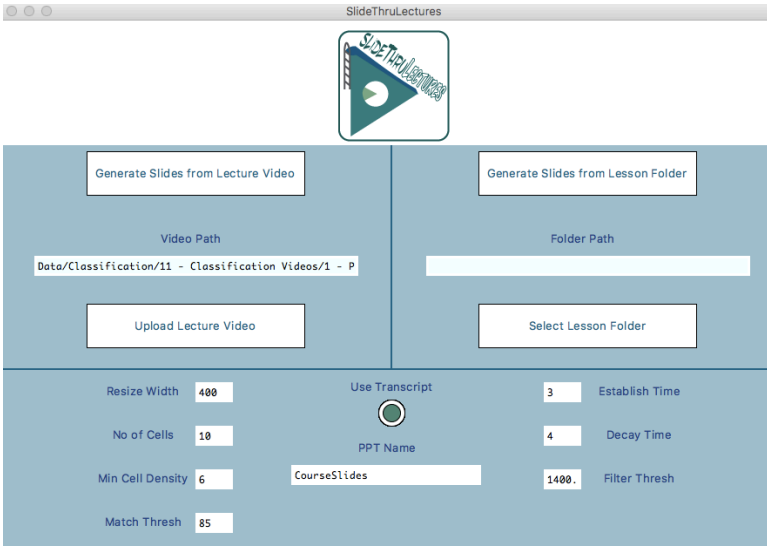

Figure 5. GUI of SlideThruLectures

The system is in the form of a cognitive agent architecture with a deliberation process similar to SOAR[16], containing procedural knowledge of production rules, semantic knowledge about visual content and ROI Frame knowledge representations and episodic knowledge about finding matches of tracked Frames and detection of new Frames. The knowledge representations of tracked visual content is stored in working memory and knowledge is used for reasoning, updating the memory through actions and eventually generating output in the form of visual transcript slides from video and transcript file inputs. Metacognition process validates the success of process of extraction of optimal video frame by comparing it with last extracted video frame to add a layer of reasoning about reasoning, along with making decision about integrating transcript text with slides based on selected option.

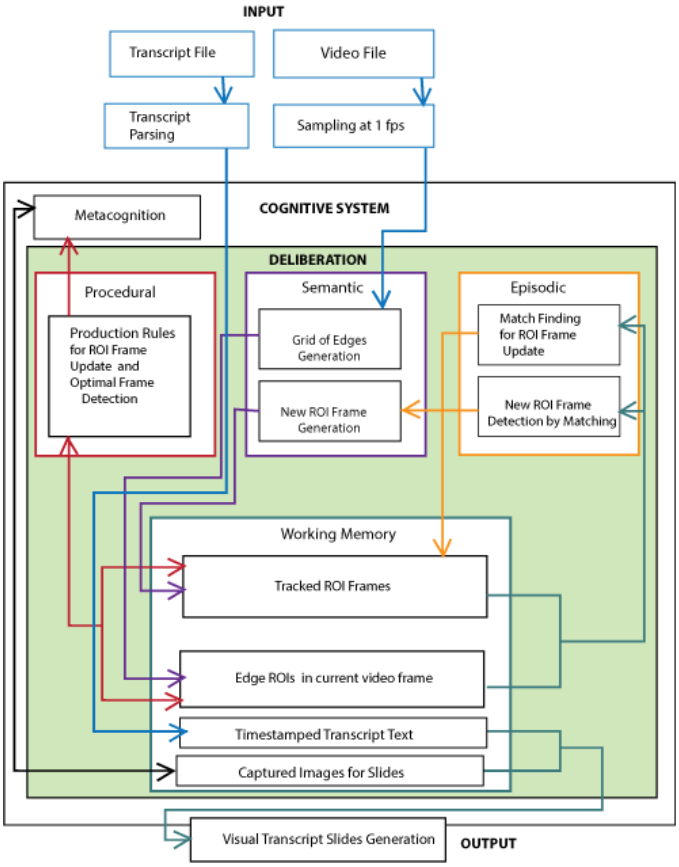

Figure 6. Cognitive System Architecture

EXPERIMENTAL RESULTS

Output Results

Following are some results representing the visualization of algorithm execution and the generated output slides.

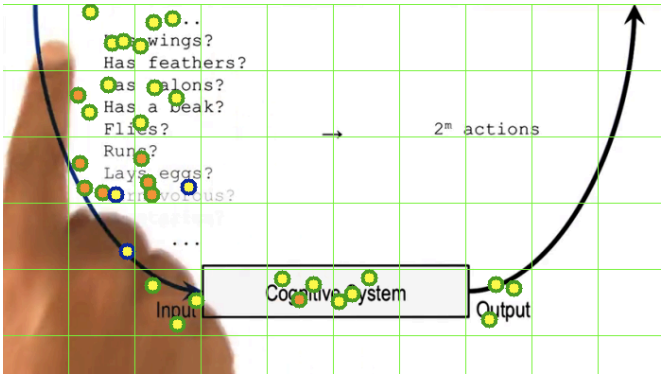

Figure 7. Algorithm Execution and Visualization

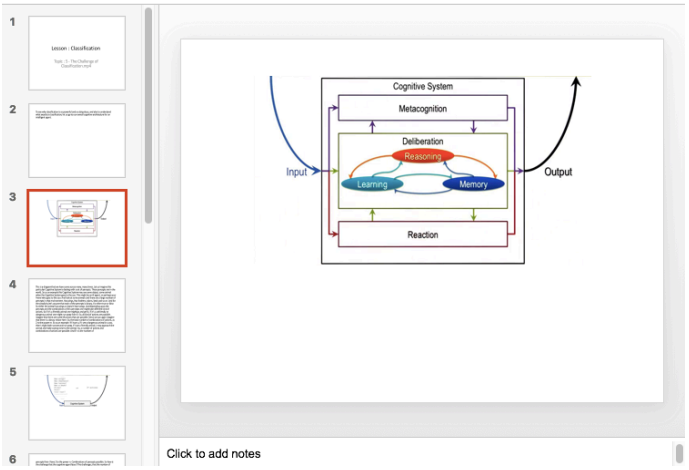

Figure 8. Output Visual Transcript Slides

Emperical Results

SlideThruLectures was tested on 2 hours of video content from 18 random lecture videos with 6 videos each from Udacity, Coursera and edX. The videos were from 17 different courses and contained all the types of situations and challenges like content build-ups, writing, typing, transition fades, moving instructor on screen, small deformities, occlusions by instructor, embedded videos and dynamic content etc. The algorithm parameters were kept constant for all tests.

Ground Truth was established by manually choosing optimal frames to capture. The task can be considered a binary classification problem with prediction on video frames sampled at 1 fps. The empirical results represent Accuracy, Precision and Recall. True Positives are video frames which were supposed to be captured and got captured while False Positives are video frames that were not supposed to be captured but got captured. Following are the results for tests:

Video Type	Average Accuracy	Average Precision	Average Recall
Udacity	**1.0**	**1.0**	**1.0**
edX	**0.999**	**0.963**	**1.0**
Coursera	**0.998**	**0.945**	**1.0**
Total Average	**0.999**	**0.969**	**1.0**

Table 1. Empirical Results

The results indicated that the algorithm is accurate and robust even for the challenging scenarios. It almost never misses true positives i.e. frames that should be captured but rarely gives false positives i.e. redundant (building-up)

frames in complex dynamic scenarios. This behavior is good for the intended purpose.

CONCLUSION

The paper described SlideThruLectures [17] an open-source software for helping online learners by generating visual transcript slides from online lecture videos by applying a novel algorithm based on Knowledge-based AI and Image Processing. The algorithm is in the form of a cognitive architecture which uses knowledge representation and tracking of edges in localized regions of video frames using rule-based reasoning, based on procedural knowledge about sequence of events related to finding tracked content and detecting new content to capture. This approach is based on the principle that optimal frame to capture is when note-worthy visual content that stayed long enough on screen starts disappearing permanently. The system proved to be very effective in dealing with the challenges in capturing most useful frames in online course/MOOC videos and generated visual transcript files for a large variety of online course videos.

ACKNOWLEDGMENTS

I would like to thank Dr. David Joyner, my instructor for the Educational Technology course at Georgia Tech, for designing the course based on an open-ended Ed-Tech project and efficiently managing it and hence giving me the opportunity and time to think of this idea and implement it. I would also like to thank David Somocurcio, my designated mentor for the project for his valuable feedback, motivation and guidance and Dr. Ashok Goel for his great course on KBAI, which I took along with Ed-Tech and got the inspiration and ideas behind my algorithm.

REFERENCES

1. Schoen, I. 2012. Effects of Method and Context of Note-taking on Memory: Handwriting versus Typing in Lecture and Textbook-Reading Contexts-Thesis. Retrieved April 29, 2018 from http://scholarship.claremont.edu/pitzer_theses/20/
2. Brief Tube [Computer software]. 2017. Retrieved from http://brieftube.com/
3. IVAS Video Synopsis [Computer software]. Retrieved from http://www.ivassystems.com/video-synopsis-3/
4. Brief Cam [Computer software]. Retrieved from http://briefcam.com/
5. Elbex Video Synopsis [Computer software]. Retrieved from http://www.elbextechnologies.com/video-synopsis/
6. Shin, H. V., Berthouzoz, F., Li, W., & Durand, F. 2015. Visual Transcripts : Lecture Notes from Blackboard-Style Lecture Videos. ACM Transactions on Graphics.
7. Jeong, H. J., Kim, T., Kim, H. G., & Kim, M. H. 2014. Automatic detection of slide transitions in lecture videos. Multimedia Tools and Applications.
8. Zhao, B., Lin, S., Qi, X., Wang, R., & Luo, X. 2017. A novel approach to automatic detection of presentation slides in educational videos. Neural Computing and Applications.
9. Yang, H., Siebert, M., Luhne, P., Sack, H., & Meinel, C. 2011. Automatic Lecture Video Indexing Using Video OCR Technology. 2011 IEEE International Symposium on Multimedia.
10. Choudary, C., & Liu, T. 2007. Summarization of Visual Content in Instructional Videos. IEEE Transactions on Multimedia.
11. Liu, T., & Kender, J. 2002. Rule-based semantic summarization of instructional videos.Proceedings. International Conference on Image Processing.
12. Ngo, C., Pong, T., & Huang, T. 2002. Detection of slide transition for topic indexing. Proceedings-IEEE International Conference on Multimedia and Expo.
13. Choudary, C. 2007. Extracting content from instructional videos by statistical modelling and classification. Article in Pattern Analysis and Applications.
14. Canny, J. 1987. A Computation Approach to Edge Detection. Readings in Computer Vision.
15. Goel, A., Joyner, D., Thaker, B. 2016. Knowledge-based Artificial Intelligence.
16. Lehman, J. F., Laird, J. E., & Rosenbloom, P. S. 1996. A gentle introduction to Soar, an architecture for human cognition. Invitation to Cognitive Science.
17. SlideThruLectures [Computer software]. 2018. Retrieved from https://github.gatech.edu/imalik8/Vid2Doc_EdTechProject