ASEE 2022 ANNUAL CONFERENCE
Excellence Through Diversity

MINNEAPOLIS, MINNESOTA, JUNE 26TH-29TH, 2022 ASEE
Paper ID #37344

Creating TikToks, Memes, Accessible Content, and Books from Engineering Videos? First Solve the Scene Detection Problem

Lawrence Angrave (Teaching Professor)

Lawrence Angrave is computer science teaching professor at University of Illinois who playfully creates and researches the use of new software and learning practices often with the goals of improving equity, accessibility, and learning.

Jiaxi Li

Jiaxi Li is a 5-year BS-MS student at the University of Illinois at Urbana-Champaign (UIUC). He is co-advised by Professor Lawrence Angrave and Professor Klara Nahrstedt. He has research interests in Artificial Intelligence for Human Computer Interaction. He has experience in Machine Learning, Computer Vision, and Text Mining.

Ninghan Zhong

Ninghan Zhong is a senior student in Computer Science at the University of Illinois at Urbana-Champaign. His research interests are in (but are not limited to) Autonomous Systems, Human-Robot Interactions, Human-Computer Interactions, and Artificial Intelligence. He has experience in computer vision, robotics, and machine learning.

Optimizing Scene Detection of Engineering Videos to Create TikTok Videos, Memes, Books, and Accessible Content

Abstract

To efficiently create books and other instructional content from videos and further improve accessibility of our course content we needed to solve the scene detection (SD) problem for engineering educational content. We present the pedagogical applications of extracting video images for the purposes of digital book generation and other shareable resources, within the themes of accessibility, inclusive education, universal design for learning and how we solved this problem for engineering education lecture videos. Scene detection refers to the process of merging visually similar frames into a single video segment, and subsequent extraction of semantic features from the video segment (e.g., title, words, transcription segment and representative image). In our approach, local features were extracted from inter-frame similarity comparisons using multiple metrics. These include numerical measures based on optical character recognition (OCR) and pixel similarity with and without face and body position masking. We analyze and discuss the trade-offs in accuracy, performance and computational resources required. By applying these features to a corpus of labeled videos, a support vector machine determined an optimal parametric decision surface to model if adjacent frames were semantically and visually similar or not. The algorithm design, data flow, and system accuracy and performance are presented. We evaluated our system using videos from multiple engineering disciplines where the content was comprised of different presentation styles including traditional paper handouts, Microsoft PowerPoint slides, and digital ink annotations. For each educational video, a comprehensive digital-book composed of lecture clips, slideshow text, and audio transcription content can be generated based on our new scene detection algorithm. Our new scene detection approach was adopted by ClassTranscribe, an inclusive video platform that follows Universal Design for Learning principles. We report on the subsequent experiences and feedback from students who reviewed the generated digital-books as a learning component. We highlight remaining challenges and describe how instructors can use this technology in their own courses. The main contributions of this work are: Identifying why automated scene detection of engineering lecture videos is challenging; Creation of a scene-labeled corpus of videos representative of multiple undergraduate engineering disciplines and lecture styles suitable for training and testing; Description of a set of image metrics and support vector machine-based classification approach; Evaluation of the accuracy, recall and precision of our algorithm; Use of an algorithmic optimization to obviate GPU resources; Student commentary on the digital book interface created from videos using our SD algorithm; Publishing of a labeled corpus of video content to encourage additional research in this area; and an independent open-source scene extraction tool that can be used pedagogically by the ASEE community e.g., to remix and create fun shareable instructional content memes, and to create accessible audio and text descriptions for students who are blind or have low vision. Text extracted from each scene can also used to improve the accuracy of captions and transcripts, improving accessibility for students who are hard of hearing or deaf.

1 Introduction

Recent advancements in educational technologies have made available many innovative approaches to engage students with the course materials. In addition to standard-classroom teachings, efficient and reliable educational tools have been developed to make the content more accessible to all students. ClassTranscribe is an educational web application that is designed to offer accessible video-based lectures to engineering college students. Equipped with user-friendly functionalities such as real-time speech-to-text transcription and caption search, the video player was found to improve the final exam scores for students in a computer science course, with the largest effect size for lowest scoring students[2]. One key feature ClassTranscribe offers is the automatic generation of digital books, available in multiple formats (pdf,epub,html) which are composed of lecture clips, slideshow text, and audio transcription content extracted from uploaded lecture videos. The `ePub` format is an open standard format that can be further edited and transformed into other common formats. The structure and chapters of the digital books were automatically extracted from the video scene change detection model and a title detection algorithm. This functionality facilitates an alternative learning-pathway to consume video-based content. However, despite these benefits, the digital book generation process was limited to videos based on discrete slideshow content. If a lecture was recorded in a different format, for example if the instructor's talking face was included or content was constructed incrementally, the scene detection model performed sub-optimally negatively affecting the quality of the generated digital books. Our work aimed to address this limitation and create an improved scene detection method that was adaptable to a wider range of lecture video recording formats employed in modern engineering education video content.

As a brief aside, an early question to the researchers was often "Why - Surely instructors just share their slides as a pdf?" However, only a subset of instructors prepared traditional slides; others constructed live content in editors, used document cameras, or employed a wide variety of sources during the lecture. Even when instructors used Microsoft PowerPoint or equivalent, the linear slide sequence was an incomplete representation of the presented material. In short, slides, and annotated slides were not an *equivalent learning resource* to the recorded lecture video. We stress this is not just a technical limitation, rather that from a student perspective, the slides if available and provided, did not meet the standard of an equivalent, alternative learning pathway.

In this paper, we present and evaluate a new video scene detection framework, which was designed and evaluated for a corpus of engineering educational video content, to facilitate automatic digital book generation and other instructional content. We also discuss the educational potential of digital book generation under the motivation of providing accessible and inclusive education, in accordance with the principles of Universal Design for Learning.

Scene detection refers to the task of partitioning visually similar and semantically related sequential frames into groups, representing video segments. Then, subsequent operations can be applied to these segments individually to extract important semantic features, such as titles, key words, transcription segments and images.

In our framework, features that indicate potential scene changes were extracted from each pair of sequential sample frames using metrics that represented multiple similarity comparisons between two video frames. These features were based on the structure and text aspects of the image, and also by excluding the estimated face and upper body area when a face was detected. The accuracy, performance and computation resources needed for our platform were evaluated and optimized. By creating a corpus of labeled videos, approximately $30,000$ entries of feature measurements were collected as training data. A support vector machine was trained to determine the optimal parametric decision boundary to predict if a pair of adjacent frames signaled a frame change. Experimental evaluation of the system was based on engineering lecture videos from multiple disciplines. The testing videos used were a mixture of different presentation formats, including hand-written notes, PowerPoint slides, digital ink annotations, and live code demos. For each educational video, a comprehensive digital textbook comprising of the extracted lecture clips, slideshow text and transcription content could be generated, with the scene detection framework serving as the backbone of the generation process. Providing a equivalent learning resource to facilitate multiple learning pathways is a principle of Universal Design for Learning, which is discussed in the background section. We also show how this work improves accessibility and equity for students with visual impairments and students who are hard of hearing or deaf.

The main contributions of this work are:

1. Identification of multiple causes why automated scene detection of engineering lecture videos is challenging and creation of a labeled corpus of videos representative of multiple undergraduate engineering disciplines and lecture styles.
2. Creation of a set of image metrics which were then used as input to a support vector machine (SVM)-based classification approach.
3. Evaluation of the accuracy, recall and precision of our SVM-based approach.
4. Creation of a performance optimization to obviate the need for GPU resources.
5. Subject to instructor permission, publication of a scene-labeled corpus of engineering video content to encourage additional engineering education research in this area.
6. Publication of an open source scene-detection and extraction python tool to efficiently extract unique images from lecture videos. We demonstrate how the output can then be the basis for additional manual and automated activities to improve accessibility and equity, and can be used for content creative remixing to improve student engagement.

The rest of the paper is organized as follows. Section 2 describes the general background of the problem and its importance from a Universal Design for Learning perspective. Section 3 provides a formal problem statement, describes the challenges and related works. Section 4 presents our scene detection framework, along with its evaluation and optimization. Section 5 addresses the book generation application of the scene detection framework. Section 6 provides students' early feedback on the new book and note taking opportunities based on this work. Section 7 describes accessibility and content remixing applications and provides examples of both. Section 8 discusses future work and concludes this paper.

This is a relatively long and detailed paper; we also wanted to highlight that math, computation, heuristics, optimization, and engineering-focused analysis and iterative design can all be employed together to improve inclusivity, equity, and accessibility.

2 Background

2.1 Universal Design for Learning

Universal Design for Learning (UDL) is a set of educational design principles that seek to improve the education accessibility and inclusiveness for course content and student assessment. A UDL course provides varied and multiple learning pathways and modalities for a student to acquire course knowledge and skills. For example, a student may learn a topic by attending lectures, asking questions, reading course notes, or reviewing lecture videos, reading lecture transcript, or most likely, a combination of a subset of these items. Instructors that adopt a UDL approach will usually ensure that their videos are accurately captioned, visually described, and look for additional methods to provide further learning opportunities. The UDL approach places a strong expectation of inclusivity, i.e. that educational technologies will support widely-accessible content delivery methods that can benefit all students, including students with disabilities [12]. Accessible content also helps international students with imperfect English, to obtain better learning results[6]. With such a motivation, our work here aims to improve the automatic digital textbook generation feature in ClassTranscribe, because digital textbooks provide an alternative learning pathway and provide equity i.e., help "level the playing field" particularly for students with attention, hearing and/or visual difficulties and find it hard to learn from video-based resources.

2.2 Introduction to ClassTranscribe

ClassTranscribe is a new web-based video platform developed at the University of Illinois, and previously introduced at ASEE [11] [1]. The goal of ClassTranscribe is to enable students to learn, search, review the lecture videos efficiently and conveniently in an equitable and accessible interface. Designed to offer

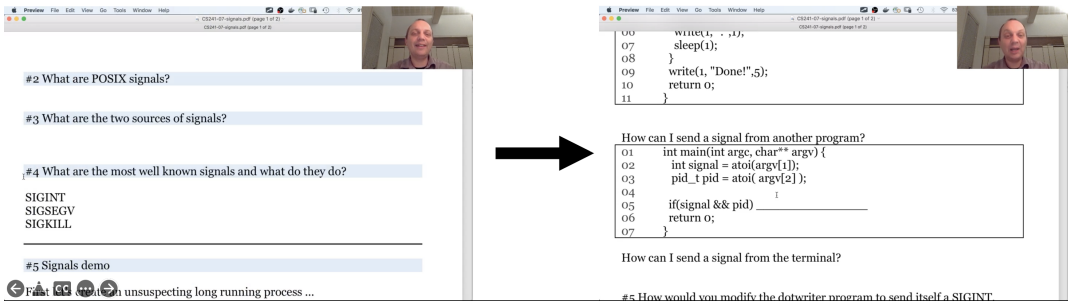


Figure 1: An example scene change problem. A scene detector should identify change in information content from one video frame to the next but ignore superfluous visual changes due to facial and hand movements.

accessible video lectures to engineering college students, ClassTranscribe is equipped with user-friendly features, such as automated transcriptions, automatic digital book generation, and the ability to resume watching a longer video when a student has difficulty focusing. ClassTranscribe also allows students to search through transcriptions and to fix transcriptions errors efficiently. ClassTranscribe has been used as an instruction tool for computer science classes and other engineering classes at the University of Illinois. This includes multiple large-enrollment classes with more than 300 students. ClassTranscribe videos have been watched by over 6,200 students. With usage data available from the large number of users, ClassTranscribe has also been used as educational research platform. This includes, for example, exploring the effects of ClassTranscribe by students on improving learning outcomes [2] and characterizing how students utilize transcription-search across the whole course content to find video content [25].

ClassTranscribe can create digital books from video content. The first stage of automatic digital book generation is video segmentation based on scene changes, and all the subsequent processing is based on the output from this stage. Thus, the eventual quality of the generated digital books depends on the accuracy of the scene change detection during the video segmentation. To guarantee the quality and usefulness of the generated digital books, an accurate and robust scene detection framework is needed. Nonetheless, such scene change detection is a challenging task as the framework should be able to adapt to a wide range of video lecture formats, and is discussed next.

3 Problem Description

3.1 The Scene Detection Problem

A video is composed of a series of images, or frames. Adjacent frames can be either similar or visually different. Given a video, the objective is to find all the moments when neighboring frames change and provide new information. Then, from either sides of the frame changes, locate the segments of repetitive frames, and extract a representative frame, or *scene*, for each segment.

Figure 1 represents two typical scene changes in a video lecture of a college-level Computer Science course. In the first example, the instructor switches from the camera view to the screen view. In the second example, the professor changes the presentation material. Those two frames are examples of typical frames for their corresponding segments.

3.2 Challenges in Scene Detection

Scene detection is a "diamond-in-the-rough" filtering problem because nearly all neighboring video frames are not a scene change. The performance of an automated scene detector has two kinds of failure modes: False Positives - incorrectly identifying a scene change - this would result in many superfluous visual images being included in the digital book, effectively cluttering the book with unnecessary and similar images. False Negatives - not detecting a scene change would cause visual information, e.g. a slide, to be excluded from a digital book. For example, a naive scene detector that emitted an image every second of video lecture content would produce nearly 4000 images for an hour content, with nearly all images being false-positives,

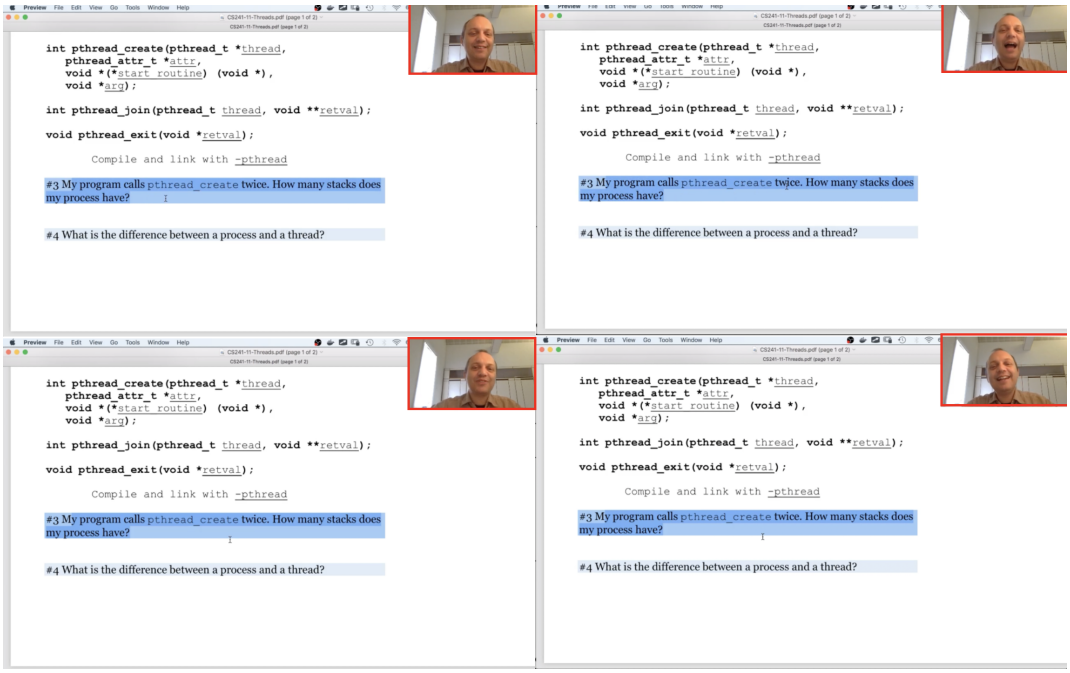


Figure 2: A scene detection challenge: Facial movement in the red box can mislead a pixel-based approach.

whereas only sampling the video every minute could miss important content (i.e. have false negatives). A more sophisticated solution is required.

For the simplest slide-based content, a pixel-similarity approach is sufficient; specifically to average the difference in brightness of each pixel from one frame to the next, and then compare the result with a threshold value. This approach was employed in the initial version of ClassTranscribe. However it was impossible to determine a reasonable threshold value that did not cause too many false negatives or false positives, and thus was the motivation for the work described here.

By collecting multiple videos where the simple pixel-change described above failed we identified the following common video elements in engineering education videos that caused scene detection to be a non-trivial problem.

Firstly, the instructor's face movement in the lecture created false negatives when only a pixel-similarity-based approach was used. For example, in many lecture videos, the instructors' face appeared in a rectangle box at the edge of the screen, as indicated by the red box in Figure 2. Although the content in the presentation stayed the same, the pixel changes from the instructors' face movements caused the simple scene detection algorithm to incorrectly declare a scene change. To address this, a face detection library was used to locate the face on the screen, and compared structural similarity with the face pixels masked in both frames. The mask area overrides the original pixel colors of both frames to the same color RGB value, effectively removing the original pixel differences of the mask area from the similarity measure.

Slight scrolling of a presentation text page also misled the pixel-based similarity detector. In Figure 3, the presenter scrolled up the page, as indicated by the upward movement of the red box. Although most of the information stayed the same, all the pixels have shifted by some amount, which mislead the pixel-based method to classify it as a scene change. To solve this issue, an Optical Character Recognition (OCR) lirbary was used to detect text information in both frames and compared the text difference; this is discussed in more detail later in the paper.

Incremental writing and typing were used when the presenter introduced a problem step by step. Figure 4 represents a typical example of this case, where the presenter typed the code one line at a time. An optimal scene detector would extract only the first frame and last frame in this example. Scene detection

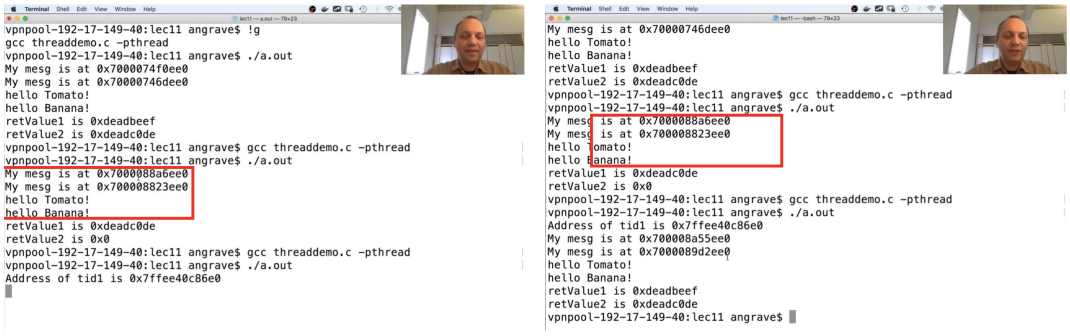


Figure 3: A scene detection challenge: Scrolling the page can mislead the pixel-based approach.

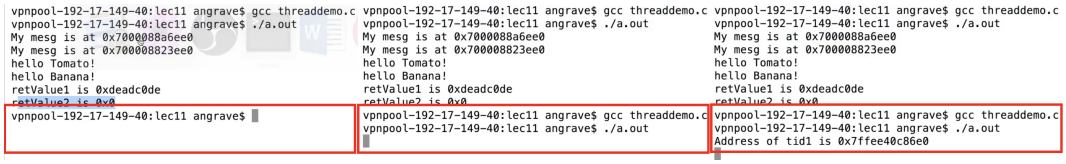


Figure 4: A scene detection challenge: Incremental Writing – An optimal detector would extract the start and end frames and ignore unnecessary intermediate frames.

on incremental annotation is the subject of ongoing work and we look forward to reporting on our progress in a future paper.

3.3 Related Works

In this section, we discuss some of the related work to video scene detection. Most of the existing frameworks can be categorized into one of the three approaches: clustering approach, graph-based approach, and learning-based approach.

Contemporary existing works based on clustering approaches transform video frames into a feature space. Then, clustering algorithms are performed under the assumption that frames within the same scene have close proximity in the feature space and will be clustered together. For instance, Baraldi *et al.* [5] and Panda *et al.* [13] utilize a spectral clustering approach to group adjacent shots together.

Previous works also used a graph-based approach [15] [8] to represent a video shot as a node in a graph. Then, graph partition algorithms are applied to find the best cuts that determine scene transitions. Further, some works, such as [18], first construct Scene Transition Graphs, which considers a cluster of shots as a node. Then cumulative confidence estimates are computed to find the optimal scene divisions.

More advanced frameworks also incorporate learning into the scene detection process. For instance, Rotman *et al.* [17] proposed a learnable Optimal Sequential Grouping method, in which a fully connected neural network is applied to the video content to learn the visual and audio embedding, and distance metrics are computed from the embedding. Then, Optimal Sequential Grouping [16] is applied to calculate the scene division probability. Further, Baraldi *et al.* [4] proposed a framework that learns a similarity measure using a Siamese Neural Network and then performs spectral clustering based on the learned similarity matrix.

The majority of the existing works on video scene detection were designed for general purpose production videos, and the ultimate goals vary widely based on the applications. However, to the best of our knowledge, none of the existing frameworks are optimized toward educational lecture videos or modern engineering educational videos. The method proposed in [5] has the most in common with our motivation and problem context, as it also aims to improve re-using media content in an educational setting. However, the scene detection method in [5] is not optimized for lecture videos and it takes a clustering approach, while our method incorporates machine learning by training a support vector machine model that is more efficient and robust.

4 Scene Detector Design

4.1 Algorithm Design

Our approach to the scene detection problem is to compare both structural and text disparity between neighboring frames.

4.1.1 Candidate Frame

Modern popular video formats have a typical frame per second (fps) of 24 or 30. Thus, a medium-length lecture of 50 minutes typically includes 72,000 to 90,000 frames. However, a large amount of those are redundant and equivalent to the previous frame. For example, the video frames of a recorded Microsoft PowerPoint presentation slides are identical to the previous frame for most of the lecture. To reduce computation time we assumed an upper limit on visual change in instructional videos. We assume instructors will present material at a rate of no faster than 0.5 fps. In other words, the scene detector needed to process a video frame from every 2 seconds of video content. For a lecture of 50 minutes, this is a sequence of approximately 1,500 frames. These frames were called *candidate frames* as they represented a list of frames that may correspond to the end of one scene and the start of the next. For each candidate frame, the similarity of successive *candidate frames* using multiple metrics was calculated. This analysis is described below.

4.1.2 Inter-Frame Structural Similarities

The first metric *sim_structural*, computed the structural similarity index measure[23] [3] (*SSIM*) for a pair of adjacent frames. *SSIM* differentiates two images under visual similarity. Specifically, the measurement considered the optical difference in luminance, contrast and structure. An algorithm from the skimage package [19], was used for this purpose which returns a *SSIM* value between 0 and 1 for two images, with a higher value emphasizing greater similarity.

4.1.3 Inter Frame OCR Similarities

Optical Character Recognition (OCR) is the process of recognizing and extracting from an image the text that is either typed or handwritten and converting it into machine-readable text. For this work the extracted text characters and additional meta-information - position, size and a self-reported accuracy measure - was used.

Using the OCR text information from each image, we defined a new metric, *sim_ocr*, which compared the text difference between two adjacent frames. For two adjacent frames, *sim_ocr* returns a value in the range 0 and 1, with 1 implying the text content in two images is identical and 0 representing dissimilar text. To calculate *sim_ocr*, Google's Tesseract OCR[21] was used to extract a list of words together with geometric information and a confidence value that estimated the correctness of the extracted text. The formula for calculating *sim_ocr* for a pair of successive frame F_A and F_B is summarized as follows,

$$sim\_ocr(F_A, F_B) = \frac{\sum_{w=a\cap b} (C_A(w) + C_B(w))}{\sum_a C_A(a) + \sum_b C_B(b)}$$

where a is a word that appears in OCR text of F_A, and b is a word that appears in the OCR text of F_B, respectively and w is a word that appears in OCT text of both F_A and F_B. $C_X(y)$ is the OCR confidence value of the word y in the frame F_X. If no words are detected in either frame, *sim_ocr* is defined to be 1. Thus, the metric is sensitive to the fraction of words that have changed between the two frames and words that were identified with a high confidence by the OCR algorithm contribute more to the metric total. Other measures are possible and may be worthwhile researching in the future. For example, edit distance, which accounts for typographic changes required to convert one word into another, could be used to calculate a

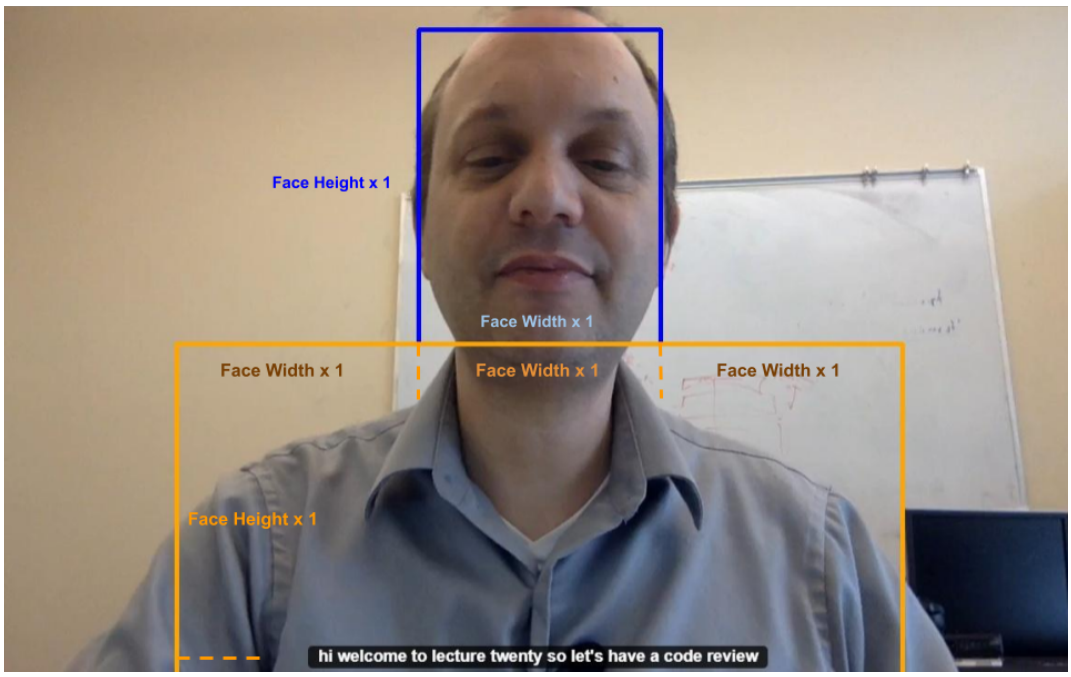


Figure 5: Face Detection Result – with an upper body.

secondary OCR metric which may be useful for OCR output with low confidence and small single-letter OCR errors.

4.1.4 Face Detection

To address the facial and body movement false-positive problem described previously, the presenters' faces and bodies were automatically located in the frames and added to the pixel mask in the predicted face and body position to calculate an adjusted *sim_structural* value. A pre-trained Multi-task Cascaded Convolutional Networks (MTCNN) [24] [9] was used to detect faces in all *candidate frames*. MTCNN was selected for two reasons. First, MTCNN is a light-weight architecture, so its efficiency guarantees a lower time consumption during scene detection. Second, MTCNN is trained on an image pyramid [24], and is able to detect faces with varied sizes in the frames. This is an important aspect because corpus of lecture videos included both large and small instructors' faces with respect to the frame size.

To ignore pixel changes due to body movements we first observed that in the corpus of lecture videos with the instructors' bodies present, only the upper bodies appeared in the frames, as shown in Figure 5. Thus, to simplify our model, only the upper bodies of the instructors were located and masked. Second, when instructors' faces appeared in the far left 20% or far right 20% of the video area, they were consistently small head shots, as illustrated in Figure 6. So, only detected faces within the 20% - 80% horizontal range of the frames were considered to also require upper body masking. To locate the upper bodies, for each detected face, a new bounding box that was 3 times wide and 3 times long as the bounding box of that face was constructed, according to standard proportions of human bodies for adults[7]. An example of the final face and upper body mask is shown in Figure 5.

The mask was then used in the construction of a another metric, *sim_structural_no_face*, which compared the structural pixel difference between two adjacent frames after masking out the pixels inside all the upper-body and face bounding boxes.

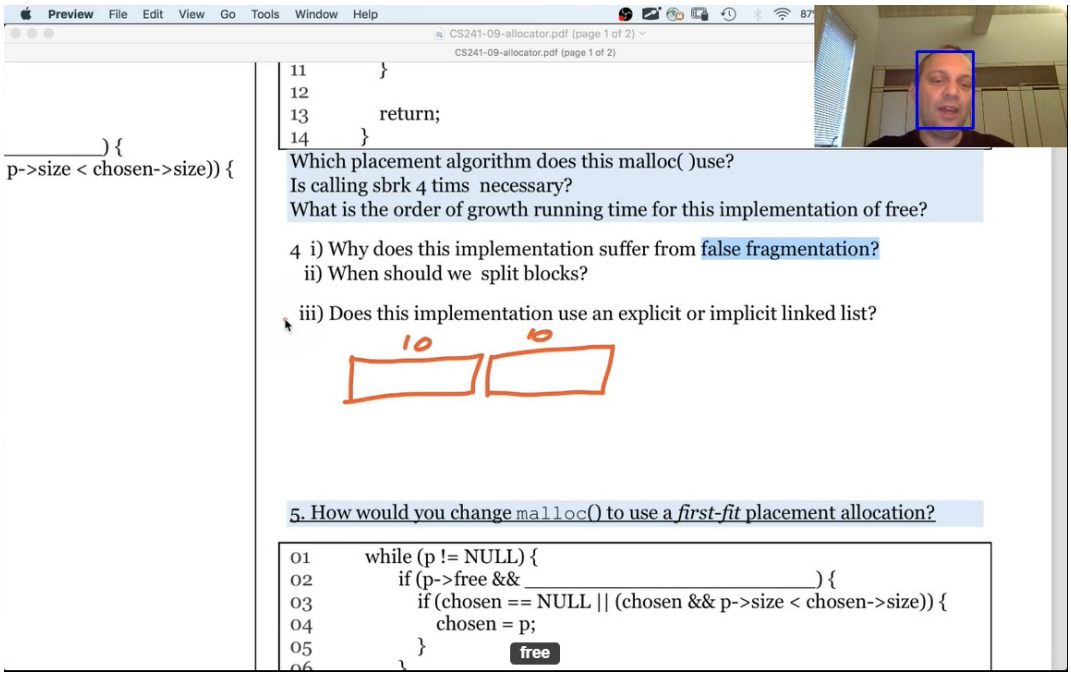


Figure 6: Face Detection Result – Face inside the blue rectangle without an upper body.

4.2 Model Training and Evaluation

4.2.1 Data

A set of 9 educational videos from ClassTranscribe's database was selected to be representative of multiple engineering disciplines and diversity of presentation styles. Using a sampling rate of 0.5 fps, all adjacent frames in the 9 videos were manually compared and labeled as either a scene change ("real positive") or not scene change ("real negative"). Then, *sim_structural*, *sim_ocr*, and *sim_structural_no_face* were computed for each pair of adjacent frames.

Each pair of adjacent frames served as one sample. In total, there were 33,244 labeled samples and the three metrics were computed for each sample.

To encourage further research, we shared our labeled training data. The dataset is available at https://uofi.box.com/v/SceneDetection-ASEE2022-shared, along with the original video content subject to the instructors' approval. Please cite this paper if you use the data set.

4.2.2 Training a Support Vector Machine

For each pair of successive frames, there were three metrics, *sim_structural*, *sim_structural_no_face*, and *sim_ocr*. The objective was to train a classifier to discriminate if a pair of successive frames from a new video was a scene change using the three metrics. The use of a Support Vector Machine (SVM) model was selected because, 1) A supervised machine learning model is appropriate because real labels for all training samples have been created, and 2) The small training set is too small for deep neural network approaches, 3) We expected the data to be linearly separable or nearly separable. 4) A SVM provides an intuitive geometric interpretation that can visualized, whereas deep learning models provide no or little insight into the classification output.

A SVM was modeled based on the three metrics of all training samples. A SVM model can fit a boundary surface directly in the metric space, which best separates positive labels from negative labels in the training samples. Figure 7 is an example of the SVM classifier modeled on the training samples, where darker, blue points refer to real positive samples, lighter, orange points refer to real negative samples. The gray plane

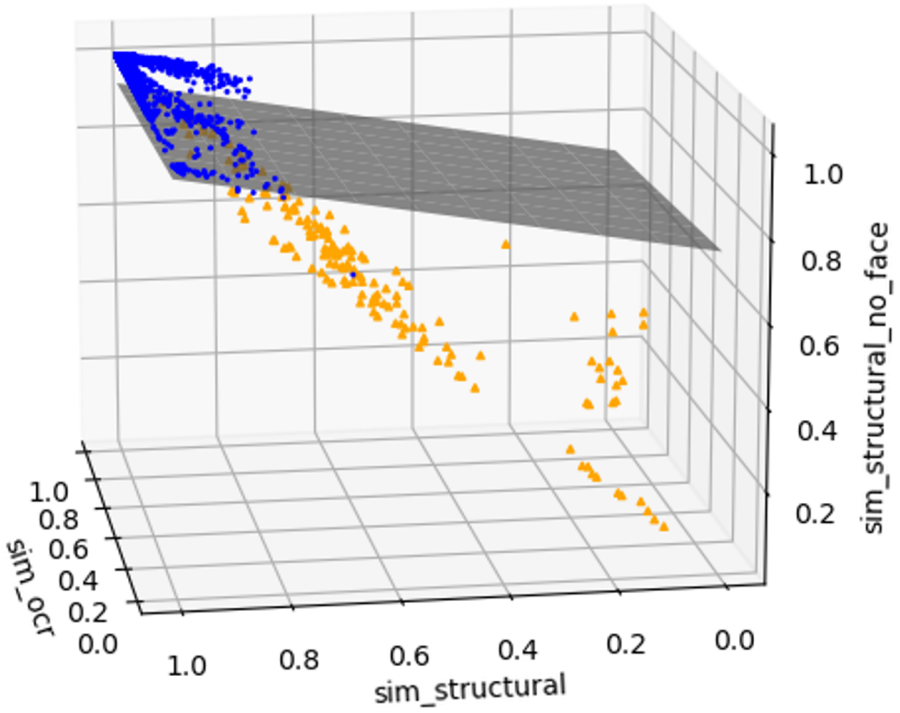


Figure 7: SVM Visualization with on linear kernel. Darker blue points represent negative samples (i.e. not a scene change); lighter, orange triangles represent positive samples; and the grey plane represents the best separating surface.

refers to the boundary surface from the SVM model. For an unlabeled pair of adjacent frames, it is now possible predict if it is a scene change based on its relative position using the 3 metrics i.e. if it is above or below the gray plane of the SVM boundary surface.

To ensure the independence of the testing data from the training data, only one video was selected as the testing set at a time. The Scene Detector was created from the other 8 videos, which were the training set, then used to predict the frame labels for the test video frames. This process was iterated for all of the 9 videos.

There were many more negative samples than positive samples in our data because most of lecture videos had a static presentation style with one slide change every several minutes, hence the number of scene changes (positive) was fewer than that of stationary frames (negative). To address this imbalanced-training-data issue, and to further account for our preference to have higher recall than precision metrics, the class weight for positive data was set to be larger than that of negative data in the SVM model. In other words, to counteract the abundance of negative samples in the metric space where training data is not separable, there was a slight intentional bias for the SVM to favor correctness of the (fewer) positively labeled samples, resulting in a more balanced prediction.

4.2.3 Kernel Function

To train a SVM on data that is not linearly separable in the original space, a non-linear kernel function can be used to transform the space into a more desirable form. The most common kernel functions are linear, polynomial, and radial basis function (RBF). Additional details on kernel functions and support vector machines are described in [14]. Since the frame-similarity-measurement data was not linearly separable in the metric space, in addition to the common linear kernel, the polynomial and RBF kernel were evaluated.

Metric	Accuracy %	Precision %	Recall %	F1 Score
sim_structural	98.8	57.3	85.0	68.4
sim_structural_no_face	99.4	74.1	92.8	82.4
sim_ocr	97.4	18.2	21.7	19.8

Table 1: Results based on different metric setting. The best single-performing metric is *sim_structural_no_face*. This is not particularly surprising; this metric was designed as a direct improvement over the original *sim_structural* metric while *sim_ocr* may not detect content changes in non-textual images.

Kernels	Accuracy %	Precision %	Recall %	F1 Score
Linear	99.3	74.0	90.3	81.3
Polynomial	99.4	76.0	93.2	83.7
RBF	99.4	76.5	92.4	83.7

Table 2: Results using all metrics under different kernel functions. The Polynomial kernel using all 3 metrics provided the best recall and similar precision to the RBF kernel.

The performance of using these kernels are presented in the next section.

4.2.4 Results

Table 1 presents the results based on the individual metric and all metrics using a linear kernel. The individual metrics did not perform as well as using all 3 metrics (See Table-2). Table 2 presents the results by combining all 3 metrics using the linear, polynomial and RBF kernels. Accuracy is the fraction of all correctly labeled samples (both positive and negative). Due to the imbalance of real negative to real positive cases, accuracy is less useful than the other 3 performance measurements. Precision is the fraction of correctly labeled samples from samples that are predicted as positive. Thus a high percentage precision implies few false positives - practically, this would result in a minimal clutter of unnecessary images in the digital book. Recall is the fraction of correctly labeled samples from samples that are labeled as real positive. A high percentage recall implies most real scene changes are identified i.e. there would be no missing slides in a digital book. F1 is the harmonic mean of precision and recall, and can be used as general comparative measure that equally balances precision and recall.

4.2.5 Discussion

The testing results show that an increase in the overall recall and precision after introducing all three metrics, which indicate the effectiveness of using OCR and Face Detection to supplement the original similarity measure.

In the context of digital book generation, the ideal performance is to detect every real scene change without cluttering the book with distracting false positives. To minimize the number of real positive samples that are not detected, Recall is an useful measurement because it represents the fraction of detected positives out of all real positives. A higher recall means fewer missed real positive images, and hence a better extraction outcome.

From Table 2, among all the kernels, the polynomial kernel exhibited the optimal performance in recall.

Table 3 listed the accuracy performance of the best SVM model on each video. The model performed best on video 6 because it involves less face movements and no handwritten annotations. The model performed worst on video 8 because incremental handwritten annotations in the lecture caused excessive scene changes detected.

Video	Accuracy	Precision	Recall	F1 Score	Comments
1)	99.9	97.5	95.2	96.3	
2)	99.2	69.0	91.8	78.8	
3)	99.7	100.0	76.1	86.4	
4)	99.8	100.0	88.4	93.8	
5)	99.7	91.6	100.0	95.6	
6)	100.0	100.0	100.0	100.0	Best result because of no annotations and few face movements
7)	99.8	83.3	100.0	90.9	
8)	95.6	34.8	97.4	51.3	Worst result due to large number of handwritten annotations
9)	99.8	97.3	97.3	97.3	

Table 3: Performance of the best SVM model for each video.

Computed Metrics	Corpus Processing Time (seconds)	Ratio	Single lecture (seconds)
All 3 metrics	2,095	10.4%	311
Only *sim_structural*	307	1.5%	45.6

Table 4: Time requirement for different metric computation strategies.

4.3 Optimization

4.3.1 Time Requirement

Although the model with all metrics has better correctness than the original scene detector, it required a significant amount of additional compute time to calculate the two additional metrics. Table 4 is a summary of the time requirement for metric computation in the 9 videos. The third column represents the ratio of computation time dividing the total video length. The total video length is about 5 hours and a half (approximately $20,000$ seconds). The fourth column represents the estimated computation time for a 50-minute lecture using a 6-core Intel Core i5-11400 CPU. Although the GPUs would be able to achieve results more quickly, fast CPU-based performance was desirable because university-owned GPU virtual machines hosts are uncommon, modern GPUs are difficult to obtain, Amazon cloud GPU VMs are relatively expensive (50+ cents/hour; $4K+ /year), whereas CPU-only VMs are readily available on older and/or free university hardware.

The algorithm described so far required 30 CPU minutes (5 minutes of wall-clock time using a 6 CPU core desktop machine) to complete the scene detection process for a 50-minute lecture. This was considered to be unacceptably slow for inclusion into a video platform, such as ClassTranscribe that was designed to support multiple courses. Though GPU resources would speed the computation, a processing time of approximately 1 minute was desired for a typical 50 minute lecture without using GPU resources. This was solved using an early dropping technique, described in the next section.

4.3.2 Early Dropping

We describe here a technique to optimize the computation process using an *Early Dropping* technique (also known as Short-circuiting or Early-exit), where a portion of a time-consuming calculation can be obviated under some conditions. Briefly, the strategy is to use a metric(s) that is fast to compute to process samples to determine if they are very similar or very dissimilar and only calculate the more expensive metric(s) when more careful arbitration is required. Figure 8 plots the distribution of *sim_structural* and *sim_ocr* metrics for all samples with the real positive and negative labels. For the majority of samples with a real-negative label, the *sim_structural* values are higher than 0.9. Therefore, for a unlabeled sample, if it has a *sim_structural* that is higher than a threshold, then the early dropping model prediction is that it is not a scene change and hence computing *sim_ocr* and *sim_structural_no_face* can be avoided. In practice,

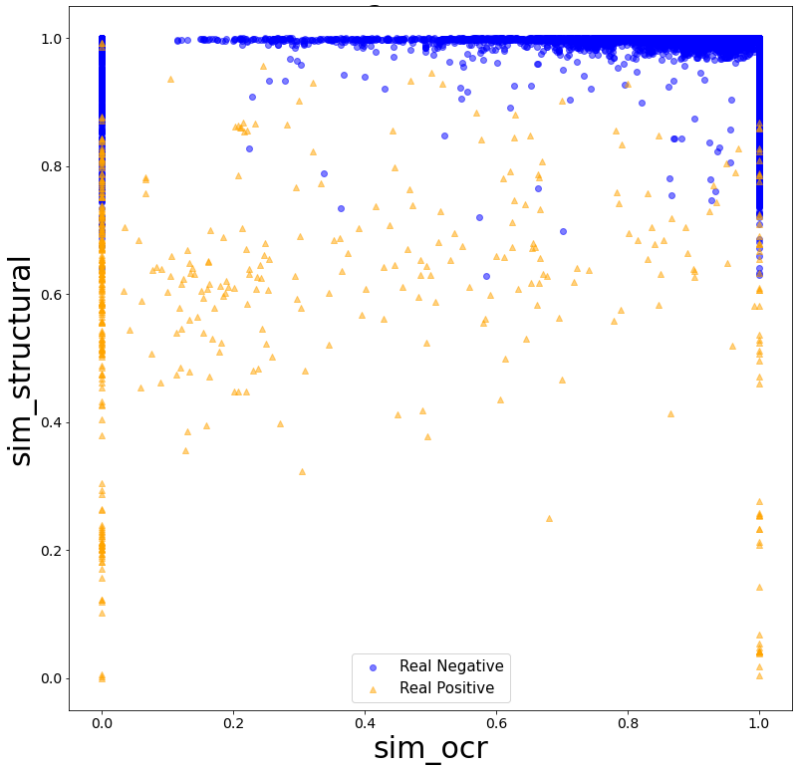


Figure 8: Distribution of *sim_structural* and *sim_ocr* values for samples using the real label. Darker blue points representing negative samples (not a scene change) and lighter orange triangles representing positive samples (a scene change).

a value of 1 is then used in place of calculating *sim_ocr* and *sim_structural_no_face* metrics for these cases. Note early dropping was employed to exclude both similar and dissimilar cases, however for brevity this paper only presents the details of the former.

To find the best value of the early-dropping threshold, experiments were performed on the $33,244$ samples from the 9 videos. Different thresholds were evaluated for their associated error rate. The results are summarised in Figure 9.

The Error rate refers to the fraction of the positive samples incorrectly dropped from all samples dropped-as-negative. With a threshold of 0.96, the error rate remains small but it is still possible to drop a large number of negative samples. Specifically, among all the frames with *sim_structural* higher than 0.964, only 0.00956% (3 out of $31,372$) were real positive frames. Therefore, we concluded a new frame with *sim_structural* higher than 0.96 will likely be a negative frame.

The same method was applied to drop samples with low *sim_structural* as a positive. Then, the process was repeated in section 4.2 with *Early Dropping*, and the performance is summarized in Table 5. The *Early Dropping* technique had minimal change to the correctness measurements, but significantly reduced the computation time. With *Early Dropping*, $31,372$ samples (94.4%) could be dropped from the full, 3-metric calculation, though in a production system, a more conservatively threshold might be used to avoid potential over-fitting to the training set. Compared to the original algorithm the computation time requirement decreased from $2,095$ seconds to 443 seconds, which was similar to the original processing time for the single *sim_structural* metric. With an improved ratio of 2.19% of processing time to video time, this implied a 50-minute lecture would require about 66 seconds to complete the scene detection process. With the improved accuracy and a 5-fold increase in performance of Scene Detection by using *Early Dropping* the new revised algorithm was ready for inclusion into ClassTranscribe and potentially other video production systems.

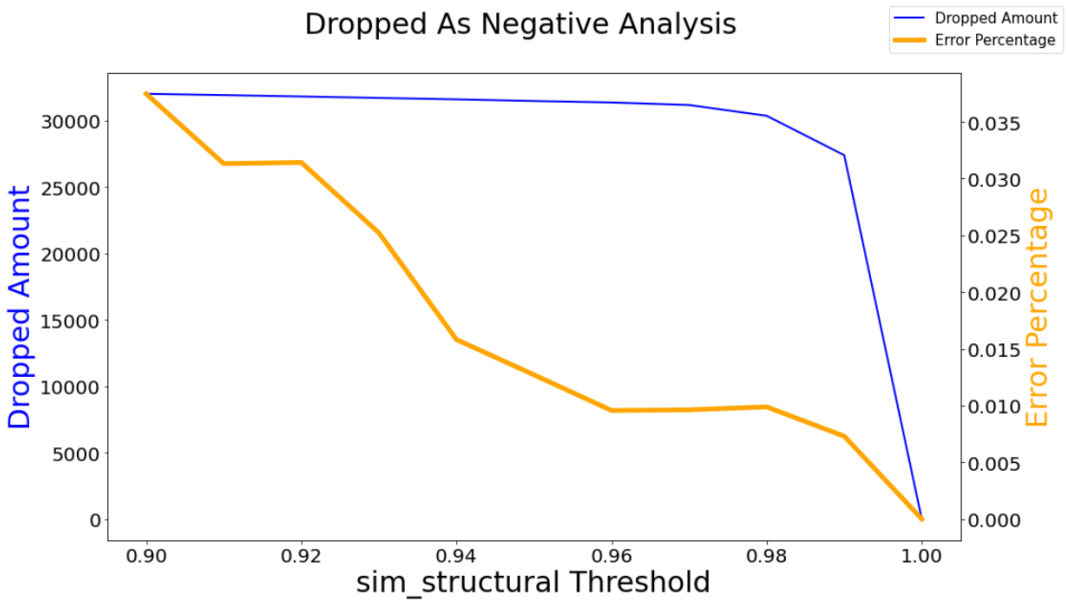


Figure 9: Dropped amount and error percentage of different *sim_structural* thresholds. When increasing the threshold, both the dropped amount and error rate decreased.

Computed Metrics	Full Corpus Processing Time (seconds)	Ratio	Single Lecture Time (seconds)
Early Dropping	443	2.2%	65.8

Table 5: Time requirement for *Early Dropping* metric computation strategy.

The next section returns to one of our original motivations for automated scene detection - book creation. We reiterate that digital book of recorded video content has the potential to provide an alternative, yet equivalent high-quality learning method that can benefit all students but may particularly benefit students with disabilities and students with limited access to IT video-capable resources or read online despite limited internet availability in impoverished and rural areas. Using the approach described above it was possible to create a digital book that automatically includes extracted images and transcribed text content from each lecture video. Digital books can also be printed as physical books. For example, a book can be printed to provide printed materials for incarcerated persons who have limited or no access to video content.

5 Digital book generation

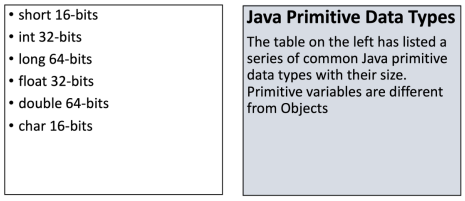
5.1 Chapter Image

The scene detection algorithm segmented the video into separate visual scenes. A scene change could be any visual or text content change that occurs during the video lecture, such as flipping a PowerPoint slide or moving to a new page on hand-written notes. After scene detection, a video has been segmented into a series of clips, each of which is defined as a *chapter*. Each digital book contained several *chapters* and each *chapter* included generated images and text content.

To avoid scrolling artefacts the middle frame between two scene change points (two boundaries of the clip) was selected as the exemplar frame of each scene.

In addition to visual content, the transcription of the audio clip inside each video scene segment was transformed into a book-like format.

Detected Title: "UNIVERSAL DESIGN FOR LEARNING"



Detected Title: "Java Primitive Data Types"

Figure 10: Examples of more challenging images for the Title Detection detection. The first one has a title of 2 lines. The second one has a title on the right middle of the screen.

5.2 Chapters Title Generation

Using OCR output - the positional, text content, and size properties of the extracted text, and combined heuristics to represent common slide layouts it was possible to suggest likely title text for each chapter.

First, a candidate word that had the maximum likelihood of being in a title sentence was located. Using the Tesseract OCR[21] library, the coordinates of the bounding box for each word were extracted. Note, simply finding the largest bounding box of the extracted text was an insufficient heuristic because the bounding box varied depending on the presence of ascender and descender letters. For example, text of the same font size with descenders, ascenders, or both (e.g., "gj", "TP", "Ty" respectively) would have a larger bounding box than a word with neither (e.g., "can").

The relative position of the word was also used to identify title words. The title sentence usually appeared near the middle top of a presentation slide. Words were assigned a score based on font size, vertical and horizontal location. After combing all three heuristics – font size, vertical position, and horizontal position – candidate word that had the best combined score is selected. Please see the source code for full details. Once a single word was identified as part of the title the algorithm searched for words in the same or nearby line with a similar font size as that of the candidate word.

In the corpus of 9 videos this relatively simply heuristic-based algorithm was surprisingly robust, though formerly evaluating this is outside the scope of this work. Figure 10 demonstrates two more-challenging examples that were still successful for the title detection algorithm. The left image had a title of more than 1 line. The right slide had a slide with a title not centered on the left top, but on the right middle.

The predicted titles were then inserted into the beginning of each *chapter*. ClassTranscribe's users were able to edit the detected titles in case the suggested titles were inaccurate.

5.3 User Interface

Figure 11 and Figure 12 are examples of the user interface when editing the digital book. Figure 11 represents the Book Structure Editing Interface, where the user can combine and split the automatically generated chapters. To be specific, there are altogether 4 working areas in this interface. Working area 1 is a bar of several buttons of useful functionalities, such as renaming the book, downloading the book, and saving the current progress. Working area 2 is a preview of the current chapter structure with their titles. In working area 3, users can further combine or split chapters accordingly if the Scene Detector does not function as expected. Working area 4 is a preview of the images and text content of the currently selected chapter.

Figure 12 represents the Chapter Editing Interface, where users can edit different content in the chapter. In working area 1, users can add or remove images, change the title, and edit text content. The text content is initially the transcript of the video segment that corresponds to the chapter or sub-chapter.

Course lecturers can simply upload a newly recorded video lecture to `classtranscribe.illinois.edu`

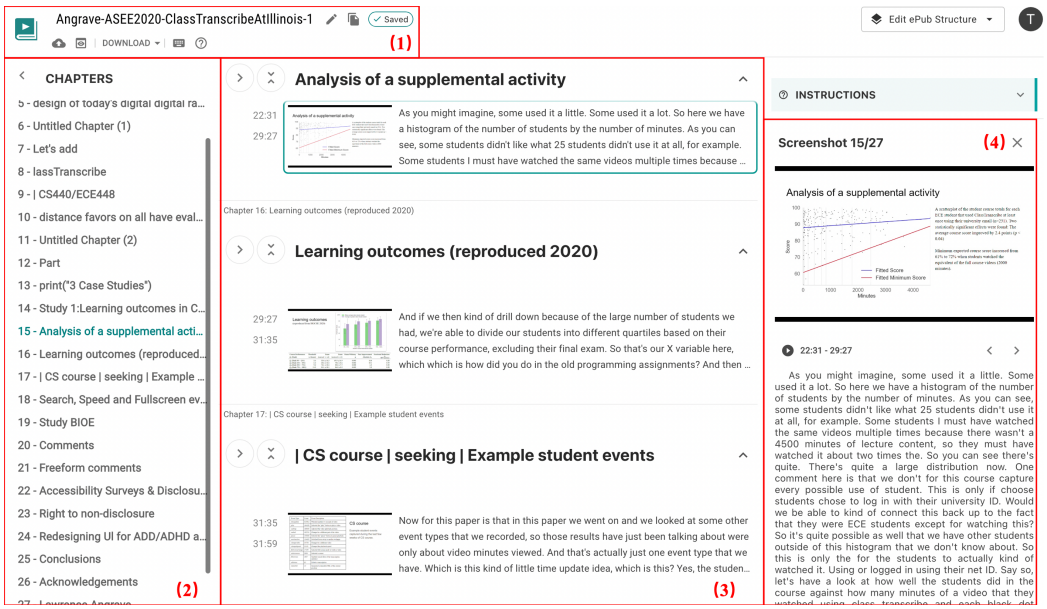


Figure 11: Digital Book Structure Editing Interface. The user is presented with the output of the scene detector and can select which images should be used as new chapters or new sub-chapters.

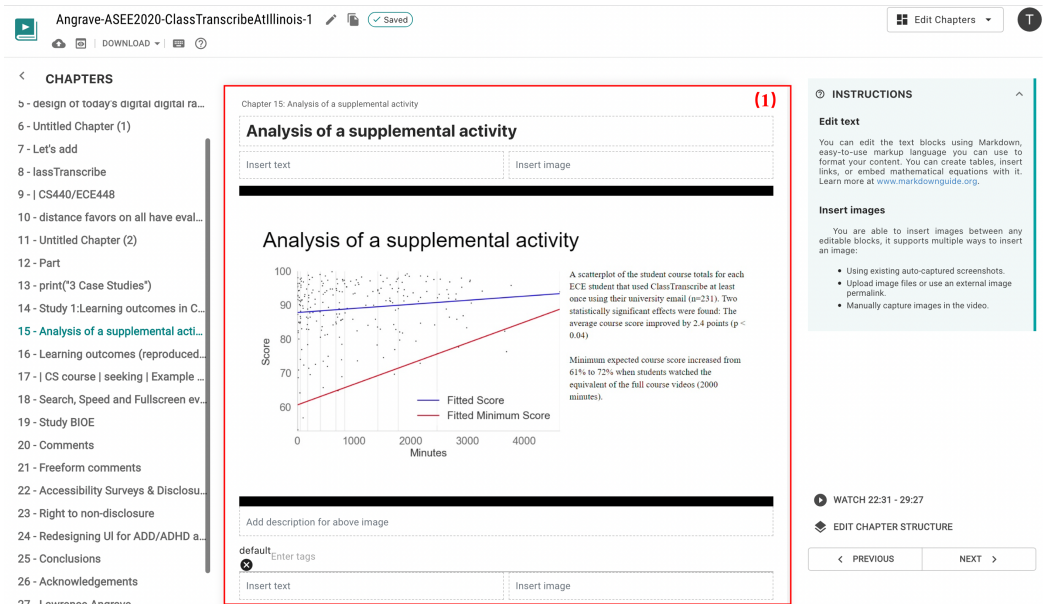
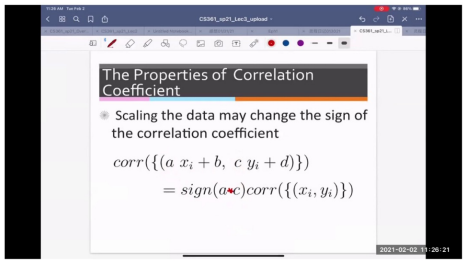


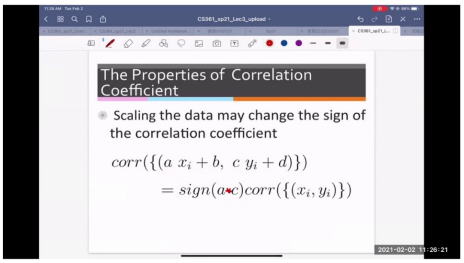
Figure 12: Digital Book Chapter Editing Interface where content can be added and edited.

30 The Properties of Correlation Coefficient



You can prove that very easily and you can see that if you scale the data. The correlation coefficient would only possibly change the sign or remain the same. But the value the magnitude will be the same. I wouldn't call it scale parameter because this is a sign. So either positive one or negative one. No matter how big or how small a NCR, it's a sign. Yes 8 * C. This is a time seat. And this is a sign. So you see that when you look at the two data features, we don't care about the unit and location anymore and the correlation coefficient would only give you any quantitative or value that indicate how there are there related together in terms of the trend or assign, it doesn't show. The value doesn't care about the value of X&Y. Magnitude. Yeah, scaling only affected it's negative or positive, yeah? The magnitude will be the same. You can prove this. Yeah, the magnitude is the same and then this is negative or positive one. If NCR positive are the same sign, then you have positive correlation.

31 The Properties of Correlation Coefficient



So that they can prove that using the definition of the correlation coefficient. So now let's talk about this next property. That is a very interesting correlation. Coefficient is always bonded within negative one and one. That's why we call it coefficient. It's always between negative one and one. When does it have one? When does it have a negative one? When the correlation coefficient is one, we know that these two features after you standardize them, they're the same. If you know that correlation is negative, one that obtuse angled a lie, we know that as I had, is negative why I had. We know that this is a case that correlation is negative one. So basically this means a linear perfect linear positive linear relationship, perfect negative linear relationship. We can prove that.

Figure 13: A two page sample from a digital book created from a lecture video.

and click "Create New Book" to generate a content-equivalent digital book from that video. Once the digital books are generated, they can be immediately downloaded in a desirable format by clicking the button in the bar. Popular download formats (pdf, html, and epub) are supported.

5.4 Sample Book

Figure 13 gives two example pages extracted from a generated book of a computer science course. The digital book interface of ClassTranscribe, how the generated books were further enhanced to improve the accessibility for students with disabilities, and a survey of student textbook needs are discussed in the accompanying ASEE 2022 paper [10].

5.5 Limitations

Some lectures are hierarchically structured chapters that include sub-chapters. It is desirable to detect the segment hierarchy, split each segmentation into sub-segment, and structure the book based on both chapters and sub-chapters. The current scene detector implementation only suggests chapters and does not create a hierarchy of sub-chapter content.

Of the corpus of 9 videos from the University of Illinois that were reviewed handwritten content that used scrolling in digital notebook interface generated the most false positives (due to the limitations of pixel similarity matrix and ability to extract reliable text from each slide). Optimizing scene detection for this content style would benefit from additional research.

6 Content reuse, remixing and transformation: From TikTok and Memes to a equitable pathway of accessibility for everyone

Scene detection allows new representations from existing lecture content. With the key moments extracted as individual image files in to single directory, additional content creation and applications become

possible. These include integration into larger platforms for the creation of an equivalent digital book (and this is explored further in the ASEE 2022 paper [10]), and using a single image or all images as source for creative content creation and accessibility-related applications.

As an example of a creative application we assembled short (approximately 10-second) animations of lecture videos. We defined this as our "TikTok Video" experience because it was inspired by the short videos on the social media website that has become popular format for sharing videos among students. Lecture videos compressed to 10-seconds are not designed for learning the content but are a novel approach to engage students using a video modality that they watch and share daily. The shell script to automatically assemble a 10-second video is available at https://uofi.box.com/v/SceneDetection-ASEE2022-shared.

Memes are amusing popular image formats that typically contain a figure and title text, and are often shared on social media websites, instant messaging, and phone applications. An image from the scene detection output can be repurposed as a meme image by the addition of large-font text, or by inclusion of an image into a standard meme format (for example, see Figure 14). The authors suggest the primary utility of this method is to engage and inspire students. The authors note that memes are purely a visual representation so it is important to provide an alternative text to describe the image.

Scene extraction also benefits students with visual and hearing difficulties, which we discuss next.

The ability to extract each scene from a video is advantageous to students who are hard of hearing or deaf because it can improve the accuracy of captions and transcriptions. The ClassTranscribe video platform performs automated text extraction using the OCR output from each scene which is used to create a set of phrase hints. The phrases hints are sent along with the audio to Microsoft Azure Cognitive Services to perform speech to text. The response is used to create captions and transcript of the audio stream. By identifying domain words presented visually from the extracted scenes and creating phrase hints for the speech to text service, the accuracy of the captions was improved (e.g., it could identify and correctly spell words that were likely not part of the original training data, e.g., "Clausius–Clapeyron relation," or was more likely to correctly identify the spoken audio phrases that were also presented in text form in the video e.g. "Electron Beam Microprobe Analyzer").

Scene data provides a starting image data set for extended audio description and equivalent text generation too, which, if created, provide the semantic equivalent of the visual content for students with low vision or who are blind. The scene image is converted into associated textual description (or multiple descriptions of varying complexity). For example audio description text can then be later rendered as audio by a volunteer narrating the text or by using automated text to speech technology. A full discussion of enhanced audio description and is beyond the scope of this paper but the interested reader is referred to [20]. Automated creation of a text description of images (e.g., "A bar graph" , "A circuit diagram", do not yet convey desired, useful and insightful pedagogical information. Thus, the creation of equivalent text and audio descriptions is still a manual task that is often overlooked in course content creation; for example it appears that faculty may equate the existence of captions, accurate or not, as sufficient to me accessibility and equity goals. However, with the availability of scene images that can now be extracted from engineering videos there are new opportunities – some authors of this paper suggest a mandate – to develop and share inclusive educational practices that includes the generation of audio descriptions and equivalent textual content. For example, if the scene detection output includes an image from a video of an equation or computer code then these items should be usable directly as an equation or editable and executable code, respectively.

7 Student Feedback

We demonstrated the new digital book feature to a small number (N=10) of engineering undergraduate students from Physics, Computer Science and Neural Engineering. Their informal comments and feedback included, "May save the time... It's a good technique for preserving ... valuable lecture contents, which is extremely helpful in a situation like pandemic ... shortens the time of students' recitation of course videos, and facilitates the collation of electronic notes. Moreover, the design of web interface is simple and generous... reading a book is much more efficient than watching a video, and it's easy to take notes by copying the content directly to our notebook... It gives instructors an alternative and customized way to teach, providing students with an auxiliary learning method... The digital book and the editing feature are

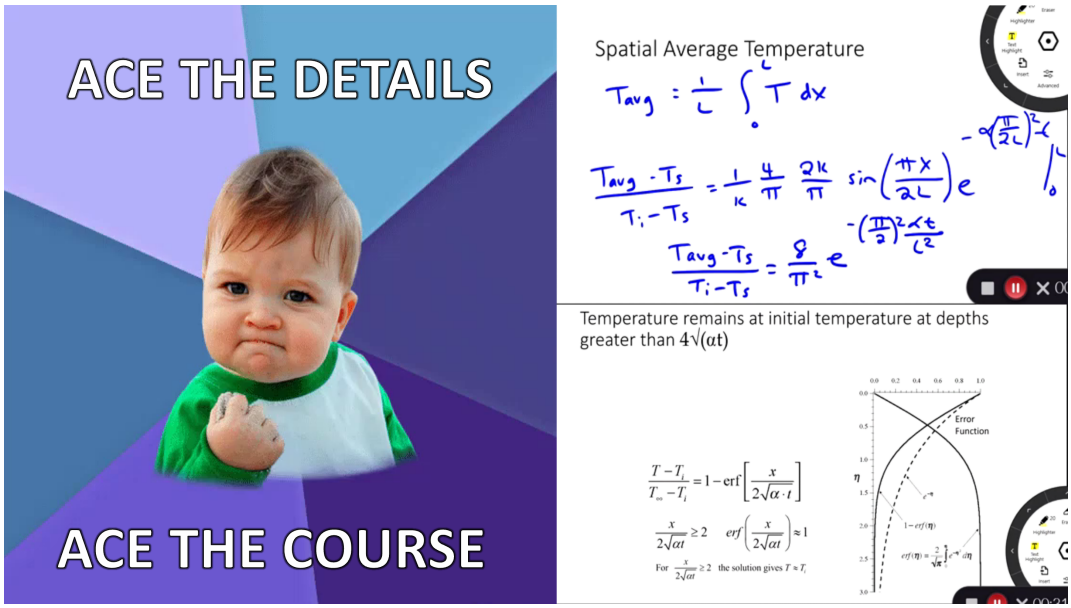


Figure 14: Example of a meme with large-font text juxtaposed with lecture content on the right into a single shareable image. For accessibility, the accompanying caption, or the Alt-text description – which should be accurate and succinct – could be, "Success-Kid meme and course slides; Ace the details - Ace the course."

very useful for student for note-taking... The user interface is intuitive and easy to understand... Students can save time with a condensed version of the lecture video". All students shared positive opinions towards the experience of editing and using digital books as a learning resources.

Some students suggested that 1) the layout of the output book is overly complicated and could be improved, 2) more languages should be supported by the digital book feature, 3) drag and drop editing for book structure should be supported, and 4) words extracted from speech recognition with low confidence level should be highlighted so that users can easily identify and modify them when needed. We look forward to reporting on students' use of the digital books for learning in a future paper.

8 Conclusion and Future Work

This paper introduced a new approach to the scene detection problem which was optimized for common types of engineering lecture videos. Scene detection that is fast and provides high precision and recall is an important goal because it provides a foundational role for automatic digital textbook generation process and other accessibility related processes. The trade-offs in accuracy, performance and computational resources required were analyzed. An evaluation of our framework found 93.24% in recall, 76.09% in precision, and 99.47% in overall accuracy. Scene detection remains an open problem, however, for example there is room for improvement for lecture videos that use incremental annotation using digital ink or chalk.

The scene extraction python tool is available as a source library on GitHub https://github.com/classtranscribe/SceneExtractor-2022 and has also been embedded into the ClassTranscribe web application where it was used to create digital books from existing lecture videos. The former extracts a sequence of images from one or more videos into a local directory which can be further combined with other videos, used as basis to create content-driven memes, or visual index, to further reach, engage and inspire students. The image set can also be used to create audio descriptions and alternative text for students who are blind or have low vision, to further make engineering courses more inclusive and accessible.

Our approach and open-source implementation are adaptable; they can also be embedded into other video-based educational platforms and workflows. Finally, we hope the promising results of our work will inspire more innovative advancements in creating inclusive, accessible educational environment for all students.

Successful scene detection for engineering videos enabled the creation of equivalent digital book content, which is of value from the perspective of Universal Design Learning, to support multiple learning pathways and modalities. We discussed and demonstrated other content transformations that scene detection supported including phrase hinting and audio descriptions for equity and video and meme creation for engagement and review. We encourage the ASEE community to adopt and explore the ideas and practices outlined in this paper, in particular to find novel ways to remix and reuse content for engagement and accessibility. Future research opportunities were also discussed including improving scene detection for scrolling content, developing best practices for creating and using audio descriptions and equivalent content in engineering courses, and research into optimizing the accuracy of automated caption and transcript generation using phrase hinting from lecture content.

The ClassTranscribe video platform is available for use by instructors at other institutions to create digital books; please email classtranscribe@illinois.edu for further information. We look forward to discovering how the ASEE community and engineering students will use the scene extraction tool and the ClassTranscribe web application both pedagogically and creatively.

9 Acknowledgements

The research reported here was supported by a Microsoft Corporation gift to the University of Illinois as part of the 2019 and 2020 Lighthouse Accessibility Microsoft-Illinois partnership, GIANT award (GIANT2021-03) from the IDEA institute [22], an award from Center for Innovative Teaching and Learning, and the Institute of Education Sciences, U.S. Department of Education through Grant R305A180211 to the Board of Trustees of the University of Illinois. The opinions expressed are those of the authors and do not represent views of the Institute or the U.S. Department of Education. We acknowledge support of NCSA, and the many former students who worked on the ClassTranscribe source code and early implementations of the scene detection and phrase hinting code including, Akhil Vyas and Vicky Cai.

References

[1] Lawrence Angrave et al. "Improving Student Accessibility, Equity, Course Performance, and Lab Skills: How Introduction of ClassTranscribe is Changing Engineering Education at the University of Illinois". In: *2020 ASEE Annual Conference*. June 2020. DOI: 10.18260/1-2--34796.

[2] Lawrence Angrave et al. "Who Benefits? Positive Learner Outcomes from Behavioral Analytics of Online Lecture Video Viewing Using ClassTranscribe". In: *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. New York, NY, USA: Association for Computing Machinery, 2020, pp. 1193–1199. ISBN: 9781450367936. DOI: 10.1145/3328778.3366953. URL: https://doi.org/10.1145/3328778.3366953.

[3] Alireza Avanaki. "Exact global histogram specification optimized for structural similarity". In: *Optical Review* (2009). DOI: https://doi.org/10.1007/s10043-009-0119-z.

[4] Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara. "A Deep Siamese Network for Scene Detection in Broadcast Videos". In: *Proceedings of the 23rd ACM international conference on Multimedia* (2015).

[5] Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara. "Analysis and Re-Use of Videos in Educational Digital Libraries with Automatic Scene Detection". In: *IRCDL*. 2015.

[6] Margaretha Izzo and William Bauer. "Universal design for learning: enhancing achievement and employment of STEM students with disabilities". In: *Universal Access in the Information Society* 14 (Mar. 2013). DOI: 10.1007/s10209-013-0332-1.

[7] Devin Larson. *STANDARD PROPORTIONS OF THE HUMAN BODY*. URL: https://www.makingcomics.com/2014/01/19/standard-proportions-human-body/. (accessed: 01.22.2022).

[8] Chao Liang et al. "A Novel Role-Based Movie Scene Segmentation Method". In: vol. 5879. Dec. 2009, pp. 917–922. ISBN: 978-3-642-10466-4. DOI: 10.1007/978-3-642-10467-1_82.

[9] Github User linxiaohui. *mtcnn-opencv*. https://github.com/linxiaohui/mtcnn-opencv. 2021.

[10] Hongye Liu et al. "A Digital Book Based Pedagogy to Improve Course Content Accessibility for Students with and without Disabilities in Engineering or other STEM courses". In: *2022 ASEE Annual Conference*. June 2022.

[11] Chirantan Mahipal et al. ""What did I just miss?!" Presenting ClassTranscribe, an Automated Live-captioning and Text-searchable Lecture Video System, and Related Pedagogical Best Practices". In: *2019 ASEE Annual Conference*. June 2019. DOI: 10.18260/1-2--31926.

[12] Stephanie Moore. "David H. Rose, Anne Meyer, Teaching Every Student in the Digital Age: Universal Design for Learning". In: *Educational Technology Research and Development* 55 (Oct. 2007), pp. 521–525. DOI: 10.1007/s11423-007-9056-3.

[13] Rameswar Panda, Sanjay Kuanar, and Ananda Chowdhury. "Nyström Approximated Temporally Constrained Multisimilarity Spectral Clustering Approach for Movie Scene Detection". In: *IEEE Transactions on Cybernetics* PP (Feb. 2017), pp. 1–12. DOI: 10.1109/TCYB.2017.2657692.

[14] Arti Patle and Deepak Singh Chouhan. "SVM kernel functions for classification". In: *2013 International Conference on Advances in Technology and Engineering (ICATE)* (2013), pp. 1–9.

[15] Stanislav Protasov et al. "Using deep features for video scene detection and annotation". In: *Signal, Image and Video Processing* 12 (July 2018). DOI: 10.1007/s11760-018-1244-6.

[16] Daniel Rotman, Dror Porat, and Gal Ashour. "Robust and Efficient Video Scene Detection Using Optimal Sequential Grouping". In: *2016 IEEE International Symposium on Multimedia (ISM)* (2016), pp. 275–280.

[17] Daniel Rotman et al. "Learnable Optimal Sequential Grouping for Video Scene Detection". In: *Proceedings of the 28th ACM International Conference on Multimedia* (2020).

[18] Panagiotis Sidiropoulos et al. "Temporal Video Segmentation to Scenes Using High-Level Audiovisual Features". In: *IEEE Transactions on Circuits and Systems for Video Technology* 21 (Aug. 2011), pp. 1163–1177. DOI: 10.1109/TCSVT.2011.2138830.

[19] Alexandre Siqueira et al. *Scikit-image*. https://github.com/scikit-image/scikit-image. 2021.

[20] University of South Carolina. *Writing Audio Descriptions*. URL: https://www.sc.edu/about/offices_and_divisions/digital-accessibility/guides_tutorials/audio_descriptions/writing-audio-descriptions/index.php. (accessed: 05.13.2022).

[21] Github User stweil. *Tesseract OCR*. https://github.com/tesseract-ocr/tesseract. 2022.

[22] *The Grainger College of Engineering Institute for Inclusion, Diversity, Equity and Access*. URL: https://idea.illinois.edu/.

[23] Zhou Wang et al. "Image Quality Assessment: From Error Visibility to Structural Similarity". In: *IEEE TRANSACTIONS ON IMAGE PROCESSING* 13 (Apr. 2004). DOI: 10.1109/TIP.2003.819861.

[24] Kaipeng Zhang et al. "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks". In: *IEEE Signal Processing Letters* 23 (Apr. 2016). DOI: 10.1109/LSP.2016.2603342.

[25] Zhilin Zhang et al. "How Students Search Video Captions to Learn: An Analysis of Search Terms and Behavioral Timing Data". In: *2021 ASEE Virtual Annual Conference*. https://peer.asee.org/37257. ASEE Conferences.