2020 25th International Conference on Pattern Recognition (ICPR)
Milan, Italy, Jan 10-15, 2021

Automated Whiteboard Lecture Video Summarization by Content Region Detection and Representation

Bhargava Urala Kota, Alexander Stone, Kenny Davila, Srirangaraj Setlur, Venu Govindaraju
Dept. of Computer Science and Engineering
University at Buffalo, State University of New York, Buffalo, NY, USA
Email: [buralako, awstone, kennydav, setlur, govind]@buffalo.edu

***Abstract*—Lecture videos are rapidly becoming an invaluable source of information for students across the globe. Given the large number of online courses currently available, it is important to condense the information within these videos into a compact yet representative summary that can be used for search-based applications. We propose a framework to summarize whiteboard lecture videos by finding feature representations of detected handwritten content regions to determine unique content. We investigate multi-scale histogram of gradients and embeddings from deep metric learning for feature representation. We explicitly handle occluded, growing and disappearing handwritten content. Our method is capable of producing two kinds of lecture video summaries - the unique regions themselves or so-called key content and keyframes (which contain all unique content in a video segment). We use weighted spatio-temporal conflict minimization to segment the lecture and produce keyframes from detected regions and features. We evaluate both types of summaries and find that we obtain state-of-the-art peformance in terms of number of summary keyframes while our unique content recall and precision are comparable to state-of-the-art.**

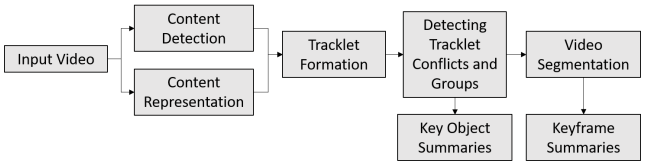

Fig. 1: Our pipeline for producing keyframe and key object summaries is shown above. We first take every frame of input video and detection content regions using a neural network, features are obtained from detected regions and this information is used to form an initial tracklet graph, where each node corresponds to a unique content region. Then we detect conflicting region tracklets as well as group occluded/growing/disappearing tracklets. At this point, we generate all unique groups of tracklets as key object summaries. Further, we use the conflict information to segment the video and produce a keyframe per segment.

I. Introduction

Educational videos are an abundant resource thanks to the proliferation of Massively Online Open Courses and university online offerings. These resources hold the promise of democratizing education by ensuring that distance and economy do not prevent access to quality content. It has been shown that breaking down a lecture into topically coherent segments increases viewer engagement [1]. Further, studies have found that students tend to return multiple times to certain points in the video to review concepts [2]. Thus, there is a need to *summarize* the content so that a student can a) quickly decide which lecture to view and b) navigate to specific content.

Information extraction from lecture videos is a challenging problem. Several lectures are self-recorded and do not have extensive production and transcript annotation and often involve a single fixed camera covering the whiteboard. Thus, there is a need for a pipeline to summarize lectures via visual text as opposed to text transcript. Further, whiteboard lectures tend to be a semi-structured arrangement of phrases, math expressions and sketches making direct recognition challenging. Therefore, there is a need to explore methods for handwritten content representations such as visual features.

In our work, we use term 'summarization' to describe video summarization by keyframes which is the process of reducing a video to a subset of frames that capture all of the content within the video [3]. This method of summarization has been adapted in both presentation and whiteboard lecture videos. Some methods for presentation videos extract image descriptors from consecutive frames and detect slide transitions by measuring sharp changes in frame descriptors [4, 5]. While this approach has been used for whiteboard lectures [6], it was found to be susceptible to noise induced by presence of lecturer as well as by handwritten content changing gradually over time, as opposed to instant slide changes.

Thus, state-of-the-art whiteboard summarization approaches [7, 8] use the paradigm of content detection and binarization, tracking binary connected component (BCC) lifetimes across the video using pixel-wise overlap as matching metric and finally segmenting the video based on these lifetimes. In particular, Davila and Zanibbi proposed *conflict minimization* - a greedy algorithm to segment a lecture video, based on spatial conflicts between dissimilar BCCs [7]. However, such approaches typically use a specialized binarization network that needs to be trained on lecture data [9, 10]. In this work, we propose a region-based framework that does not need explicit binarization to perform summarization.

Content regions need robust descriptors to represent the con-

978-1-7281-8808-9/20/$31.00 ©2020 IEEE

2020 25th International Conference on Pattern Recognition (ICPR) | 978-1-7281-8808-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICPR48806.2021.9412386

tent. Accurate recognition of the content region would yield a satisfactory descriptor, however, it is not always feasible due to lack of annotation. Furthermore, the variety of handwritten content encourages us to explore visual descriptors for whiteboard lecture content. In our prior work, we investigated deep metric learning methods to produce these descriptors. Although this method showed promise, it was used to produce 'key object' summaries and *not* keyframe summaries. In our current work, we investigate high level descriptors obtained from deep metric learning techniques and low level descriptors (using Histogram of Gradients) and propose a framework to produce *both* keyframe and key object summaries.

Methods for associating detected text regions across a video have been shown in video scene text literature using a mixture of recognition and visual features [11], however in whiteboard lecture videos, there is a need to be aware of occlusions of text due to lecturer as well as handling growing/disappearing content due to writing/erasure events. In our work, we propose a graph-based technique that can perform association and conflict detection of content regions even under occluded and partial conditions.

The evaluation of whiteboard lecture video summarization involves two metrics, size of produced summary and extraction of content with respect to ground truth. The goal is to produce as compact a summary as possible (usually measured in terms of number of summary objects) while extracting all the unique content within the lecture video (usually measured in terms of recall and precision). In our work, we introduce *weighted conflict minimization* by modifying the original algorithm proposed by Davila and Zanibbi [7] in order to obtain a finer control on the trade-off between the two evaluation metrics.

Our contributions in this work, are as follows:

- A content region-based summarization method that produces both keyframes and 'key objects' as summaries.
- An efficient graph-based tracking scheme for associating content regions and detecting content conflicts which is robust to occluded, growing and disappearing content.
- A modified scheme of conflict minimization which allows for a fine-grained control on the trade-off between compactness of summary and recovery of unique content within summaries.

Figure 1 shows an overview of our summarization method which is explained in detail in Section III.

II. Background

General Video Summarization: Truong and Venkatesh have broadly classified video summarization approaches into static (*keyframes*) and dynamic (*one or more skims*) [3]. Keyframes are a set of frames from the video that capture the most important content whereas skims are short segments of the videos that capture the most important events. Recently, Meng et al. [12] proposed a method to summarize videos by so-called *key objects*, obtained using representation selection methods from candidate detections on all video frames. Summarization methods could be supervised, where each frame has an importance score allotted by experts which can be used to train models. In unsupervised methods, frame representations are extracted and dynamic programming or similar approaches are used to maximize coverage or minimize summary size [13].

Whiteboard Lecture Video Summarization: Most recent approaches summarize lecture videos by providing a set of keyframes that capture all unique content. Lecture summaries are of two types (i) *extractive*,- focusing on all content elements; (ii) *abstractive* - focusing on content elements relevant to search queries or user navigation.

Most whiteboard approaches are extractive and are evaluated by number of keyframes and recall, precision of content [5, 7, 8, 10, 14, 15, 16, 17]. Summarization of content is carried out by analysis of detected content (typically binary connected components) [7, 8, 10] by minimizing a global objective (content conflicting for the same space at different times) or using local content difference [15]. Further, labelling every frame with an importance score is challenging, therefore, unsupervised methods are preferred.

Whiteboard lecture videos are typically preprocessed by background removal and binarization followed by content extraction and summarization [7, 16, 18]. After preprocessing, handwritten content is extracted and grouped into meaningful sets, primarily using spatiotemporal cues [7, 8, 10, 14, 16] or OCR [16]. Explicit modelling of lecturer actions have also been used for content extraction and video segmentation [15, 17]. Neural networks have recently been used for direct content extraction [8, 10, 19, 20].

Recently, we proposed a triplet loss based embedding for representing content regions (as opposed to binary connected components in prior work), method to summarize lectures by *key content* and a corresponding evaluation scheme [21]. In our current work, we use neural network to detect content and investigate low-level descriptors and deep metric learning to extract features from content. We then propose a framework sensitive to occluded, growing and disappearing content to obtain *both* keyframe and key content summaries.

Text Representation in Images and Videos: While there have been many methods based on direct recognition [22, 23] and learning vector representations of recognized text [24, 25], there are few works that use visual features to represent detection regions. Phan et, al. use SIFT and Stroke Width Transform descriptors matching to align regions across frames and augment recognition based representation [26].

Dataset and Evaluation: AccessMath is the largest, publicly available, benchmarked dataset for whiteboard lecture video summarization [7]. It consists of 12 lecture videos (5 training and 7 testing), recorded with a single still camera at 1920×1080 resolution spanning the whole whiteboard. AccessMath consists of ground truth summary keyframes and is evaluated by the average number of keyframes produced and the average recall and precision of all binary connected components (CC) in the summary as well as in all frames of

the video. The matching scheme for binary CCs is detailed along with benchmarking procedure by the creators of the dataset [7] and allows split and merged matches. Additionally, content region ground truth bounding boxes are also provided. The boxes are drawn around content that is created and erased at roughly the same time [8]. In our previous work [21], we proposed a method to summarize whiteboard lectures by key content regions. Further, an evaluation scheme based on DetEval [27] was proposed to evaluate summary regions. In this work, our method generates both key content and keyframe summaries from the same framework.

III. Lecture Video Summarization

We summarize whiteboard videos by detecting and representing handwritten content regions on the video frames. The feature representations are used to group and track lifetimes of each unique region within the video. We use partial region features to ensure occlusions due to lecturer movement and additions/deletions made to content are correctly handled during tracking. Finally, we propose a scheme to weigh the so-called spatio-temporal content conflicts (STCC) i.e. two different tracklets occupy the same region of the whiteboard but at different times. The video is then segmented such that the total weight of STCCs within all segments is minimized and a binarized keyframe is constructed for each segment using all tracklets that are active within that segment. Figure 1 shows an overview of this pipeline. The values of hyperparameters used in different stages are specified in Section IV.

A. Content Detection

We adapt PSENet, a deep neural network model proposed by Wang et. al. [28] to detect whiteboard content. This network is initialized using scene text detection weights and trained on lecture videos annotated at the bounding box level [8]. The architecture of the neural network is as follows:

Feature Extractor Block: employs the Feature Pyramid Network (FPN) with ResNet-50 backbone. It consists of a downsampling path with consecutive blocks of convolutional layers with residual connections and an upsampling path using deconvolution layers. Feature maps at downsampling stage are smoothed via lateral convolutional layers and concatenated with the corresponding upsampled feature maps. Finally, all upsampled feature maps are concatenated along the channel axis to produce multi-scale features that have proven effective for object and text detection tasks [28, 29].

Classification Block: a two layer convolutional network that operates on the concatenated upsampled feature maps to produce k dense pixel level text/non-text prediction mask. During training, the masks are trained against k target masks generated using increasing amount of shrinking applied to the ground truth text area polygons. During inference, a region growing algorithm starting with the mask with the highest amount of shrinking is applied progressively for all k maps to obtain text masks and polygons using connected components.

TABLE I: Quantitative evaluation of isolated handwritten content detector in our prior work and current work by measuring pixel-wise recall and precision against test set video frames.

Method	Frame-wise AVG BBoxes	Pixel-wise AVG Rec.	Prec.	F-meas.
Prior work 1 [8]	12.25	81.87	76.20	76.48
Prior work 2 [10]	12.35	**88.43**	68.39	75.27
Current work	23.69	86.75	**84.62**	**85.68**

TABLE II: Evaluation of feature extraction methods by measuring area under the curve (AUC) of the ROC curve

Method	AUC
Deep Metric Learning	85.82
Histogram of Gradients	89.10

This detection architecture was shown to be successful at handling text of different scales, orientations and layouts including curved text while the progressive expansion allows the network to handle instance separation when text is close by or overlapping. Both of these properties are desirable in whiteboard lectures where the content is loosely structured. Finetuning on lecture data was necessary since the content type includes math expressions and figures which are not typically seen in scene text datasets.

PSENet is finetuned on the AccessMath [7] training lecture videos for detection of handwritten content on frames sampled at 1 frame per second. In order to save time, bounding box annotations for this dataset were created by marking the frame when the lecturer completes writing a unit of content and the frame when the lecturer begins to erase the same content [8]. This results in regions that are varied in layout (multi-line, sketches with text labels) and content type (math expressions, phrases, words, sentences, sketches). During annotation, every unit of content is assigned a unique ID. Training procedure and hyperparameters are detailed in Section IV. After training, the learnt weights are used to predict regions in the AccessMath test lecture video frames sampled at 1 frame per second.

B. Feature Representation

Given the lack of recognition labels for supervised training and the variety of text content including math expressions and sketches, we have opted for a visual feature representation approach. We compare a histogram of gradients (HoG) based baseline feature extractor with deep metric learning methods learnt using the training set of the AccessMath dataset.

1) Histogram of gradients: Every detected region is reshaped into multiple aspect ratios in order to extract features robust to shape and size variations. In this work, we have chosen 32×32, 64×16, 16×64, 54×18, 18×54. HoG features with cell size 8×8, and block size 2×2 are extracted from each aspect ratio and concatenated. The normalized area of the input region is appended to these to form the final feature descriptor with 1189 descriptors. Table II shows the results of using the HoG features to perform matching of ground truth regions in the test video lectures.

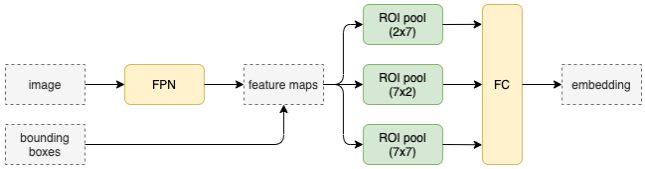

Fig. 2: Neural network structure for deep neural network feature extraction. The feature maps are extracted from the FPN block of our detector. Detected bounding boxes are used to crop regions from the feature maps and ROIPool layers are used to get fixed aspect ratio representations of cropped areas. These representations are flattened, concatenated and passed to a fully connected layer followed by L^2 normalization to produce the final feature embedding.

2) Deep metric learning: The feature extractor block of the PSENet trained for the detection task is re-used to compute detected region representations. The output feature maps with 256 channels are compressed to 64 channels using a 1x1 convolution layer. From here, we use the detected region bounding box to crop the feature map and resize it to 2x7, 7x2, and 7x7 regions using an ROI pooling layer. These features are flattened, concatenated and fed into a fully connected layer which is trained using multi-similarity (MS) loss [30].

MS loss is a deep metric loss which selects an anchor sample and minimizes the weighted distance to positive samples within a mini-batch and maximizes weighted distance to negative samples with the mini-batch. These weights are controlled using a hyper-parameter λ, and the loss function is designed to give higher weight to positive sample distances that are further away from the anchor and negative sample distances that are closer to the anchor. Like many deep metric learning methods, larger mini-batch size and effective sampling, and data augmentation is critical to ensure convergence.

We use the unique region IDs assigned during annotation to compose mini-batches with sufficient positive and negative samples. Data augmentation is carried out by slightly perturbing ground truth bounding boxes as well as color space variations. We make sure that each mini-batch is only composed of samples from a single lecture so that the unique IDs are not confused across lectures.

Table II shows comparison of HoG-based features and deep metric learning features on the AccessMath test dataset.

C. Spatio-temporal Content Graph Creation

We first provide an overview of our spatio-temporal analysis procedure before diving into the details. The content graph creation happens in two stages. In the first stage, an initial detection graph G_d is built where each detection is a node in the graph. Other metadata such as frame number, video time, bounding box etc. are also noted in the node. In the second stage, another graph G_t is composed where each node is a tracklet and consists of a list of nodes of G_d. Ideally, each tracklet node groups visually similar and spatially proximal detection nodes and can be considered as a representative for that region. After this, we consider all tracklet node pairs which have positive area overlap as interactions of interest.

For these pairs, we compute partial features (i.e. only from area of overlap) to compare similarity of the two regions. The pair is considered to be a spatio-temporal content conflict (STCC) if dissimilar, or is considered to be occluded, growing or disappearing content if similar. Conflicting nodes indicate that new content has replaced older content and is a cue for video segmentation. On the other hand, if there is no conflict, then the nodes are part of the same content and the video need not be segmented.

1) Initial graph formation: The detection network is used to predict handwritten content regions from every frame. Features are extracted from each detected region. An object detector trained on VOC2007 dataset is used to detect the speaker and any content detection node which spatially overlaps with the speaker is removed. Each remaining detection is considered node in a graph G_d. An edge is drawn between nodes if their normalized spatial distance and feature distances are below a threshold and they occur within $85s$ of each other. These thresholds are obtained using the ground truth detections from the training video lectures. We compute the feature distance $d^f_{ij} = \|f(r_i) - f(r_j)\|^2_2$ and spatial distance $d^s_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|^2_2$. Where, f represents normalized feature extraction and $\mathbf{x}_i$, $\mathbf{x}_j$ are the top-left and bottom-right corner coordinates of detection nodes n_i and n_j normalized with respect to frame height and width.

2) Tracklet nodes creation: Connected components are extracted from G_d which are designated as nodes of a new graph G_t. Each node of G_t represents a unique region tracklet t_i, consisting of a set of detection instances N_i and corresponding features F_i which ideally capture the same content. The metadata associated with these tracklet nodes are the lists of frame numbers, video times and bounding boxes corresponding to its constituent detection nodes. Further, we can compute representative features, lifetimes and bounding box for each tracklet node by aggregating constituent node metadata.

3) Characterizing tracklet node interactions: Illumination changes, occlusions due to speaker movement result in inconsistent detections. Thus, it is possible that different tracklets overlap in space or one completely contains the other. We exhaustively consider pairs of tracklet nodes t_i and t_j and if they overlap spatially, we compute the area of intersection $A_{ij} = A_i \cap A_j$ between aggregate bounding boxes of tracklets t_i and t_j, denoted as A_i and A_j respectively.

For every detection node $n^i_k \in N_i$ within the tracklet t_i, partial visual features $P_i = \{p^i_k\}$ are computed from the region A_{ij} using the meta-data information to obtain list of frame numbers to get the corresponding frame images. This process is repeated for every detection node $n^j_l \in N_j$ in tracklet t_j to obtain partial feature set $P_j = \{p^j_l\}$. Means and standard deviations of these features p^μ and p^σ are computed for tracklet pair t_i and t_j. Spatial overlap is measured using intersection over union $\text{IOU}_{ij} = \frac{A_{ij}}{A_i \cup A_j}$ as well as inter-

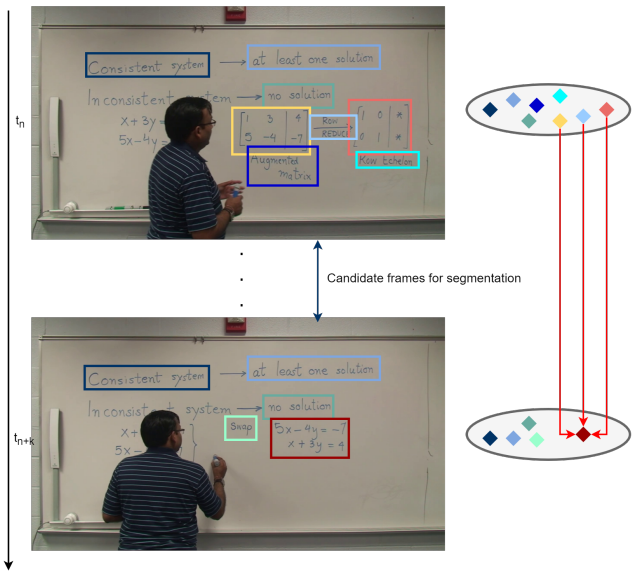

Fig. 3: Demonstration of conflicting regions across time. Two frames ordered in time from top to bottom are shown with a snapshot of the state of the tracklet graph, each node (tracklet) is indicated by a rhombus. Red arrows indicate spatio-temporal content conflict. Strength of each conflict is computed using the scheme in Section III-D. These conflicts are resolved by creating a video segment such that the two frames belong to different segments. Selected detected regions and nodes in the tracklet graph are shown for clarity.

section over minimum of aggregate tracklet bounding boxes $\text{IOMin}_{ij} = \frac{A_{ij}}{\min(A_i,\ A_j)}$.

Two nodes t_i and t_j are *grouped* if they overlap temporally and the difference between partial feature means $p^{\mu}_{ij} = |p^{\mu}_i - p^{\mu}_j|$ is low and the spatial overlap IOU_{ij} or IOMin_{ij} is high. This means that there is a common region between the two tracklets that have similar features and indicate occluded/growing/disappearing content regions. On the contrary, if the p^{μ}_{ij} value is high, this indicates that either one of the tracklets has merged multiple unique content regions into one; then tracklet with the higher standard deviation p^{σ} is marked to be split. If there is no temporal overlap, tracklets which overlap spatially and are have distant visual features are marked as *conflicting* to denote that they occupy the same space on the whiteboard at different timestamps. An edge is drawn between t_i and t_j if they need to be grouped or are in conflict. Edge metadata also includes difference between feature means p^{μ}_{ij} and area of overlap A_{ij} normalized to image area. These values are noted in order to later compute weight of the conflict/grouping. Examples of conflicting and non-conflicting regions are shown in Figures 3 and 4 respectively.

D. Video Summarization

At this stage of the algorithm we can generate two kinds of summaries - *key content* and *keyframes*. Key content regions can be computed by using the partial feature difference and

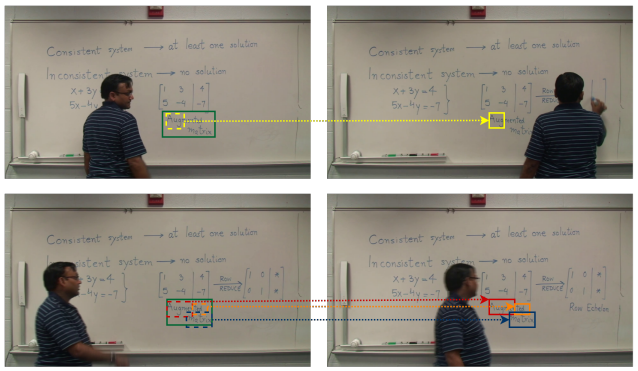

Fig. 4: Occlusion due to lecturer movement or growing handwritten content produces tracklets that cover overlapping regions. Direct feature comparison would produce different tracklets and could lead to oversegmentation during the summarization as well as lower content recall and precision. By extracting partial features from regions of overlap, we merge these separate tracklets into the same. The frames above are ordered in time starting from left to right on the top row and then left to right on the bottom row.

area of overlap between nodes stored as edge metadata in G_t. This computed weight allows us to control the number of final content regions produced. This summary is evaluated by average recall with respect to ground truth content regions at intersection-over-union (IOU) tolerance of 0.5 for matching, as well as number of content regions compared to ground truth [21]. Table IV shows the compares the performance of our current method with prior art.

Keyframe summaries are produced by first segmenting the video using *weighted conflict minimization*, a variant of the approach proposed by Davila and Zanibbi [7]. Each content node present within every interval is binarized and placed on a virtual frame to produce the summary keyframe. It is evaluated by the recall-precision of binary connected components in the output keyframes and number of keyframes.

Weighted Conflict Minimization: This is a greedy algorithm used to segment the video into intervals containing unique content. The initial video segment is set to be entire video from the first to the last frame. We identify the lifetimes (in terms of frame number) of every pair of conflicting tracklets that are present on the whiteboard within the selected interval. The frames between these lifetimes are candidates for segmenting the video. For every such frame, we sum the weights of the conflict (derived from edge metadata computed in Section III-C3) that would be resolved if the video was segmented at that frame. The frame with highest conflict resolution weight is chosen greedily and the algorithm is recursively repeated in the left and right intervals until a stopping criterion is reached.

After identifying the frame numbers to segment the video, we gather all tracklets that are active within each interval, obtain cropped regions using the stored frame numbers and

TABLE III: Comparison of different methods of conflict minimization strategies by measuring recall (R), precision (P), f-score (F), average number of frames (N_f) and standard deviation (σ). The first half shows results for uniform weights while the second half shows results for weights computed from product of normalized area of conflict and mean partial feature difference. With the latter strategy, we get more control on the trade-off between compression and content recall. w_{min} indicates the minimum conflict weight required to create a segment.

	AVG	AVG GLOBAL			AVG PER FRAME		
w_{MIN}	N_f (σ)	R	P	F	R	P	F
5	10.71 (1.48)	94.98	93.77	94.36	93.92	92.50	93.18
3	11.86 (1.25)	95.2	93.97	94.57	94.13	92.85	93.47
1	15.57 (2.19)	95.52	94.18	94.84	94.45	93.36	93.89
0.025	10.29 (1.67)	95.01	93.84	94.41	93.98	93.69	93.84
0.0025	12.57 (1.50)	95.37	94.00	94.68	94.24	93.09	93.64
0.0001	15.57 (1.99)	95.61	94.56	95.08	94.6	93.77	94.17

TABLE IV: Evaluation of key content summaries by comparing average number of content regions, N_c (ground truth $N_c = 87.43$), and average summary recall. Intersection-over-Union threshold of 0.5 is used to check if ground truth region has been recalled.

Method	N_c	Avg. Recall
Prior Work [21]	127.14	92.09
Our Work	114.57	94.10

bounding boxes and binarize them using a random forest (RF) in hysteresis with Otsu binarization as described by Davila and Zanibbi [7]. Thus, a summary keyframe is produced for each interval by compositing all binarized tracklet regions within that video segment. It must be noted that we use binarized summary keyframes in order to compare fairly with existing methods in the literature that are evaluated on recall and precision of unique ground truth binary components. We can also produce summary keyframes by selecting a video frame with maximum content from within a segment.

Table V shows the number of keyframes produced along with recall-precision of summary binary connected components obtained using our proposed framework and comparison to existing methods.

IV. EXPERIMENTS AND DISCUSSION

PSENet structure of the handwritten content detector is initialized with weights trained on a combination of scene text datasets from ICDAR 2015 and 2017 Robust Reading Competitions [31, 32]. It is then fine-tuned end-to-end on the AccessMath dataset for 200 epochs at a learning rate of 0.001 for 200 epochs using a stochastic gradient descent optimizer with momentum of 0.9 and a weight decay of 0.0005. The training frames are randomly cropped to sizes of 512×512 and the batch size is 8.

Pixel-wise recall and precision of the detector is compared to other existing methods on the AccessMath test lecture dataset in Table I. We can see that the detector demonstrates higher F-measure than both existing methods despite producing more regions per-frame. This is possibly attributed to more pre-training data and instance segmentation as opposed to regression-based detection.

The feature extraction network is initialized using the same weights as the detector for all common layers and the rest are initialized using Kaiming-normal initialization. These layers are then trained for 200 epochs at a learning rate of 0.0001 which is dropped by a factor of 0.1 after 100 epochs. In our experiments, using MSLoss hyperparameter $\lambda = 0.3$ gave us best results, whereas under triplet loss this network did not converge. We test the matching performance of the two feature extraction algorithms by using receiver operating characteristic (ROC) curves at various thresholds. Features are extracted from ground truth test set regions using the two methods and for each region, cosine distance is computed to every other region feature. We then measure the area under the curve (AUC) for genuine and impostor distances to compare the two methods, which is shown in Table II.

While deep metric learning (DML) approaches show promise, the histogram of gradients features outperforms this approach. DML approaches are often sensitive to sampling and thus, require large batch sizes and multiple epochs to converge to meaningful representations. This forces trade-offs with respect to number of parameters and computational hardware use. Further, neural network features extract high-level features due to multiple downsampling layers and are often further reduced by pooling operations, which may not be sufficient to perform fine-grained distinction of handwritten content regions without additional supervision requiring extensive annotations.

During computation of spatio-temporal graphs (STG), we use distance threshold $d^s_{ij} = 0.04$ and feature threshold $d^f_{ij} = 0.5$ for both full and partial matches. If spatial overlap, $\text{IOU}_{ij} \geq 0.70$ or $\text{IOMin}_{ij} \geq 0.90$, two tracklets t_i and t_j are considered as occluded/growing/disappearing regions and are candidates for grouping. For conflicts checks, any non-zero values of overlap are considered. These thresholds are obtained empirically by measuring final performance on the training lecture video set. Though, the first two stages of STG creation (Sections III-C1 and III-C2) are analogous to work by Davila and Zanibbi [7], we have designed an algorithm that uses partial region feature extraction (detailed in Section III-C3) to merge region tracklets or mark them as conflicting, thus enabling us to perform spatio-temporal content analysis at region level.

For keyframe generation via video segmentation, we ex-

TABLE V: Comparison of different methods of lecture video summarization by measuring recall (R), precision (P), f-score (F), average number of frames (N_f) and standard deviation (σ). Lower N_f is better, whereas higher (R, P, F) is better. Our results in the lower part of the table are obtained by tuning our system for two different objectives - 1) minimum number of summary keyframes (best compression) and 2) maximum unique content f-measure (best f-measure). In configuration 1, we are able to achieve the best N_f with competitive f-measure whereas in configuration 2 we show near state-of-the-art f-measure with competitive N_f. Depending on the end use of summarization, our framework can be tuned to optimize either objective.

	AVG	AVG GLOBAL			AVG PER FRAME		
METHOD	N_f (σ)	R	P	F	R	P	F
AccessMath [7]	17.29 (4.54)	96.28	93.56	94.90	95.73	92.21	93.93
Maximum Content Sum [15]	34.42 (10.15)	**96.49**	94.51	**95.49**	**96.13**	91.95	93.99
Prior work 1 [8]	19.43 (5.32)	92.33	94.16	93.23	91.69	93.45	92.56
Prior work 2 [10]	21 (5.17)	95.80	92.88	94.32	95,40	92.44	93.90
Xu et al. [17]	12.29 (2.14)	95.89	86.28	90.83	94.18	85.15	89.44
Combined Area-Feature (best compression)	**10.29** (1.67)	95.01	93.84	94.41	93.98	93.69	93.84
Combined Area-Feature (best f-measure)	15.57 (1.99)	95.61	**94.56**	95.08	94.60	**93.77**	**94.17**

perimented with different conflict weighing schemes namely, uniform conflict weight (all conflicts receive a weight of 1.0 which is identical to the scheme used by Davila and Zanibbi [7]), and product of partial feature mean difference and area of overlap (computed as described in III-C3). Table III shows the differences between the approaches. We found that non-uniform conflict weights allowed a more fine-grained control over the segmentation process than uniform weights. This is because in the uniform scheme, the minimum weight of conflicts to create a segment w_{min}, can be varied in discrete steps, whereas in the weighted scheme, we have a continuous range. Further, we note that for similar number of summary frames, we get slightly higher content f-measure with the weighted scheme.

The minimum conflict weight (w_{min}) to decide if segmentation should occur was determined empirically using the training video lecture set. Table V shows the final results we obtained, compared to other work on the AccessMath dataset. We present two sets of results, one configuration with $w_{\text{min}} = 0.0025$ which achieved compression closest to ground truth on training lectures and another with $w_{\text{min}} = 0.0001$ which achieved highest f-measure.

We note that our best compression method (row 1 in lower part of Table V) achieves lowest number of summary frames in the literature while maintaining global f-measure comparable to the highest in literature [6], requiring about 24 summary keyframes lesser than this method. Further, the method by Xu et. al. which uses lecturer pose to perform summarization [17] has the next best summarization performance while being about 3.5% points lower in f-measure. Our best f-measure method (row 2 in lower part of Table V) achieves second highest global f-measure falling short by 0.4% of the highest in literature [6] while requiring about 19 frames lesser to summarize the content.

This shows that our framework is flexible with respective both the objectives of whiteboard summarization - i.e. more compact summary as well as high extraction measure of unique content and can in fact be tuned to achieve the right trade-off depending on the downstream application such as search and retrieval and design of a navigable user interface. Our failure cases in the recall largely arise from detector errors especially smaller content regions that may be entirely missed. Some content regions that change very subtly and also missed by the conflict detection resulting in undersegmentation. Precision errors are caused by spurious detections sometimes induced by lecturer or other false positives that cause extraneous conflicts to be detected creating oversegmentation.

V. CONCLUSION

In this work, we have proposed a fully bounding box detection oriented approach to summarizing whiteboard lecture video in terms of both key content and keyframes. We investigated and compared deep metric learning (DML) and histogram of gradient (HoG) feature approaches to represent detected regions and found that HoG achieved better performance although DML shows promise. Further, we proposed an efficient spatio-temporal graph (STG) based tracking scheme using partial region feature regions that can handle growing and occluded handwritten content to find unique regions which are presented as key content summaries. The STG along with a weighted conflict minimization scheme to segment the video led to keyframe summaries. Our methods lead to performance comparable to state-of-the-art methods in terms of average recall, f-measure as well as average number of summary keyframes. Additionally, the meta-data information such as frame number and timestamps stored along with the nodes and tracklets in our spatio-temporal graphs could help in video navigation.

In the future, we wish to further investigate deep metric approaches to represent handwritten content especially using additional supervision such as recognition or classification labels. Semi-supervised methods for handwritten content are also an attractive avenue for future research. We also wish to unify lecturer action detection and text detection approaches in order to obtain higher f-measure and similar or lower compression ratios. Finally, we wish to explore these methods on other handwritten lecture datasets, especially in other domains.

ACKNOWLEDGMENT

This research was supported by the National Science Foundation under Grant No.1640867 (OAC/DMR).

REFERENCES

[1] P. J. Guo, J. Kim, and R. Rubin, "How video production affects student engagement: An empirical study of mooc videos," in *Proceedings of the first ACM conference on Learning@ scale conference*, 2014, pp. 41–50.

[2] J. Kim, P. J. Guo, D. T. Seaton, P. Mitros, K. Z. Gajos, and R. C. Miller, "Understanding in-video dropouts and interaction peaks inonline lecture videos," in *Proceedings of the first ACM conference on Learning@ scale conference*, 2014, pp. 31–40.

[3] B. T. Truong and S. Venkatesh, "Video abstraction: A systematic review and classification," *ACM transactions on multimedia computing, communications, and applications (TOMM)*, vol. 3, no. 1, pp. 3–es, 2007.

[4] A. Biswas, A. Gandhi, and O. Deshmukh, "Mmtoc: A multimodal method for table of content creation in educational videos," in *Proceedings of the 23rd ACM international conference on Multimedia*, 2015, pp. 621–630.

[5] K. Li, J. Wang, H. Wang, and Q. Dai, "Structuring lecture videos by automatic projection screen localization and analysis," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 6, pp. 1233–1246, 2015.

[6] C. Choudary and T. Liu, "Extracting content from instructional videos by statistical modelling and classification," *Pattern analysis and applications*, vol. 10, no. 2, pp. 69–81, 2007.

[7] K. Davila and R. Zanibbi, "Whiteboard video summarization via spatio-temporal conflict minimization," in *International Conference on Document Analysis and Recognition (ICDAR)*, 2017.

[8] B. U. Kota, K. Davila, A. Stone, S. Setlur, and V. Govindaraju, "Automated detection of handwritten whiteboard content in lecture videos for summarization," in *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, 2018, pp. 19–24.

[9] K. D. Castellanos, *Symbolic and Visual Retrieval of Mathematical Notation using Formula Graph Symbol Pair Matching and Structural Alignment*. Rochester Institute of Technology, 2017.

[10] B. Urala Kota, K. Davila, A. Stone, S. Setlur, and V. Govindaraju, "Generalized framework for summarization of fixed-camera lecture videos by detecting and binarizing handwritten content," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 22, no. 3, pp. 221–233, 2019.

[11] X.-C. Yin, Z.-Y. Zuo, S. Tian, and C.-L. Liu, "Text detection, tracking and recognition in video: A comprehensive survey," *IEEE Transactions on Image Processing*, vol. 25, no. 6, pp. 2752–2773, 2016.

[12] J. Meng, H. Wang, J. Yuan, and Y.-P. Tan, "From keyframes to key objects: Video summarization by representative object proposal selection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1039–1048.

[13] A. Sahoo, V. Kaushal, K. Doctor, S. Shetty, R. Iyer, and G. Ramakrishnan, "A unified multi-faceted video summarization system," *arXiv preprint arXiv:1704.01466*, 2017.

[14] M. Onishi, M. Izumi, and K. Fukunaga, "Blackboard segmentation using video image of lecture and its applications," in *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, vol. 4. IEEE, 2000, pp. 615–618.

[15] C. Choudary and T. Liu, "Summarization of visual content in instructional videos," *IEEE Transactions on Multimedia*, vol. 9, no. 7, pp. 1443–1455, 2007.

[16] S. Vajda, L. Rothacker, and G. A. Fink, "A method for camera-based interactive whiteboard reading," in *International Workshop on Camera-Based Document Analysis and Recognition*. Springer, 2011, pp. 112–125.

[17] F. Xu, K. Davila, S. Setlur, and V. Govindaraju, "Content extraction from lecture video via speaker action classification based on pose information," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2019, pp. 1047–1054.

[18] G. C. Lee, F.-H. Yeh, Y.-J. Chen, and T.-K. Chang, "Robust handwriting extraction and lecture video summarization," *Multimedia Tools and Applications*, vol. 76, no. 5, pp. 7067–7085, 2017.

[19] W. Jia, L. Sun, Z. Zhong, and Q. Huo, "A cnn-based approach to detecting text from images of whiteboards and handwritten notes," in *Frontiers in Handwriting Recognition (ICFHR), 2018 16th International Conference on*. IEEE, 2018, p. to appear in.

[20] K. Dutta, M. Mathew, P. Krishnan, and C. Jawahar, "Localizing and recognizing text in lecture videos," in *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, 2018, pp. 235–240.

[21] B. U. Kota, S. Ahmed, A. Stone, K. Davila, S. Setlur, and V. Govindaraju, "Summarizing lecture videos by key handwritten content regions," in *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, vol. 4. IEEE, 2019, pp. 13–18.

[22] S. Tian, W.-Y. Pei, Z.-Y. Zuo, and X.-C. Yin, "Scene text detection in video by learning locally and globally." in *IJCAI*, 2016, pp. 2647–2653.

[23] X.-H. Yang, F. Yin, and C.-L. Liu, "Online video text detection with markov decision process," in *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*. IEEE, 2018, pp. 103–108.

[24] J. Almazán, A. Gordo, A. Fornés, and E. Valveny, "Word spotting and recognition with embedded attributes," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 12, pp. 2552–2566, 2014.

[25] S. Sudholt and G. A. Fink, "Phocnet: A deep convolutional neural network for word spotting in handwritten documents," in *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, 2016, pp. 277–282.

[26] T. Q. Phan, P. Shivakumara, T. Lu, and C. L. Tan, "Recognition of video text through temporal integration," in *2013 12th International Conference on Document Analysis and Recognition*. IEEE, 2013, pp. 589–593.

[27] C. Wolf and J.-M. Jolion, "Object count/area graphs for the evaluation of object detection and segmentation algorithms," *International Journal on Document Analysis and Recognition*, vol. 8, no. 4, pp. 280–296, 2006.

[28] W. Wang, E. Xie, X. Li, W. Hou, T. Lu, G. Yu, and S. Shao, "Shape robust text detection with progressive scale expansion network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9336–9345.

[29] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection." in *CVPR*, vol. 1, no. 2, 2017, p. 4.

[30] X. Wang, X. Han, W. Huang, D. Dong, and M. R. Scott, "Multi-similarity loss with general pair weighting for deep metric learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5022–5030.

[31] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu *et al.*, "Icdar 2015 competition on robust reading," in *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*. IEEE, 2015, pp. 1156–1160.

[32] N. Nayef, F. Yin, I. Bizid, H. Choi, Y. Feng, D. Karatzas, Z. Luo, U. Pal, C. Rigaud, J. Chazalon *et al.*, "Icdar2017 robust reading challenge on multi-lingual scene text detection and script identification-rrc-mlt," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1. IEEE, 2017, pp. 1454–1459.