Summarizing Lecture Videos by Key Handwritten Content Regions

Bhargava Urala Kota, Saleem Ahmed, 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, sahmed9, awstone, kennydav, setlur, govind]@buffalo.edu

***Abstract*—We introduce a novel method for summarization of whiteboard lecture videos using key handwritten content regions. A deep neural network is used for detecting bounding boxes that contain semantically meaningful groups of handwritten content. A neural network embedding is learnt, under triplet loss, from the detected regions in order to discriminate between unique handwritten content. The detected regions along with embeddings at every frame of the lecture video are used to extract unique handwritten content across the video which are presented as the video summary. Additionally, a spatiotemporal index is constructed from the video which records the time and location of each individual summary region in the video which can potentially be used for content-based search and navigation. We train and test our methods on the publicly available AccessMath dataset. We use the DetEval scheme to benchmark our summarization by recall of unique ground truth objects** (92.09%) **and average number of summary regions** (128) **compared to the ground truth** (88)**.**

I. Introduction

The ubiquity of cameras has resulted in the availability of large amounts of video data captured across many domains. Lecture videos are a subset of this growing data stream that contain handwritten or printed text elements in the scene which can typically be classified into white or black board data, or slides-based data. Lecture videos enable quality educational content to be broadcast anywhere in the world, acting as a valuable tool for students and educators across the globe. While current search engines primarily support meta-data based search and retrieval of lecture videos, effective *video summarization* techniques are needed to extract key content and condense this data into an easily searchable form, to facilitate content based search.

In this work, a framework for automated lecture summarization by key handwritten content is provided. The generated summary is in the form of a small set of handwritten content objects which represent all the unique content on the whiteboard. Further, a spatio-temporal data structure is provided, mapping each summary content object to all of its instances within the video. The summary and mapping can potentially be used for further downstream applications like recognition and indexing for visual search.

Lecture content on a whiteboard is often loosely structured and exhibits large variances in semantic grouping. Examples include sentences, multi-line phrases, sketches, plots and mathematical expressions. Further, background noise, illumination changes and occlusions are also present. Limited public annotated datasets, time and cost of annotation, further amplify these issues.

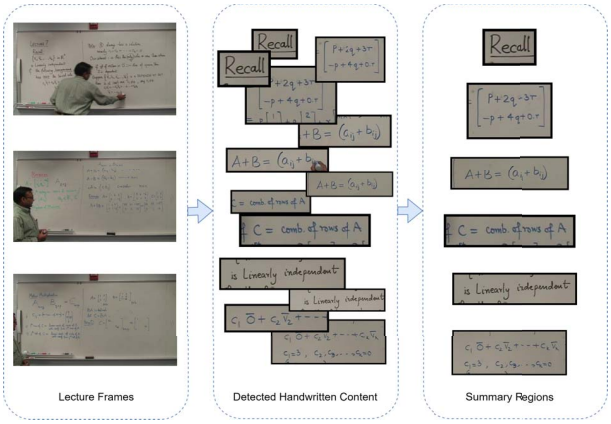


Figure 1: An overview of our proposed summarization method. We detect handwritten content regions on all frames of a lecture video and extract feature representations from these regions. We then generate a small subset of summary regions for the entire lecture video which can be used for later recognition or indexing for search.

Another challenge for lecture videos is to arrive at a robust representation given the variety of text content. Recognition of unstructured handwritten text is a hard problem and requires extensive annotation. Further, the usual assumptions about text granularity at word/line level are also not reasonable for whiteboard text. Therefore, in our work we employ a triplet loss based method to extract visual embeddings from detected content regions.

After handwritten content and features are extracted from all frames of videos, the next task is to analyze the detected regions by extracting visual features and reducing the set of all detected text into unique content regions, which falls under the paradigm of video summarization by extraction of **key objects**. For such work, the number of key objects produced as well as the recall of key content (at the level of bounding boxes) with respect to all unique text content in the lecture video, are used as evaluation metrics.

The main research questions being investigated in our

978-1-7281-5054-3/19/$31.00 ©2019 IEEE
DOI 10.1109/ICDARW.2019.30058

work include 1) Can lecture videos be summarized by key handwritten content instead of the prior art of key frames and how do we evaluate them? 2) Can we learn an embedding for content regions under lack of explicit targets such as recognized text?

II. Background

We broadly discuss the field of video summarization, in particular, lecture video summarization and some work on representing and tracking text in images and videos.

Video Summarization: Video summarization methodologies can be classified by the nature of the summary. Keyframes that contain the highlights of the video content are a typical form of summary found in the literature [1]. Shorter versions of the videos - skims, montages and synopses have also been used [2].

Many existing methods generate video summaries driven by detection and feature representation of objects [1, 2]. Recently, a method to summarize videos by key objects instead of keyframes was proposed [3]. In this work, summarization is posed as a representative selection problem from detected candidate regions. In our work, we aim to summarize lecture videos by key handwritten content regions. We use neural network based detection and feature extraction, and match extracted features guided by spatiotemporal constraints in order to produce key content regions.

Lecture Summarization: Whiteboard lecture videos are typically preprocessed by background removal and binarization followed by content extraction and summarization [4, 5, 6]. After preprocessing, handwritten content is extracted and grouped into meaningful sets, primarily using spatiotemporal cues [4, 6, 7] or OCR [4]. Neural networks have recently been used for direct content extraction [7]. We also use deep learning based detector for direct content extraction and combine spatiotemporal constraints and visual features to summarize content.

The final stage is the summarization of the lecture video. Keyframes (which contain most of the unique content within a video segment) [5, 6, 7], recognized text lines [4] and production of composite images that contain all content [5] are some of the typical methods of summaries for whiteboard lecture videos. As far as we know, there is no evaluation scheme in the literature for summarizing lecture videos by key content regions.

Spotting Text in Images and Videos: Mapping bagged keypoint-based [8] or deep neural network based features from word images, to pyramidal hierarchy of character label (PHOC) embeddings of corresponding text strings [8] has been used for both handwritten words [9] and scene text [10]. In these methods, either the word segmentation information is accessible during testing or the text transcript is available during training or both.

Earlier video text tracking methods followed the paradigm of text segmentation and spatiotemporal enhancement [11]. Recently, tracking by detection and/or recognition along with local and global spatiotemporal analysis methods such as using dynamic programming [12] and Markov Decision Process [13] have been explored. Evaluation of these methods rely on Multi-Object Tracking metrics [11].

In our work, we need to simultaneously detect and learn to represent irregularly structured handwritten content without access to the text transcriptions. Thus, we use a triplet loss based metric learning scheme to overcome this challenge.

Datasets and Evaluation: AccessMath is the largest, publicly available, benchmarked dataset for whiteboard lecture video summarization [6]. 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 [6] 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 [7].

Meng et al. summarize general videos by key objects [3]. They test their methods on a set of 10 commercial videos from YouTube [14] sampled at two frames per second. Topical objects (such as products and logos) are annotated. The ten videos have a duration of a few minutes each and on average contain 36 instances of roughly 8 key objects per video. However, in AccessMath, each test video has on average 88 key content regions and total instance count per video is of the order of ten thousand.

In our work, we propose a new evaluation scheme which is independent of binarization and keyframes while shifting the focus onto regions of handwritten content. Similar to Meng et al., we use number of proposed regions and average recall [3] of unique handwritten content across frames to evaluate summarization. The DetEval scheme [15] is used to obtain average recall as it is more suited for text.

III. Lecture Video Summarization

We use a deep neural network for detection and propose a feature extraction network trained using triplet loss learning. Feature similarity along with spatiotemporal constraints that model detector uncertainty are used to summarize the video by generating *key handwritten content* regions.

A. Detection of Content

We adapt EAST [16] for this task, mainly because it is anchor-free, i.e. it does not assume any priors on text content areas and aspect ratios, which allows handling the variety of text shapes found in lecture videos.

1) Structure: The general EAST detector consists of a convolution-deconvolution feature extraction block, a convolutional region proposal network consisting of sub-networks for text/non-text classification, regression to top, right, bottom, left edges and angle of rotation of a ground truth minimum bounding rotated rectangle. We use a Feature Pyramid Network (FPN) for feature extraction [17] with ResNet [18] backbone, deconvolution layers and activations as originally prescribed for FPN [17]. This structure is chosen because it extracts multi-scale features, has readily available initialization weights from ImageNet training and has state-of-the-art object detection performance.

The region proposal sub-network is designed as prescribed by EAST [16] with sigmoid activations. The angle prediction output of this sub-network is suppressed since the Access-Math annotations are axis-aligned bounding boxes. The final structure of the detector is shown in Figure 2.

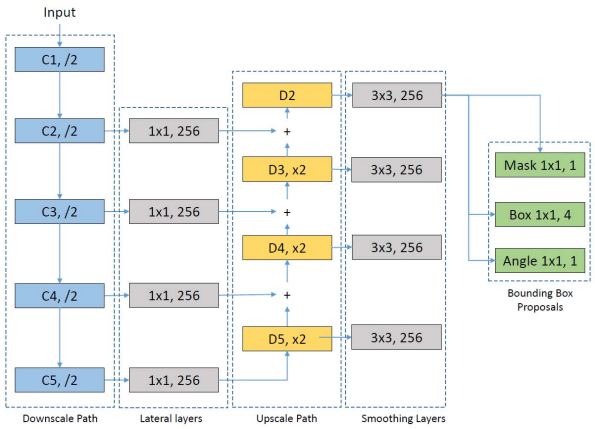


Figure 2: Our structure for EAST Detector. Here, $k \times k$, c indicates kernel size (k) and the number of channels (c) of a convolutional layer, $/n$ or $\times n$ indicates downsampling or upsampling by factor of n. C and D stands for block of multiple convolutional layers and transpose convolutional layers respectively according to the Feature Pyramid Network [17].

2) Training Labels: Lecturer bounding box for every frame of the training videos are obtained using an SSD [19] detector trained on the PASCAL VOC object detection dataset [20]. Ground truth text boxes which overlap with the bounding box of the lecturer are removed if their area of intersection is greater than 25% of the text box area and training targets for pixel mask and edge displacements are generated as described in the original EAST paper [16].

3) Loss Functions: A generalized DICE coefficient loss is computed between text pixel predictions and ground truth targets for every iteration. This has shown good segmentation performance under unbalanced labels for bio-medical images [21]. We compute the intersection over union (IOU) loss as described by Zhou et. al. with respect to the ground truth [16]. The sum of DICE and IOU losses is the total loss for the network. We chose DICE loss over weighted Binary Cross Entropy (BCE) loss because we observed better numerical stability during training, possibly due to exponential and logarithmic computation in the BCE loss.

4) Inference: After training (see Section IV for details), the detector produces a set of redundant and overlapping bounding boxes due to the dense prediction layers. The predictions are scanned row-wise and greedily merged if the IOU between two successive bounding boxes are greater than a threshold (θ_{loc}). Non-maximum suppression, with threshold (θ_{nms}), is used on the remaining predictions to obtain the final set of proposed regions.

B. Feature Extraction

The proposed regions from the detector stage need to be represented by a feature vector that can distinguish between instances of different unique content regions. Since Access-Math does not include text transcriptions, we use a metric learning method to learn this feature embedding. We choose triplet loss formulation since it has shown good performance on disjoint face identities in biometric applications [22].

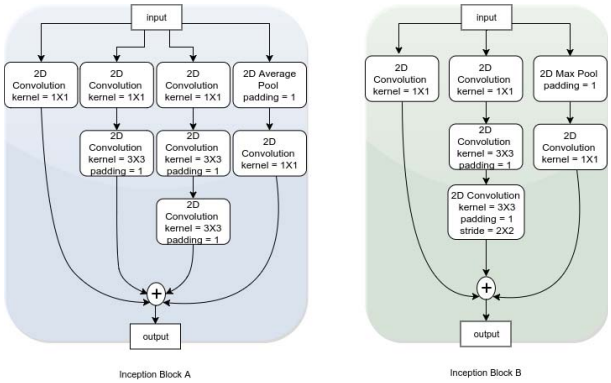


Figure 3: Our proposed feature extraction model uses Inception Block A (left) and Inception Block B (right) as the basic units. The layers are chained together and followed by s linear layer and $L2$ normalization to obtain embedding from resampled content regions.

1) Structure: The extracted bounding boxes from the detector model are first passed through a bilinear interpolation layer to get proposals of uniform size. We use interpolated size 64×64 with channel size, $c = 256$, retained from the detector D2 layer. This size is chosen to accommodate all the variety in aspect ratios seen in the lecture content, where text is not predominantly horizontal like in natural images.

These uniformly shaped feature maps are then passed to a series of inception [23] blocks stripped of any non-linearity functions. The structure of each block is illustrated in Figure 3. The blocks are connected as A-A-B-A-A-A-A-B-A-A. The architecture of the feature extractor is based on TextSpotter [24], which has state-of-the-art end-to-end text

recognition performance in natural scene images. Each block has added LeakyReLU activations (negative slope of 0.001) at the end. The last inception block output is propagated through a 1×1 convolutional layer with stride (2×2) and a fully connected layer of size 2048 followed by a $L2$ normalization generating an embedding for the input region.

2) Loss Function: A generalized triplet loss is computed between a pair of positive samples with anchor and negative sample with anchor using Equation 1.

$$L = max(0, m + \|f(x_a) - f(x_p)\|_2^2 - \|f(x_a) - f(x_n)\|_2^2) \quad (1)$$

where, $f(x_a)$ is the anchor embedding and $f(x_p)$ and $f(x_n)$ are the positive and negative sample embeddings respectively, m is a margin which indicates the ideal minimum separation between the distances computed from the positive-anchor and anchor-negative pairs of embeddings.

3) Triplet Sampling Strategy: Triplet sampling strategies are often key to learning robust representations of content regions. Typically, in every iteration, positive anchor and negative samples are chosen such that positive-anchor and anchor-negative pair distances fall within the margin.

In our case, additional priority is given during sampling to select negative regions within a certain distance of the anchor sample. For further augmentation, the triplet bounding boxes from the ground truth are perturbed with random noise and the negative samples have a 20% chance to be dropped in favor of a background region sample.

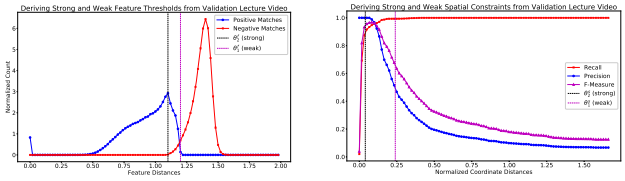
4) Inference: The feature embedding layers are trained using the ground truth triplet samples generated from the training lecture video sets. For testing, the proposed bounding boxes from the detector, after the non-maximum suppression stage, are used to generate features for each region.

C. Content Summarization

Detected regions and corresponding feature embeddings are extracted from the test set of lecture videos. Detected regions are filtered using person detection using the same procedure as during training (Section III-A2).

The video frames are then grouped in intervals of 60 seconds such that each successive interval overlaps the previous by 30 seconds. The summarization by key objects takes place in two passes. In the first pass, seed summary regions are obtained on the basis of strong feature similarity and strong spatial constraints. Then, weakly matched regions are re-examined and grouped or merged appropriately to produce the summary content. As a last step, summary content is pruned based on number of regions.

1) Generating seed summary content: For every 60 second interval of the test video, we examine all pairs (r_i, r_j) from the set of detected regions R for spatial proximity using weak and strong spatial thresholds (θ_1^s and θ_2^s) as well as for feature proximity using weak and strong thresholds (θ_1^f and θ_2^f). We compute the feature distance $d_{ij}^f = \|f(r_i) - f(r_j)\|_2^2$ and spatial distance $d_{ij}^s = \|\mathbf{x}_i - \mathbf{x}_j\|_2^2$. Where, f represents the feature extractor network and $\mathbf{x}_i$, $\mathbf{x}_j$ are the top-left and bottom-right corner coordinates of r_i normalized with respect to frame height and width.



(a) Feature Thresholds (b) Spatial Thresholds

Figure 4: Subfigures (a) and (b) shows derivation of strong and weak thresholds for feature and spatial distance respectively, from validation lectures. Best seen in digital.

- If $d_{ij}^f \leq \theta_2^f$ and $d_{ij}^s \leq \theta_2^s$, r_i, r_j are considered a strong match and merged into the same summary identity.
- If $d_{ij}^f \leq \theta_2^f$ and $\theta_2^s < d_{ij}^s \leq \theta_1^s$ OR $d_{ij}^s \leq \theta_2^s$ and $\theta_2^f < d_{ij}^f \leq \theta_1^f$, r_i and r_j are placed in separate summary identities and the two identities are marked as strong-weak matches.
- If $\theta_2^f < d_{ij}^f \leq \theta_1^f$ and $\theta_2^s < d_{ij}^s \leq \theta_1^s$, r_i and r_j are placed in separate summary identities and the two identities are marked as weak-weak matches.
- In all other cases, r_i, r_j are placed in separate summary identities and marked as non-matches.

We obtain a set of summary regions $\mathbf{S}$, where each S_m is a unique content region with instances $\{r_{1m}, r_{2m}, ..., r_{nm}\}$.

2) Growing summary content: The initial set of summary seed content regions $\mathbf{S} = \{S_1, S_2, ..., S_M\}$ consist of discrete sets of strongly matched regions within 60 second intervals. We recursively check all summary regions that overlap temporally (with a tolerance of 120 seconds) for spatial and feature based match, stopping when no changes are observed. During each recursion, the bounding box and features of each summary region are aggregated by computing the union and the mean respectively. Surviving weak matches are grouped via spatial constraint.

3) Determining thresholds: The distances between embeddings for all regions within a 60 second interval is computed and Otsu's algorithm is used to find the threshold θ_1^f. The underlying assumption is that the positive and negative matches form a bimodal distribution. The peak of the positive matches distribution is also recorded as a secondary threshold θ_2^f. Figure 4(a) shows accumulated bimodal distribution estimated for all intervals of a single lecture video along with selection of θ_1^f and θ_2^f.

The variation in illumination between frames causes jitter in the detector predictions. On the validation lecture, we analyze the displacement between detected bounding boxes that correspond to the same ground truth box and derive thresholds on the basis of recall and precision of true and

false matches. Figure 4(b) shows the recall-precision at different displacements along with the thresholds θ_1^s and θ_2^s.

4) Finalizing Summaries: Groups of summary content with low instance counts are removed to produce a more compact summary. We prune each summary so that 90% of the detections remain. From this, an inverted index is constructed, where the entries are the unique summary regions, and for each entry we store links to all the corresponding detection instances across the lecture video. The effective recall for all instances of ground truth content regions in the video is measured using our modified DetEval scheme detailed in Section III-D and shown in Table I.

D. Summary Evaluation

Video summarization by key objects use average recall as the final evaluation metric [3]. Given a video with $\mathbf{P}$ object proposals, *t* unique key objects and $\mathbf{G}_i$ is the set of instances of the *i*-th key object, average recall r is defined as follows:

$$S(\mathbf{P}, \mathbf{G}_i) = \max_{p \in \mathbf{P}, g \in \mathbf{G}_i} S(p, g) \tag{2}$$

$$r = \frac{\sum_{i=1}^{t} \mathbf{1}(S(\mathbf{P}, \mathbf{G}_i) \geq \theta)}{t} \tag{3}$$

where, $S(p, g)$ is the intersection-over-union (IOU) of the two regions p and g and $\mathbf{1}$ is the indicator function.

However, in case of text, where segmentation can happen at multiple levels of granularity, we need to accommodate merging or splitting of the ground truth into multiple regions. Therefore, we use the recall as defined in the DetEval scheme [15] which allows for ground truth to match against multiple predicted regions provided some minimum threshold of area recall and area precision is met (see Equations 4 and 5). The number of unique summary regions proposed is also compared to the number of unique ground truth regions.

$$R(g, \mathbf{P}, t_r, t_p) = \begin{cases} 1, \text{ if one-one match with any } p \\ \frac{1}{1+\log(k)}, \text{ if } g \text{ matches } k \text{ boxes} \\ 0, \text{ otherwise} \end{cases} \tag{4}$$

where, one to many (k) matches are allowed if,

$$\forall p_j \in \mathbf{P_k} \; \frac{g \cap p_j}{p_j} \geq t_p \text{ and } \sum_{p_j \in \mathbf{P_k}} \frac{g \cap p_j}{g} \geq t_r \tag{5}$$

where, $\mathbf{P_k}$ is a subset of k summary proposals, t_r and t_p are area-recall and area-precision thresholds which are set equal to the IOU threshold for one-one match in our experiments.

IV. Experiments

The AccessMath dataset was used to train and test our summarization methodology. The training and test videos are annotated to provide bounding boxes [7]. We randomly split the training set of 5 videos into 4 and 1 for training and validation and train the detector using the procedure mentioned in Section III-A2.

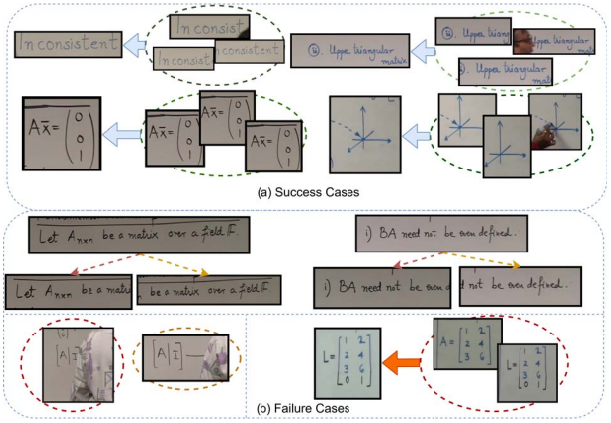


Figure 5: Subfigure (a) shows some of the correctly grouped summary regions for different type of content detection instances - words, multiline phrases, sketches and math expressions. Note that the summarization is robust to minor occlusions by lecturer, Subfigure (b) shows some of the failure cases. Top left and top right show long sentences split into two summary regions; bottom left shows lecturer induced separation of same content and bottom right shows merging of different yet similar looking content instances under same summary region due to overwriting by lecturer.

The ResNet portion of the detector is initialized with pre-trained ImageNet weights and Kaiming-normal initialization is used for all other layers. Training is carried out for 20 epochs with a batch size of 16 using a stochastic gradient descent (SGD) optimizer with an initial learning rate of 0.001. Each sample is augmented as described by EAST [16], with random 512×512 crops. The learning rate is reduced at a constant rate of $\gamma = 0.7943$ per epoch. This ensures that the learning rate drops by a factor of approximately 0.1 every 10 epochs.

The feature extractor was trained with feature maps generated from the pre-trained detector network and sampled triplet bounding boxes (as described in Section III-B), with a triplet loss margin $m = 0.5$. Initialization and optimization scheme is same as those of the content detector. Total accuracy plateaued to around 88% in $8 - 12$ epochs. Accuracy is measured as the total sum of all positive embeddings per iteration having a distance lesser than margin from the anchor as compared to negative embeddings.

Weak and strong feature thresholds were determined as the average of computed thresholds for each test lecture video, and we found that the values were consistently around $\theta_1^f = 1.2$ and $\theta_2^f = 1.0$ for all test videos. Weak and strong spatial distance thresholds were computed from the validation lecture and found to be $\theta_1^s = 0.24$ and $\theta_2^s = 0.04$ as seen in Figures 4(a) and (b).

The final summarization recall and per-frame text box recall was measured for all videos in the AccessMath test

	Avg. Per-frame			Avg. Summ.
IOU	R	P	F	R
0.50	0.7995	0.4230	0.5504	0.9209
0.60	0.6816	0.2974	0.4109	0.9024
0.70	0.4778	0.1733	0.2524	0.8535
0.80	0.2256	0.0738	0.1097	0.7633
0.90	0.0324	0.0092	0.0137	0.6254

Table I: Average per-frame Recall, Precision and F-measure are measured using detector alone. Average Summary Recall is the recall of ground truth objects by summary regions. $N_t = 87.43$ and $N_s = 127.14$ are the average number of unique ground truth and summary regions respectively.

set at different values of intersection-over-union thresholds starting from 0.50 to 0.90. As seen in Table I, our feature-based spatiotemporal analysis and summarization technique is able to recall key content despite detector noise and jitter even under stricter IOU matching thresholds.

The limited field of view of the detector induces the summarization to split regions of extreme aspect ratio. Similar looking but different content tends to get grouped together due to overwriting. False positives due to lecturer also contribute to higher number of summary regions. Figure 5 shows some success and failure cases of our summarization.

V. Conclusion

We have proposed a novel way of summarizing lecture videos by key handwritten content regions. The regions are semantically meaningful and can be lines, words, math expressions or sketches. A neural network based feature extractor, under triplet loss, is trained to learn text region embeddings. These, along with spatiotemporal constraints, are used to match detected regions that localize the same content across the video and a summary by key handwritten content is generated.

Our summarization also generates a spatiotemporal data structure (inverted index described in Section III-C4) which links each summary region to all its constituent detections. We evaluate the summary in terms of recall of ground truth unique content using DetEval in order to accommodate different granularity of text. The summary data structure could potentially be used for downstream applications like recognition and indexing for visual search.

Our initial efforts focused on a framework for summarization by key content which shows promise. We would like to test retrieval performance of the summary and study, in detail, the relationship between summary by key objects and keyframes. Making the detector temporally aware by training on contiguous frames and robust representations of whiteboard handwritten content under weak or no supervision are attractive avenues for subsequent work.

Acknowledgments: This material was partially supported by the National Science Foundation under Grant No.1640867 (OAC/DMR).

References

[1] Y. J. Lee, J. Ghosh, and K. Grauman, "Discovering important people and objects for egocentric video summarization," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1346–1353.

[2] Z. Lu and K. Grauman, "Story-driven summarization for egocentric video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2714–2721.

[3] 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.

[4] 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.

[5] 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.

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

[7] 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.

[8] 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.

[9] 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.

[10] L. Gómez, A. Mafla, M. Rusinol, and D. Karatzas, "Single shot scene text retrieval," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 700–715.

[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] 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.

[13] 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.

[14] G. Zhao, J. Yuan, and G. Hua, "Topical video object discovery from key frames by modeling word co-occurrence prior," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1602–1609.

[15] 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.

[16] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "East: an efficient and accurate scene text detector," in *Proc. CVPR*, 2017, pp. 2642–2651.

[17] 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.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[19] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[20] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, Jan. 2015.

[21] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. J. Cardoso, "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Springer, 2017, pp. 240–248.

[22] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[23] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*. IEEE Computer Society, 2015, pp. 1–9.

[24] T. He, Z. Tian, W. Huang, C. Shen, Y. Qiao, and C. Sun, "An end-to-end textspotter with explicit alignment and attention," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 2018, pp. 5020–5029.