Automated Detection of Handwritten Whiteboard Content in Lecture Videos for Summarization

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

***Abstract*—Online lecture videos are a valuable resource for students across the world. The ability to find videos based on their content could make them even more useful. Methods for automatic extraction of this content reduce the amount of manual effort required to make indexing and retrieval of such videos possible. We adapt a deep learning based method for scene text detection, for the purpose of detection of handwritten text, math expressions and sketches in lecture videos. We detect handwritten elements on the whiteboard to generate a summary of all content over time in the lecture, while also dealing with occluded content due to motion of the lecturer. We train, test on the publicly available AccessMath lecture video dataset and evaluate our framework on the basis of number of summary frames, as well as recall and precision of all whiteboard content in the set of test lecture videos. We found that our method increases the precision of the state-of-the-art while there is potential to increase recall as well. We have added to the existing ground truth in the AccessMath dataset by providing timestamp-based, semantically meaningful bounding box annotations for the handwritten whiteboard content, which has been released.**

I. Introduction

Nowadays there exist thousands of hours of lecture videos online and these have become a useful resource for students across the world. Despite being so common, lecture videos are still not properly indexed by most common search engines. The main reason is that, in many cases, search engines depend on existing text annotations of the video in order to add them to their indices. If such annotations are unavailable, then search engines do not have a principled way to retrieve them. Manually producing such annotations is a hard task given the scale of lecture video content available. In this paper, we provide an important initial step required for automated lecture indexing, which is the detection and extraction of handwritten content from the video. We then use the extracted handwritten content to provide a 'summary' of the lecture, in the form of a small set of binary images of the handwritten content on the whiteboard which are ready for further recognition and indexing processes. For this work, we concentrate on videos where a single lecturer conducts a class while explaining and producing handwritten content on a whiteboard.

Most existing approaches for lecture video content extraction use a combination of heuristic rules and image processing techniques to preprocess and extract the content from the video [1], [2], [3], [4]. However, in our method we use a deep learning based approach for initial detection of handwritten content from a given video frame. We fine-tune TextBoxes [5], a model designed to detect words in natural scenes, for the task of detecting handwritten whiteboard content. This approach removes the need for explicit detection of the speaker during testing, and simplifies the processing of detected handwritten content in later stages - e.g. automatic summarization.

The handwritten content is often loosely structured and exhibits significant variance in content such as sentences, math expressions, matrices, sketches and plots. This, when combined with background noise, illumination changes and occlusions due to the lecturer, presents a significant challenge for automatic extraction of handwritten content. While our proposed content extraction method is currently tested only on lecture videos recorded with a single fixed camera, it can easily be extended to work on lecture videos using multiple cameras and/or zooming and panning.

After handwritten content is extracted from all frames of videos, our next task is to analyze the detected regions and identify association relationships of text regions across space and time - including cases where text regions are occluded due to motion of the lecturer. In prior work, binarized connected components are used to track the presence and absence of content on the whiteboard. In our work, we present some preliminary strategies to perform this task at the level of text bounding boxes and discuss their merits and demerits.

Lastly, we produce a summary of the lecture video in terms of 'keyframes', which refers to frames that contains all the handwritten content that was present on the whiteboard during a certain time interval in the video. We evaluate the performance of our system on the number of keyframes produced as well as the recall and precision of keyframe content with respect to all unique text content in the lecture video. In this work, we use the *conflict minimization* approach described by Davila and Zanibbi [1] to produce summary keyframes after spatio-temporal content associations. Figure 1 shows an overview of our proposed approach.

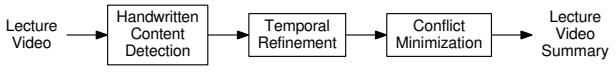


Fig. 1: An overview of the lecture video summarization pipeline used in our work

In summary, the main research questions being investigated in our work include:

978-1-5386-5875-8/18/$31.00 ©2018 IEEE
DOI 10.1109/ICFHR-2018.2018.00013

- How well does a machine-learning oriented approach perform for whiteboard content extraction compared to more traditional heuristic-based approaches?
- Can we adapt a general scene text detection approach to detect and extract handwritten text from lecture videos?
- How well can we recover occluded handwritten text using temporal information and our text detection method?

II. Background

Most lecture videos can broadly be classified into videos with pure handwritten content (HC), a lecturer producing handwritten content on a board or a lecturer referring to a slide deck during the class. For our work, we concentrate on lecture videos with handwritten whiteboard content.

A. Dataset and Evaluation

AccessMath is the largest, publicly available, benchmarked dataset for this purpose. It was created from a collection of linear algebra lecture videos [1]. These HD videos (1920x1080 pixels) were recorded using a single, still camera covering the entire whiteboard with no zooming, tilting or panning. It consists of 12 lecture videos - 5 for training and 7 for testing. The average length of all videos is about 49 minutes.

AccessMath uses the 'keyframe' method of lecture summarization and evaluation is carried out by measuring the number of keyframes produced by the summarization methodology. Apart from this, the average recall and precision of all 'matching' binary connected components (CC) are measured across the entire video ('global') as well as per frame. The AccessMath dataset is annotated at the binary level in order to facilitate this evaluation scheme and the benchmarking methodology at the CC level [1].

To determine if one or more ground truth CCs correspond(s) to one or more predicted CCs, the predicted summary frames are translated and aligned with the ground truth frames, such that overall pixel-wise recall is maximized. Then, overlapping CCs are selected and the pixel-wise recall and precision is computed. One-to-many, many-to-one and many-to-many overlaps are handled by grouping CCs appropriately [1]. These measures are designed to compensate for variations in thickness and focus on readability of extracted summary CCs.

B. Lecture Summarization

We provide an overview of prior work on our main focus, lecture video with handwritten content on a whiteboard. Most approaches in the literature follow the general pipeline of preprocessing, content extraction and summarization. Image processing and computer vision techniques are used extensively in each stage to obtain better performance.

At first, an off-the-shelf binarization technique such as Otsu's [6] algorithm on every frame is used in simple videos with not too many challenges in illumination and background [2]. Segmentation of region of interest or some background subtraction followed by specialized binarization techniques are employed for preprocessing when the lecture video poses illumination and background challenges [7], [3], [4], [1].

After preprocessing, handwritten content is extracted or separated from background content. A common approach is to divide the video frame into a grid of cells followed by rule-based or statistical classification of each cell as content, background and noise [8], [2], [9], [10]. Grouping and refining of handwritten content using OCR based methods [2] or temporal analysis [11], [1] to handle noise and occlusion are commonly used as well. Some work uses contrast enhancement [2], [8] while others use super-resolution based methods [9] to improve readability of whiteboard content.

Several methods exploit computer vision techniques to take advantage of the characteristics of lecture videos. Explicitly modeling the speaker allows better handling of occluded content [11], [3], [8], [12], [4], [9], while detecting erasure events is useful for segmentation and content extraction [3].

The final stage after preprocessing and content extraction is the summarization of the video. Video summaries in general could be of the 'keyframe' variety (described in Section I) or a 'video skim' which is a shorter version of the input video containing the highlights of the entire video. We concentrate on keyframe based summaries for our work. Keyframes are typically decided by analyzing content peaks in frames over time and segmenting the video when the content drops from a maxima, which generally corresponds to erasure events in the lecture [3], [4]. A recursive algorithm to find the correct frames to segment based on spatial conflicts of extracted content is presented in the work [1] which results in state-of-the-art summarization performance for the AccessMath dataset. Other forms of summaries of lecture videos include recognized text lines extracted from the video [9], [2] and production of composite images that contain all content [3], [4].

In our work, we use a handwritten content detector (HCD) adapted from a scene text word detector model, and use it in lieu of the preprocessing and content extraction stages. We believe this step is necessary to build a generalized pipeline to handle videos with multiple cameras and production effects.

C. Scene Text Detection

In order to detect handwritten content from video frames directly, we focus on some relevant prior work carried out in the domain of scene text detection in images and videos. A comprehensive survey of methodologies and evaluation strategies for text detection, tracking and recognition in video images is presented by Yin et al. [13]. In general, evaluation of detecting text content in video is done by treating text regions as objects and using multiple object tracking (MOT) metrics [14]. Specifically detecting handwritten content as a specialized case of scene text detection in video is covered in a survey by Ye and Doermann [15].

Earlier methods extract pixel or component level features to identify text candidates which are then post-processed using statistical learning models [16], [17]. A list of prior methods can be found in the survey by Zhu et al [18]. Of late, deep learning based methods have been adopted due to their effectiveness in taking advantage of large annotated

datasets [19], [20], and most of these treat text detection as a specialized case of object detection.

Popular strategies in deep object detection use a fixed set of predefined anchor windows that slide across convolutional feature maps in a deep neural network. At each location in the feature map, a detector network makes a prediction of whether an anchor window contains an object and if so it regresses the offsets to the anchor window dimensions to fit the object in a tight bounding box. This is the basic principle in state-of-the-art object detectors like Faster-RCNN [21], YOLO [22] and SSD [23]. SSD, in particular, carries out detection on feature maps at multiple depths and combines the predictions using non-maximum suppression (NMS). Text detection neural networks generally adapt object detection networks by making modifications to size and aspect of ratio of anchor windows and search locations across feature maps [24], [20], [5] and then retraining on scene text datasets.

We choose TextBoxes [5] as the base neural network for adaptation in our methodology because it is recent, and has implicit multi-scale feature extraction due to its SSD-based structure. Further, model weights trained on the VGG Synthetic Scene Text Database [20] and ICDAR 2015 competition dataset [25] along with code are made publicly available. We feel it is an interesting study to adapt a detector trained under for object detection into a pixel-based HC retrieval task.

III. Lecture Video Summarization

Our lecture video processing pipeline takes as input a whiteboard lecture video recorded using a still camera, and generates as output a small set of binary images of the extracted handwritten content. It is illustrated in Figure 2. First, we use our proposed handwritten text content detector to obtain text regions in frames sampled at 1 fps (frames per second). Further, we carry out coarse-grained temporal refinement (CTR) of content bounding boxes to account for variations in detector output due to occlusions and illumination changes in the video. We then binarize the detected content and reconstruct occluded content based on temporal stability of binarized CCs. This constitutes our fine-grained temporal refinement (FTR). Finally, we apply an existing lecture summarization approach [1] to further reduce the set of binary images to a smaller set containing all unique identified text regions using a set of keyframes.

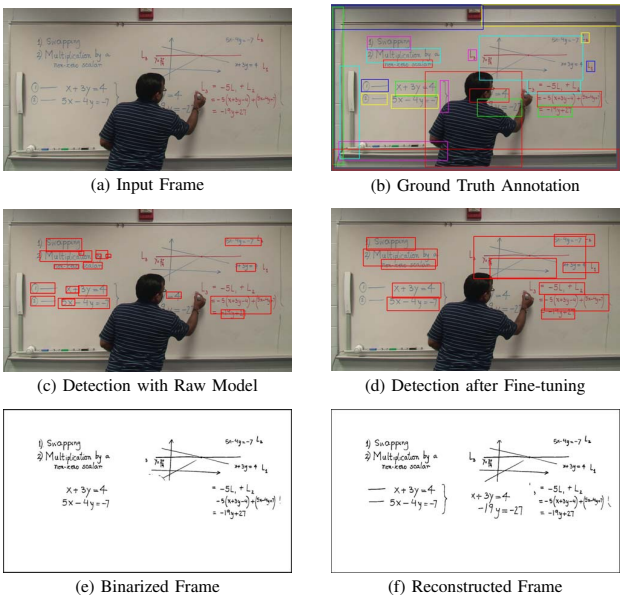
A. Handwriting Detection on Lecture Videos

TextBoxes is a 28-layer fully convolutional network. The first 13 layers are taken directly from the SSD architecture [23]. On top of these, 9 more convolutional layers are stacked in a feed-forward fashion. TextBox layers [5] are connected to 6 of the 9 additional convolutional layers. On every map location, a TextBox layer predicts a 72-dimensional vector, which are the text presence softmax scores (2-d) and coordinate offsets (x, y, w, h) for 12 default anchor boxes. The network is used at 3 image input scales - 300, 600 and 900. A non-maximum suppression is applied to the aggregated outputs of all text-box layers and scales.

Out of the box, the model trained on detecting English words in natural scenes [25], [20], did not perform well on lecture video data. Particularly, we observed that the model failed to detect the variety of handwritten content and layouts which are expected in a lecture video. This established the need to train a specialized HC detector.

We decided to 'fine-tune' TextBoxes to a create a dedicated model for detecting HC. This is a commonly used technique for domain co-adaptation which has proven to work well for diverse computer vision applications with limited training data [26]. Deep learning models are initialized with weights pre-trained on standard, larger datasets of a similar nature as that of the target application; then the models are trained on the dataset at hand. It should be noted that AccessMath does not contain annotations of the content in terms of bounding boxes nor the timestamps when it was written/erased, since it was intended to be evaluated at the binary CC level.

Therefore, we annotated the approximate frames in which a single unit of handwritten content was just written completely and when it began to get erased. This lets us quickly and automatically annotate the content in all frames in between, instead of manually annotating every frame which is time-consuming and challenging. To mitigate the effect of occlusion due to movement of the lecturer, we eliminate all text boxes which overlap with the bounding box of the lecturer if their area of intersection is a greater fraction of the text box area than 25%. Sample frame and its annotations are shown in Figure 2(a) and (b) respectively.



(a) Input Frame (b) Ground Truth Annotation

(c) Detection with Raw Model (d) Detection after Fine-tuning

(e) Binarized Frame (f) Reconstructed Frame

Fig. 2: Illustration of proposed pipeline. Sample outputs of out-of-the-box and fine-tuned handwritten content detection models for input frame 2(a) are shown in Figures 2(c) and 2(d). Figure 2(e) and 2(f) are binarized and reconstructed via temporal refinement, generated using the fine-tuned model.

We attempted to annotate the lecturer bounding box in a similar fashion by marking the beginning and end of major movement events, and interpolating intermediate frames. However, due to large number of quick movements, gesticulation and writing/erasing events this was time-consuming and hard to verify. Therefore, we used an SSD [23] network (trained on the VOC dataset [1]) on the video sampled at 1 fps to obtain lecturer bounding boxes, and linearly interpolated bounding boxes in unsampled frames. In case of multiple detections, we retain the one with largest Jaccard overlap index, also called 'Intersection Over Union' (IOU), above a threshold of 0.25 with respect to the previous frame's detection. If there are no overlapping person detections above this threshold, we take the box with highest confidence.

The TextBoxes model was fine-tuned using the training procedure described in detail in Section IV. Since the recall and precision at the level of binary CCs are evaluated in the final stage of the pipeline, we decided to tune the content detector to be biased towards slightly larger bounding boxes in order to maximize handwritten pixel recall later in the pipeline. Therefore, we double the multiplicative scaling parameter (also termed `variance` in some implementations [2]) of all bounding box regression output layers. Qualitative results comparing the two models are shown in Figure 2(c) and (d) for a sample lecture frame and quantitative results are provided in Table I and discussed in detail in Section IV.

TABLE I: Quantitative evaluation of out-of-the-box TextBoxes (TB) and fine-tuned (FT) models. The metrics are explained in Section IV.

Model	Avg. No. per Frame	Avg. per pixel recall	Avg. per pixel precision	Avg. per pixel F-score
TB	17.67	38.66	83.98	48.59
TB + FT	12.25	81.87	76.20	76.48

B. Temporal Refinement

Our temporal refinements are based on the CC stability approach from an existing method [1]. We call this method Temporal Analysis Algorithm (TAA). The input of this method is a set of individual objects detected for each frame and the output is a spatio-temporal structure that groups together stable elements that overlap in space and time. The algorithm runs in two major passes: first for identifying stable elements, and second for spatio-temporal grouping.

Temporal Analysis Algorithm: During the first pass, for a given matching criteria and every sampled frame in sequential order, each element is tested for a match against every other element present in all previous frames that are within time t before the current frame's timestamp ($t = 85$s). Matched elements are treated as instances of unique objects. After finding all the unique elements from the input frames, only those that appear in more than n frames ($n = 3$) are marked as stable and are kept for further processing.

In the second pass, elements that have an overlap in space and with duration intervals that overlap or have an intermediate gap of at most 5 seconds, are grouped into larger 'temporally stable' groups. These can be used as the final units for content summarization and further temporal processing. Other elements that only overlap in space but do not belong to the same temporal group are marked to be in *conflict*, and this information is used for summarization in order to produce temporal splits of the video that minimize the number of conflicts present on each video segment.

Coarse-grained Temporal Refinement: The output of our HCD is a set of bounding boxes. Due to false positives, variations in illumination and occlusion by the speaker, these bounding boxes might change for contiguous frames. We obtain a coarse-grained temporal refinement for the detected text bounding boxes based on the application of the temporal analysis algorithm described above. Two bounding boxes are accepted as a match if they have an IOU value above 0.5. Designing content-based matching criteria for bounding boxes is out of the scope of our current work.

Binarization. Frame binarization is achieved by processing the text detection bounding boxes and combining the partial results using pixel-wise OR operations. First, background is estimated using a median filter [1] and is subtracted from each bounding box region. Then, Otsu's binarization is applied over the resulting edge image.

Fine-grained Temporal Refinement. Since we currently work under the assumption that only one still camera is used, without zooming or panning, we also assume that missing CCs from intermediate frames, between their first and last known location are caused by occlusion due to the lecturer. We apply a fine-grained temporal refinement over the detected CCs from the binary frames using the Temporal Analysis Algorithm, to recover occluded content and remove any noise caused by HCD false positives. The expected output is a set of reconstructed binary video frames as though the lecturer was never occluding any of the handwritten content. The combined effect of both stages of temporal refinement on the overall task of lecture summarization can be seen in Table II.

C. Content Summarization

The binarized, reconstructed video has several redundant frames which record the text being written gradually until the whiteboard is filled, erased and rewritten. In our work, we summarize lecture videos by finding those 'keyframes' that completely contain all unique CCs from the previous erasure event up to the time new text appears in the same spot. In their work [1], the authors identify all conflicting CC pairs during a video segment and deploy *conflict minimization*. This greedy algorithm segments a video lecture so as to maximize the number of conflicts resolved per cut, recursively on the reconstructed video. We use the same algorithm to produce lecture summaries. The final results can be found in Table II.

[1] http://host.robots.ox.ac.uk/pascal/VOC/voc2012/index.html
[2] https://github.com/MhLiao/TextBoxes

IV. Experiments

We used the AccessMath dataset to train and test our HCD, which is based on the TextBoxes neural network structure [5]. We annotated the training videos using the procedure described in III-A. We split the annotated training frames into train and validation sets in the ratio 4:1. This resulted in about 10,000 training frames and about 2500 validation frames. The model is then fine-tuned for about 10,000 mini-batch update iterations with batch-size 32 on the train split. The learning rate decreased uniformly, so as to effectively decrease the learning rate by a factor of 0.1 every 5000 iterations. We used a stochastic gradient descent (SGD) optimizer with a base learning rate of 0.0001 and weight decay of 0.0005. The validation performance was measured using 11-point mean average precision (P_{11pt}) over all validation frames and was 77.67% at the end of training. Further training with lower learning rates yielded little gain in validation performance.

$$\tilde{P}_i(r_j) = \begin{cases} \max_{r_j \le r \le r_{j+1}} P_i(r) & \text{if } P_i(r) \text{ exists} \\ \tilde{P}_i(r_{j+1}) & \text{otherwise} \end{cases}$$
$$P_{11pt} = \frac{1}{11} \sum_{j=0}^{10} \frac{1}{N} \sum_{i=1}^{N} \tilde{P}_i(r_j);\ r_j = j \times 0.1 \quad (1)$$

Here, $P(R = r)$ indicates precision at recall r and N is number of training images.

Even though the number of frames seems large, it should be noted that many training frames will include redundancies in terms of text boxes with minor changes in illumination and speaker positions. Therefore, the training frames can be viewed as a naturally augmented smaller dataset, and along with the initialization from scene text detection network, could have caused the earlier convergence.

We evaluate the isolated HCD based on pixel-wise recall, precision and f-measure with respect to annotated content bounding boxes. We search for the optimal confidence threshold in the range of 10% - 95% using f-measure obtained on the training video frames. Quantitative comparison of out-of-the-box TextBoxes (TB) and fine-tuned (FT) models on testing videos for these metrics are shown in Table I. We found that for the optimal confidence threshold for the TB model was 10% and 65% for the TB + FT model. Although we expect the fine-tuned model to do better than the out-of-the-box model, the stark difference in performance and model confidence assures us that annotating the data and retraining a specialized model was worthwhile.

For the lecture summarization task, the same evaluation procedure described for the AccessMath dataset was followed (see Section II-A). The performance of our pipeline and state-of-the-art method with respect to all these metrics at various stages of summarization are presented in Table II. It must be noted that the average recall and precision mentioned in this table are different from the ones in Table I. We measure fine-grained content recall and precision (at the binary CC level) for summarization as opposed to coarse-grained content recall and precision (at bounding box level).

We observe in the *Binarization* section of Table II, that replacing the heuristic-based whiteboard segmentation approach of the current state-of-the-art method [1] with our deep learning based handwriting detection model, has a considerable improvement on the pipeline performance. This difference is amplified by the fact that we use only Otsu's binarization [6] in instead of the hybrid binarization used in [1]. We can see that our HCD consistently has higher f-score than the out-of-the-box model which can be attributed to higher recall by recovering a variety of HC on the whiteboard.

The reduction in recall in our method at later stages, compared to state-of-the-art, arises from the fact that our HCD model currently does not benefit from temporal information. The detector produces spurious bounding boxes in a few frames, due to illumination changes, which adversely affects performance of the temporal refinement stage, especially at the coarse-grained level which operates on bounding boxes themselves. Currently boxes are grouped purely based on area metrics which might cause many undesirable merges. In the future, we will investigate content-based temporal refinement to refine detected bounding boxes to mitigate this effect.

V. Conclusion

We set out to investigate the impact of *replacing heuristic and image processing based stages* on the state-of-the-art pipeline with *machine learning based methods*. The purpose of the pipeline was to summarize lecture videos by recovering all unique handwritten whiteboard content. We focused on the aspect of detecting handwritten content using a vision-based deep learning model treating the lecture video dataset as a specialized case of text detection in videos.

A state-of-the-art scene text detection model was adapted for detecting handwritten whiteboard content in lecture videos. This was necessary in order to capture diverse handwritten

TABLE II: Comparison of fine-tuned (FT) TextBoxes (TB) model based lecture video summarization pipeline, including temporal refinement (TR) by measuring recall (R), precision (P), f-score (F) and number of frames (N_f).

Method	AVG N_f	AVG Global R	AVG Global P	AVG Global F	AVG Per Frame R	AVG Per Frame P	AVG Per Frame F
Binarized							
Baseline [1]	296	98.96	64.01	77.73	98.69	63.30	77.12
TB	296	88.52	81.80	85.02	86.40	**84.52**	85.44
TB+TR	296	93.96	69.70	80.03	93.93	75.32	83.60
TB+FT	296	97.03	**86.05**	**91.21**	95.35	83.95	**89.28**
TB+FT+TR	296	98.27	62.61	76.48	97.60	66.32	78.97
Reconstructed							
Baseline [1]	296	96.95	94.28	95.59	96.49	90.51	93.40
TB	296	83.16	93.13	87.86	82.39	91.59	86.74
TB+TR	296	88.74	**96.26**	92.34	88.78	**93.96**	91.29
TB+FT	296	91.44	95.63	93.48	90.28	92.38	91.31
TB+FT+TR	296	92.76	95.46	*94.09*	92.09	92.37	*92.22*
Summarized							
Baseline [1]	18	96.28	93.56	94.90	95.73	92.21	93.93
TB	22	82.61	92.66	87.34	81.82	91.87	86.55
TB+TR	17	88.29	**95.39**	91.70	88.41	**94.22**	91.22
TB+FT	22	90.98	94.77	92.83	89.89	93.93	91.86
TB+FT+TR	20	92.33	94.16	*93.23*	91.69	93.45	*92.56*

content such as phrases, sentences, mathematical expressions, multi-line sentences and expressions, sketches, labeled plots and matrices, which are typically expected in lecture videos. The problem was further challenged by occlusions and illumination changes induced by lecturer movement.

In terms of summarization metrics, we showed that the predicted bounding boxes were able to recall globally most of the binarized whiteboard content with high precision. This shows promise that a specialized *handwritten content detector trained end-to-end along with temporal information could perform better* on the task of lecture video summarization. Better temporal refinement and binarization techniques will be required to achieve state-of-the-art, in terms of recall.

AccessMath dataset was enriched by additional handwritten content ground truth in the form of bounding boxes. The annotation process was carried out manually for both training and testing set of videos. The new annotations along with the tools used to create them, the trained detection models, and the code for deployment and evaluation have been released. We believe that the handwriting research community can benefit from this contribution [3].

In the future, we plan to investigate temporal refinement using content of bounding boxes, end-to-end detection of content in video frames with LSTM-based text tracking networks. We want to focus on lecturer pose estimation to better handle occlusions and perform text detection via recognizing writing and erasing actions. In addition, we want to collect and annotate lectures, with multiple cameras incorporating pan, zoom and tilt effects, while generalizing the summarization stages to extract semantically meaningful information. We also plan to incorporate material from subjects other than math, to build a comprehensive video lecture dataset.

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

REFERENCES

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

[2] 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.

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

[4] 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.

[5] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "Textboxes: A fast text detector with a single deep neural network." in *AAAI*, 2017, pp. 4161–4167.

[6] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[7] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[8] P. E. Dickson, W. R. Adrion, and A. R. Hanson, "Whiteboard content extraction and analysis for the classroom environment," in *Multimedia, 2008. ISM 2008. Tenth IEEE International Symposium on*. IEEE, 2008, pp. 702–707.

[9] L. Tang and J. R. Kender, "A unified text extraction method for instructional videos," in *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, vol. 3. IEEE, 2005, pp. III–1216.

[10] P. Banerjee, U. Bhattacharya, and B. B. Chaudhuri, "Automatic detection of handwritten texts from video frames of lectures," in *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*. IEEE, 2014, pp. 627–632.

[11] 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.

[12] R. R. Shah, Y. Yu, A. D. Shaikh, S. Tang, and R. Zimmermann, "Atlas: automatic temporal segmentation and annotation of lecture videos based on modelling transition time," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 209–212.

[13] 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.

[14] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *Journal on Image and Video Processing*, vol. 2008, p. 1, 2008.

[15] Q. Ye and D. Doermann, "Text detection and recognition in imagery: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 7, pp. 1480–1500, 2015.

[16] L. Neumann and J. Matas, "A method for text localization and recognition in real-world images," in *Asian Conference on Computer Vision*. Springer, 2010, pp. 770–783.

[17] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2963–2970.

[18] Y. Zhu, C. Yao, and X. Bai, "Scene text detection and recognition: Recent advances and future trends," *Frontiers of Computer Science*, vol. 10, no. 1, pp. 19–36, 2016.

[19] A. Veit, T. Matera, L. Neumann, J. Matas, and S. Belongie, "Coco-text: Dataset and benchmark for text detection and recognition in natural images," *arXiv preprint arXiv:1601.07140*, 2016.

[20] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2315–2324.

[21] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[22] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[23] 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.

[24] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, "Detecting text in natural image with connectionist text proposal network," in *European Conference on Computer Vision*. Springer, 2016, pp. 56–72.

[25] 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.

[26] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in neural information processing systems*, 2014, pp. 3320–3328.

[3] https://github.com/bhargavaurala/accessmath-icfhr2018