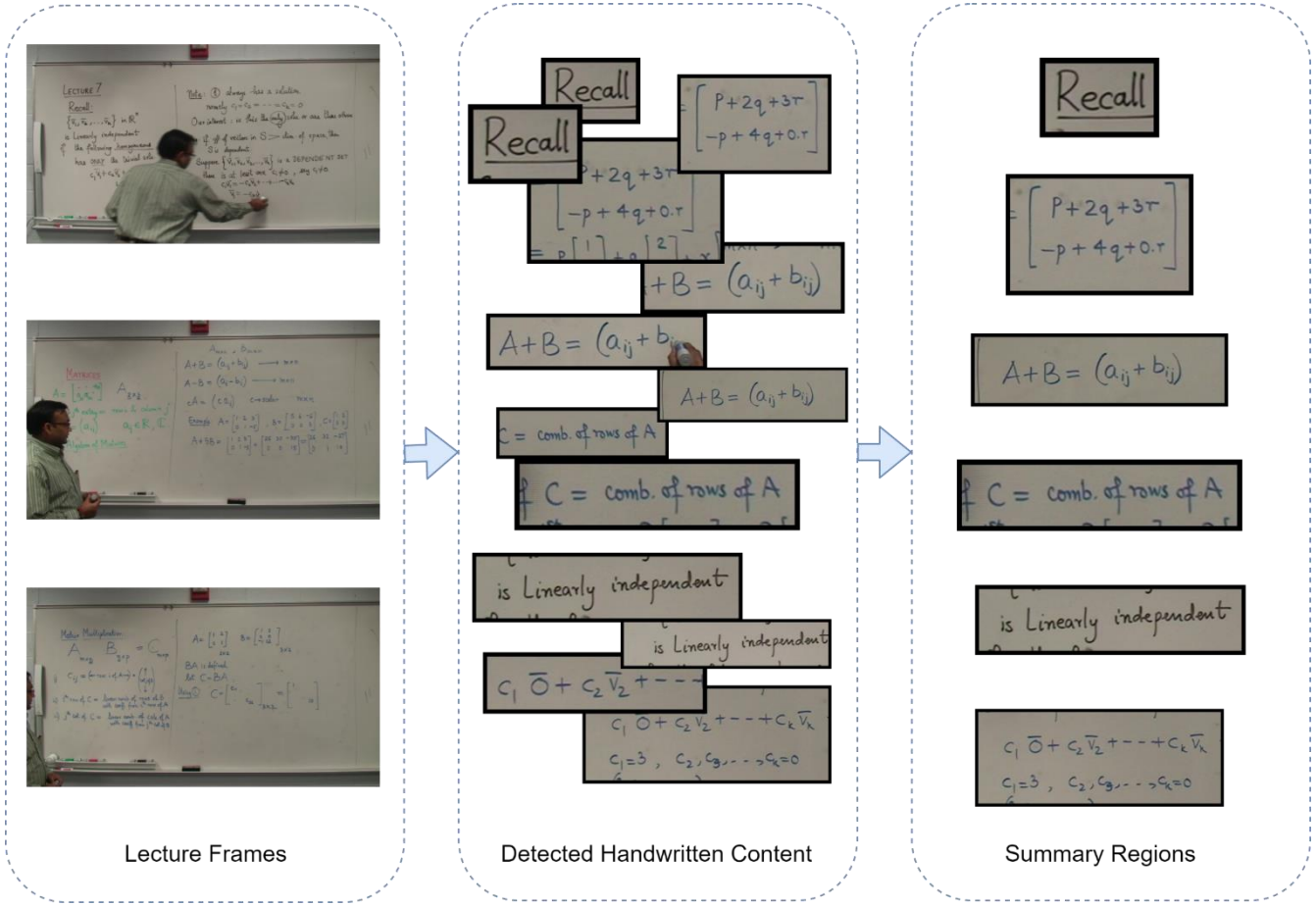
Summarizing Lecture Videos by Key Handwritten Content Regions

Bhargava Urala Kota, Saleem Ahmed, Alexander Stone, Kenny Davila, Srirangaraj Setlur, Venu Govindaraju

22/09/2019

Overview

- Motivation
- Background
- Approach
  - Content Detection
  - Feature Representation
  - Content Summarization
  - Summary Evaluation
- Experimental Results
- Conclusion and Future Directions

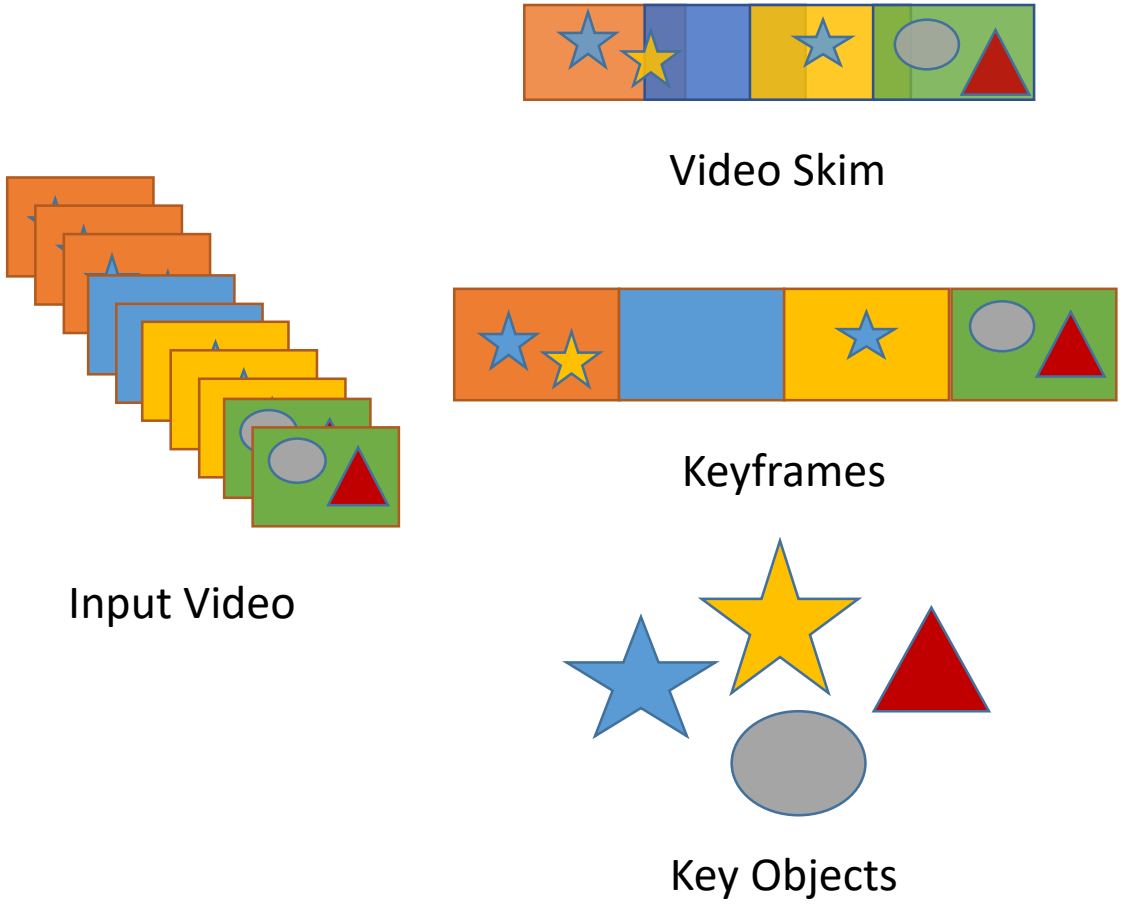


Motivation

- Lecture videos are useful for students and working professionals alike
- Current search engines rely on keyword/meta-data for indexing
- Extracting and summarizing content would facilitate more powerful search and retrieval systems

Background: Video Summarization

- Video summaries can broadly be divided into ‘keyframes’ or ‘skims’
- Recently, there are approaches attempting to summarize videos using key objects
- In our work, we extend this concept to lecture videos



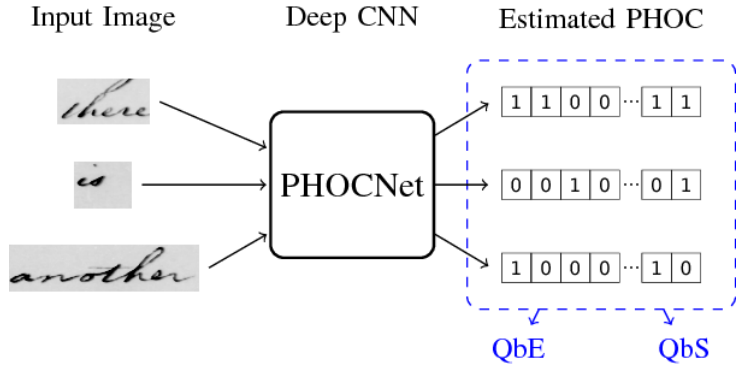
Background: Lecture Video Summarization

- Prior work generally follows the paradigm of content extraction, binarization and summarization.
- Summarization is mostly keyframes-based. Transcript and compositing of frames have also been studied.
- One publicly available dataset – AccessMath. It is evaluated on basis of:
  - Number of keyframes compared to ground truth
  - Pixel-wise Recall/Precision of binary connected components vs. ground truth

Background: Feature Extraction from Text

- Local features + Sequential Analysis (DTW/HMM/LSTM) have been used to match word image queries to word instances.
- Learning to map word region features to PHOC embedding supports both query by example and by string.
- In our work, we do not have access to transcriptions and annotated regions have different granularities.

T. M. Rath and R. Manmatha: Word Image Matching Using Dynamic Time Warping. In: Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR), Madison, WI, June 18-20, 2003, vol. 2, pp. 521-527.



Sudholt, Sebastian and Gernot A. Fink. "PHOCNet: A Deep Convolutional Neural Network for Word Spotting in Handwritten Documents." *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)* (2016): 277-282.

Our Approach

- Handwritten Content Detection
  - Neural network based on EAST
- Feature Extraction
  - Inception based sub-network on interpolated detected regions
- Summarization by Key Content
  - Spatio-temporal analysis + feature distance to find unique content
- Evaluation metric for summarization
  - Number of unique summary content regions vs. ground truth
  - Recall of ground truth in generated summaries

Handwritten Content Detection

- Detector based on EAST
- We use Feature Pyramid Network as base network
- Dice loss is used for mask prediction

$$L_{dice} = 1 - 2 \times \frac{\sum_{l=1}^{2} w_l \sum_n r_{ln} p_{ln}}{\sum_{l=1}^{2} w_l \sum_n r_{ln} + p_{ln}}; \; w_l = (\sum_{n=1}^{N} r_{ln})^{-2} \qquad (1)$$

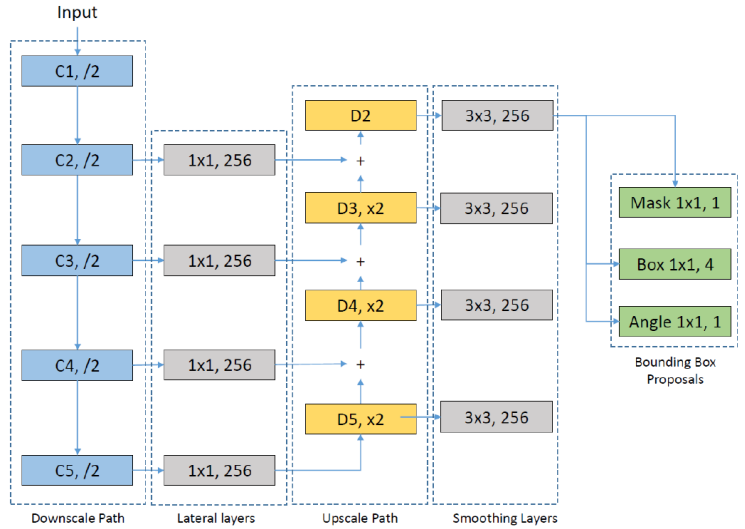
where, r and p are the targets and predictions respectively, $n \in [1, N]$ are the pixels in the target and $l \in [0, L-1]$ are the set of pixel labels.

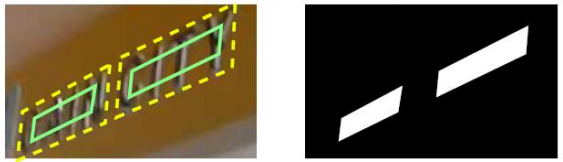
- IOU Loss is used for regression prediction

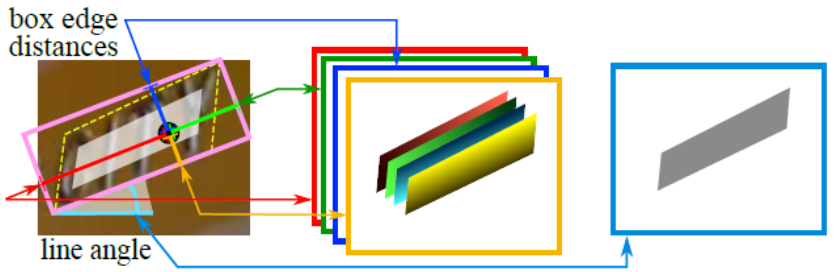
$$L_{\text{AABB}} = -\log \text{IoU}(\hat{\mathbf{R}}, \mathbf{R}^*) = -\log \frac{|\hat{\mathbf{R}} \cap \mathbf{R}^*|}{|\hat{\mathbf{R}} \cup \mathbf{R}^*|}$$

$$w_{\text{i}} = \min(\hat{d}_2, d_2^*) + \min(\hat{d}_4, d_4^*)$$

$$h_{\text{i}} = \min(\hat{d}_1, d_1^*) + \min(\hat{d}_3, d_3^*)$$



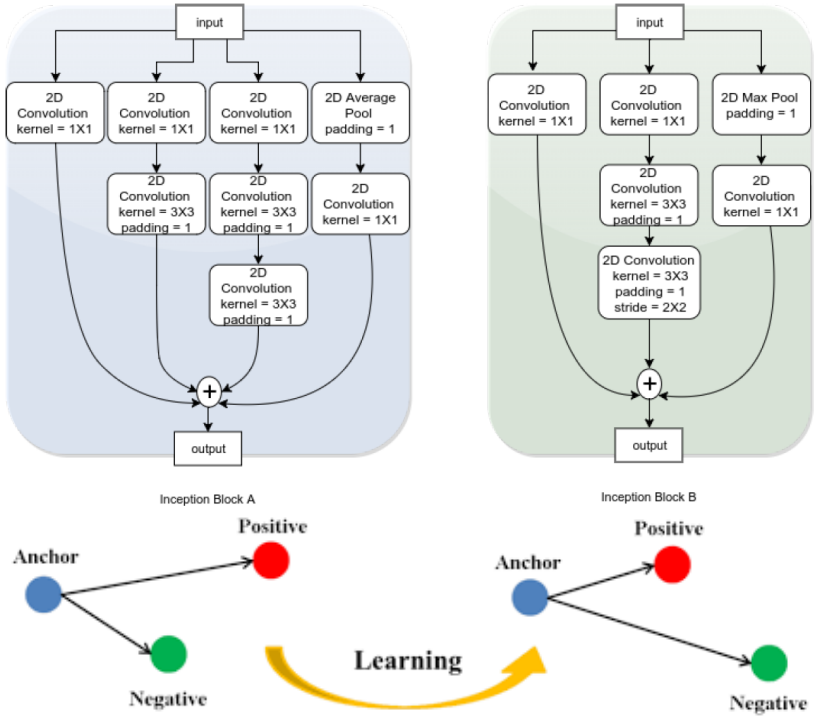




Zhou, Xinyu, et al. "EAST: an efficient and accurate scene text detector." *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2017.

Feature Representation of Content

- We do not have transcription annotation for our lecture dataset
- Region annotation on video do not have consistent granularity like words or sentences
- We use triplet loss to learn feature representations for detected content regions after interpolation
- Learnt features can be compared in a Euclidean sense to establish (dis)/similarity between two content regions
- Sampling strategy:
  - Sample from local content peaks (interval=30s)
  - Sample randomly from background regions for negative examples
  - Perturb bounding boxes



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

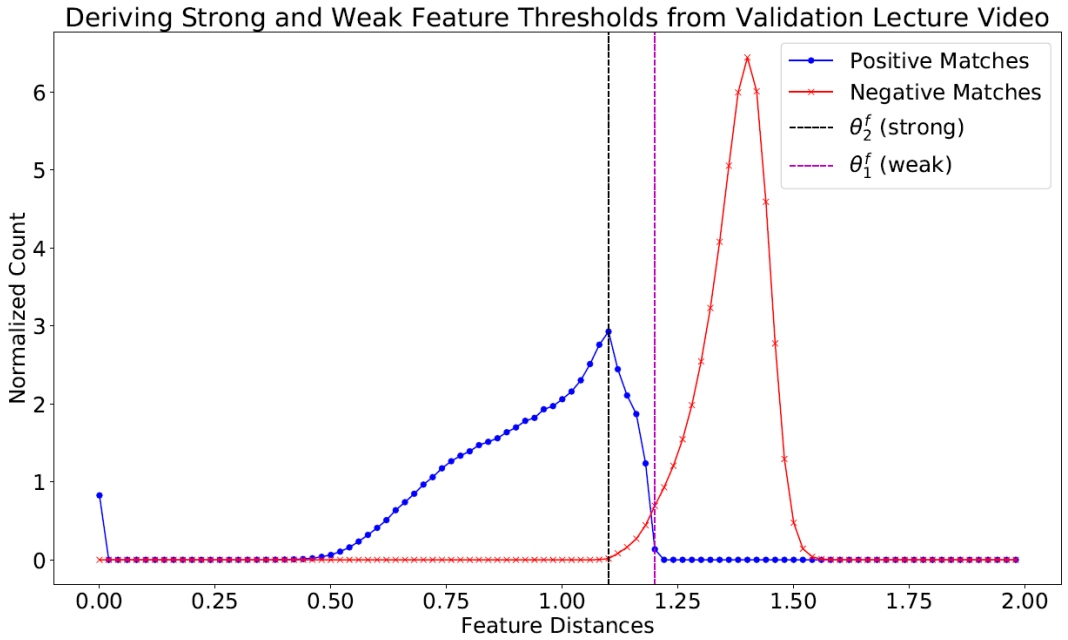
where, $f(x_a)$ is the anchor embedding and $f(x_p)$ and $f(x_n)$ are the positive and negative sample embeddings respectively, m is a margin which indicates the ideal minimum separation between the distances computed from the positive-anchor and anchor-negative pairs of embeddings.

Content Summarization

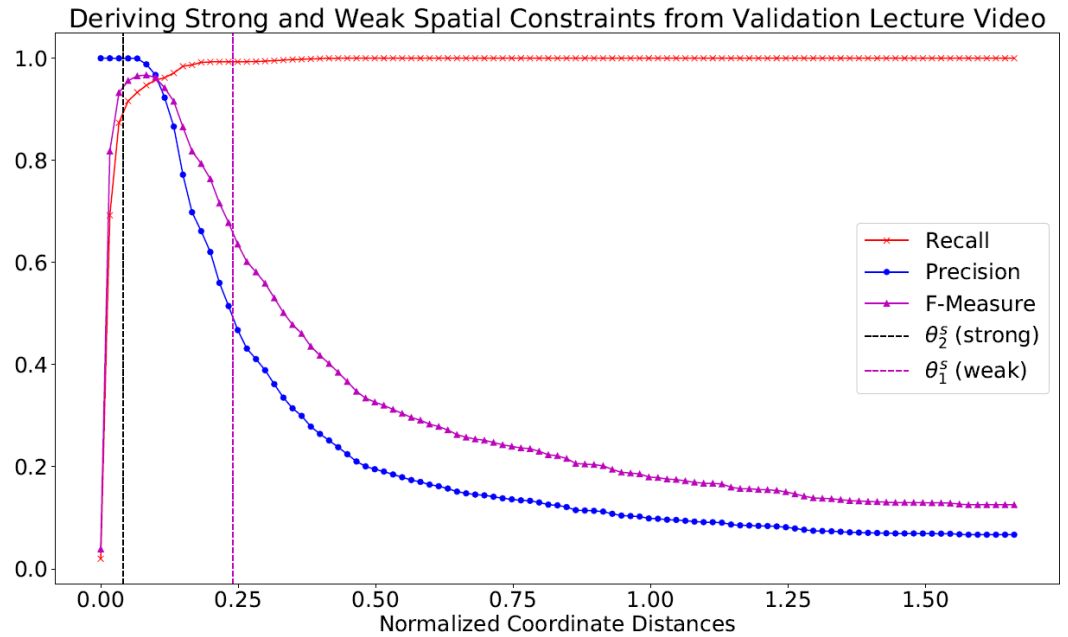
- After regions and corresponding features are extracted, we need to find the unique content regions to generate summaries

- Video is broken into 60s intervals with 30s overlap. Summarization happens over two passes:
  - Detections from every consecutive frame are compared to each other, strong feature and strong spatial matches are merged into same content region to generate seed summary content
  - Seed summary content is recursively grown across intervals by re-examining strong-weak matches until no more changes occur

- Finally, a data structure with unique content regions mapping to a list of each instance of its occurrence is obtained



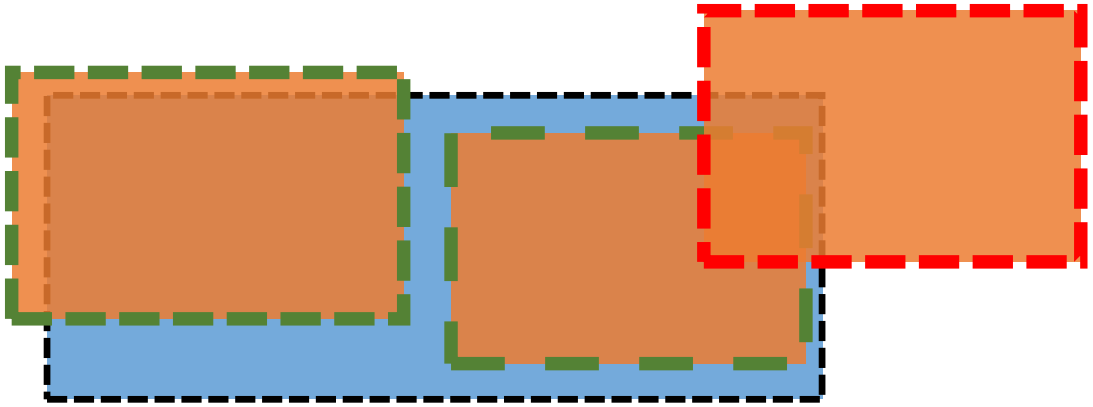
(a) Feature Thresholds



(b) Spatial Thresholds

Summary Evaluation

- Unique content regions act as our video summary
  - Analogous to key objects or key entities for general videos
- We evaluate video summaries by:
  - Comparing number of summary regions to ground truth regions
  - Measuring recall of ground truth regions against obtained summaries
  - Avg. recall for objects vs. DetEval for text regions



Ground Truth Box Predicted Boxes

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

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

where,

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

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

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

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

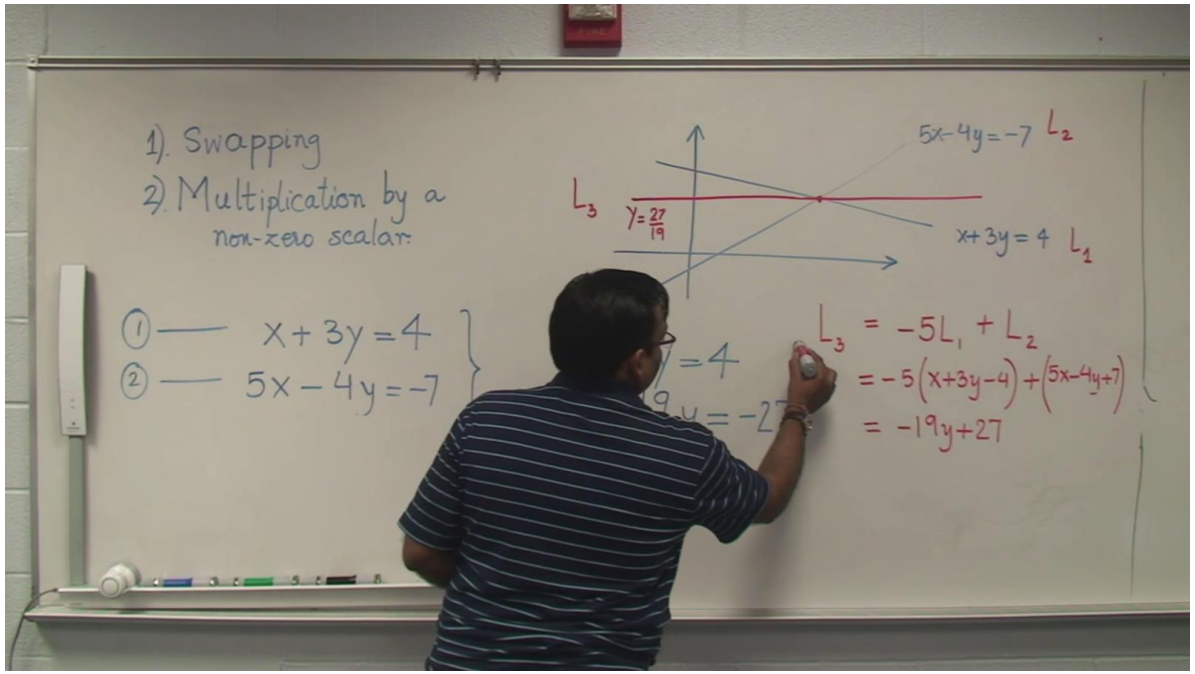
$$\forall p_j \in \mathbf{P_k} \quad \frac{g \cap p_j}{p_j} \geq t_p \text{ and } \sum_{p_j \in \mathbf{P_k}} \frac{g \cap p_j}{g} \geq t_r$$

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.

The DetEval scheme accommodates recall of segmented text region predictions with appropriate penalties

Experiments: Dataset

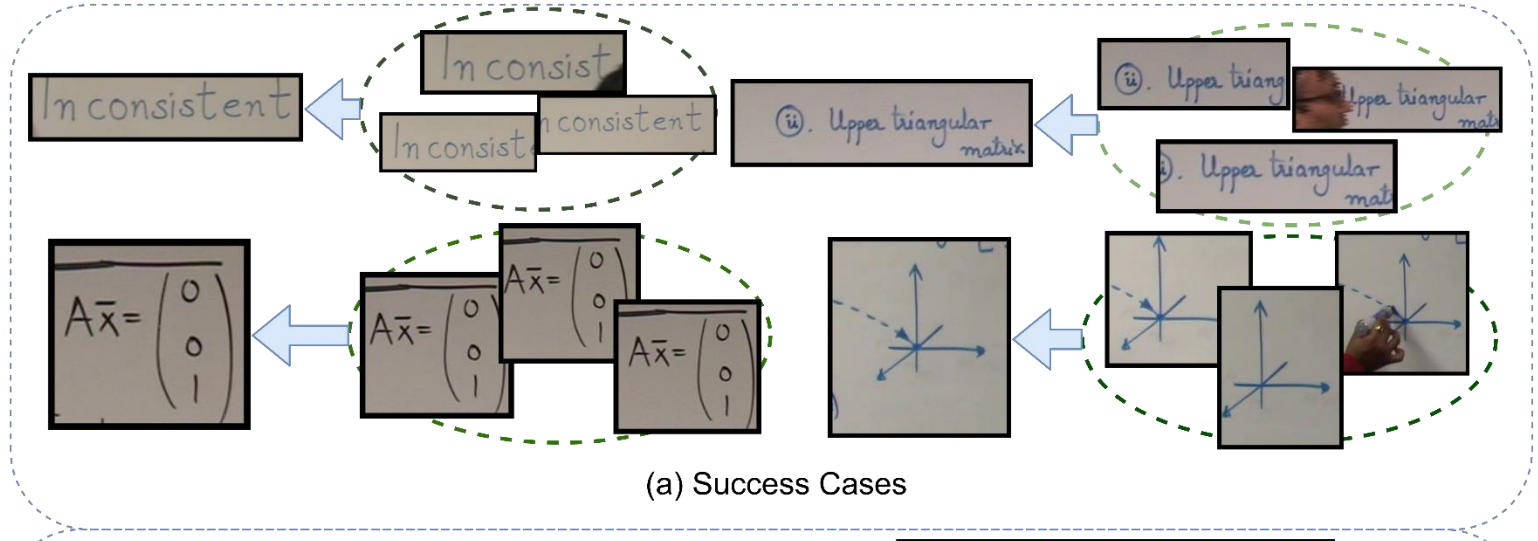
- We test our summarization on AccessMath
- Largest, publicly available, benchmarked dataset
- Collection of linear algebra lectures
- Single, still, full HD camera covering the entire whiteboard
- 12 lecture videos - 5 for training and 7 for testing.
- The average length of each video is about 49 minutes.
- Ground truth annotation:
  - Binary connected component level
  - Bounding boxes of content
- 87 unique content regions per lecture on avg. with thousands of total instance count per video.

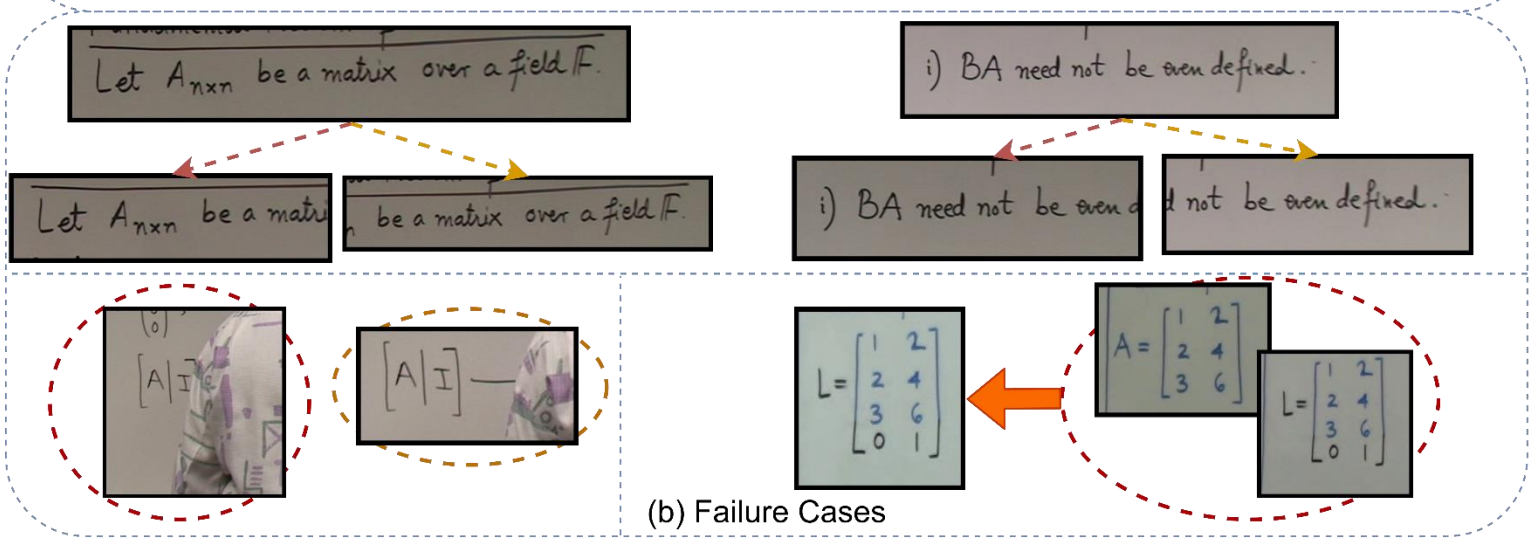


Experiments: Results

IOU	Avg. Per-frame R	P	F	Avg. Summ. 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.





Conclusions and Future Work

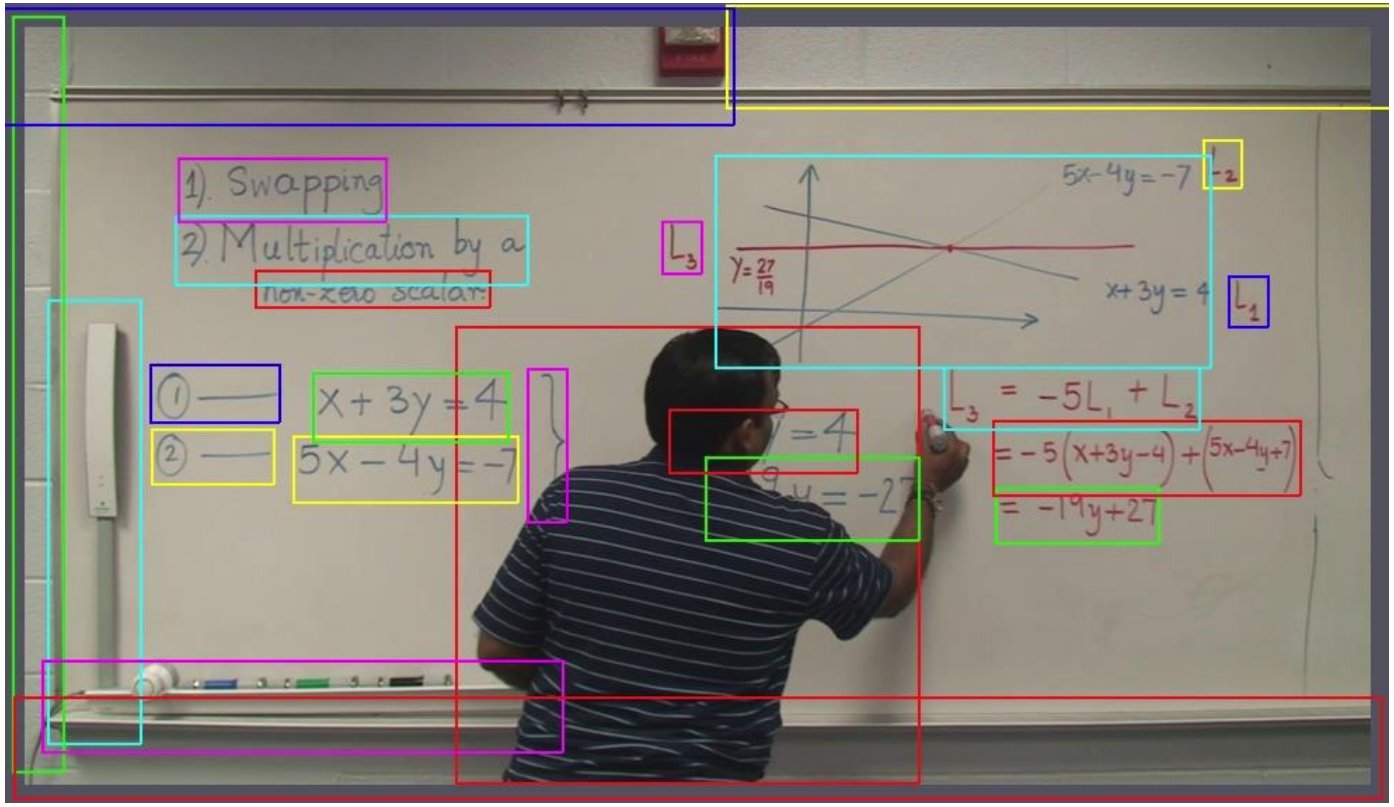
- Detection needs to be stable to illumination changes, occlusion
- Joint handling of lecturer detection and text detection
- Feature representation by triplet embedding shows promise
  - We tried visual descriptors like SIFT/SURF, BoVW – these were not robust
  - We tried global clustering methods on extracted features – this did not scale
- More fine-grained feature extraction needed
- Relation between key content and keyframes needs to explored further

Thank you

Questions?

AccessMath Bounding Box Annotation

- Manually annotating every frame in a video is too costly
- Annotate frame which marks ‘end’ of writing event and beginning of erasure event for each HC ‘unit’
- Lecturer bounding boxes are generated using SSD[3] pre-trained on VOC*
- Sentences, expressions, matrices, sketches - could be multi-line, mixed



3. Liu, Wei, et al. "Ssd: Single shot multibox detector." European conference on computer vision. Springer, Cham, 2016.

* http://host.robots.ox.ac.uk/pascal/VOC/

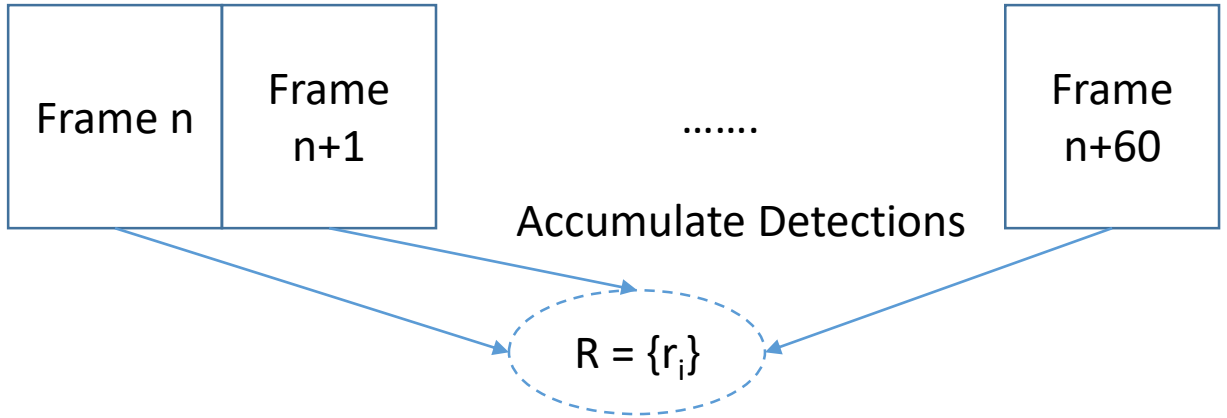
Training EAST-based HCD

- The ResNet portion is initialized with pre-trained ImageNet weights, 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.
- The learning rate is reduced at a constant rate of 0.7943 per epoch. This ensures that the learning rate drops by a factor of approximately 0.1 every 10 epochs.
- Each sample is augmented as described by EAST, with random 512x512 crops.

Design Choices for EAST-based HCD

- 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
- FPN for feature extraction with ResNet backbone; deconvolution layers and activations as originally prescribed for FPN
  - extracts multi-scale features
  - has readily available initialization weights from ImageNet training
  - state-of-the-art object detection performance
- DICE loss was found more stable numerically than BCE Loss which has exponential/log calculations

Spatio-temporal Processing Details



First pass

For every r_i, r_j in R, j ≠ i

- Compute feature distance: $d_{ij}^f = \|f(r_i) - f(r_j)\|_2^2$
- Compute spatial distance: $d_{ij}^s = \|\mathbf{x}_i - \mathbf{x}_j\|_2^2$
- Where, f is the feature extractor and **x** are the coordinates of normalized bounding box corners of r_i and r_j respectively
- Merge r_i and r_j into the same summary region S_m, if both distances are within respective thresholds
- Split r_i and r_j into different summary regions, if both distances are outside of respective thresholds
- Mark as tentative and split otherwise

Repeat till end of video

Second Pass

- Aggregate each S_m by merging bounding boxes and averaging features.
- Recompute spatial distance and feature distances for each S_m wrt every other S_n, n ≠ m:
- Merge or split according to same criteria with a temporal tolerance of 60s
- Repeat till convergence.