

Robust Text Activity Detection from Videos of Computer Monitors

Krithika Saravanan¹, Venkatesh Jatla², Marios S. Pattichis², Carlos LópezLeiva³, and Sylvia Celedón-Pattichis³

¹ krithika@unm.edu

Dept. of Computer Science

University of New Mexico, United States.

² {venkatesh369, pattichi}@unm.edu

image and video Processing and Communications Lab

Dept. of Electrical and Computer Engineering

University of New Mexico, United States.

² {callopez, sceledon}@unm.edu

Dept. of Language, Literacy, and Sociocultural Studies

University of New Mexico, United States.

Abstract—The paper introduces a robust method for detecting text regions in videos of computer monitors. The basic problem comes from the need to assess middle-school student learning when using a low-cost Raspberry Pi computer. To assess learning, we need to detect the text entered by the students without using video capture software that would severely limit the performance of the Raspberry Pi. We describe a robust approach that uses projections and quadtree decompositions to detect the monitor, the windows boundaries, and the individual characters. We provide visual assessment on ten challenging videos that exhibit non-uniform illumination, monitor shaking, and variable viewing angles.

I. INTRODUCTION

There is strong interest in the development of robust methods for video activity recognition [1]. Many of the problems come from strong variations in image acquisition and the complexities associated with different video activities.

The current paper considers the detection of text regions from digital videos as shown in Fig. 1. The basic challenges come from non-uniform illumination, camera and monitor motions, variable viewing angles, and significant changes in video content.

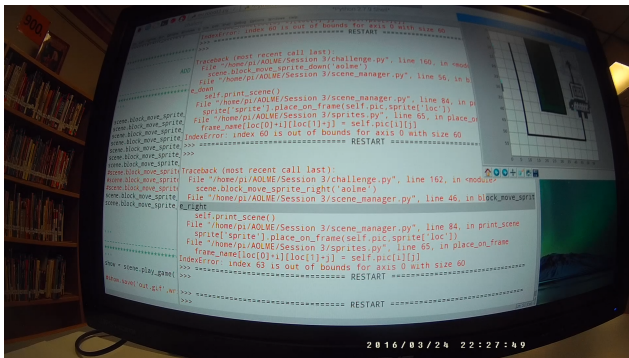


Fig. 1. Text region detection in digital videos of computer monitors.

II. METHOD

We summarize our basic approach in Fig. 2. In what follows, we provide detailed descriptions of each step. We also demonstrate each step on an image example in Fig. 3.

First, we note that the digital camera is positioned in front of each monitor. Thus, we expect that the screen and the text will be displayed along mostly horizontal lines through the image.

To allow for fast implementations, we consider the use of horizontal and vertical projections. To define the projections, let $I(i, j)$ denotes the input image. We then have that $H_i = \sum_j I(i, j)$ denotes a horizontal projection along the i -th row. Similarly, $V_j = \sum_i I(i, j)$ denotes the vertical projection along the j -th column. Our approach is motivated by the fact that black boundaries or the black pixels in letters will cause local minima along H_i or V_j (see Fig. 1). Thus, transitions are detected by looking local minima (see Fig. 3b).

The screen is detected as the largest collection of adjacent rectangles (see Fig. 3d). The text regions are then detected as the union of the small quad-tree decomposition boxes as shown in Fig. 3f.

III. RESULTS

We trained our algorithm on 25 digital images extracted from videos with $k = 5$ for KNN. We provide results on a separate set of 10 videos. The results are summarized in Table I.

Overall, the algorithm worked for about 90% of the time. We present several examples of successful detections in Fig. 4. In Figs. 4h and Fig. 4j, we show that pop-ups can lead to partial failures.

IV. CONCLUSION

We presented a robust method for detecting text regions from computer monitors. Our approach is currently being revised for processing 100s of hours of videos for assessing the

- ▷ Grid computation of regions of interest
- 1: **Load** video.
- 2: **Read** a frame.
- 3: **Compute** horizontal and vertical projections.
- 4: **Detect** transitions from projections.
- 5: **Construct** grid using transition positions.
- 6: **for** each frame **do**
 - ▷ Detect monitor using KNN classifier.
- 7: **Classify** rectangular regions as screen or no screen.
- 8: **Find** largest connected component by joining adjacent rectangles.
- ▷ Detect text using Quad Tree Decomposition.
- 9: **Apply Quad Tree Decomposition** on screen region.
- 10: **Select** quad tree decomposition with box sizes 2, 4.
- 11: **Dilate** selected image.
- 12: **Save** dilated image as output frame.
- 13: **end for**

Fig. 2. Text area selection algorithm

TABLE I
SCREEN DETECTION ACCURACY. ACCURATE DETECTION TIMES REFERS TO THE TOTAL AMOUNT OF TIME FOR WHICH THE TEXT REGION WAS SUCCESSFULLY DETECTED.

Video	Accurate Detection	Video length	Percentage
Video 1	16 sec	16 sec	100%
Video 2	11 sec	11 sec	100%
Video 3	3.334 sec	4 sec	83%
Video 4	22.8 sec	24 sec	95%
Video 5	11 sec	11 sec	100%
Video 6	5 sec	5 sec	100%
Video 7	19 sec	19 sec	100%
Video 8	26 sec	26 sec	100%
Video 9	2 sec	11 sec	18%
Video 10	5 sec	17 sec	30%

amount of time that students spend on typing versus running their code.

REFERENCES

- [1] A. C. Bovik, *The essential guide to video processing*. Academic Press, 2009.
- [2] C. Carranza, D. Llamocca, and M. Pattichis, "Fast and scalable computation of the forward and inverse discrete periodic radon transform," *IEEE Transactions on Image Processing*, vol. 25, no. 1, pp. 119–133, Jan 2016.
- [3] —, "Fast 2d convolutions and cross-correlations using scalable architectures," *IEEE Transactions on Image Processing*, vol. 26, no. 5, pp. 2230–2245, May 2017.

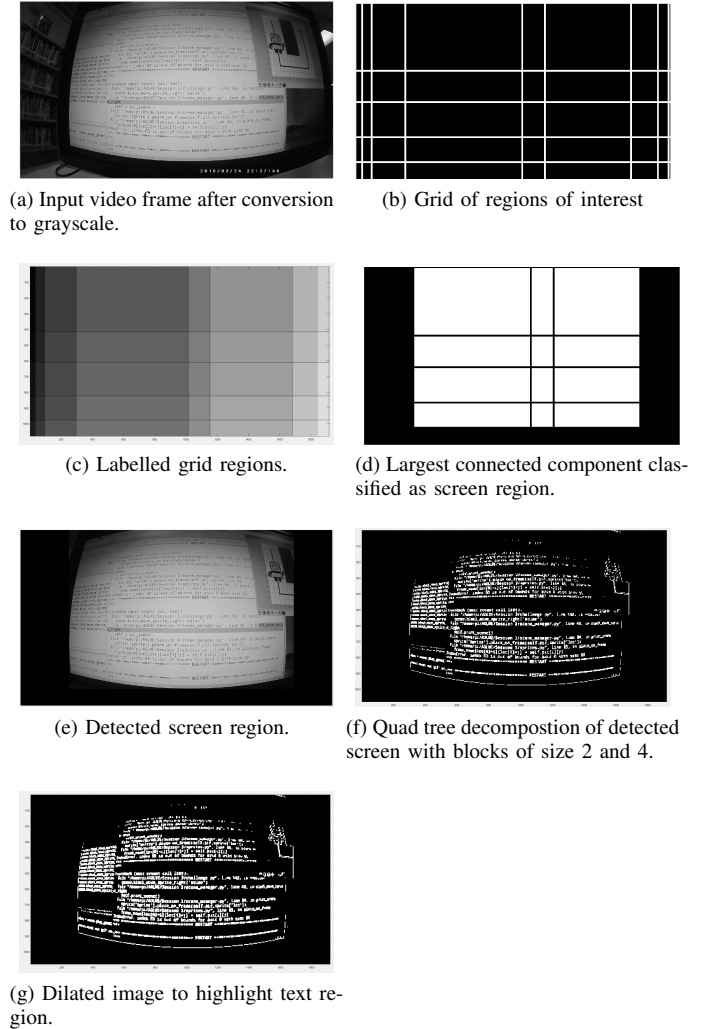
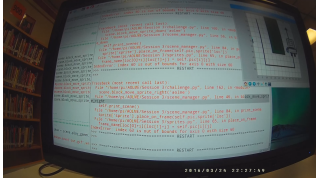


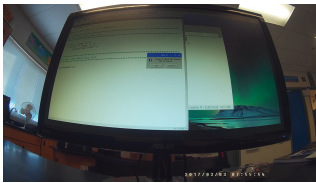
Fig. 3. Text region detection example.



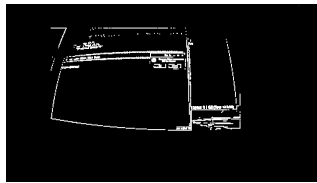
(a) Frame having lot of text.



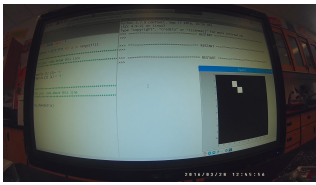
(b) Correct detection of text region.



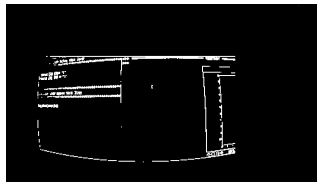
(c) Frame having two overlapping windows.



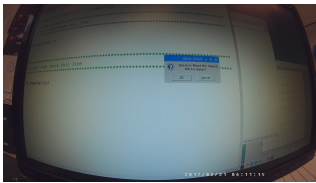
(d) Correct detection of overlapping windows.



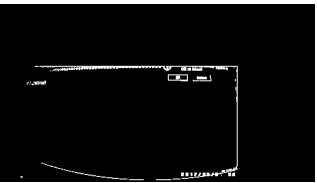
(e) Frame having figure window on top of editor.



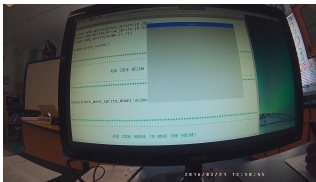
(f) Text region detection after removing figure.



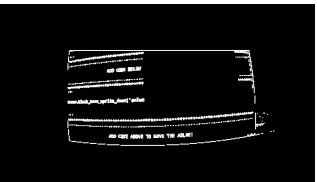
(g) Frame with popup.



(h) Top text not detected due to sudden pop up.



(i) Frame where new pop up window.



(j) Text not detected at top left due to sudden pop up.

Fig. 4. Editor text detection applied to different scenarios.