2009 International Conference on Machine Learning and Computing
IPCSIT vol.3 (2011) © (2011) IACSIT Press, Singapore

Integrated Coarse-Fine Scheme for Text Information Extraction from Instructional Videos

Ahmed Hamad[1], Said El-Ghonaimy[2], Taysir Soliman[3], Marwa Afifi[2+]

[1] Faculty of Informatics and Computer Science, British University in Egypt (BUE), Cairo, Egypt
[2] Faculty of Computer and Information Sciences, Ain Shams University, Cairo, Egypt
[3] Faculty of Computer and Information Sciences, Assiut University, Assiut, Egypt

Abstract. In this paper, we propose an effective coarse-to-fine algorithm to detect and extract text in instructional videos. Firstly, in the coarse section, an edge-based algorithm is employed to detect all candidate regions of character edges. First in the detection step, an edge map is created using the canny edge detector. Then, morphological filtering is used, based on geometrical structure element, in order to connect the vertical edges and discard false alarms. A connected component analysis is performed to the filtered edge map in order to determine a bounding box for every candidate text area. Finally in the localization step, horizontal and vertical projections are calculated on the edge map of every box and a threshold is applied, refining the result and splitting text areas in text lines. Secondly, in the fine section, correct text regions are selected from candidate ones by support vector machine (SVM) model and texture features. Finally, we segment these regions and binarize them to be fed into the OCR engine to be recognized. Experimental results show that our algorithm achieves high performance and prove that our system is highly effective and efficient for text information extraction.

Keywords: Text Detection, Text Extraction, Edge Detection, Morphological Operations, SVM, OCR

1. Introduction

E-learning system is a new emerging education approach that integrates the technologies of multimedia and network. Efficient indexing and retrieval of digital videos is an important function of video databases. videotext is very suitable for solving this problem because of; it is very useful for describing the semantic information of the multimedia content of videos, it is an efficient summary, it is very powerful for indexing and retrieval of videos, it can be easily extracted compared to other semantic contents, and it enables applications such as word-based video search, automatic video logging and text-based video indexing. There exist many challenges for text information extraction (TIE) process which are:

- **Low resolution:** Video frames are typically captured at resolutions of 320 × 240 or 640 × 480 pixels.
- **Unknown text color:** Text can have different and non-uniform color.
- **Unknown text size, position, orientation, layout:** Text lacks the structure usually associated with documents.
- **Color bleeding:** Video compression artifacts and poor visual quality cause colors to run together.
- **Unconstrained background:** The background can have colors similar to the text color. The background may include streaks that appear very similar to character strokes.
- **Low contrast:** Low bit-rate video compression causes loss of contrast between character strokes and the background.
- **Different text formats:** Text presented in different formats, such as handwriting on chalkboard, handwriting on paper, electronic slides, web pages, book pages, etc.

+ Corresponding author.
E-mail address: Marwa_afifi2002@yahoo.com

Existing text detection algorithm can be classified into four kinds: color and connected component based algorithm [1-2], edge and texture based algorithm [3-9], video temporal information based algorithm [10], and stroke based algorithm [11-12]. Edge and texture based algorithm is most popular and stroke based algorithm is the latest. Figure (1) illustrates the basic architecture of the text information extraction (TIE) system [13].

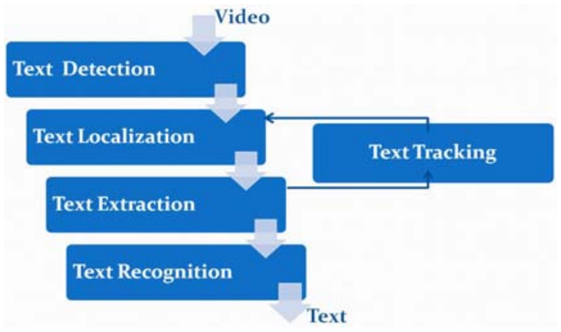


Fig. 1: The General Architecture of TIE System.

We reconstruct the scheme of this architecture and try to find more effective algorithm to detect and extract text from instructional video. Our algorithm flow chart is shown in Figure (2). We split our system into two main phases or stages. Since after the instructor has presented a visual text material or writing a block of text, there is usually a period of time spent on explaining this text material. So, this text material or block is highlighted by the camera operator and the camera is held static for a certain time. So our goal in the first phase is to format video summary consists of all the candidate text frames to minimize the number of frames processed in the second phase. We rely on the four fundamental bases for accessing the video contents shown in Figure (3) [14]. In the first phase we do motion analysis which is the most analytical feature of the instructional video to identify the candidate text frames and then detect the shot boundaries between these candidate text frames and finally select a representative key frame to form the summary video as an output of this stage. Our first phase is described in details in [15]. In the second phase we detect videotext in the summary video frames in a coarse-to-fine scheme. We detect and locate candidate text regions in the coarse section according to our edge-based algorithm. To better characterize text, texture features are extracted from candidate regions and then put into SVM model for region identification in the fine section. Finally, we segment these regions and binarize them to be fed into the optical character reader (OCR) engine.

In this paper we highlight only the details of our work in the second stage of our suggested system. The paper is organized as follows: Sections 2 describes the steps of the coarse section in the second phase, Sections 3 describes the steps of the fine section in the second phase, Section 4 shows the experiments and results, and Finally Section 5 provides the conclusion.

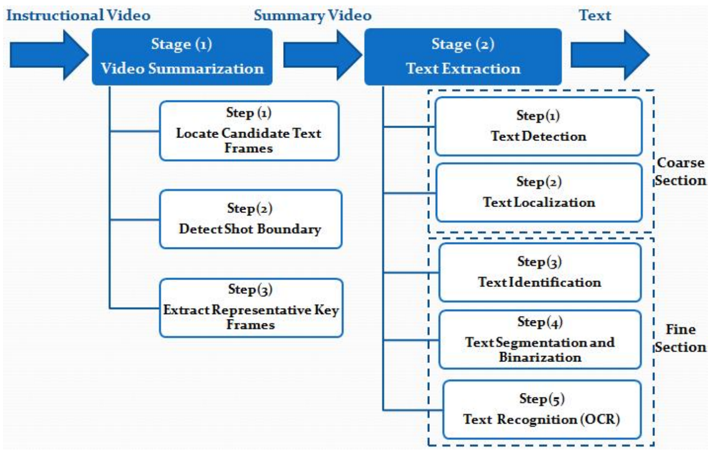

Instructional Video
Summary Video
Text
Stage (1) Video Summarization
Stage (2) Text Extraction
Step (1) Locate Candidate Text Frames
Step (2) Detect Shot Boundary
Step (3) Extract Representative Key Frames
Step(1) Text Detection
Step(2) Text Localization
Coarse Section
Step(3) Text Identification
Step(4) Text Segmentation and Binarization
Step(5) Text Recognition (OCR)
Fine Section

Fig. 2: The Architecture of Our Suggested System.

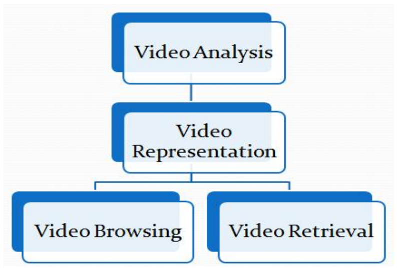
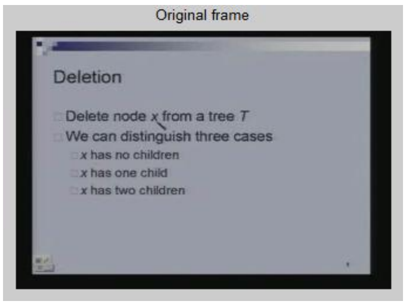


Fig. 3: The Four Fundamental Bases for Accessing the Video Content.

2. COARSE SECTION

2.1. Step (1): Text Detection

As a pre-processing step we apply median filtering on the candidate summary video frames. A median filter is more effective than convolution when the goal is to simultaneously reduce noise and preserve edges. According to the fact that text lines produce strong vertical edges horizontally aligned, using edges as the distinct feature of our system gives us the opportunity to detect characters with different fonts and colors. An example of text in a summary video frame is shown in Figure (4). Firstly, we produce the edge map of the summary video frame image after applying the median filter. Several methodologies are used computing the edge map of an image [16]. For our algorithm we use Canny [17] edge detector applied in the greyscale of frames images. This method is less likely than the others to be fooled by noise, and more likely to detect true weak edges. Ideally the created edge map is a binarized image as shown in Figure (5a). Secondly, a morphological dilation operation is performed to connect the character contours of every text region by a crossed-shaped structure element then we remove the noise by suppressing the light structures connected to frame image border and applying morphological opening operation to remove small objects. Figure (5b) shows the edge density map produced at the end of this step. Finally we do analysis on these detected connected components to define the initial bounding boxes of the candidate text regions as shown in Figure (5c).

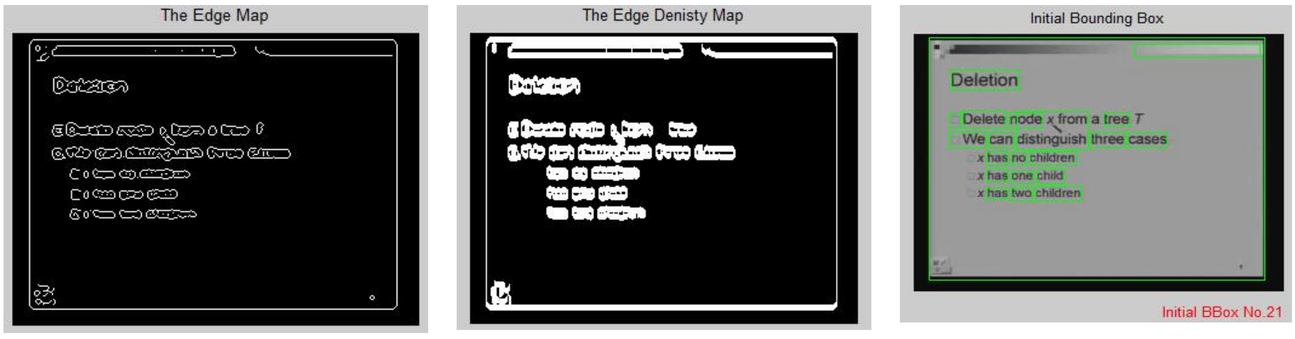


Original frame

Deletion

Delete node *x* from a tree *T*
We can distinguish three cases
x has no children
x has one child
x has two children

Fig. 4: The Candidate Text Summary Video Frame.



(a) (b) (c)

Fig. 5: Text Detection in a candidate Text Frame

(a) The Edge Map, (b) The Edge Density Map, and (c) The initial Bounding boxes.

2.2. Step (2): Text Localization

We achieve a high detection rate in the previous step but relatively low precision. This means that most of the text lines are included in the initial text boxes while at the same time some text boxes may include more than one text line as well as noise or non-text regions. This noise usually comes from non-text objects that connect to the text lines during the dilation process. And the low precision comes from detected bounding boxes which do not contain text but objects with high vertical edge density. To increase the precision and reject the false alarms we use a method based on horizontal and vertical projections. Firstly, the horizontal edge projection of every box is computed. A horizontal projection is defined as the sums of the candidate pixels over rows. Figure (6) shows an example of horizontal projection of a candidate text box. To divide box with more than one text line, we need to find the 'valley' points where the projection value is smaller than a threshold THP and then segment the text lines at these valley points. Secondly, a similar procedure with vertical projection is applied to each candidate text line extracted from the previous step. In this way, a bounding box will split only if the distance between two words is larger than the threshold TVP. Figure (7) shows an example of the vertical projection procedure. This method would actually break every text line in words. However, the parts of the text line remains connected if the distance between them is less than a threshold TVP which depends on the height of the bounding box of the candidate text line. In this way, a bounding box will split only if the distance between two words is larger than the threshold TVP.

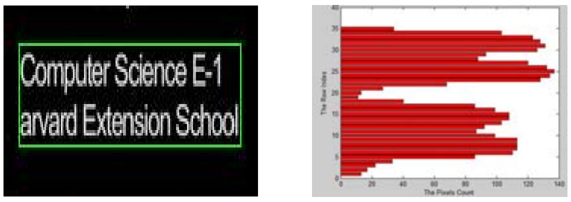


Fig. 6: The Horizontal projection of a Candidate Text Region

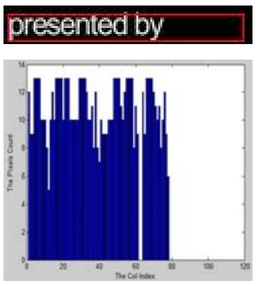
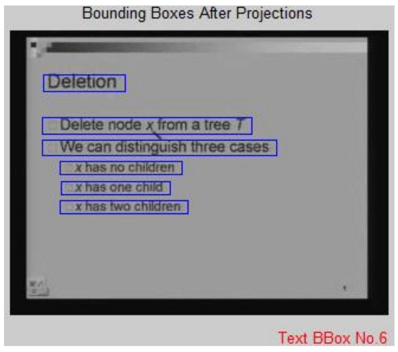


Fig. 7: The Vertical projection of a Candidate Text line

Finally, after projections analysis, boxes with noise or non-text boxes that are splitted into a number of boxes with very small height will be discarded according to these geometrical constraints. This step eliminates also short and non-important words to be used as an index such as; a, an, at, on, etc. A box is discarded if the height of the bounding box is less than 8 pixels and the ratio between the width and the height is less than 1.2. Figure (8) shows the final output of the localization step. In this step we successes to filter these bounding boxes and discard high rate of non-text bounding boxes.



Bounding Boxes After Projections

Deletion

Delete node x from a tree T

We can distinguish three cases

x has no children

x has one child

x has two children

Text BBox No.6

Fig. 8: The Final Bounding Boxes after Projections

3. FINE SECTION

3.1. Step (3): Text Identification

To correctly fine and identify the text lines from the candidate ones, we extract texture features and use SVM classifier. Support Vector Machine (SVM) is a technique motivated by statistical learning theory. The key idea is to transform the input data into high dimensional feature space and separate classes with a decision surface in this space. Extensive discussions of SVMs can be found in [18-20]. SVM is easier to train, needs fewer training samples and has better generalization ability. It is very effective for text identification [9]. Considering the limited number of training sample, we train the SVM model on a dataset consisting of 1000 text and 2000 non-text labeled samples. Since only one kind of texture feature is insufficient to model the texture pattern of a text line. We join four kinds of features to represent each candidate text line; three of them are extracted in the grayscale domain and one in the binary edge domain. Before extracting the features we first normalize the size of the candidate text lines, since they are having different resolutions, to rectangles with 64 pixels in height and a variable width, by using linear interpolation. Then extract the following features which are:

- **Edge Moment Features:**

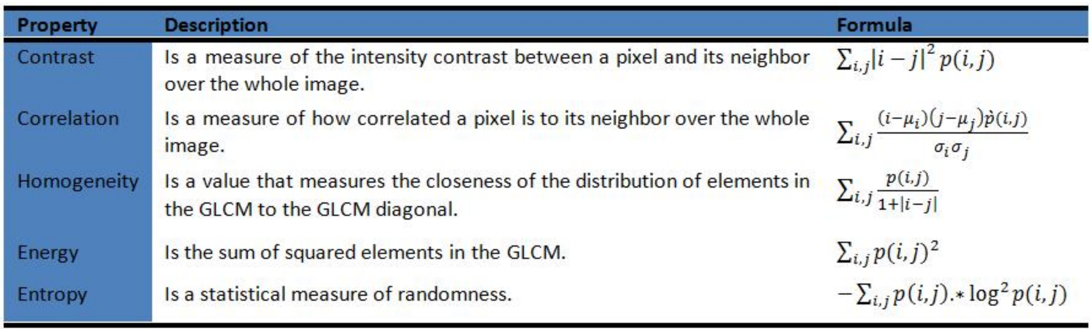
Compared with background, text region has different edge density moment distribution. We calculate the mean, second-order or the variance, and the third-order central moments from the edge map. The central moment of order k of a distribution is defined as:

$$m_k = (E - \mu)^k$$

Where, E(x) is the expected value of x.

- **Gray Level Co-Occurrence Features:**

The gray-level co-occurrence matrix (GLCM) is a statistical method of examining texture that considers the spatial relationship of pixels. The texture filter functions using standard statistical measures cannot provide information about shape, however the GLCM do. We create the p(i,j) in the GLCM by calculating how often a pixel with the intensity value i occurs in a specific spatial relationship to a pixel with the value j on a distance d in the direction ɵ. Then we extract the features of energy, contrast variance or inertia, homogeneity and correlation from the co-occurrence matrix which are shown in Figure (9).



Property	Description	Formula
Contrast	Is a measure of the intensity contrast between a pixel and its neighbor over the whole image.	$\sum_{i,j} \lvert i-j \rvert^2 p(i,j)$
Correlation	Is a measure of how correlated a pixel is to its neighbor over the whole image.	$\sum_{i,j} \frac{(i-\mu_i)(j-\mu_j)\dot{p}(i,j)}{\sigma_i \sigma_j}$
Homogeneity	Is a value that measures the closeness of the distribution of elements in the GLCM to the GLCM diagonal.	$\sum_{i,j} \frac{p(i,j)}{1+\lvert i-j \rvert}$
Energy	Is the sum of squared elements in the GLCM.	$\sum_{i,j} p(i,j)^2$
Entropy	Is a statistical measure of randomness.	$-\sum_{i,j} p(i,j).* \log^2 p(i,j)$

Fig. 9: The Statistics of GLCM.

Where μ_i, μ_j and σ_i, σ_j are the means and the variances of the co- occurrence matrix on a distance d in the direction ɵ. We set the direction ɵ of co-occurrence as 0, 45, 90 and 135 degree and set the distance d as 1, 3 and 5 pixels. So we calculate the five features in the 12 co-occurrence matrixes and obtain 60 co-occurrence features.

- **Gray Level Entropy Feature:**

Entropy is a statistical measure of randomness that can be used to characterize the texture of the input image. Entropy is defined as

$$-\sum p.* \log^2 (p)$$

Where, p contains the histogram counts of 256 bins for the intensity image of each candidate text line bounding box.

- **Gray Level Histogram Feature:**

Since the histogram is a kind of effective feature to represent the distribution of the intensity values, we calculate 8-bins histogram for the intensity image of each candidate text region.

All the texture features mentioned above make up 72 dimensions vector for each candidate text region. This vector is fed into SVM classifier to classify it into two classes: text region and non-text region. Figure (10) illustrates an example of positive text regions that is recognized and successfully classified as text regions by our trained SVM classifier model. Figure (11) illustrates an example of false text regions that is rejected and successfully classified as non-text regions by the trained SVM classifier model.

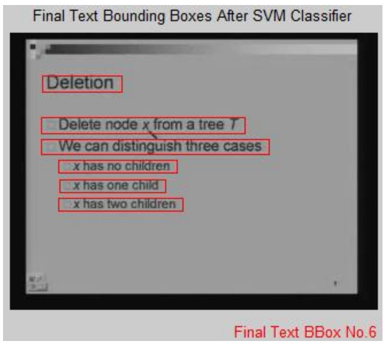
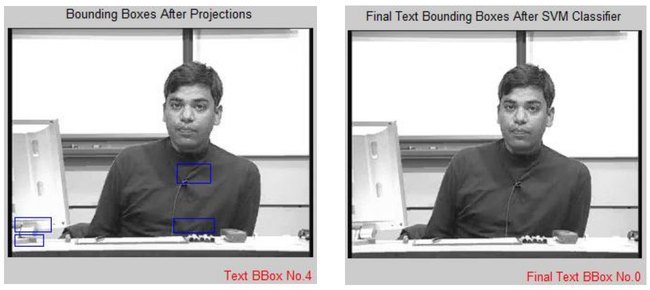

Final Text Bounding Boxes After SVM Classifier
Deletion
Delete node x from a tree T
We can distinguish three cases
x has no children
x has one child
x has two children
Final Text BBox No.6

Fig. 10: The Final Bounding Boxes after Classification by SVM.


Bounding Boxes After Projections
Text BBox No.4
Final Text Bounding Boxes After SVM Classifier
Final Text BBox No.0

(a) (b)

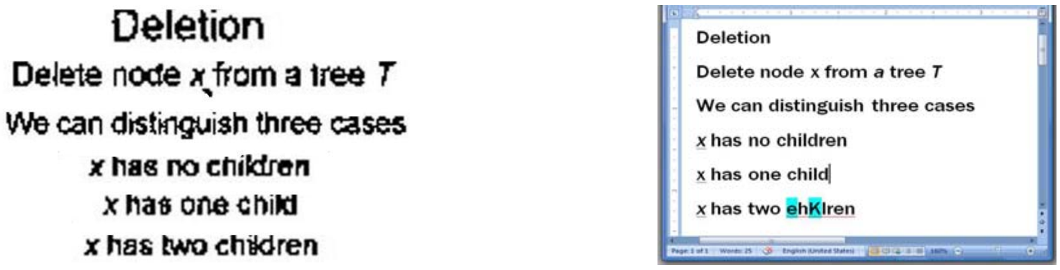
Fig. 11: (a) Candidate Bounding Boxes after Projections in the Localization Step
(b) The Final Bounding Boxes after Classification by the SVM Model

3.2. Step (4): Text Segmentation and Binarization

We crop the original image to split each final text bounding box into separate text image, and then segment the text in each separate text image from the background by locally converting it to binary using the Otsu's method [21] to facilitate its recognition by the OCR software engine. Then we calculate the intensity value inside and outside the bounding box and compare the two values to decide about inverse the binary text image or not. Figure (12a) illustrates an example of the result.

3.3. Step (5): Text Recognition

We use the optical character recognition system of ABBYY FineReader Professional Edition Software to accomplish this final step in our system, convert each binary text image into a plain text, and pull out this textual information producing keywords for video indexing. Example of our results is shown in Figure (12b).



Deletion
Delete node x from a tree T
We can distinguish three cases
x has no children
x has one child
x has two children

Deletion
Delete node x from a tree T
We can distinguish three cases
x has no children
x has one child
x has two ehKlren

(a) (b)

Fig. 12: (a) The Result of Segmentation and then Binarization.
(b) The Plain Text Produced by the OCR Software.

4. EXPERIMENTS AND RESULTS

The following results are obtained in one pass of the summary videos processing. These summary video frames have a size of 240 × 320 pixels and are obtained from the first stage of our proposed system to minimize the number of frames processed in the second stage. Designing evaluation methods for text detection is an aspect that has not be studied extensively. Very few related works have been published; moreover these works propose evaluation strategies with very complicated implementations or demand great effort for the generation of the ground truth [22, 23]. Moreover that, there is not an optimal way to define and draw the ground truth bounding boxes. Many of the researchers use their own evaluation tool to test the success of their algorithm. This fact leads to the inability to compare the performance of the different algorithms which indubitably consists a barrier to the evolution of the area. We apply cross validation on the training set to evaluate the performance of our SVM classifier model which achieves accuracy rate of 98.8%.

Since the goal of our system is to detect the maximal number of characters. To evaluate the performance of our system according to the results achieved in both the final detection after applying the SVM classifier and the recognition done by the ABBYY FineReader OCR Software with respect to the character level, we use the precision rate and the recall rate, respectively.

The precision rate; used to evaluate the final results obtained after detecting, locating, and then identifying the text regions; is defined as:

Precision = CD/TC

Where CD is the number of characters correctly detected, and TC is the total number of characters in the text frame. Our system achieved high precision rate 98.7%. Actually since that all of the characters detected are positive text characters, so there is no mean to use the recall as an evaluation parameter for this step.

The recall rate; used to evaluate the results obtained after recognizing the text regions; is defined as:

Recall = CCD/CD

Where CCD is the number of characters correctly recognized and converted to editable characters by the OCR engine, and CD is the total number of characters detected to be recognized. Our system achieved high recall rate 97.6%.

5. CONCLUSION

In this paper, we highlight our work in the second phase of our suggested system to enhance the solution of the challenging problem of text information extraction from instructional videos. In this paper, we propose our coarse-to-fine integrated scheme, which is efficient and effective. We detect the characters edge by applying canny operator to obtain the edge map. Then a morphological dilation operation and opening are applied to form the edge density map and connect the edges of characters and discard false alarms. A connected component analysis is performed to the filtered edge map in order to determine a bounding box for every candidate text area. Finally in the localization step, horizontal and vertical projections are calculated on the edge map of every box and a threshold is applied, refining the result and splitting text areas in text lines. Secondly, in the fine section, correct text regions are selected from candidate ones by support vector machine (SVM) model and texture features. Finally, we segment these regions and binarize them to be fed into the OCR engine to be recognized. Experimental results show that our algorithm achieves high performance and introduce contribution according to the effective and efficient results evaluated by the precision rate and the recall rate. In our future work, we will focus on improving the Binarization step to obtain more accurate and enhanced results from the recognition step.

6. References (This is "Header 1" style)

[1] V. Wu, R. Manmatha, and E.M.Riseman, "Finding text in images," in *Proceedings of the second ACM international conference on Digital libraries*, pp. 23-26, 1997.

[2] K. Jain and B. Yu, "Automatic text location in images and video frames," *Pattern Recognition*, vol. 31, pp. 2055–2076, 1998.

[3] T. Sato, T. Kanade, E. K. Hughes, and M. A. Smith, "Video OCR for digital news archive," in *Proceeding of IEEE Workshop on Content-Based Access of Image and Video Database*, pp. 52–60, 1998.

[4] V. Wu, R. Manmatha, and E. M. Riseman, “Textfinder: An automatic system to detect and recognize text in images,” *IEEE Transactions on Pattern Analysis and Matching Intelligence*, vol. 21, no. 11, pp. 1224–1229, Nov. 1999.

[5] R. Lienhart and A. Wernicke, “Localizing and Segmenting Text in Images and Videos,” *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 12, No. 4, April 2002.

[6] M.R. Lyu, J. Song, and M. Cai, “A Comprehensive Method for Multilingual Video Text Detection, Localization, and Extraction,” *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 15, pp. 243- 255, Feb. 2005.

[7] H. Li, D. Doermann, and O. Kia, “Automatic text detection and tracking in digital video,” *IEEE Transactions on Image Processing*, vol. 9, no. 1, pp. 147–156, Jan. 2000.

[8] K.I. Kim, K. Jung, H. Kim, “Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm,” *IEEE Transactions on Pattern Analysis and Matching Intelligence*, vol. 25, pp.1631– 1639, December 2003.

[9] Q. Ye, Q. Huang, W. Gao, and D. Zhao, “Fast and robust text detection in images and video frames,” *Image and Vision Computing*, Vol.23, pp565-576, March 2005.

[10] X. Tang, X.B. Gao, J. Liu, and H. Zhang, “A spatial-temporal approach for video caption detection and recognition”, *IEEE Transactions on Neural Networks*, Vol. 13, pp. 961–971, July 2002.

[11] Q. Liu, Ch. Jung, and Y. Moon, “Text Segmentation based on Stroke Filter,” in *Proceedings of ACM international conference on Multimedia*, pp. 129-132, 2006.

[12] Q. Liu, Ch. Jung, S. Kim, Y. Moon, and J. Kim, “Stroke Filter for Text Localization in Video Images,” *IEEE International Conference on Image Processing*, pp. 1473-1476, 2006.

[13] K. Jung, K.I. Kim, and A.K. Jain, “Text information extraction in images and video: A survey,” *Pattern Recognition*, Vol. 37, No. 5, 2004, pp. 977-997.

[14] Handbook of Image and Video Processing, by Al Bovik (Author), Alan C. Bovik (Editor), Jerry D. Gibson (Editor), Publisher: Academic Press; 1 edition (May 2000).

[15] A. Hamad, T. Hassan, and M.A. Afifi, “Automatic Summarization of Visual Text Contents in Instructional Videos,” in *Proceeding of the Fourth international Conference on Intelligent Computing and Information systems* (ICICIS’09), 2009, 19-22 March, 2009, Page(s): 480-483. http://icicis.edu.eg/

[16] R. Gonzalez and R. Woods Digital Image Processing (Addison Wesley, 1992), pp 414 – 428.

[17] Canny, John, “A Computational Approach to Edge Detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, No. 6, 1986, pp. 679-698.

[18] C. J. C. Burges. “A tutorial on support vector machines for pattern recognition,” *Data Mining and Knowledge Discovery*, 2(2):1–47, 1998.

[19] R. Collobert, S. Bengio, and Y. Bengio. “A parallel mixture of svms for very large scale problems,” *Neural Computation*, 14(5), 2002.

[20] V. Vapnik, Statistical Learning Theory. John Wiley & Sons, 1998.

[21] Otsu, N., “A Threshold Selection Method from Gray-Level Histograms,” *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 9, No. 1, 1979, pp. 62-66.

[22] Xian-Sheng Hua, Liu Wenyin, and HongJiang Zhang, “An automatic performance evaluation protocol for video text detection algorithms,” *IEEE Transactions on Circuits and Systems for Video Technology* 14(4), 498-507, 2004.

[23] Vasant Manohar, Padmanabhan Soundararajan, Matthew Boonstra, Harish Raju, Dmitry B. Goldgof, Rangachar Kasturi, John S. Garofolo, “Performance Evaluation of Text Detection and Tracking in Video,” *Document Analysis Systems* 2006, 576-587.