Text Identification in Complex Background Using SVM

Datong Chen*, Hervé Bourlard*, Jean-Philippe Thiran**

* Dalle Molle Institute for Perceptual Artificial Intelligence, Switzerland
** Signal Processing Laboratory, EPFL, Switzerland
{chen, bourlard}@idiap.ch, JP.Thiran@epfl.ch

Abstract

This paper presents a fast and robust algorithm to identify text in image or video frames with complex backgrounds and compression effects. The algorithm first extracts the candidate text line on the basis of edge analysis, baseline location and heuristic constraints. Support Vector Machine (SVM) is then used to identify text line from the candidates in edge-based distance map feature space. Experiments based on large amount of images and video frames from different sources showed the advantages of this algorithm compared to conventional methods in both identification quality and computation time.

Keywords: text identification, support vector machine, image and video OCR

1 Introduction

Text embedded in image and video usually provides brief and important information about the content, such as name of a player or speaker, title, location and date of an event, category of the product etc. This kind of embedded text, referred to as closed caption, is a powerful knowledge source in building image and video indexing and retrieval system. The extraction of closed captions has therefore gained research importance recently.

Due to the huge amount of data carried by images and video, it is of very practical importance to detect and identify the text region as accurately as possible before performing any character recognition. However, text detection and identification is a difficult task because background, color, size of text strings may vary, even in a same image. Many papers [5][14] show that available binarization methods, including global and adaptive thresholding (which has been well used in identifying characters printed on clean papers) do not work well for typical image and video frame. Furthermore, the image digitalization and compression also introduce noise that may blur the embedded text characters.

Previous work on text identification in image or video can be briefly classified into region-based, texture-based and edge-based methods.

Region-based methods detect characters as the monochrome regions satisfying certain heuristic constraints. The pixels of each character are assumed to have similar color and can be segmented from background by image segmentation [4][8][9][10][11] or color clustering [17] preprocess. The resulting monochrome regions are selected as characters under some simple heuristic constraints, such as the size, the height/width ratio of the region or baselines. Region-based methods not only identify the embedded text regions but also segment characters from background. However, the monochrome constraint can not always be satisfied and, therefore, the methods are not robust to complex background and compressed video.

Texture-based methods make use of texture features to decide whether a pixel or block of pixels belongs to text or not. Wu et al. [5][6][7] proposed an algorithm based on K-means to identify text pixels on the basis of nine second order derivatives of Gaussians at three scales. Li et al. [15] used a neural network to extract text blocks in Haar wavelet decomposition feature space. Zhong and Jain [8] [16] presented a text region identification approach to combine spatial variance (texture feature) and connected component (regions) analysis together. Texture-based method is able to detect text in complex background but is very time consuming [15] and cannot always perform accurate localization [8].

Edge-based method detects the text by finding vertical edges. In [12] and [18], vertical edges are first detected and connected into text clusters by using a smoothing filter. As with region and texture-based methods, the text clusters are then selected by using heuristic constraints. Edge-based method performs fast text detection but also results in many false detections.

In the present paper, we introduce a fast and robust algorithm of text identification in image and video frame. As illustrated in Figure 1, the algorithm consists of two steps: text line extraction and SVM-based text identification. In the first step, candidate text regions are quickly extracted by using edge analysis, and further segmented into text lines on the basis of baseline location and heuristic constraints at the aim of producing high text location rate and reasonable false alarms. In the

0-7695-1272-0/01 $10.00 © 2001 IEEE

second step, the candidate text lines are identified by using a support vector machine (SVM) trained in a distance map feature space to lower the false alarm rate. The detail of the two steps are described in Section 2 and 3 respectively. Experiments and results are shown in Section 4. The discussion and conclusion are in the last Section.

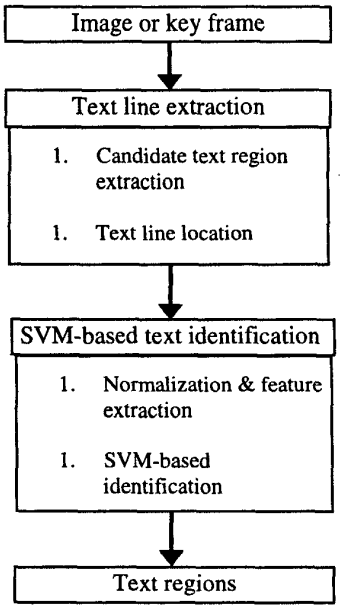


Figure 1 Algorithm of text identification

2 Text Line Extraction

In this section, we present an algorithm to quickly extract text lines in images and key frames by exploiting two characteristics of closed captions. First, a visible character always forms some edges against its background. Second, a text string has a special kind of texture pattern, a rectangle shape and horizontal alignment.

2.1 Candidate text region extraction

In our algorithm, the texture pattern of text string is simply regarded as a group of short vertical and horizontal edges mixed together. Although the compression process may blur parts of characters in image, the visible part of the text still has quite a different intensity compared to its neighbor's background. Varying-orientation edges can therefore always be detected in a text embedded region. These text edges are generally short and connected with each other in different orientations.

To detect this kind of short edge mixture pattern, we first detect the vertical and horizontal edges individually using "Canny" edge detector. Morphological dilation is then employed to connect edges into clusters. According to the type of edge (vertical or horizontal), different dilation operators are used so that the vertical edges are connected in horizontal direction while horizontal edges are connected in vertical direction. The dilation operators are designed to have rectangle shapes; in our case, 5×1 for the vertical operator and 3×6 for the horizontal operator.

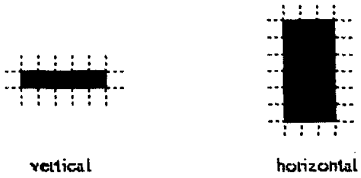


Figure 2 Vertical and horizontal edge dilation operators

Since non-text areas usually formed by isolated or long vertical and horizontal edges do not occupy the same places in both vertical and horizontal dilated edge images at the same time, they can be removed by using an "AND" operation to the vertical and horizontal dilated edge images. Figure 3 illustrates the clusters resulting from this detection process.

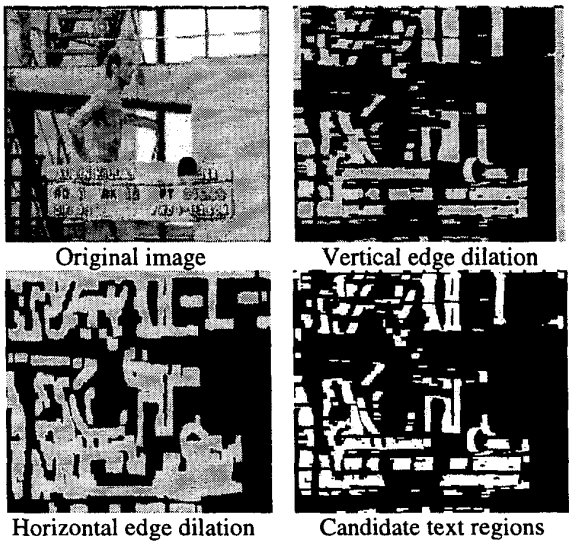


Figure 3 Candidate text region extraction

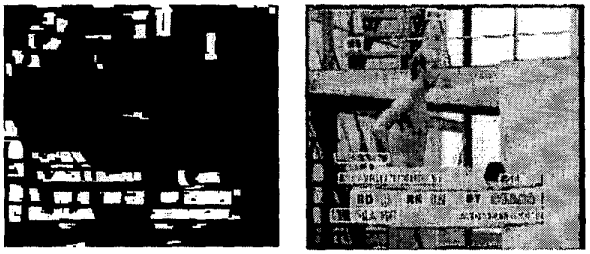
2.2 Text line location

The connected pixels in resulting clusters are grouped into candidate text regions. If a candidate region contains text lines, the top and bottom baselines of these text lines can be located.

Baseline location is based on the horizontal projection of the target region. Here, we simply regard a candidate text region as binary image and project it onto the Y-axis. Baselines can be located at the lowest values or the peak of the first order derivative of this Y-axis projection. In case that a candidate region may contains more than one text line, the candidate region is first segmented by the resulting baselines into several smaller regions. Baseline locating process is then performed iteratively on these new born regions.

Very small regions and non-baseline regions are removed in this process and the updated regions bounded with its baselines are illustrated in Figure 4 (left).

The typical heuristic character of a text string is then employed to select the text lines. In our experiments the text line should satisfy the following constraints: (1) the size of region is between 75 to 9000; (2) the horizontal-vertical aspect ration is more than 1.2; (3) the height of the region is between 8 to 35. In general, the size of the text can vary greatly (more than 35 pixels high). Large characters can be detected by using the same algorithm on scaled image pyramid. Figure 4 (right) shows the extracted text lines.



Baseline location Extracted text lines

Figure 4 Text line location

3 SVM-Based Text Identification

3.1 Support vector machine

SVM is a technique motivated by statistical learning theory and has been successful applied to numerous classification tasks. The key idea is to separate two classes with a decision surface that has maximum margin. The extensive discussion of SVM can be found in [21]. In the present paper, we will only consider a binary classification task with m labeled training examples: $(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)$, where $y_i = \pm 1$ indicating two different classes $i = 1,2,....,m$.

For the linear separable case, we have a hyperplane $w \cdot x + b = 0$ (decision surface) that separates all the training examples:

$$y_i(x_i \cdot w + b) \geq 1 \quad \forall i \quad (1)$$

where w is normal to the hyperplane.

The margin of such a hyperplane: $w \cdot x + b = 0$ is defined by the sum of the shortest distance from hyperplane to the closest positive example and the shortest distance from hyperplane to the closed negative example. Since this margin is simply $2/\|w\|$, where $\|w\|$ is the Euclidean norm of w, the maximum margin can be given by minimizing $\|w\|^2$ subjecting to the constraints Eq. (1).

For the nonlinear non-separable case, the learning task involves the minimization of $\|w\|^2 + C\sum_{i=1}^{m}\xi_i$, subject to the constraints:

$$y_i(x_i \cdot w + b) \geq 1 - \xi_i \quad \forall i$$
$$\xi_i \geq 0 \quad \forall i$$

where C is the penalty to errors and ξ_i are positive slack variables that measure the amount of constraint violations. The training examples $x_i \cdot x_j$ are mapped into an alternative space $\phi(x_i) \cdot \phi(x_j)$, so called feature space, by choosing kernel $K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$. The learning task therefore equals to the maximization of the Lagrangian:

$$W(\alpha) = \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{m}\alpha_i\alpha_j y_i y_j K(x_i \cdot x_j)$$

subject to constraints:

$$\alpha_i \geq 0,$$

$$\sum_{i=1}^{m}\alpha_i y_i = 0.$$

$W(\alpha)$ can be solved using quadratic programming techniques. Once we have found the optimal α_j, the classification of an unknown example z is based on the sign of function:

$$G(z) = \sum_{j=1}^{m}\alpha_j y_j K(z, x_j) + b.$$

SVM can be regarded as an alternative training technique for Radial Basis Function, Multi-Layer Perceptron and Polynomial classifiers. One of

advantages of SVMs is that the learning task is insensitive to the relative numbers of training examples in positive and negative classes. For example, in our case, the candidate text lines usually involve 15.4% false alarms (in terms of regions). The number of positive examples thereby is roughly 6 times as the negative examples. Most learning algorithm based on Empirical Risk Minimization will tend to classify only the positive class correctly to minimize the error over data set. Since SVM aims at minimizing a bound on the generalization error of a model in high dimensional space, so called Structural Risk Minimization, rather than minimizing the error over data set, the training examples that are far behind the hyperplane will not change the support vectors. Therefore, SVM is used to identify text regions in the candidate text lines for achieving a lower false alarm rate.

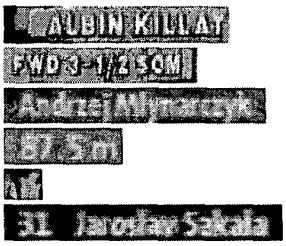


Figure 5 Normalized text lines

3.2 Normalization and feature extraction

We first normalize the candidate text lines, which may have varying resolutions, to rectangles with 16 pixels in height by using bilinear interpolation (8 pixels between the baselines, 8 pixels for top and bottom boundary). Some examples are shown in Figure 5.

The feature space we used has 256 dimensions corresponding to a 16×16 slide window in the normalized text region. Since the text may have varying gray-scales in images, the gray value is not a robust feature in this case. As explained below, we therefore use the distance map of each slide window as input feature of SVM.

The distance map [19] *DM(z)* of window z is defined as the set of all the associated distance values $v(x,y)$ in the window z with respect to a distance function *dis* according to

$$\forall (x,y) \in z : v(x,y) \stackrel{def}{=} \min_{(x_i,y_i)\in B} dis[(x,y),(x_i,y_i)].$$

Here, B is a set of strong edge points extracted in window z. The distance function used here is Euclidean. Figure 6 illustrates an example of distance map.

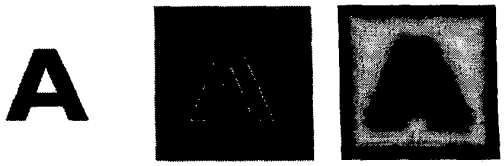


Figure 6 Illustration of original image (left), strong edges in the original image (middle), and its distance map (right)

3.3 SVM training and identification

The SVM was trained on a database consisting of 6000 samples labeled as text or non-text, using the software package developed at IDIAP and called SVMTorch [22]. We used radial basis function kernel:

$$K(X, X_j) = \exp\left\{-\|x-x_j\|^2 / 2\sigma^2\right\}$$

where the kernel bandwidth σ was determined through cross-validation. The details of the requirement of the kernel and construction of the hyper-plane via a dual optimization process can be found in [21].

The output of the SVM $G(z)$ estimates the confidence of the block of pixels in the 16×16 window z to be text. In the identification process, we slide the window every four pixels from left to right in each normalized text region and compute the confidence of each window. The confidence $Conf(R)$of a text region R was defined as:

$$Conf(R) = \sum_{z\subseteq R} G(z) \cdot \frac{1}{\sqrt{2\pi}\sigma_0} \exp\left(\frac{d_z^2}{2\sigma_0^2}\right)$$

where d_z is the distance between the center of window z and the center of the text region R, and $\sigma_0 = 10$. A candidate region R is identified as a text region if $Conf(R) \geq 0$.

Figure 7 Valid baseline ranges: the shading parts indicate the valid baseline range

4 Experiments and Results

Experiments were carried out on a database consisting of 18,000 video frames extracted from video of advertisements, sports, interviews, news, movies and 50 compressed images including the covers of journals, maps, and flyers. Each video frame or image has

352x288 resolution in JPEG or MPEG format and was decompressed and converted into grayscale before applying text identification. Some video frames contain the same closed captions but with different backgrounds.

Performance of the text identification was measured in term of identification rate (IR) and false alarms. A text string is considered to be correctly identified if and only if the located baselines are in the valid ranges, which is labeled by human visual inspection as shown in Figure 7. The false alarms are reported in terms of false region alarms and false pixel alarms. The false region alarm rate (FRR) is measured by the percentage of the number of false alarm regions in all the identified regions. The false pixel alarm rate (FPR) is defined as the area inside of false alarm region as a percentage of the whole area of the image.

Table 1 summarizes the performance of the proposed algorithm, where text line indicates the number of text strings containing at least two characters. We list both the performance for both text line extraction and identification. The fast text extraction algorithm correctly extracts about 99.3% text lines, but at the cost of high FRR and FPR. After applying the SVM identification, the FRR drops to 1.7% and FPR to 0.38%, while preserving a high IR as 98.7%.

Table 1 Identification performance

23037 text lines	IR	FRR	FPR
without SVM	99.3%	15.4%	2.13%
With SVM	98.7%	1.7%	0.38%

Table 2 compares the performance and running time cost of the proposed algorithm with typical region-based [11], texture-based [5] and edge-based [18] methods, which is re-implemented by ourselves. It can be observed that the proposed algorithm results in a good tradeoff between high identification rate and low false alarm rates. Computation costs in Table 2 are reported for Sun UltraSPARC-II with 333 MHz without counting the I/O consumption. CPU required by this algorithm is higher than region and edge based method but much lower than texture-based method. Figure 8 shows some text identification results, including correct identifications and false alarms.

Table 2 Performances and running costs

X-based method	IR	FRR	FPR	Sec. per Image
Region	89.6	59.2%	5.8%	1.15
Texture	99.1%	11.5	3.3%	11.27
Edge	92.6%	24.1%	12.3%	0.52
Proposed	98.7%	1.7%	0.38%	2.76

5 Discussion and Conclusion

Text identification in image and video with complex backgrounds and compression effects is a difficult and challenging problem. In this paper, we have presented a fast text identification algorithm based on support vector machine. The algorithm first integrates the edge, and heuristic evidences to extract the candidate text lines and then identifies these candidates by using SVM.

The algorithm described in this paper does not use color, although many systems also make use of color information in detecting text in color images [9][17]. The main reason is that the start point of our system, the edge evidence, is mostly coming from intensity in compressed image. Transforming the RGB color image to YUV color space and performing edge detection in U or V image can easily find out this fact. No temporal information is used in our algorithm. Since text may have different movements in video, text identification is usually performed before tracking the text among the video frames.

The algorithm presented here achieves high identification rate, as well as low false alarm rates. In fast text line extraction phrase, this algorithm is faster than (or equivalent to) other fast text identification methods, although the whole identification process is more CPU intensive than region-based and edge-based methods. The evaluation criterion of the identification result presented in this paper is on the basis of correct baseline localization. This criterion is stricter than complete cover criterion used in [6]. With this criterion, we can measure the identification performance precisely without having to show the final character recognition result.

Acknowledgements

The authors thank Dr. Samy Bengio and Dr. Jean-Marc Odobez for their comments on this work.

References

[1] M. Bokser, "Omnidocument technologies", Proc. IEEE, 80(7):1066--1078, July 1992.

[2] S. V. Rice, F. R. Jenkins, and T. A. Nartker. "OCR accuracy: UNLV's fifth annual test", INFORM, 10(8), September 1996.

[3] L. O'Gorman and R. Kasturi, "Document Image Analysis", IEEE Computer Society Press, Los Alamitos, 1995.

[4] J. Ohya, A. Shio, and S. Aksmatsu, "Recognition characters in scene images. IEEE Trans. Pattern Analysis and Machine Intelligence", 16(2):214--220, 1994.

[5] V. Wu, R. Manmatha, and E. M. Riseman, "Finding text in images", In Proc. ACM Int. Conf. Digital Libraries, 1997.

[6] 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 Machine Intelligence, 20(11):1224--1229, 1999.

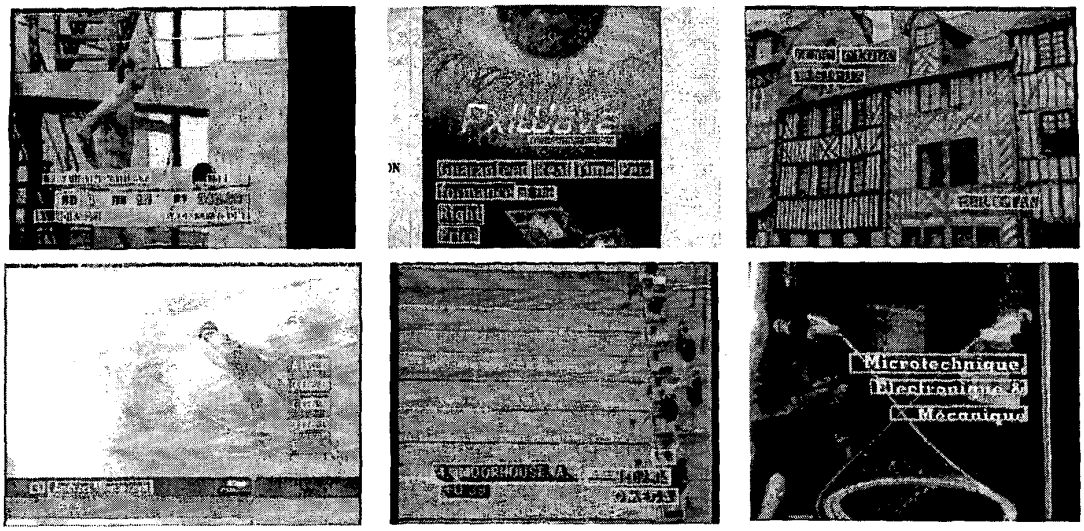


Figure 8 Identified text lines and false alarms in images or video frames

[7] V. Wu and R. Manmatha, "Document image clean-up and binarization", In Proc. SPIE Symposium on Electronic Imaging, 1998.

[8] Y. Zhong, K. Karu, and A. K. Jain, "Locating text in complex color images", Pattern Recognition, 28(10):1523--1536, 1995.

[9] R. Lienhart and W. Effelsberg, "Automatic text segmentation and text recognition for video indexing", Technical Report TR-98-009, University of Mannheim, Mannheim, 1998.

[10] R. Lienhart, "Automatic text recognition in digital videos", In Proc. SPIE, Image and Video Processing IV, January 1996.

[11] R. Lienhart, "Indexing and retrieval of digital video sequences based on automatic text recognition", In Proc. 4th ACM International Multimedia Conference, Boston, November 1996.

[12] T. Sato, T. Kanade, E. K. Hughes, and M. A. Smith, "Video ocr for digital news archives", In IEEE Workshop on Content Based Access of Image and Video Databases, Bombay, January 1998.

[13] T. Sato, T. Kanade, E. K. Hughes, M. A. Smith, and S. Satoh, "Video OCR: indexing digital news libraries by recognition of superimposed caption", In ACM Multimedia System Special Issue on Video Libraries, Feb. 1998.

[14] H. Li and D. Doermann, "Text enhancement in digital video using multiple frame integration", ACM Multimedia 1999.

[15] H. Li, D. Doermann, and O. Kia, "Automatic text detection and tracking in digital video", Maryland Univ. LAMP Tech. Report 028, 1998.

[16] K. Jain and B. Yu, "Automatic text localisation in images and video frames", Pattern Recognition, 31(12):2055--2076, 1998.

[17] K. Sobottka, H. Bunke, H. Kronenberg, "Identification of text on colored book and journal covers", ICDAR, pp: 57-63, 1999.

[18] M. A. Smith and T. Kanade, "Video skimming for quick browsing based on audio and image characterization", Carnegie Mellon University, Technical Report CMU-CS-95-186, July 1995.

[19] J. Toriwaki and S. Yokoi, "Distance transformations and skeletons of digitized pictures with applications", in L. N. Kanal and A. Rosenfeld, editors, Progress in pattern recognition, North-Holland, Amsterdam, 1981.

[20] C. Burgess, "A tutorial on support vector machines for pattern recognition", Data mining and knowledge discovery, 1998.

[21] V. Vapnik, "Statistical learning theory", John Wiley & Sons, 1998.

[22] R. Collobert, and S. Bengio, SVMTorch: Support Vector Machines for Large-Scale Regression Problems , in Journal of Machine Learning Research, 2001. (http://www.idiap.ch/learning)