RC 22064 (98981) 21 May 2001 Computer Science

IBM Research Report

Study of Embedded Font Context and Kernal Space Methods for Improved Videotext Recognition

Hrishikesh Aradhye[*], Chitra Dorai[1], Jae-Chang Shim[2]
[*]The Ohio State University
140 W. 19th Avenue
Columbus, Ohio 43210

[1]IBM Research Division
Thomas J. Watson Research Center
P. O. Box 704
Yorktown Heights, NY 10598

[2]Andong National University
388 Song-Chun Dong
Kyungpook
760-749 South Korea

Research Division
Almaden - Austin - Beijing - Haifa - T. J. Watson - Tokyo - Zurich

STUDY OF EMBEDDED FONT CONTEXT AND KERNEL SPACE METHODS FOR IMPROVED VIDEOTEXT RECOGNITION

Hrishikesh Aradhye[†], *Chitra Dorai*[‡], *Jae-Chang Shim*[⊙]

The Ohio State University[†]
140 W 19th Avenue
Columbus
Ohio 43210, USA
aradhye.1@osu.edu

IBM T.J. Watson Research Center[‡]
P.O. Box 704
Yorktown Heights
New York 10598, USA
dorai@watson.ibm.com

Andong National University[⊙]
388 Song-Chun Dong
Kyungpook
760-749 South Korea
jcshim@andong.ac.kr

ABSTRACT

Videotext refers to text superimposed on video frames. A videotext based Multimedia Description Scheme has recently been adopted into the MPEG-7 standard. A study of published work in the area of videotext extraction and recognition reveals that, despite recent interest, a reliable general purpose video character recognition (VCR) system is yet to be developed. In our research and development of a character recognition algorithm designed specifically for the low resolution output from automatic videotext extractors, we observed that raw VCR accuracies obtained using various classifiers including kernel space methods such as SVMs, are inadequate for accurate video annotation and browsing. Intelligent post-processing mechanisms that are supported by general data characteristics of the domain are hence, required for performance improvement. We describe one such method, referred to as the Font Context Analysis, which works independently of the raw character recognition technique. As a result, it can be easily implemented in conjunction with other VCR algorithms being developed elsewhere, and offer the same performance gains. Experimental results on various video streams show notable improvements in recognition rates with our system incorporating a SVM-based character recognition mechanism and font context analysis.

1. INTRODUCTION

The evolution of the Internet has led to a proliferation of digital media available for search and retrieval. Manual indexing of multimedia archives is an expensive and time-consuming task given the volume and richness of their content. Algorithms for automated content analysis of digital images and video are, thus highly desirable. Text superimposed on video provides important information such as the identity of the speaker, his/her location, the topic under discussion, scores, associated data, and so on. Videotext extraction and recognition, therefore forms an important task in an automated video content annotation system. Unlike scanned paper documents, videotext is superimposed on varying backgrounds with moving objects with a rich variety of colors and textures. In addition, text extracted from digital video is often of low resolution and suffers from compression artifacts. Due to these difficulties, our experiments and other previous work (see Table 1) evidenced low accuracy of character recognition (about 50%) when commercially available OCR algorithms were applied to text extracted from various digital video streams. A video character recognition (VCR) algorithm designed specifically for the low resolution output from videotext extractors is thus needed for better results.

A study of published research in the area of videotext extraction and recognition reveals that, in spite of recent interest, a general purpose VCR system is yet to be developed (Table 1). Most of the existing methods focus on a specific genre of digital video. The best results reported are those of Sato et.al. [1]. They report a recognition accuracy of 73% to 89% for CNN news videos. However, they make domain-specific assumptions such as stationary text, brightly illuminated text on a dark background, and so on. Our video character extraction and recognition system is designed to achieve similar or better results, without making any such restrictive assumptions.

Our previous work [2] presented an algorithm for automatic videotext extraction. This paper focuses on the subsequent steps of feature extraction from character regions, SVM-based classification, videotext object synthesis, font context analysis, and temporal contiguity analysis for accurate video character recognition. These individual steps can also be implemented in conjunction with other VCR algorithms being developed elsewhere, and in turn, can improve their recognition performance.

2. TEXT EXTRACTION FOR VCR

Extraction of videotext in a frame is the result of image analysis involving text character segmentation and location. The algorithm briefly described here is designed to extract superimposed text and also scene text which possesses typical (superimposed) text attributes. No prior knowledge about frame resolution, text location, font styles, and text appearance modes such as normal and inverse video are assumed. Some common characteristics of text are exploited in the algorithm including monochromaticity of individual characters, size restrictions (characters cannot be too small to be read by humans or too big to occupy a large portion of the frame), and horizontal alignment of text (preferred for presentation for ease of reading).

Our videotext extractor [2] works by segmenting and analyzing regions in a video frame. The processing stages in this system are: (i) isolating regions that may contain text characters, (ii) separating each character region from its surroundings, and (iii) verifying the presence of text by consistency analysis across multiple text blocks. Optionally, if consecutive frames in videos are being processed together in a batch job, then text regions determined from say, five consecutive frames can be also analyzed together to

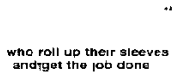
Table 1. Review of videotext recognition systems.

Reported Work	Recognition Method	Accuracy	Remarks
Baseline OCR [3]	Conventional OCR	50%	No frame analysis
CMU [3, 1]	Nearest Neighbor	73%-89%	CNN News, Cannot handle moving text
SRI ConTEXTrack [4]	Preprocessing + commercial OCR	Unknown (estimated to be 70%)	CNN News, Works in real-time
UMann [5]	Iterative comm. OCR	41%- 76%	Results obtained from JPEG
WLDC [6]	Local direction feature + cityblock distance classifier	76%	Works on Japanese
SOCR [7]	machine-learning techniques	60%	Font dependent performance

add missing characters in frames and to delete incorrect regions posing as text. The output is a binary image of detected text lines with text as foreground in black on a white background. This can be the input to a VCR system which recognizes the characters and outputs international character codes for further use in video analysis and annotation.

Figure 1 shows a sample result obtained by applying our videotext extraction algorithm to a frame from an IBM commercial video.

(a) Video frame



(b) Extracted text

Fig. 1. Result of our videotext extraction algorithm.

3. FEATURES FOR VCR

We have developed a rich feature set for video character recognition. Our features are designed to capture coarse shapes of the characters to handle low resolution of video characters. Rather than extracting precise character shapes and their properties, we have devoted our attention to macro font features.

The features, *new* and *old* developed for our VCR system include: (i) Vector template features which consist of measurements obtained from the slope histogram [8] computed from the vectorization of a character outline; (ii) Regional features which characterize a character blob's area; (iii) Run features encoding longest vertical and horizontal run lengths in three positions within a character MBR; (iv) Balance and symmetry features which capture both the balance and the symmetry that may be present in the characters; (v) Occupancy feature which encodes the positional presence of the character in its MBR block; (vi) Skeleton features which describe interrelationships between end and junction points on character skeletons obtained by thinning; (vii) Corner features that code corner presence of a character in its MBR; (viii) Convex deficiency feature details which encode the positions, sizes and directions of each convex deficiency detected in MBRs; (ix) MBR deficiency feature details which describe bi-open direction vectors and the positions and sizes of each MBR deficiency detected in the MBR; (x) Local direction distribution feature that locally measures the length of the character components in four major orientations of the character MBR; and (xi) topological features that represent holes in the characters and their attributes.

The values of these features (a total of 172) for each character extracted constitute the feature vector input to a VCR classifier. The classifier is then trained to learn the mapping from this 172-dimensional input feature space to an output recognition space consisting of 62 classes, corresponding to 26 lower case letters, 26 upper case letters, and 10 digits.

4. VCR USING SUPPORT VECTOR MACHINES

We use Support Vector Machines (SVMs) as our basic classifier for video character recognition. An SVM is a new classification algorithm under development by Vapnik and co-workers [9]. The advantages of SVMs over other learning methods were demonstrated with the USPS handwritten character recognition dataset. They include ease of parameter selection and configuration, faster training, and global optimization. Our research establishes the utility of kernel space methods such as SVMs for recognizing video characters where the problem domain is complicated by lower character resolution, a bigger output recognition space, and a parsimonious input feature set.

4.1. SVMs for video character recognition

Our experimental results are based on two datasets: Dataset D1 was generated using the *drawString* Java function on separate image buffers and consisted of 8000 characters generated using 3 different fonts, 12 sizes, and 4 styles. The characters were in black on a white background and were subjected to compression using 4 different JPEG quality factors (Q). Dataset D2, on the other hand, consisted of characters extracted from various MPEG-1 video streams using our videotext extraction algorithm. Image buffers from both the datasets were automatically processed for background removal and feature extraction. Recognition rates are reported on a case-sensitive basis, with numbers reported in parentheses for accuracies calculated on a *case-independent* basis.

Pattern recognition problems where the number of classes is greater than two are typically solved with a combination of many binary SVM classifiers. Our results indicate that the One-vs-One architecture with a linear kernel function provides the best performance in terms of both speed and accuracy. The Sequential Minimal Optimization algorithm was used for SVM training with a conservative margin parameter (C) of 1.0. The SVM classifier

was trained on 4000 characters chosen randomly from Dataset D1 and tested on remaining characters from D1 and on the characters from D2.

4.2. Raw VCR Performance

We compared the raw SVM classification performance on D1 with that of a Perceptron, decision trees (C4.5), and a Naive Bayes classifier. SVMs are seen to outperform the other three algorithms as shown in Table 2.

Table 2. VCR Performance comparison with dataset D1, 4000 test samples and Q = 0.5.

Algorithm	1000 Training Samples	4000 Training Samples
SVM	78.7%	89.3%
Perceptron	64.8%	
C4.5		80.3%
Naive Bayes		84.0%

The results for our SVM-based VCR system on the dataset D2 are provided in Table 4. A typical range of recognition accuracies is from 65% to 78%. The errors are due to SVM misclassification as well as due to missed, merged, or broken characters from the text extraction phase. Further, the videos often contained characters with fonts different from the 3 basic fonts used for training.

5. SMART POSTPROCESSING

Although the raw VCR results are unacceptable for the purpose of accurate video content annotation, they indicate a reliable baseline performance which can be improved further by postprocessing. The following sections describe the postprocessing steps for improving the recognition accuracy in our VCR scheme as well as any other character classifier.

5.1. Text Object Synthesis

Video characters are intended to convey a meaningful message to the viewer. As a result, videotext seldom occurs as isolated characters. In addition to the removal of the occasional falsely detected character region by the text extraction algorithm, analysis of *groups* of *semantically and syntactically related* characters has certain advantages in terms of improvement in the recognition accuracy. We refer to such groups as *videotext objects* (Figure 2). Characters belonging to the same text object share properties such as horizontal alignment, spatial proximity, parallel motion, and so on. Some of these properties can be used to synthesize text objects from individual characters, and others can then be used to isolate the misfit or inconsistent members from the groups. Inconsistencies can occur due to text extraction errors such as merged, broken, and/or non-text characters; or at a later stage due to misclassification by the VCR classifier. Localization of such errors can help correct them by iteratively applying text extraction and/or classification until all members of a text object are consistent.

In our present implementation, two video characters are concluded to be *neighbors* if (i) they are horizontally aligned; (ii) they are separated by a distance that is within an acceptable maximum and there is no other character between them; (iii) the ratio of their heights is within an acceptable range, and so is the ratio of their widths; and (iv) their gray scale values match within an acceptable error margin.

A text object is then synthesized as a chain that starts from a character that has no left neighbor and ends at a character that has no right neighbor. Text object attributes such as their bounding box locations and sizes are output in XML as part of their descriptions based on the MPEG-7 Videotext MDS.

Fig. 2. Text object synthesis.

5.2. Font Context Analysis

Font context analysis (FCA) is motivated by the fact that certain pairs of characters, such as the letter `g` and the number `9`, are very similar in terms of features, albeit very easily distinguishable by their placement above or below the horizontal position baseline.

Due to the possibility of errors in text extraction and/or classification, the baseline, ascent and descent portion of a text object cannot always be exactly determined. We have used the following technique. Each character *votes*, according to its classification determined by the VCR system (in our case SVMs), for a particular distribution of its text object's vertical space into ascent, body, and descent. A distribution that is consistent with a majority of the member characters of the text object is then calculated. Let such information be referred to as the *Font Context*. Characters that are inconsistent with respect to the estimated distribution of the object's font context are potential classification errors. For example, if a character is judged to be comprised of no ascent, a body and a descent, a label of `S` would be inconsistent since an `S` is expected to have an ascent and a body with no descent. For each incorrect character, we revisit the classification stage. Of the 62 possible character labels, all class labels that are inconsistent with the estimated font context are eliminated. The VCR system is then made to choose among the remaining classes.

For example, consider the extracted text from Figure 3(top). The raw SVM classification wrongly labels the letters `g` and `h` as `S` and `R`, respectively (Figure 3(middle)). Considering the baseline alignment of the majority of the characters from this text object, the label `S` is readily diagnosed to be incorrect and is correctly replaced by the label `g` (Figure 3(bottom)). The other wrong label, `R`, cannot be detected by font context analysis, since it satisfies the appropriate constraints. Table 3 provides some more examples of the application of the font context analysis. Results from Table 4 show that the font context analysis can significantly improve the recognition accuracy of the raw SVM-based VCR system. A similar analysis can be applied across character widths as well.

Our FCA assumes that the raw classifier output is of *reasonable* reliability so that its estimates of font context characteristics

Table 3. Effect of font context analysis on our VCR system with One-vs-One SVMs trained on 4000 D1 characters with Q = 0.5.

Original Text	Raw SVM Recognition	+ Font Context Analysis	Mistakes Detected	Mistakes Corrected
Tough day mac	TouSR day mac	TougR day mac	1	1
Man Im so stressed	Man t o atressed	Man t o atressed	0	0
Call IBM	Cail lBM	Call lBM	1	1
IBM consultants	iBM con6ui1ants	lBM consul1ants	3	2
Experienced team players	Experi4nced tegm PiByerD	Experienced team playera	6	5
Right	Rt9ht	Rtght	1	1

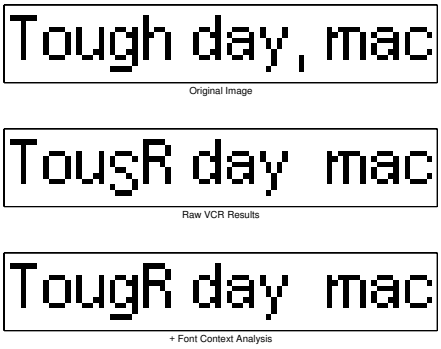


Fig. 3. Example of font context analysis.

(such as the location of the baseline) are accurate. It can then improve upon the accuracy of the raw classifier output by suggesting certain changes, if necessary. Secondly, it is only applicable when the text object contains sufficient number of characters so that the majority-basis estimates of font context are meaningful.

5.3. Temporal Contiguity Analysis

Fusion of recognized labels from three consecutive video frames further improves the recognition accuracy. In the presence of moving videotext, recognized text is aligned using an approach based on dynamic programming. It optimizes an enhanced edit distance metric that penalizes mismatches in videotext motion over consecutive frames. Table 4 summarizes the improvements in results.

Table 4. Performance of our VCR system with One-vs-One SVMs trained on 4000 D1 characters with Q = 0.5.

Test Video	CNN-1	T5	Bangkok
Type	News Captions	Scrolling Credits	Subtitles
Raw SVM Accuracy	(67%)	(65%)	(78%)
+ Font Context Analysis	72%	68%	87%
+ Temporal Analysis	80%	76%	89%

6. CONCLUSION

This work presents a novel scheme for the task of video character recognition. The core of this scheme is a Support Vector Machine based classifier. SVMs outperformed other classification algorithms, such as a Naive Bayes classifier, when tested on generated characters. We also propose a set of novel postprocessing steps that significantly improve the recognition performance. Our method for font context analysis exploits syntactic information such as placement of characters relative to the string baseline. Video characters are grouped into text objects based on characteristics such as proximity and alignment. Potential recognition errors are identified via mismatches in the placement of characters with respect to the collective coordinates of the text object. These errors are often fixed by forcing the classifier to propose alternate labels for the erroneously classified characters until no mismatches are detected. Since this scheme is independent of the character classifier, it can be used to improve recognition accuracies of other VCR algorithms as well. Our end-to-end SVM-based VCR scheme performs comparably well with respect to recent methods without resorting to any domain-specific restrictive assumptions.

7. ACKNOWLEDGEMENT

We thank Irina Rish for running the Naive Bayes experiments on our VCR data.

8. REFERENCES

[1] T. Sato, T. Kanade, E. Hughes, M. Smith, and S. Satoh, "Video OCR: Indexing news libraries by recognition of superimposed caption," *ACM Multimedia Systems*, vol. 7, no. 5, pp. 385–395, 1999.

[2] Jae-Chang Shim, Chitra Dorai, and Ruud Bolle, "Automatic text extraction from video for content-based annotation and retrieval," in *Proc. International Conference on Pattern Recognition*, Brisbane, Australia, August 1998, vol. 1, pp. 618–620.

[3] T. Sato, T. Kanade, E. Hughes, and M. Smith, "Video OCR for digital news archives," in *IEEE Intl. Workshop on Content-Based Access of Image and Video Databases*, January 1998.

[4] G. Myers and P. Gallagher, "Video text recognition," 2000, http://www.erg.sri.com/automation/video_recog.html.

[5] Rainer Lienhart and Wolfgang Effelsberg, "Automatic text segmentation and text recognition for video indexing," *ACM Multimedia Systems*, vol. 8, pp. 69–81, January 2000.

[6] H. Kuwano, Y.Taniguchi, H. Arai, M. Mori, S. Kurakake, and H. Kojima, "Telop-on-demand: Video structuring and retrieval based on text reocgnition," in *ICME 2000*, New York City, July 2000, pp. 759–762.

[7] M. Bertini, A. Del Bimbo, and P. Pala, "Content based annotation and retrieval of news video," in *ICME 2000*, New York City, July 2000, pp. 483–486.

[8] J.R. Parker, *Alogirthms for image processing and computer vision*, John Wiley & Sons, New York, 1997.

[9] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, N.Y., 1995.