TEXT DETECTION AND EXTRACTION FROM COMPLEX VIDEO SCENES USING MORPHOLOGY

[1]Mr. Binay Kumar Chaudhary, [2]Dr. Shambhu Kumar Singh
[1]Research Scholar, [2]Assistant Professor & Head
CSE Department, Sandip University, Madhubani, BIHAR, INDIA
vinaychaudhary54321@gmail.com, shambhu.singh@sandipuniversity.edu.in

Abstract

Video text provides crucial information that aids in understanding, annotating, and searching video scenes. Previous methods for extracting text from videos often rely on low-level features such as edges, color, and texture. However, these methods struggle with texts that have varying contrasts or are embedded in complex backgrounds. In this paper, we present a new framework for detecting and extracting text from video scenes. We generate a morphological binary map by calculating the difference between the closing image and the opening image. Candidate regions are then connected using a morphological dilation operation, and text regions are identified based on the presence of text within each candidate. The text regions are accurately localized by projecting text pixels onto the morphological binary map, and text extraction is subsequently performed. Our method is robust to different character sizes, positions, contrasts, and colors, and it is language-independent. To reduce processing time, text region updates between frames are also implemented. We conducted experiments on various videos to demonstrate the effectiveness of our proposed method.

I. INTRODUCTION

The amount of images and videos available on the internet and in databases is rapidly growing. Broadcasters are increasingly interested in creating extensive digital archives of their content to reuse in TV programs, make available online to other companies, and share with the public. To meet this demand, there is a need for systems capable of efficiently indexing and retrieving video segments based on the extraction of content-related information from visual data. While content-based retrieval of images is effectively achieved using low-level image features, this approach is not as successful for videos, except in very limited contexts. Instead, successful video retrieval relies on high-level content descriptors [1].

Broadcasting videos frequently use text to provide direct summaries of their content and enhance the viewing experience. For instance, headlines summarize news reports, and subtitles in documentaries help viewers understand the content. Sports videos also include text to display scores and the names of teams or players [2]. Generally, text in videos can be categorized into scene text and overlay text [3]. Scene text appears naturally in the background as part of the scene, such as on advertising boards and banners. Overlay text, on the other hand, is superimposed on the video scene to aid viewer comprehension. This overlay text is often compact and structured, making it useful for video indexing and retrieval [4]. However, extracting text for video optical character recognition (OCR) is more challenging than OCR for document images due to the complexities of backgrounds, unknown text colors, and varying text sizes.

The remainder of this paper is organized as follows. Section II reviews related work. Section III describes the generation of the morphological binary map and the refinement of detected text regions.

Section IV explains the text extraction process from the refined text regions. Section V presents experimental results on various videos, and Section VI concludes the paper.

II. RELATED WORK

Many existing methods for detecting text in videos rely on features such as color, edge, and texture. Color-based methods typically assume that video text has a uniform color. For example, Agnihotri and Dimitrova's approach [5] detects and binarizes horizontal caption text in white, yellow, and black within video frames. After preprocessing, edge pixels are identified using a fixed-threshold edge detector. Regions with excessively high edge density are deemed too noisy and are discarded. Remaining regions undergo connected component analysis, merging edge components based on spatial heuristics to localize text regions. Binarization is then performed by thresholding at the average pixel value of each localized text region. Kim et al. [6] utilize Euclidean distance clustering in the RGB space, employing 64 clustered color channels for text detection. However, video text rarely maintains a uniform color due to compression artifacts and low contrast between text and background.

Edge-based methods are effective for video text detection since text regions typically contain rich edge information. These methods often apply an edge detector to the video frame and identify regions with high edge density and strength. This approach works well in scenes without complex backgrounds but becomes less reliable as background complexity increases. Lyu et al. [7] use a modified edge map to detect text regions, which are then localized using coarse-to-fine projection. Text strings are extracted through local thresholding and inward filling. Xi et al. [8] propose an edge-based method that employs the Sobel operator to create an edge map, followed by smoothing filters, morphological operations, and geometrical constraints.

Texture-based methods, including salient point detection and wavelet transforms, have also been employed for text detection. Bertini et al. [9] detect corner points in video scenes and identify text regions based on the similarity of corner points between frames. Zhong et al. [10] use texture features from DCT coefficients to detect text in JPEG/MPEG compressed domains. They first identify blocks with high horizontal spatial intensity variation as text candidates, refining these into regions using spatial constraints. Potential caption text regions are verified by vertical spectrum energy, though this method's robustness in complex backgrounds may be limited due to spatial domain feature constraints.

After text detection, text extraction is necessary before applying OCR. Text extraction methods can be categorized into color-based [11] and stroke-based [12] methods. Color-based methods, such as Otsu's algorithm [11], are widely used due to their simplicity and efficiency. However, Otsu's method is not robust when text and background colors are similar because it uses global thresholding. To address this, detected text regions can be divided into blocks, and adaptive thresholding, such as that introduced in [7], can be applied locally. Stroke-based methods use filters based on stroke direction to enhance stroke-like shapes and suppress others. The four-direction character extraction filters [12] are an example, but they can suppress characters without clear stripe shapes due to their language dependence.

In this paper, we introduce a novel text detection and extraction method that utilizes the transition region between text and background. We generate a morphological binary map based on the observation that transient colors exist between text and its adjacent background. Text regions are roughly detected by computing the density of transition pixels and the consistency of texture around these pixels. The detected text regions are then accurately localized using the projection of the morphological binary map, with an improved color-based thresholding method [7] to correctly extract text strings.

III. TEXT REGION DETECTION

The proposed method is based on our observations that there exist contrast colors between text and its adjacent background. The relative contrast between texts and their background is an important feature for text region detection. The overall procedure of proposed text detection method is shown in Fig. 1.The text extraction method will be clearly explained in Section IV.

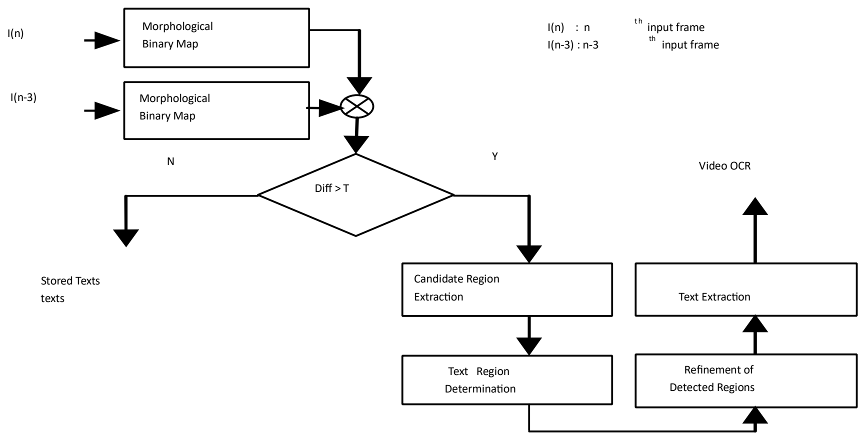


Fig. 1. Overall procedure of the proposed detection method.

A. Morphological Binary Map

In order to detect text regions from complex background a morphology based approach is used to extract high-contrast feature [13].

Let $I(x,y)$ denote a gray-level input image. Let $S_{m,n}$ denote a structuring element with size m×n. where m,n are odds and larger than zero. Besides, let $\oplus$ denote a dilation operation, and $\ominus$ denote an erosion operation.

Closing Operation:

$$I(x,y)\bullet S_{m,n}=(I(x,y)\oplus S_{m,n})\ominus S_{m,n} \quad (1)$$

Opening Operation:

$$I(x,y)\circ S_{m,n}=(I(x,y)\ominus S_{m,n})\oplus S_{m,n} \quad (2)$$

Difference:

$$D(I_1,I_2)=|I_1(x,y)-I_2(x,y)| \quad (3)$$

Thresholding:

$$T(I(x,y))=\begin{cases} 255, & \text{if } I(x,y) > T \\ 0, & \textit{otherwise} \end{cases} \quad (4)$$

To obtain the morphological binary map, closing (1) and opening (2) operations are performed using a disk structural element $S_{3,3}$. The difference (3) obtained from subtracting both images are the result of the following step. Then, a threshold procedure (4) is applied followed by a labeling process to extract the text segments. In the threshold procedure a parameter T is defined dynamically according to the background of the image. This parameter is responsible to determine the limit value of the binarization operation.

The whole procedure of our morphology-based technique to extract the contrast features is shown in Fig 2.

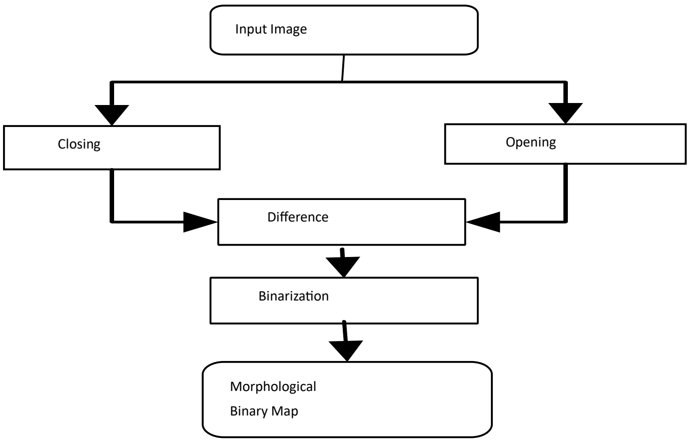


Fig.2. Flowchart of the proposed method to extract contrast features for text region detection.

An example of the result of this process is shown in Fig.3(b).

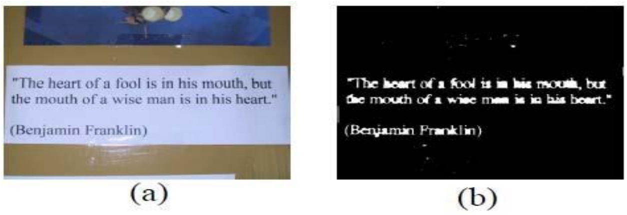
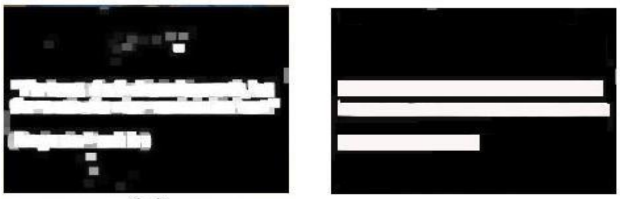


Fig.3. Generation of morphological binary map (a) Input image (b) morphological binary map

B. CandidateRegionExtraction

A morphological dilation operator can easily connect the very close regions together while leaving those whose positions are far away to each other isolated. In our proposed method, we use a morphological dilation operator [14] with a 7× 7 square structuring element to the binary image obtained from the previous step to get joint areas referred to as text blobs. Fig.4 (a) shows the result of feature clustering. If a gap of consecutive pixels between two nonzero points in the same row is shorter than 5% of the image width, they are filled with 1s. If the connected components are smaller than the threshold value, they are removed. The threshold value is empirically selected by observing the minimum size of text region. Then each connected component is reshaped to have smooth boundaries. Since it is reasonable to assume that the text regions are generally in rectangular shapes, a rectangular bounding box is generated by linking four points, which correspond to (min_x, min_y), (max_x, min_y), (min_x, max_y), (max_x, max_y) taken from the text blobs. The refined candidate regions are shown in Fig. 4(b).



(a) (b)

Fig.4. Extraction of candidate regions (a) Connected components through dilation (b) Smoothed candidate regions

C. Text Region Determination

The next step is to determine the real text region among the boundary smoothed candidate regions by some useful clues, such as the aspect ratio of text region. Since most of texts are placed horizontally in the video, the vertically longer candidates can be easily eliminated. Based on the observation that intensity variation around the transition pixel is big due to complex structure of the text, we employ the dominant local binary pattern (DLBP) introduced in [15] to describe the texture around the transition pixel. DLBP effectively capture the dominating patterns in texture images. Unlike the conventional LBP approach, which only exploits the uniform LBP [16], given a texture image, the DLBP approach computes the occurrence frequencies of all rotation invariant patterns defined in the LBP groups. These patterns are then sorted in descending order. The first several most frequently occurring patterns should contain dominating patterns in the image and, therefore, are the dominant patterns.

LBP is a very efficient and simple tool to represent the consistency of texture using only the intensity pattern. LBP forms the binary pattern using current pixel and its all circular neighbor pixels and can be converted into a decimal number as follows:

$$\mathrm{LBP}_{P,R} = \sum_{i=0}^{p-1} s(g_i - g_c)2^i, \text{where} \quad s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \qquad (5)$$

Where, P and R denote the user's chosen number of circular neighbor pixels of a specific pixel and the radius of circle, respectively. g_c and g_i denote the intensity of current pixel and circular neighbor pixels, respectively.

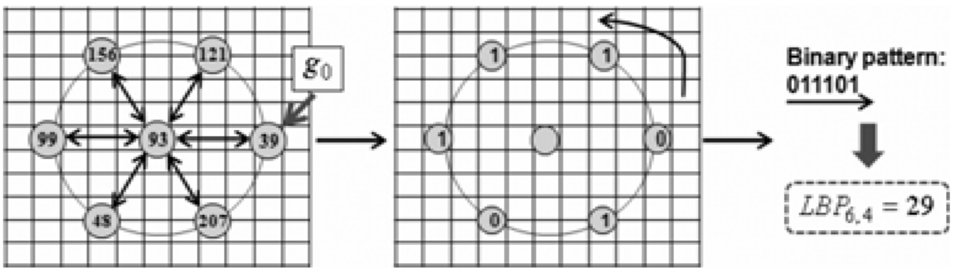


Fig.5. Example of LBP computation

We can obtain the binary pattern as shown in Fig. 5, and the resultingLBP$_{6,4}$=29(=2^4+2^3+2^2+2^0).

DLBP consider the most frequently occurred patterns in an image. It is shown that the DLBP approach is more reliable to represent the dominating pattern information in the images.

It avoids the aforementioned problems encountered by merely using the uniform LBPs or making use of all the possible patterns, as the DLBPs are defined to be the most frequently occurred patterns.

The pseudo codes on determining the number of dominant patterns of DLBP and extracting DLBP feature vectors are presented in Algorithm 1 and Algorithm 2, respectively.

Algorithm 1

Determining the number of dominant patterns of DLBP

Input: Input image, and the parameters P and R for DLBP Output: The required number of patterns for 15% pattern occurrences

1. Initialize K_{temp}=0.
2. FOR each Candidate region *I* in the image
3. Initialize the pattern histogram, $H[0...(2^m-1)]=0$.
4. FOR each center pixel $g_c \in I$
5. Compute the pattern label of g_c, $LBP_{P,R}$ (1)
6. Increase the corresponding bin by 1, $H[LBP_{P,R}]$++
7. END FOR
8. Sort the histogram in descending order
9. Find the number of patterns *k* for 15% pattern occurrences in *I* $K = \arg_k \left(\frac{\sum_{i=0}^{k-1} H[i]}{\sum_{i=0}^{(2m-1)} H[i]} \right) \geq 15\%$
10. K_{temp}+=k
11. END FOR
12. $K_{15\%}$=Number of different Dominant Patterns having occurences more than 15%.
13. 13. Return $K_{15\%}$.

Algorithm 2

Extracting a DLBP feature vector

Input: Input image, the required number of dominant patterns $K_{15\%}$, and the parameters P and R for DLBP

Output: The DLBP feature vector corresponding to

1. Initialize the pattern histogram, $H[0...(2^m-1)]=0$.

2. FOR each center pixel $g_c \in I$

3. Compute the pattern label of g_c, $LBP_{P,R}$ (1)

4. Increase the corresponding bin by 1, $H[LBP_{P,R}]$++

5. END FOR

6. Sort the histogram in descending order

7. Return $H[0...(K_{15\%}-1)]$ as the feature vector of DLBP.

Now we define the probability of text (POT) using the operator as follows: The LBP operator is first applied to every transition pixel in each candidate region. We use the 8 neighbor pixels to obtain the DLBP value. Then, we compute the number of different DLBPs to consider the intensity variation around the transition pixel by algorithm 1. Thus the total number of potentially different DLBPs is K. Algorithm 2 explains the extraction of DLBP feature vector.

Let w_i denote the density of transition pixels in each candidate region and can be easily obtained from dividing the

number of transition pixels by the size of each candidate region. POT is defined as follows:

$$\mathrm{POT}_i = w_i \times \mathrm{NOD}_i, \qquad i=1\ldots\ldots\ldots\ldots N \qquad (6)$$

Where N denotes the number of candidate regions as mentioned. NOD_i denotes the number of different DLBPs, which is normalized by the maximum of the number of different DLBPs (i.e.,$K_{15\%}$) in each candidate region. If POT of the candidate region is larger than a predefined value, the corresponding region is finally determined as the text region. The detection result is shown in Fig. 6. The thresholding value in POT is empirically set to 0.05 based on various experimental results.We can see that the text region is well identified from other candidates.

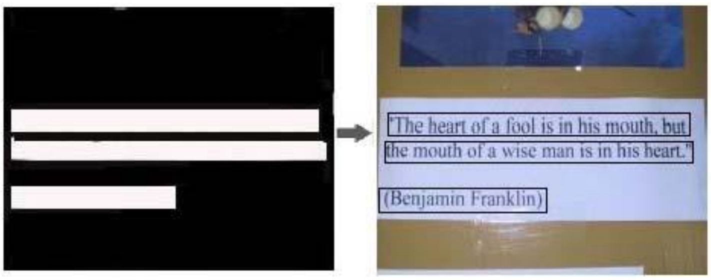


Fig.6.Text Region Determination

D. Section Headings

The text region or the bounding box obtained in the preceding subsection needs to be refined for better accurate text extraction. In this subsection, we use a modified projection of transition pixels [17] in the morphological binary map to perform the text region refinement. First, the horizontal projection is performed to accumulate all the transition pixel counts in each row of the detected text region to form a histogram of the number of transition pixels. Then the null points, which denote the pixel row without transition pixels, are removed and separated regions are re-labeled. The projection is conducted vertically and null points are removed once again. Compared to the coarse-to-fine projection proposed for edge-based scheme in, our projection method is applied to the detected text regions only, making the process simpler. The result of refinement is shown in Fig. 7.

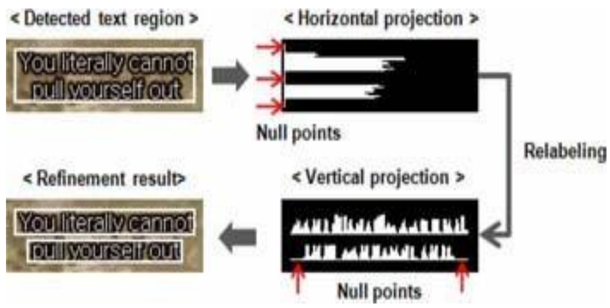


Fig.7.Refinement process of detected text region

E. TextRegionUpdate

Once the text regions are detected in the current frame, it is reasonable to take advantage of continuity of text between consecutive frames for the text region detection of the next frame. If the difference, which can be obtained by XOR of current morphological binary map and previous morphological binary map, is smaller than a predefined value, the text regions of previous frame are directly applied as the detection result without further refinement.

In order to deal with such changes, we compare the current morphological binary map with the morphological binary map obtained 3 frames earlier and the dissimilarity measure between these maps is defined as follows:

$$d\,(\mathrm{M_n}, \mathrm{M_{n-3}}) = \sum_{(x,y)\in T} (\mathrm{M_n}(x,y) \otimes \mathrm{M_{n-3}}(x,y))$$

$$\text{if } (d(m_n, M_{n-3}) < T)\ TR_n = TR_{n-3}$$
$$\text{Otherwise, find new } TR_n \qquad (7)$$

Where M_n and M_{n-3} denote the morphological binary map obtained from *n*th frame and the (*n*-3)th frame, respectively. TR_n and TR_{n-3} denote the detected text regions in the *n*th frame and (*n*-3)th frame, respectively. ⊗ denotes the XOR operator. If the values on the *n*th frame and the (*n*-3)th frame morphological binary map are same, the result of ⊗ between

two values is set to be 0.Otherwise the result of ⊗ between two values is set to be 1. The text region update method can reduce the processing time efficiently.

IV. TEXT EXTRACTION

Before applying video OCR application, the refined text regions need to be converted to a binary image, where all pixels belonging to text are highlighted and others suppressed.Since the text color may be either brighter or darker than the background color, an efficient scheme is required to extract the text dealing with complex backgrounds and various text appearances. In this section, we propose a fast and efficient text extraction technique, which is based on Lyu's approach [7].

A. Color Polarity Computation

Color based text extraction technique [17] is proposed for text extraction. The goal in this subsection is to check the color polarity and inverse the pixel intensities if needed so that the output text region of the module can always contain bright text compared to its surrounding pixels. We observe that this goal can be simply attained owing to the morphological binary map obtained in the preceding section. First of all, the binary image obtained by thresholding with average intensity value can be effectively utilized. Given the binarized text region, the boundary pixels, which belong to left, right, top, and bottom lines of the text region are searched and the number of white pixels is counted. If the number of white boundary pixels is less than 50% of the number of boundary pixels, the text region is regarded as "bright text on dark background" scenario, which requires no polarity change. In other words, the text is always bright in such scenarios. If the number of white pixels is greater than that of black pixels, we conduct a task to turn on or off the "bright_text_flag" as expressed in (8).

$$\text{Bright\_text\_flag} = \begin{cases} 1, & \text{if } I_B(x_F, y_F) = 1 \text{ and } I_B(x_F+2, y_F) = 0 \\ 0, & \text{Otherwise} \end{cases} \quad \textbf{(8)}$$

Where (x_F,y_F) denotes the position of the first encountered transition pixel in each row of the text region and I_B denotes the value on the binary image.

The flag is set to 1 if the first encountered transition pixel belongs to 1, whereas the pixel apart by two pixel distance belongs to 0. If such case happens at least once, the pixel values in the text region is inverted to make the text brighter than the surrounding background. Note that the inversion is simply done by subtracting the pixel value from the maximum pixel value. The process of color polarity computation is shown in Fig.8.

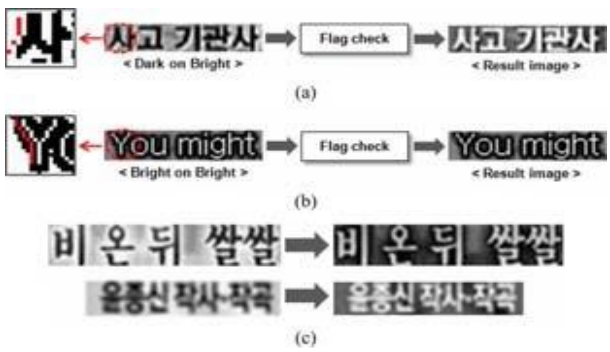
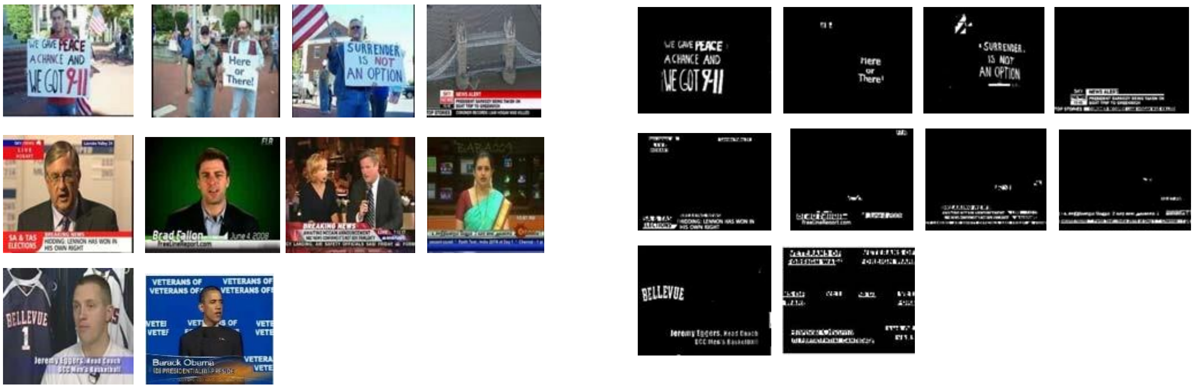


Fig.8. Process of inversing image by the color polarity. (a) Dark text on bright background. (b) Bright text on bright background. (c) Examples of inverted text by "bright_text_flag".

As shown in Fig. 8, the flag is set to 1 for Fig. 8(a) since the first encountered transition pixel belongs to 1, whereas the pixel apart by two pixel distance belongs to 0. The first transition pixels in each row on the binary image are represented by red color in Fig. 8. Examples with "bright_flag_text" are also shown in Fig. 8(c).

V. EXPERIMENTAL RESULTS

The proposed approach has been tested on real-life videos. The experiments have been performed on a Pentium PC with 333 MHz CPU. The program is implemented in MATLAB v7.0.As there is no standard database available we created our own dataset consisting of 15 MPEG-1 video sequences with 320 × 240 pixel resolution. There were a total of 5,299 frames (about 75MB of data). There were 156 overlay text events and 144 scene text events in the video data. All text had horizontal orientation and most text events were stationary. The dataset contained a wide variety of video captured from television channels, including television commercials and news broadcasts (domestic and foreign). A wide variety of text fonts, colors, languages, and scripts were represented. Video sequences were captured at 30 frames per second. The proposed method is processed in frames.



(a) (b)

Fig.9.Experimental results of overlay text and scene text detection and extraction (a) original frames (b) Result

Scene texts are difficult to detect than overlay texts because some of them are irregularly aligned and with different size. Fig.9 gives some result of the proposed algorithm for overlay text and scene text.

A. Performance Evaluation

In order to confirm the superiority of our proposed text detection and extraction method, we compare our approach with other methods; Lyu's method [7] and Kim's method [17].The accuracy of text extraction shown in Fig. 8 is evaluated using the probability of error (PE) [18] as follows:

$$PE = P(T)P(B\backslash T) + P(B)P(T\backslash B) \quad (9)$$

Where P(T) and P(B) denote the probabilities of text pixels and background pixels in the ground-truth image ,respectively. P(B\T) denotes the error probability to classify text pixels as background pixels. P(T\B) denotes the error probability to classify background pixels as text pixels. The comparison of PE for each extraction result and Total Processing Time is shown in Table 1.

TABLE1
COMPARISON OF EFFICIENCY OF EXTRACTION

IMAGE	KIN'S METHOD	PROPOSED METHOD
1	0.086	0.072
2	0.076	0.065
3	0.059	0.059
4	0.057	0.055
5	0.074	0.067
6	0.070	0.068
7	0.102	0.087
8	0.132	0.098
9	0.078	0.067

10	0.092	0.088

The comparison of proposed method with Kim's method is shown in Table 1 and Fig 10. It clearly shows that Average PE is reduced.

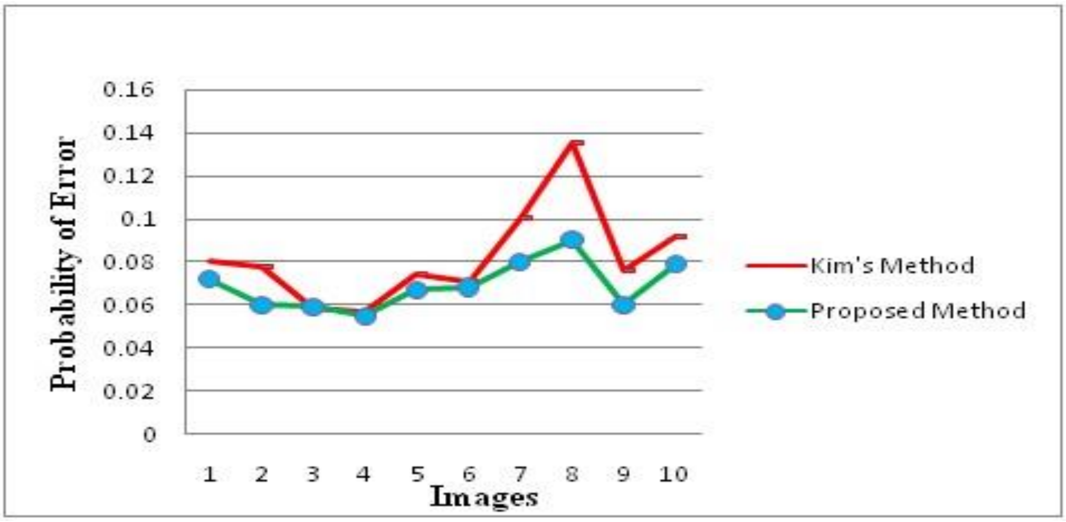


Fig.10. PE for Each Extraction Result

DLBP is used to describe the texture around the transition pixel. Unlike LBP, which makes use of all possible patterns, DLBP consider only the most frequently occurred patterns in a texture image. The dominant patterns around 80% of the total pattern occurrences, which can effectively captures the image textural information is used to describe the texture. Thus by using DLBP and updation between frames are employed the time complexity is reduced.

VI. CONCLUSION

Text embedded in videos often carries the most important information, such as time, place, name or topics, etc. This information may do great help to video indexing and video content understanding. A novel method for text detection and extraction from complex videos is proposed in this paper. Our detection method is based on the observation that there exist contrast colors between text and its adjacent background. The morphological binary map is first generated by obtaining the difference between closing and opening image. Connected components for each candidate region are generated and then each connected component is reshaped to have smooth boundaries. The dominant local binary pattern is used to find the intensity variation around the transition pixel. The boundaries of the detected text regions are localized accurately using the projection of text pixels in the morphological binary map. Text region update between frames is also exploited to reduce the processing time. Based on the results of text detection, the texts are extracted based on color polarity computation method. To validate the performance of our detection and extraction method, various videos have been tested. The proposed method is very useful for the real-time application. Our future work is to detect and extract the text with different orientations to extend the algorithm for more advanced and intelligent applications.

REFERENCES

1. A. Del Bimbo, C. Colombo and P. Pala. "Semantics in visual information retrieval". IEEE Multimedia, 6(3), 1999.
2. C. G. M. Snoek and M. Worring, "Time interval maximum entropy based event indexing in soccer video," in Proc. Int. Conf. Multimedia and Expo, vol. 3, pp. 481–484, Jul. 2003.

3. J.Gllavata,R.Ewerth, and B. Freisleben, "Text detection in images based on unsupervised classification of high-frequency wavelet coefficients,"in Proc. Int. Conf. Pattern Recognition, vol. 1, pp. 425–428 ,Aug. 2004.
4. J. Cho, S. Jeong, and B. Choi, "News video retrieval using automatic indexing of korean closed-caption," Lecture Notes in Computer Science, vol. 2945, pp. 694–703, Aug. 2004.
5. L. Agnihotri and N. Dimitrova, "Text detection for video analysis," in Proc. IEEE Int.Workshop on Content-Based Access of Image and Video Libraries, Jun. 1999, pp. 109–113.
6. K.C.K.Kim,H.R.Byun,Y.J.Song,Y.W.Choi,S.Y.Chi,K.K. Kim, Y. K. Chung "Scene text extraction in natural scene images using hierarchical feature combining and verification," in Proc. Int. Conf. Pattern Recognition, Aug. 2004, vol.2, pp. 679–682.
7. M. R. Lyu, J. Song, and M. Cai, "A comprehensive method for multilingual video text detection, localization, and extraction," IEEE Trans.Circuit and Systems for Video Technology, vol. 15, no. 2, pp. 243–255,Feb. 2005.
8. Xi Jie, Xian-Sheng Hua, Xiang-Rong Chen, Liu Wenyin, HongJiang Zhang, "A Video Text Detection And Recognition System", 2001.
9. M. Bertini, C. Colombo, and A. D. Bimbo, "Automatic caption localization in videos using salient points," in Proc. Int. Conf. Multimedia and Expo, Aug. 2001, pp. 68–71.
10. Y. Zhong, H.J. Zhang, A.K. Jain, Automatic caption localization in compressed video, IEEE Transactions on PAMI 22 (2000) 385–392.
11. N. Otsu, "A threshold selection method from gray-levl histograms,"IEEE Trans. Syst., Man, Cybern., vol. 9, no. 1, pp. 62–66, Mar. 1979.
12. T. Sato, T. Kanade, E. K. Hughes, and M. A. Smith, "Video OCR for digital news archive," in Proc. IEEE International Workshop on Content Based Access of Image and Video Libraries, Jan. 1998, pp. 52–60.
13. Jui-Chen Wu, Jen-Wei Hsieh and Yung-Sheng Chen", Morphology based text line extraction", pp.1195-1200, 2006.
14. Jagath Samarabandu and Xiaoqing Liu," An Edge-based Text Region Extraction Algorithm for Indoor Mobile Robot Navigation", pp.273-280. 2007.
15. S. Liao, Max W. K. Law, and Albert C. S. Chung," "Dominant Local Binary Patterns for Texture Classification", IEEE Transactions on Image Processing, Vol. 18, No. 5, May 2009.
16. T. Ojala, M. Pierikainen, and T. Maenpaa, "Multiresolution gray- scale and rotation invariant texture classification with local binary patterns,"IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 7, pp. 971–987, Jul. 2002.
17. Wonjun Kim and Changick Kim, "A New Approach for Overlay Text Detection and Extraction from Complex Video Scene", IEEE Transactions on Image Processing, Vol. 18, No. 2, February 2009.
18. S. U. Lee, S. Y. Chung, and R. H. Park, "A comparative performance study of several global thresholding techniques for segmentation," Comput. Vis., Graph., Image Process., vol. 52, pp. 171–190, 1990.