AN AUTOMATIC LYRICS RECOGNITION SYSTEM FOR DIGITAL VIDEOS

Hadi Hadizadeh†‡*, Mehrdad Fatourechi*‡*, and Ivan V. Bajić*†

†School of Engineering Science, Simon Fraser University, Burnaby, BC, V5A 1S6, Canada
‡BroadbandTV Corporation, Vancouver, BC, V6E 2P4, Canada

ABSTRACT

We present a system for automatic lyrics recognition in digital video. The system incorporates a text detection and extraction algorithm, a commercial optical character recognition (OCR) engine, as well as a method for text comparison and similarity measurement. The proposed system was tested on a database of 200 lyrics videos of different resolutions and lengths downloaded from YouTube. Experimental results show that the proposed system is able to detect and recognize lyrics in digital videos with a true positive rate of 94% and a false positive rate of 3.8%.

Index Terms— Video analysis, text detection, lyrics recognition, optical character recognition, OCR

1. INTRODUCTION

Thousands of videos are uploaded on the Internet every day, and shared by millions of users through social networking websites such as Facebook and YouTube. For example, over 72 hours of new video are uploaded every minute to YouTube, resulting in about 8 years worth of video content uploaded every day [1]. A significant number of these videos are either illegal copies of existing music videos or display song lyrics on them. This makes copyright protection and management a very sophisticated yet important task for the music industry and individual content owners [2]. Legally, lyrics are considered as "literary work." Hence, if anyone other than the lyrics owner reproduces any part of the lyrics (without the owner's consent), this is considered as copyright infringement [3]. This argument signifies the need for developing fast and high performance video copy-detection and video-based lyrics recognition algorithms. In this paper, our focus will be on detecting videos which contain lyrics on them.

From a technical point of view, a video-based lyrics recognition system is composed of the following three main sub-systems: (1) video text detection and extraction; (2) optical character recognition (OCR); and (3) text comparison or text similarity measurement. Over the past two decades, various methods have been proposed for text detection and recognition in digital images and videos [4, 5, 6, 7]. However, to the best of our knowledge, there is no existing work that directly addresses the problem of lyrics recognition in videos. Hence, in this paper, we present an overall system for automatic lyrics recognition in digital videos. Our experimental results show that the proposed system is able to detect and recognize lyrics with high accuracy (a true positive rate of 94% and a false positive rate of 3.8%) at a relatively low computational cost. The proposed system can be used for several applications such as automatic pirated content detection, content monetization, video retrieval, and advertisement. The current on-going work is mainly dedicated to improving the accuary and the performance of the proposed system, adding a feature to support different languages, and detection of lyrics in different orientations.

The organization of the paper is as follows. We first review some related works in Section 2. Our proposed lyrics recognition system is then presented in Section 3. Experimental results are given in Section 4, followed by conclusions and discussions in Section 5.

2. RELATED WORK

Several methods exist for text detection and recognition in digital images and videos. However, due to the existence of very accurate and robust commercial OCR engines such as TESSERACT [8], many of the existing methods for text detection and recognition address only the text detection and localization problems [4],[7].

In general, existing methods for text detection and localization can be classified into two main groups: region-based and connected component (CC)-based [5]. In region-based methods [4, 5], candidate text regions are detected by low-level image analysis methods such as texture analysis and image filtering. In these methods, a feature vector is first extracted from each local region. The features are then fed into a classifier to detect potential candidate text regions. On the other hand, CC-based methods directly segment candidate text regions by performing edge/corner detection and color clustering [6, 7]. Region-based methods are usually very slow and sensitive to the text orientation, while CC-based methods are faster but more sensitive to false alarms (non-text regions).

This work was supported in part by the NSERC grant EGP 411937.

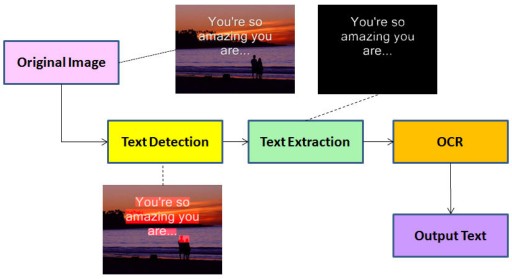
Original Image
You're so amazing you are...
You're so amazing you are...
Text Detection
Text Extraction
OCR
You're so amazing you are...
Output Text

Fig. 1. A block diagram of the proposed system.

3. THE PROPOSED LYRICS RECOGNITION SYSTEM

In our proposed lyrics recognition system called Lyrics Recognition Module (LRM), we utilize a commercial optical character recognition (OCR) engine called TESSERACT [8], and focus on the design and implementation of other modules needed for lyrics recognition. It should be noted that many of the current commercial OCR engines expect their input image to be a binary image in which text regions are already segmented from the background. In other words, they cannot directly be applied to images with complex background, otherwise the performance is degraded significantly [9]. For this purpose, we first apply a text localization and extraction algorithm on the input image as a pre-processing step to create an OCR-ready binary image.

LRM then consists of the following four main modules: (1) text detection and localization; (2) text extraction; (3) OCR; and (4) text comparison or similarity measurement for lyrics recognition. LRM takes a video sequence $\mathbf{V}$ consisting of M video frames $\mathbf{F}_j$ $(j = 1, \cdots, M)$ as input. To reduce the computational complexity, it then sub-samples $\mathbf{V}$ at a sampling frequency f_s to select a smaller set of video frames $S = \{\mathbf{F}_i\}$, where $i = 1, \cdots, N$ and $N << M$. Each of the above four modules is then sequentially applied to every frame in S. In the sequel, we describe the function of each of the above modules. A block diagram of LRM is depicted in Fig. 1.

3.1. Text Detection and Localization

This module detects and localizes potential text regions within an input RGB video frame $\mathbf{F}_i$ $(i = 1, \cdots, N)$ of size $H \times W$ pixels. Motivated by the CC-based text detection methods such as [4], we first compute the edge map of the input frame in each of the three color channels (i.e., $\mathbf{F}_i^R, \mathbf{F}_i^G$, and $\mathbf{F}_i^B$) separately using an appropriate edge detection method such as Canny [10]. We then combine the obtained three edge maps with a logical "OR" operator to get a single edge map $\mathbf{E}_i$ as follows

$$\mathbf{E}_i = edge(\mathbf{F}_i^R) + edge(\mathbf{F}_i^G) + edge(\mathbf{I}_i^B), \tag{1}$$

where $edge(.)$ is the utilized edge detection method, and "$+$" denotes the logical "OR" operator.

The obtained edge map is then processed to obtain an "extended edge map" $\hat{\mathbf{E}}_i$ using a horizontal edge extension algorithm. The extension algorithm starts scanning the input edge map $\mathbf{E}_i$ line by line in a raster-scan order, and connects every two non-zero edge points whose distance is smaller than a specific threshold t_1. Here t_1 is set experimentally to a fraction of the input image width (e.g., $t_1 = 0.04 \times W$). Because the edge density in text-like regions is high, and the vertical edges of characters in a text region are very close to each other (especially in lyrics videos where text is usually aligned along the horizontal direction), different characters in a potential text region can horizontally be connected to each other by the algorithm described above. The horizontal edge extension algorithm can be implemented by a horizontal dilation operator [10] of size $1 \times t_1$ as well.

A connected component analysis [10] is then performed on the extended edge map $\hat{\mathbf{E}}_i$ to find isolated binary objects (blobs or connected components). We then extract the following geometric properties of the obtained blobs: width, height, area, and aspect ratio. To remove potential noisy blobs, those blobs whose geometric properties satisfy one of the following conditions are considered as false alarms and discarded: (1) if a blob's width or height is smaller than a specific threshold t_2 (pixels); (2) if the aspect ratio (width/height) of a blob is smaller than a pre-determined threshold t_3 or larger than a threshold t_4; and (3) if the area of a blob is smaller than a threshold t_5 or larger than a threshold t_6. These thresholds can be set experimentally based on a fraction of width or height of the input image as described in Section 4. After discarding the potential unwanted blobs, a smaller set of candidate blobs is obtained. The bounding boxes of the remaining blobs are then used to localize the obtained candidate text regions, where the bounding box of a blob is the smallest rectangle that encloses the blob completely.

3.2. Text Extraction

We then segment (extract) the text from the background within the bounding box of each candidate text region. The output of this module is a binary image, which is then fed to the OCR module.

We note that characters in a lyrics string usually share the same (or very similar) color content while the background usually contains various colors (possibly very different from the color of characters). Therefore, one can expect to find the pixels of all characters in the input text region in one class, and the background pixels in another. Motivated by this fact, the text segmentation task in our proposed system is performed by a thresholding algorithm, which gets the RGB image within each candidate text region, considers each color pixel as a vector, and clusters all vectors (or pixels) in the given text region into two separate clusters using the K-Means clustering algorithm [11].

To figure out which of the obtained two classes contains the characters of interest, we create two binary images. In the first binary image, we set all pixels that fall in the first class to one, and others to zero. Similarly, in the second binary image, we set all pixels that fall in the second class to one, and others to zero. We then perform a separate connected-component analysis on each of these two binary images, and count the number of valid blobs inside them. We use the same criteria as described in Section 3.1 for finding the valid blobs. Because the background is usually uniform, and has fewer isolated binary objects, the class whose corresponding binary image has more valid blobs is then considered as the class that contains the characters. Using this approach, we can create an OCR-ready binary image, which is then fed to the OCR module.

3.3. Optical Character Recognition (OCR)

In our proposed lyrics recognition system, we used the TESSERACT OCR engine [8] for character recognition, which can be trained with different fonts and languages. The OCR module gets the binary image produced by the text extraction module as its input, and returns the actual text (string) within the image as its output.

3.4. Text Comparison

After extracting the text (string) within a video frame, we can compare it against the lyrics in our database.

Let T_i be the extracted text of the i-th frame $\mathbf{F}_i$, and R be a given lyrics file. In order to find the similarity of T_i to R, we scan R by a moving window of length L_i with a step of one word, where L_i is the length of T_i. Here, we assume that words are separated by one space (all potenital consecuitve spaces are considered as just one space). Let R_j be the text (the lyrics part) that falls within the j-th window over R. The Levenstein distance [12] between T_i and R_j, $LV(T_i, R_j)$, is then calculated. The minimum distance of T_i with respect to R, d_i, is then computed as

$$d_i = \min_j LV(T_i, R_j), \tag{2}$$

where j is taken over all possible overlapping windows of length L_i over R. The above process is repeated for all N frames in S. The final relevance/matching score between the video $\mathbf{V}$ and the lyrics R, d_{total}, is then calculated as the average of the obtained N minimum distances, d_i's ($i = 1, \cdots, N$).

4. EXPERIMENTS

We implemented the proposed lyrics recognition system in MATLAB R2010b, and tested it on a database of 200 video clips downloaded from YouTube along with their corresponding original lyrics. The resolution of the videos were 320 × 240, 640 × 480, and 1240 × 1024. All videos had 30 frames-per-second (fps). The minimum video length in the database was about 2 minutes, and the maximum video length was about 9 minutes. About 50% of the videos clips had the same lyrics (from the same singer) but with different fonts or backgrounds. The sampling frequency f_s was set such that each input video sequence $\mathbf{V}$ was sampled every 100 frames. The values of the thresholds described in Section 3.1 were experimentally set as follows to get maximum detection performance: $t_1 = 0.04 \times W$, $t_2 = 0.08 \times H$, $t_3 = 0.01$, $t_4 = 1$, $t_5 = 100$, and $t_6 = 1000$, where W and H were the frame width and height (in pixels), respectively.

For the purpose of lyrics recognition, we are interested in checking whether the lyrics of interest exist on a given video sequence or not. For this purpose, we need to compare the total similarity/relevance score, d_{total}, of a given video sequence with a specific pre-determined threshold, t_0. In order

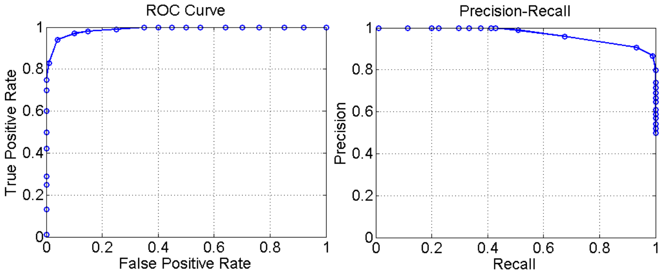

Fig. 2. ROC and Precision-Recall graphs on the test dataset.

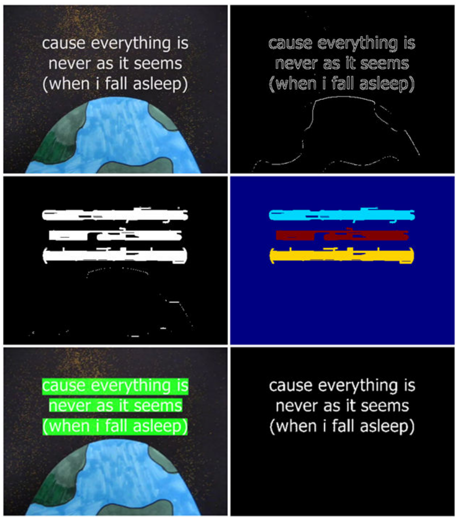

Fig. 3. An example of text detection and extraction. Top row: original image (left), edge map (right); Middle row: extended edge map (left), candidate blobs (right); Bottom row: detected text regions (left), OCR-ready binary image (right).

to obtain t_0, we randomly selected 75% of the videos in the database as our training dataset, and considered the remaining videos as our test dataset. We then computed the total relevance/matching score d_{total} between each video $\mathbf{V}$ and lyrics in the training dataset using the method proposed in Section 3.4. Afterwards, we plotted the precision-recall and ROC (Received Operating Characteristic) curves [13] based on all the computed total similarity scores. To generate these curves, the threshold t_0 was varied over a wide range of different values in the range $[0-50]$, and at each specific threshold, the false positive rate (FPR) and the true positive rate (TPR) were computed. The best threshold t_0 was then experimentally selected as the one whose FPR was the smallest among those thresholds whose TPR were above 90%. This gave $t_0 = 14.3$ with a precision of about 91% and a recall of about 93% on the training dataset. The precision and recall on the test dataset using the obtained t_0 was about 88% and 90%, respectively. The results are shown in Fig. 2. Note that a perfect recognition results in 100% precision and 100% recall. The TPR on the training dataset was about 95% and the FPR was 2.9%. Also, the TPR on the test dataset was about 94% and the FPR was 3.8%. These results confirm the good performance of the proposed lyrics recognition system.

Fig. 3 is an example showing the function of each step in the text localization and extraction modules.

The average processing time of the proposed system for a 640×480 video frame under MATLAB on an Intel Core 2 Duo @ 3.33 GHz, with 8 GB RAM was about 1.2 seconds.

5. CONCLUSIONS

In this paper, we presented a lyrics recognition system for digital video. To the best of our knowledge, this is the first research paper to directly address the problem of lyrics recognition in videos. Experimental results showed that the proposed system is able to recognize lyrics with high performance and accuracy at a relatively low computational cost. Although the system was proposed for digital video, it can also be utilized for lyrics recognition in digital images, as well as for any other application that requires detection of known text in images and video. The proposed system can be used in several applications such as pirated content detection, content monetization, video retrieval, and advertisement. As for future work, we aim at improving the accuarcy of our text localization and extraction modules, adding a feature to detect texts in different languages and orientations, and optimizing the code for speed and performance.

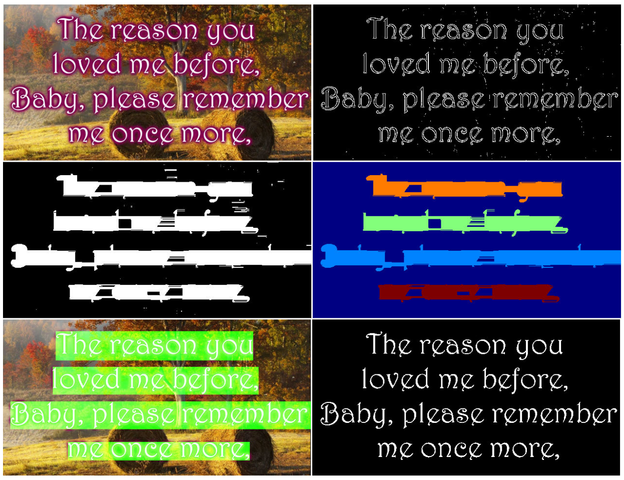

Fig. 4. An example of text detection and extraction. Top row: original image (left), edge map (right); Middle row: extended edge map (left), candidate blobs (right); Bottom row: detected text regions (left), OCR-ready binary image (right).

6. REFERENCES

[1] "Youtube - frequently asked questions," [Online] Available: http://www.youtube.com/faq.

[2] M. Malek Esmaeili, M. Fatourechi, and R. K. Ward, "A robust and fast video copy detection system using content-based fingerprinting," *IEEE Trans. on Info. Foren. and Secu.*, vol. 6, no. 1, pp. 213–226, Mar. 2011.

[3] "Canada copyright and moral rights in works (R.S.C., 1985, c. C-42)," [Online] Available: http://laws-lois.justice.gc.ca/eng/acts/C-42/page-4.html#h-4.

[4] D. Chen, J.-M. Odobez, and H. Bourlard, "Text detection and recognition in images and video frames," *Pattern Recognition*, vol. 37, pp. 595–608, 2004.

[5] Y.-F. Pan, X. Hou, and C.-L. Liu, "A hybrid approach to detect and localize texts in natural scene images," *IEEE Trans. on Image Proc.*, vol. 20, no. 3, pp. 800–813, 2011.

[6] X. Zhao, K.-H. Lin, Y. Fu, Y. Hu, Y. Liu, and T. S. Huang, "Text from corners: A novel approach to detect text and caption in videos," *IEEE Trans. on Image Proc.*, vol. 20, no. 3, pp. 790–799, 2011.

[7] M. R. Lyu, J. Song, and M. Cai, "A comprehensive method for multilingual video text detection, localization, and extraction," *IEEE Trans. on Circuits & Syst. Video Tech.*, vol. 15, no. 2, pp. 243–255, Feb. 2005.

[8] "The Tesseract OCR engine," [Online] Available: http://code.google.com/p/tesseract-ocr/.

[9] R. Lienhart, "Automatic text recognition in digital videos," *Proceedings SPIE, Image and Video Processing IV*, p. 26662675, 1996.

[10] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition)*, Prentice Hall, 2007.

[11] C. Bishop, "K-means clustering," *Pattern Recognition and Machine Learning*, pp. 423–430, 2006.

[12] D. Gusfield, *Algorithms on strings, trees, and sequences: computer science and computational biology*, Cambridge University Press, 1997.

[13] C. D. Manning, P. Raghavan, and H. Schtze, *Introduction to Information Retrieval*, Cambridge University Press, 2008.