

# An Automatic Text Region Positioning Method for the Low-Contrast Image

## Gongqin Liu, Murong Jiang\*, Helin Cun, Zhenzhong Shi, Jianyu Hao

School of Information Science and Engineering, Yunnan University, Kunming, China Email: \*jiangmr@ynu.edu.cn

How to cite this paper: Liu, G.Q., Jiang, M.R., Cun, H.L., Shi, Z.Z. and Hao, J.Y. (2017) An Automatic Text Region Positioning Method for the Low-Contrast Image. *Journal of Computer and Communications*, **5**, 36-49.

https://doi.org/10.4236/jcc.2017.510005

**Received:** June 30, 2017 **Accepted:** August 15, 2017 **Published:** August 18, 2017

Copyright © 2017 by authors and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

http://creativecommons.org/licenses/by/4.0/

## Abstract

Text extraction is the key step in the character recognition; its accuracy highly relies on the location of the text region. In this paper, we propose a new method which can find the text location automatically to solve some regional problems such as incomplete, false position or orientation deviation occurred in the low-contrast image text extraction. Firstly, we make some preprocessing for the original image, including color space transform, contrastlimited adaptive histogram equalization, Sobel edge detector, morphological method and eight neighborhood processing method (ENPM) etc., to provide some results to compare the different methods. Secondly, we use the connected component analysis (CCA) method to get several connected parts and non-connected parts, then use the morphology method and CCA again for the non-connected part to erode some noises, obtain another connected and non-connected parts. Thirdly, we compute the edge feature for all connected areas, combine Support Vector Machine (SVM) to classify the real text region, obtain the text location coordinates. Finally, we use the text region coordinate to extract the block including the text, then binarize, cluster and recognize all text information. At last, we calculate the precision rate and recall rate to evaluate the method for more than 200 images. The experiments show that the method we proposed is robust for low-contrast text images with the variations in font size and font color, different language, gloomy environment, etc.

## **Keywords**

Low-Contrast Image, Text Region Positioning, Connected Component Analysis, SVM

## **1. Introduction**

Text information extraction from images and video is a very important subject

in computer vision, it's widely used in specific applications including page segmentation, address block location, license plate location, etc. Because there are so many possible sources of variation when extracting text from a shaded or textured background, from low-contrast or complex images, or from images having variations in font size, style, color, orientation and alignment. These variations make the problem of automatic text information extraction extremely difficult. Numerous of existing methods have been proposed to detect and recognize text in scene imagery, which can be categorized into edge-based detection, connected component based detection and texture based detection. The connected component based methods assume that the text pixels belonging to the same connected region share some common features such as color or gray intensity [1] [2]; texture based methods may be unsuitable for small fonts and poor contrast text [3]; edge based methods return a lot of false alarms and are not robust to complex background images. In Ref. [4], Wei Fan presented a novel text segmentation method which is independent of variations in text font style, size, intensity, and polarity, and of string orientation with separating the pixels of a document image into four categories: "dark text/lines", "bright text/ lines", "dark figure/graphics" and "white background". But this method is only valid for text embedded in a simple background. Antani et al. [5] assumed that text and background in localized region had consistent gray levels that all characters were either lighter than or darker than the background, so the detection rate may get reduced. Lowell L. Winger [6] used a fast thresholding scheme which could deal with the texture background images better. It was unable to deal with the low-contrast text images with both English and Chinese character, especially when a background of the text images is various, including building, objects, clouds, houses and so on. These methods only do well for the text images with high resolution, huge text or simple background, which not for the low contrast text images, even text in small size or in different color, etc., for appearing a series of problems such as location bias, imperfect or fault, etc., and finally causing poor text recognition results.

In order to solve the problem of text recognition in low-contrast colorful images, we proposed a new method for positioning text region automatically. This method includes the image color space transform, edge detection, image enhancement, morphological, connected component, SVM, etc. In order to evaluate our method efficiently, we collect a new data set with 200 images with various low-contrast, compute their precision rate and recall rate. Experiments show that our method not only can accurately position the text area but also gain good results on the low-contrast images with different sizes and languages.

The paper is organized as follows. In Section 2, four steps for positioning the text region location with image pre-processing, connected component analysis, edge detection and text region merge are introduced in detail. In Section 3, text extraction and recognition results using OCR system are given. In Section 4, some experiments and evaluations are presented. In Section 5, what we have done is

summarized.

## 2. Text Region Positioning

The goal of our approach is to detect low contrast text images without being affected by the language, font color or font size. For the simplify, we assume that text present in images is in the horizontal direction with uniform spacing between words. The processing to locate the text region is divided into four steps: pre-processing, gain the connect component part using CCA method, first and second layer judge to locate the text location coordinates, merging the text regions.

### 2.1. Step 1: Pre-Processing

1) Convert Original Image to YUV Color Space Image

RGB color space is complex in describing the color pattern and has redundant information between each component. Since pixel values in RGB color space are highly correlated, RGB color space is converted into other color spaces. The YUV color model defines a color space in terms of one luminance and two chrominance components. Because of the low-contrast images we choose have similar color between character and background, meanwhile, the luminance information can get a better result than colorful information for further processing, so, we convert the input image (see **Figure 1**) into YUV color space (luminance + chrominance), and use only luminance (Y) channel (see **Figure 2**) for further processing.



Figure 1. Original image.



Figure 2. Y channel image.

2) Enhance the Above Y Channel Image to Increase Image Contrast

Owing to the text images we choose are low-contrast, so, in order to get a better result, we first select some methods to enhance the image, comparing with multi-scale retina algorithm, adjust image intensity, histogram equalization and contrast-limited adaptive histogram equalization, finally finds that contrast-limited adaptive histogram equalization (see Figure 3) can get a good results than other algorithms.

3) Use Sobel Edge Detector Operator to Detect the Above-Enhanced Image

After doing some enhancement for the image, we need to select a kind of edge operator to detect the image edge. In the experiment, we can see Sobel operator [7] [8] (see **Figure 4**) is able to detect all the text edge and meanwhile remove the non-text edge.

4) Use Morphology Method (MM) and Eight Neighborhood Processing Method (ENPM) to Enhance Edge Information

Observing the Sobel detector results, we found the edge density is very weak, so we decide to adopt morphology method [9] and eight neighborhood processing method proposed by us to enhance edge information. About method ENPM, it is that the value in edge map gets by its eight neighborhood value, only if there is one appearance in its neighborhood, its value will be replaced by one. Figure 4 displays Sobel operator detector results, and Figure 5 and Figure 6 respectively show the results processed by morphology method and eight neighborhood processing method. Obviously, edge density is enhanced after using morphology method and eight neighborhood processing method.



**Figure 3.** Enhanced image using contrast-limited adaptive histogram equalization.



Figure 4. Sobel edge detector image.



Figure 5. Image processed by MM.



Figure 6. Image processed by ENPM.

Following is the algorithm about ENPM (see Table 1).

## 2.2. Step 2: Use Connected Component Analysis (CCA) to Gain Connected Components

CCA could be regarded as a graph algorithm, where subsets of connected components are uniquely labeled based on heuristics about feature consensus, *i.e.*, color similarity and spatial layout. In the implementation of CCA, syntactic pattern recognition methods are often used to analyze the spatial and feature consensus, and to define text regions. Considering the complexity of fine turning the syntactic rules, a new trend is to perform CCA with statistical models.

The basic steps of the connected-component text extraction algorithm are given below. 1) Convert the input image to YUV color space (luminance + chrominance), use only luminance channel for further processing; 2) Convert the image gray scale using only Y channel; 3) Compute the edge image for Y channel gray image; 4) Sharpen the edge image by convolving it with sharpening filter; 5) Compute horizontal and vertical projection files considering the sharpened edge image as the input intensity image; 6) Segment the candidate text regions based on adaptive threshold values calculated for vertical and horizontal projections; 7) Perform gap filling to eliminate possible non-text regions [10].

Through the above steps, we can use CCA method to remove part overlap blocks from connect component.

In Figure 7(a), a lot of non-text region or overlap regions are been selected, which is useless and time-consuming for us, so we need to take some measures to remove these overlap blocks. Figure 7(b) shows results after removing part

**Table 1.** The algorithm about ENPM.

#### Input:

*ImgSobelMorph: ImgSobelMorph* is the image, which the source image has been processed by step one to step three and morphology method.

*r*: *r* is the row of the image.

*c: c* is the column of the image.

#### Algorithm:

ImgENP is a two-dimensional array with r rows and c columns, whose value set to be zero.

For i set 2 to r-1

<u>For</u> *j* set 2 to *c*-1

If the sum of *ImgSobelMorph* (*i*-1:*i*+1, *j*-1:*j*+1) is greater than 1

Then ImgENP (i, j) set to be 1

```
End If
```

End For

#### <u>End For</u> Output:

ImgENP: ImgENP is the output image after being processed by ENPM algorithm.



(a) Un-removed overlap image

(b) Removed part overlap blocks

Figure 7. (a) is the test images; (b) is the correspondence image removed part overlap blocks.

overlap blocks. After removing the overlap regions, we need to extract the remaining regions except for the regions gained by CCA.

The type of split overlap region is as follows. In which the yellow color region the first block and the white color region is the second block. According to dual law, there are four types (see **Figure 8**) for overlap which the second block is on the border or inside in the first block.

Figure 9 presents the extracted results from image processed by CCA removed part overlap blocks.

# 2.3. Step 3: Locate the Text Region Coordinates with Two Layer Judge

#### 1) First Layer Judge (FLJ)

Reading 200 text images, and using the above method to gain a lot of connected component regions, by observing the regions difference between text regions and non-text regions, we find that regions whose row or column is less than 22, or the value for row divide column is more than 10 or less than 0.1 generally would be non-text regions. The results see in **Figure 10** and **Figure 11**. In **Figure 9**, there are 36 blocks extracted from the images, while after using first layer judge method, the number of blocks reduced to 14 in **Figure 10**.

## 2) Second Layer Judge (SLJ)

a) Get the Edge Detector Image



Figure 8. Types of overlap we removed.



**Figure 9.** Extracted results from image processed by CCA removed part overlap blocks.



Figure 10. Extracted results after using first layer judge.



Figure 11. The result for used first layer judge.

Edge is the distinct characteristic which can be used to find possible text areas. The text is mainly composed of the strokes in horizontal, vertical, up-right, upleft direction, so it can be considered that the region with higher edge strength in these directions is the text region. In order to get a better edge strength, we made a lot of experiments to see which style of computing would be better.

In **Figure 12** the number a from g shows the block extracted from the Y channel image and a1 from the g1 shows the correspondence binary image.

**Figure 13** gives the edge detection results with four angles. We find that 45 with 135 averages are better than others. For saving time, undoubtedly, we choose these two directional edge maps to represent the density and strength for edge enhancement.

b) Based Edge Map Image to Detection Unsupervised Text Regions

To some extent, the text has weak and irregular texture property, so we can look text as a special texture. We employ the statistical features in the edge maps to capture the texture property. The features are mean, standard deviation, energy, inertia, local homogeneity and correlation of edge maps [11].

After feature computing, we get 5 features form the feature vector representation for each block, then use Support Vector Machine Classifier Algorithm to classify the feature vectors into two clusters: text regions and background regions (see Table 2).

**Figure 14** is the flow chart of an overall process for a specific image to realize automatically text position.

## 2.4. Step 4: Text Regions Merge

Text region merges mainly rely on the location of the text region, finding the neighborhood region, and compute the distance in near neighborhood regions. Then merge with those regions which their distance is small. **Figure 15** shows the start state before merging to end state which had successfully located all the text regions.

## 3. Text Extraction and Recognition

Through the method we proposed in Section 2, we can successfully locate the



Figure 12. The flow chart of region extract and process.

original image-a1 45 angle edge detector 135 angle edge detector original image-c1 45 angle edge detector 135 angle edge detector



**Figure 13.** Sobel edge detector results in different direction. (a1) Results for labeled at a1 block; (c1) Results for labeled at c1 block; (d1) Results for labeled at d1 block; (e1) Results for labeled at e1 block.



Figure 14. Each step result for special image during automatically position text image.

text, then, based on the coordinate of text to extract the text regions. For each text region, first use global threshold method to gain binarization images, then, use means filter to filter the small spot pixels so as to get a smooth image. After doing that, we use OCR system to recognize text, including Chinese or English. **Figure 16** presents the flow chart of recognizing text in low-contrast images in detail.

## **4. Experiment Results**

## 4.1. Text Region Position

Figure 17 shows the experimental results of various kinds of images selected from www image or advertisements. In Figure 17(a), Figure 17(b) and Figure 17(d), Chinese texts with different size in images are successfully detected. Figure 17(c) demonstrates that this approach is effective for the texts in a gloomy environment.



Figure 15. Shows the original text region result and final position results.

**Table 2.** The results of regions show.



Figure 16. Flow chart of recognizing text in low-contrast images.



**Figure 17.** Some experiments result. (a) Middle process result 1; (b) Middle process result 2; (c) Middle process result 3; (d) Middle process result 4; (e) Final text position result about middle process 1; (f) Final text position result about middle process 2; (g) Final text position result about middle process 4.

## 4.2. Evaluation

Here we collect a new dataset with 200 images of various low-contrast images. In addition, in order to evaluate our method, we compute the precision rate and recall rate, meanwhile, makes comparisons with other methods.

The performance of each algorithm has been evaluated based on its precision rate, an average run time obtained. The precision and recall rates are calculated as [12]

Precision Rate = 
$$\frac{\text{Correctly detected}}{\text{Correctly detected} + \text{False positives}} \times 100\%$$
(1)

Table 3	3. Text	location	results.
---------	---------	----------	----------

Method	Test Images	Precision Rate	Recall Rate	Average
	Number	(%)	(%)	Runtime in (s)
Our Method	200	76.5	60.39	27

Table 4. The results of text detection.

Text	Chinese	English
Number of characters	2281	1304
Correctly detected	1924	1271
Missed	357	33
Number of false alarm regions		490
Missed rate (%)	15.6	2.5

Recall Rate =  $\frac{\text{Correctly detected}}{\text{Correctly detected} + \text{False negatives}} \times 100\%$  (2)

We compute the results in **Table 3**. After counting the recognition results, we get the **Table 4**.

## **5.** Conclusion

In this paper, we provide a new method to find the text region in low-contrast image automatically. In the first, we use the only luminance (Y) channel for further processing, and use the contrast-limited adaptive histogram equalization to enhance the image. Then we use the connected component analysis (CCA) method to analysis the location of connected parts to remove inner or border parts so as to reduce the connected parts. Thirdly, we compute the edge feature for all connected parts, and combine Support Vector Machine (SVM) to obtain the real text region. Finally, we merge the text region to extract the block including the text, and use OCR system to recognize all text informations. In order to evaluate our method efficiently, we collect a new data setting with 200 images with various low-contrast, and compute their precision rate and recall rate. Experiments show that our method can not only accurately position the text area but also gain good results on the low-contrast images with different sizes and languages.

## Acknowledgements

This work was supported by Chinese National Natural Science Foundation (No. 11161055) and Program for Innovative Research Team (in Science and Technology) in University of Yunnan Province.

## References

- [1] Jain, A.K. and Yu, B. (1998) Automatic Text Location in Images and Video Frames. *Pattern Recognition*, **31**, 2055-2076.
- [2] Lee, C.W., Jung, K. and Kim, H.J. (2003) Automatic Text Detection and Removal in

Video Sequences. Pattern Recognition Letters, 24, 2607-2623.

- [3] Ye, Q., Huang, Q., Gaoand, W. and Zhao, D. (2005) Fast and Robust Text Detection in Images and Video Frames. *Image and Vision Computing*, **23**, 565-576.
- [4] Fan, W., Sun, J. and Naoi, S. (2015) Separation of Text and Background Regions for High Performance Document Image Compression. *DRR*, San Francisco, 8 February 2015, 94020K.
- [5] Antani, S., Crandall, D. and Kasturi, R. (2000) Robust Extraction of Text in Video. *Proceedings* 15th International Conference on Pattern Recognition, Barcelona, 3-7 September 2000, Vol. 1, 831-834. <u>https://doi.org/10.1109/ICPR.2000.905537</u>
- [6] Winger, L.L., Jernigan, M.E. and Robinson, J.A. (1996) Character Segmentation and Thresholding in Low-Contrast Scene Images. In: *Electronic Imaging: Science and Technology*, International Society for Optics and Photonics, 286-296.
- [7] Sahoo, T. and Pine, S. (2016) Design and Simulation of SOBEL Edge Detection Using MATLAB Simulink.
- [8] Gonzalez, C.I., Melin, P., Castro, J.R., Mendoza, O. and Castillo, O. (2016) An Improved Sobel Edge Detection Method Based on Generalized Type-2 Fuzzy Logic. Soft Computing, 20, 773-784. <u>https://doi.org/10.1007/s00500-014-1541-0</u>
- Hasan, Y.M. and Karam, L.J. (2000) Morphological Text Extraction from Images. *IEEE Transactions on Image Processing*, 9, 1978-1983. https://doi.org/10.1109/83.877220
- [10] Bo, T., Tang, J. and Chan, C.C.K. (2016) Enhanced Blind Modulation Formats Recognition Using Connected Component Analysis with Quadruple Rotation. *Opto Electronics and Communications Conference (OECC) Held Jointly with* 2016 *International Conference on Photonics in Switching (PS)*, Niigata, 3-7 July 2016, 1-3.
- [11] Liu, C., Wang, C. and Dai, R. (2005) Text Detection in Images Based on Unsupervised Classification of Edge-Based Features. *Proceedings of 8th International Conference on Document Analysis and Recognition*, 31 August-1 September 2005, 610-614.
- [12] Das, M.S., Bindhu, B.H. and Govardhan, A. (2012) Evaluation of Text Detection and Localization Methods in Natural Images. *International Journal of Emerging Technology and Advanced Engineering*, 2, 277-282.

Scientific Research Publishing

# Submit or recommend next manuscript to SCIRP and we will provide best service for you:

Accepting pre-submission inquiries through Email, Facebook, LinkedIn, Twitter, etc. A wide selection of journals (inclusive of 9 subjects, more than 200 journals) Providing 24-hour high-quality service User-friendly online submission system Fair and swift peer-review system Efficient typesetting and proofreading procedure Display of the result of downloads and visits, as well as the number of cited articles Maximum dissemination of your research work Submit your manuscript at: http://papersubmission.scirp.org/

Or contact jcc@scirp.org