A Learning-based Text Detection Method in Camera Images

Kai Chen Yi Zhou Chenxuan Li Li Song Xiaokang Yang
Institute of Image communication and Information Processing, Shanghai Jiaotong University, China.

Abstract

This paper proposed a learning-based text detection method in camera images. First, we find 280 pictures of book covers, CD covers and movie posters shot with cameras on Internet. We manually label and extract text regions in them. Second, based on statistical analysis of the difference between text and non-text samples, we get three sets of features which are used to produce weak classifiers. Third, Ada-boost is utilized to select and combine these weak classifiers into two-stage attentional cascade. At last, this two-stage cascade can detect text area in images by classifying sub-regions of images as text and non-text. Compared with previous works, this method is robust in detecting single characters, skewed and even vertical lines.

I. Introduction

Applications of image text detection are useful and significant in many situations, including identifying products by reading text (such as books, movie posters, CD covers), indexing images in digital databases for retrieval and so on. However, Optical Character Recognizer (OCR) can only recognize texts with simple background. So, if text areas in images been detected in advance, it'll be useful for many applications.

Approaches for text detection can be divided into two categories based on feature utilized: region-based and texture-based methods. [1]

Region-based methods utilized the feature that text area has distinct intensity or color compared with background. Jain and Yu [2] used a set of geometry features to classify generated connected-components. Hasan and Karam [3] utilized morphological operations to extract text regions. S. Messelodi and C.M. Modena [4] proposed a cover oriented method which can estimate the skew of text lines. These methods need a lot of experimental threshold and cannot support robust detection.

Texture-based methods assumed that text area has distinct texture apart from background. These methods often involve techniques like Gabor filter, wavelet, FFT, spatial variance and so on. Wu [5] used a multi-scale texture segmentation scheme to segment images which includes 9 second-order Gaussian derivatives. This kind of methods is sensitive to font and size of texts and is hard to select a threshold manually. So, many approaches manipulate machine learning algorithm to do this. Kim [6] used SVM to learn texture feature of text. Chun [7] combined FFT with neural network. They all achieved a much better result after using machine learning algorithm.

The method proposed is a learning-based one, which utilizes texture feature to discriminate text areas from non-text areas. First, we find 280 pictures on Internet, including book covers, CD covers and movie posters shot with cameras. We manually label and extract text regions in them. Positive training samples are from these labeled text regions while negative training samples are selected randomly from the remaining area. Second, based on statistical analysis on these samples, on and off text, we modify and improve two feature sets, namely derivative features [8] and histogram features [9]. By testing these features on training data sets, we get weak classifiers and each classifier corresponds to one feature. Third, Ada-boost is used to select and combine these weak classifiers into a powerful classifier. And we create a two-stage attentional cascade. First stage consists of only derivative features, as they are easy to compute and can reserve almost all text areas while eliminating lots of non-text area. Second stage is histogram features, which is much complicated and computation exhausted. At last, this generated two-stage cascade can detect texts in images by classifying sub-regions of images as text and non-text.

II. Datasets generation

We downloaded 280 pictures from Internet including book covers, CD covers and movie posters shot with cameras, 130 for training dataset and 150 for testing dataset.

(a) (b)

978-1-4244-8728-8/11/$26.00 ©2011 IEEE

(c) (d)

Figure 1: Pictures in dataset. (a)(b) Book covers (c)(d) Posters

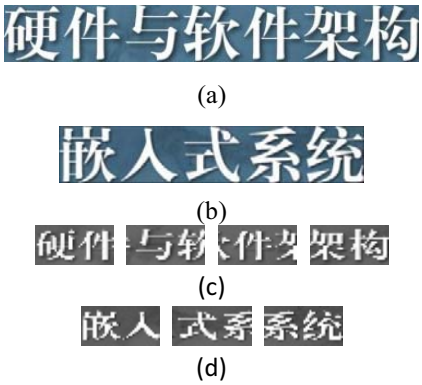


Figure 2: (a)(b)Text regions labeled (c)(d)Positive training samples extracted from labeled text regions

In order to make training and test environment more similar with application scenarios, the selection of these pictures is quite arbitrary. See Figure 1. We manually label and extract text regions in them. From these text regions, positive training samples are generated. Considering what discriminate text from non-text, generation of positive training samples is quite special.

As you can see in Figure 2, the labeled text regions are quite rigid, whose boundaries are tightly around the text. But at the step of training sample generation, the vertical boundaries are enlarged by 1/3 height of the original text regions, with 1/6 height at both upper and lower boundaries. By doing this, it'll be much easier to discriminate text from non-text, which will be explained in detail in next section.

III. Feature extraction

In fact, image text detection is a binary classification problem, classifying windows or areas as text or non-text. In classification problems, features are of vital importance for results' accuracy. We specified two sets of robust features based on statistical analysis of datasets.

A. Histogram Features.

Histogram features are based on this observation: in an image or a sub-region which contains text, there must be a large number of horizontal, vertical or diagonal lines. And in those don't, number of these lines is usually small. [10]

So, we can define patterns like these, which represent vertical, horizontal, vertical, horizontal, and diagonal lines respectively. See Figure 3.

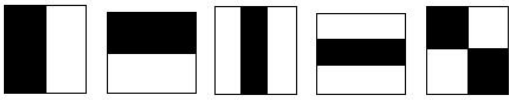


Figure 3: Five patterns

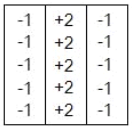


Figure 4: Feature used in test

A histogram feature contains these parameters:

1. Type. Which type of pattern this feature belongs to.
2. Height and width.
3. Value interval. An interval includes an upper threshold and a lower threshold. These thresholds refer to pixel value threshold of filtered image.

Given an input image or an image window $\{x_{ij}\}$ and a predefined feature f_k, a feature value is generated after the following steps:

1. Filter image $\{x_{ij}\}$ with pattern defined in f_k and get another image $\{y_{ij}\}$.
2. Get percentage of pixels of $\{y_{ij}\}$ whose values are between certain intervals. These intervals are also defined in f_k.

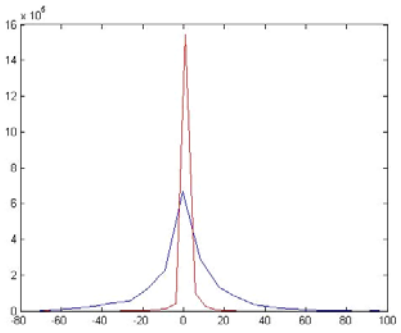


Figure 5: Test results. For non-text images, their responses to this feature are mostly around zero and have low entropy. For text images, their responses are much more scattered

And it's known that the selected features must be informative, which is having low entropy on and off text. We tested feature in Figure 4 on the training samples and got results like Figure 5.

B. Derivative-based Features

The second feature set is from Chen [11], but has been improved. Based on observation, around most text lines, background is rather smooth compared with text area. And there is a vivid

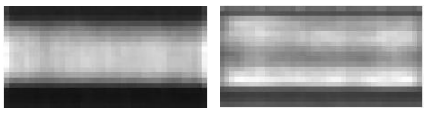


Figure 6: The means of response of x and y derivatives. X derivatives are small at top and bottom while pattern of y derivatives not that obvious

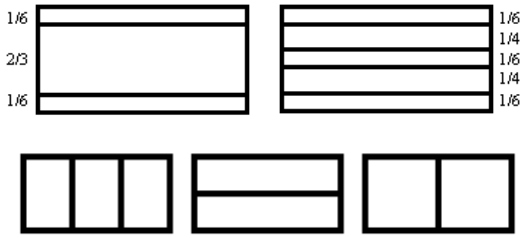


Figure 7: Block patterns (up) and symmetric block patterns (down) used by Chen [11]. Correspondingly, characters in training samples must adapt to these block patterns

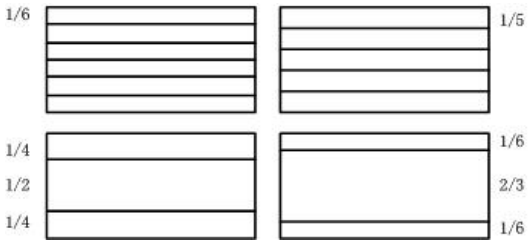


Figure 8: Four of block patterns we use. The height of each sub-block is randomly selected from 1/6 to 2/3.

saying that, in a zonal area, if variance of middle area is large while other area small, in most cases, this zonal area would be a text area. So in last section, we specify a special way of sample generation, which is enlarging both upper and lower boundary by 1/6. And by testing on positive training samples, the average response of x and y derivatives have obvious patterns shown in Figure 6. X derivatives tend to be large in middle of text area while pattern of y derivatives not so obvious. And the variance of x derivatives is large while y derivatives are small.

Chen [11] designed symmetric block patterns for English words, see Figure 7, so that each sub-window contains a character. This put forward high requirements for training samples especially in multi-language circumstances. So, we abandon symmetric block patterns and design a set of inclusive block patterns. See Figure 8. The height of each sub-block is randomly selected from 1/6 to 2/3. With more sub-blocks or features, the possibility of getting a robust powerful classifier increases.

In summary, there are: (1)5200 first class features based on histogram and (2)384 second class features based on x and y derivatives. Therefore, a large amount of features have been specified and Ada-boost can be utilized to generate a powerful classifier.

IV. Text detection

Given a training set of positive and negative samples and a set of features, any machine learning algorithm could be used to train a strong classifier. But Ada-boost's performance on detecting faces [12] has proved that it's the most effective algorithm for detecting target object in images.

First, Ada-boost learning requires a set of training data labeled manually as text or non-text. We use the 280 images labeled ourselves. From this data set, we divide each text window into several overlapping samples with fixed aspect ratio of 2:1 and get 3670 positive samples. The negative samples are extracted randomly from the non-text area of this data set and we get 10012 negative samples. See Figure. 8.

Second, we transform features described in previous section into weak classifiers. A weak classifier $w_i(x)$ usually consists of a feature $f_i(x)$, a threshold t_i and a parity p_i which indicates the direction of the inequality sign:

$$w_i(x) = \begin{cases} 1, p_i f_i(x) < t_i \\ 0, otherwise \end{cases}$$

Here x is a 40×20 pixel sub-window of an image. We selected these weak classifiers with standard Ada-boost learning procedure combined with an attentional cascade [12]. A cascade could drop those sub-windows which are apparently non-text in early stages. This brings a significant boost in processing speed compared with standard Ada-boost algorithm [13]. Our algorithm had 2 cascade layers. The first layer consists of only 5 block-based weak classifiers. The second layer includes only one histogram classifiers, which are much more computational exhausted. By applying generated powerful classifier on image, we can get its text area.

Figure 9: Positive samples used in Ada-boost training. These samples include various type of text which appear in book covers

Figure 10: Testing samples that cannot be correctly classified

V. Experiments

In the test stage, we applied generated powerful classifier to testing samples, including 2177 positives and 2931 negatives. See Table.1.

Table 1. Experiment results

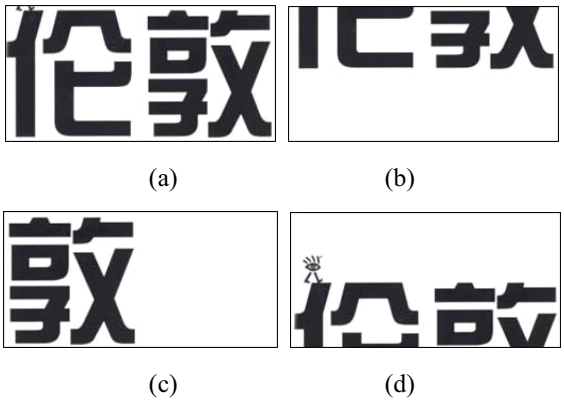
Feature	Positive precision	Negative precision	Weak classifier number	Time
First stage	94.6%	85.0%	5	0.3s
Second stage	91.8%	93.0%	1	7s
Overall	90.7%	95.2%	6	3s

The resolution of testing samples is 1024*768. First we tested two stages separately. The first stage, derivative-based feature, reserves most text areas and eliminate most non-text areas with only 0.3s. The second stage runs rather slow with 7s an image, but much more accurate when detecting text.

Then, we apply the powerful classifier generated to all our datasets and get the result like Figure 9 and Figure 10. The results show that the powerful classifier behaves well when locating single characters and vertical lines, though there are some false positives and negatives. And this thanks to two feature sets we adopt.

The first stage, including 5 weak classifiers from derivative feature, is fast and tries to retain all text area in images. The remaining 5.4% text areas are listed below, which are either blur or similar to background.

It's all for the second stage that the powerful classifier can detect single characters and vertical lines. This feature is anti-rotating, for it counts number of the horizontal, vertical and diagonal lines in a small window. If there are sufficient lines in a small window, whatever it contains, single character or vertical text lines, this window shall be classified as text region.



(a) (b)

(c) (d)

Figure 11: (a) A window whose feature value is 2T. (b)(c)(d) The windows adjacent that would be classified as text areas too

But based on experiments, we discovered a shortcoming of histogram feature. After training on training samples, each histogram weak classifier will get a threshold T representing percentage of specified lines. If the feature value of a window is 2T, then the adjacent windows would be classified as text areas too. See Figure 11. So, you would find out that the boundaries around detected text area are quite loose. See Figure 12~14.

VI. Conclusion & Discussion

This paper presents a learning-based text detection method in camera images. By combining derivative and histogram feature sets, we get an effective attentional cascade. Compared with other researchers' work, such as Chen's [11], this classifier is robust to single characters and vertical lines.

Due to histogram feature's computational exhaustive and relative high false-positive detection rate, there should be more kinds of easy and simple features in a cascade containing histogram feature. So our future work is to find out other useful statistical feature sets, which could be very helpful in overcoming the short points of histogram feature.

Acknowledgment

The work is supported by the National Grand Fundamental Research 973 Program of China with grant No.2010CB731406, 2010CB731401.

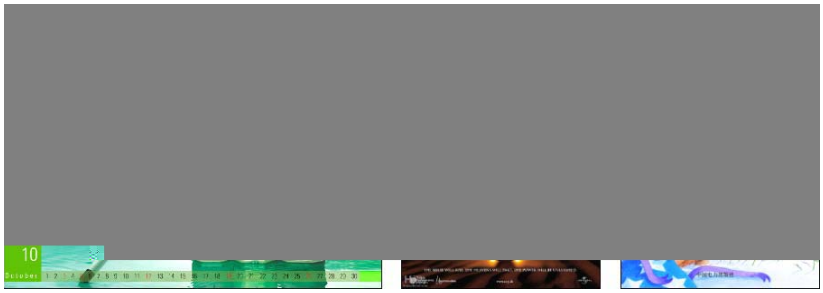


Figure 12: Original images

Figure 13: Detected regions

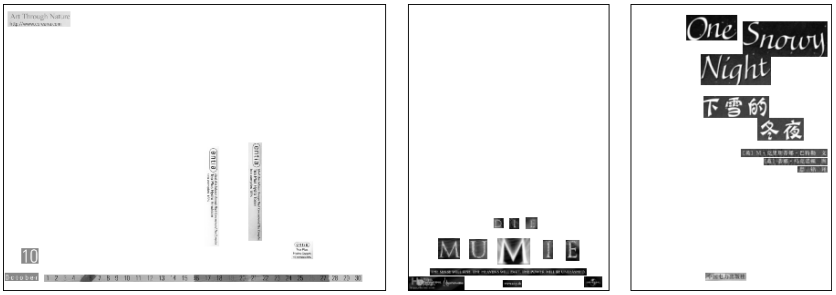


Figure 14: Ground-truth

Reference

[1] K. Jung, K. I. Kim, A. K. Jain, Text information extraction in images and videos: A survey, Pattern Recognition 37 (2004) 977–997

[2] A. K. Jain, B. Yu, Automatic Text Location in Images and Video Frames, Pattern Recognition 31 (12) (1998) 2055–2076.

[3] Y.M.Y. Hasan, L.J. Karam, Morphological text extraction from images, IEEE Trans. Image Process. 9 (11) (2000) 1978–1983.

[4] S. Messelodi, C.M. Modena, Automatic identification and skew estimation of text lines in real scene images, Pattern Recognition. 32 (1992) 791–810.

[5] V. Wu, R. Manmatha, E.M. Riseman, TextFinder: an automatic system to detect and recognize text in images, IEEE Trans. Pattern Anal. Mach. Intell. 21 (11) (1999) 1224–1229.

[6] K.I. Kim, K. Jung, S.H. Park, H.J. Kim, Support vector machine-based text detection in digital video, Pattern Recognition 34 (2) (2001) 527–529.

[7] B.T. Chun, Y. Bae, T.Y. Kim, Automatic text extraction in digital videos using FFT and neural network, Proceedings of IEEE International Fuzzy Systems Conference, Vol. 2, Seoul, South Korea, 1999, pp. 1112–1115.

[8] X. R. Chen, A. L. Yuille, Detecting and reading text in natural scene. Proceeding of CVPR'04.

[9] C. Li, X. G. Ding, Y. S. Wu, An Algorithm for Text Location in Images Based on Histogram Features and Ada-boost, Journal of Image and Graphics, 2006.

[10] C. Li, X. G. Ding, Y. S. Wu, An Algorithm for Text Location in Images Based on Histogram Features and Ada-boost, Journal of Image and Graphics, 2006.

[11] X. R. Chen, A. L. Yuille, Detecting and reading text in natural scene. Proceeding of CVPR'04.

[12] P. Viola, M. Jones, Rapid Object Detection using a Boosted Cascade of Simple Features, CVPR'01.

[13] P. Viola, M. Jones, Fast and Robust Classification using Asymmetric Ada-boost and a detector cascade. Advances in Neural Information Processing Systems, 2002