2011 International Conference on Document Analysis and Recognition

Text Localization in Web Images Using Probabilistic Candidate Selection Model

Liangji Situ
School of Computing
National University of Singapore
Singapore
situ04@comp.nus.edu.sg

Ruizhe Liu
School of Computing
National University of Singapore
Singapore
liurz@comp.nus.edu.sg

Chew Lim Tan
School of Computing
National University of Singapore
Singapore
tancl@comp.nus.edu.sg

***Abstract*— Web has become increasingly oriented to multimedia content. Most information on the web is conveyed from images. Text localization in web image plays an important role in web image information extraction and retrieval. Current works on text localization in web images assume that text regions are in homogenous color and high contrast. Hence, the approaches may fail when text regions are in multi-color or imposed in complex background. In this paper, we propose a text extraction algorithm from web images based on the probabilistic candidate selection model. The model firstly segments text region candidates from input images using wavelet, Gaussian mixture model (GMM) and triangulation. The likelihood of a candidate region containing text is then learnt using a Bayesian probabilistic model from two features, namely, histogram of oriented gradient (HOG) and local binary pattern histogram Fourier feature (LBP-HF). Finally best candidate regions are integrated to form text regions. The algorithm is evaluated using 155 non-homogenous web images containing around 600 text regions. The results show that the proposed model is able to extract text regions from non-homogenous images effectively.**

Keywords: text extraction; text localization; web image

I. INTRODUCTION

Internet has become one of the most important information sources in our daily life. As network technology advances, multimedia contents such as images, contribute a much heavier proportion than before. Survey by Petrie et al. [1] shows that among 100 homepages from 10 websites, there are average 63 images per homepages. Text, as a high-level semantic feature in image, contains useful information of the image contents, and thus the contents of the web. Therefore, text localization in web images plays an important role in web image information extraction and retrieval. The problem of text extraction has been addressed under different contexts in the literature, such as natural scene images [2,12], document images and videos [11]. However, web image exhibits different characteristics comparing to other types of images, such as natural scene image and document image. A web image normally has only hundreds of pixels and low resolution [3]. On the other hand, although frames in video suffer the same problem of low resolution and blurring, text localization in videos can utilize the temporal information. However, this information is inherently absent in web images. Therefore, the current approaches for text extraction on general images and videos cannot be directly applied to web images.

Current works on text extraction in web images were mainly based on clustering color information to segment the original image coarsely and then apply connected component analysis to extract text strings [4-6]. These approaches generally assumed that text regions are in homogenous color and high contrast. Hence, the approaches may fail when text regions are in multi-color or imposed in complex background. The first work attempting to extract texts from non-homogeneous color web images is proposed by Karatzas et al. [3]. They present two segmentation approaches to extract text in non-uniform color and more complex situations. However, their experimental datasets consist of only a minor proportion (29 images) of non-homogeneous images, which is not able to reflect the true nature of the problem.

In this paper, we propose a text localization algorithm based on the probabilistic candidate selection model for multi-color and complex web images. Our algorithm firstly segments text region candidates from input images using wavelet and Gaussian mixture model (GMM). It then groups nearby regions by triangulation and generate segmented text candidate regions. Two features are computed from each candidate region. One is the histogram of oriented gradient (HOG), proposed by Dalal et al. [8]. The other is local binary pattern histogram Fourier feature (LBP-HF), proposed by Ahonen et al. [9]. The likelihood of a candidate region containing text is then learnt using a naïve Bayes probabilistic model. Finally we integrate best candidate regions to form text regions. Our algorithm is evaluated using 155 non-homogenous web images.

The paper is organized as follows. Section 2 introduces the probabilistic candidate selection model. Section 3 elaborates our algorithm in details. Section 4 presents the evaluation methods and experimental results. Discussion and comparison with other contexts on text localization are also illustrated in this section. Section 5 concludes our work and proposes future research directions.

II. PROBABILISTIC CANDIDATE SELECTION MODEL

The proposed model is basically a divide-and-conquer approach. Instead of answering where the text regions locate, we divide the image into candidate regions and decide the likelihood of each region being text. Then the best candidate regions are select and integrated as the final results according

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.273

to the probability. (Fig. 1) In this way, the harder question "where" is transformed to many easier "yes-no" questions.

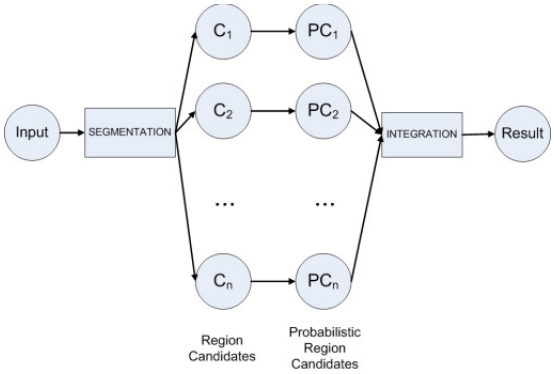


Figure 1. The probabilistic candidate selection model

A text localization algorithm is constructed based on the model (Fig. 1). Specifically, the algorithm firstly generates region candidates (Fig. 1 C_i) from input image, using wavelet, Gaussian mixture model (GMM) and triangulation. Two features are computed from each region candidate. One is the histogram of oriented gradient (HOG) [8]. The other is local binary pattern histogram Fourier feature (LBP-HF) [9]. The likelihood of a region candidate (Fig. 1 PC_i) containing text is then learnt using a naïve Bayes probabilistic model. Finally we integrate best candidate regions to be text regions. We shall give detail implementations of the algorithm in the next section.

III. ALGORITHM IMPLEMENTATION

In this section, we describe the text localization algorithm in three parts: region segmentation, probability learning and probability integration.

3.1 Region Segmentation

3.1.1 Wavelet Quantization and GMM Segmentation

The input color image is firstly quantized in gray scale and decomposed into several channels, in order to separate pixels with large different intensity values. The quantization is achieved by reconstructing the approximate coefficients from 2D wavelet decomposition. The continuous intensity histogram will be discretized into several pikes, where each pike represents certain intensity channel. (Fig. 2c) Thus one input image is decomposed into four channel images. (Fig. 2d)

Each channel image is further segmented into regions using Gaussian mixture model (GMM), based on the position and RGB intensity values. (Fig. 3) The GMM model is learnt with the Expectation Maximization (EM) algorithm and we use a boosting method to find the optimized number of Gaussian kernels. In this way, a multicolor input web image will be decomposed into regions that distribute sparsely in different intensity channels, where pixels in the same region of the same channels have similar intensity values and distribute spatially nearby.

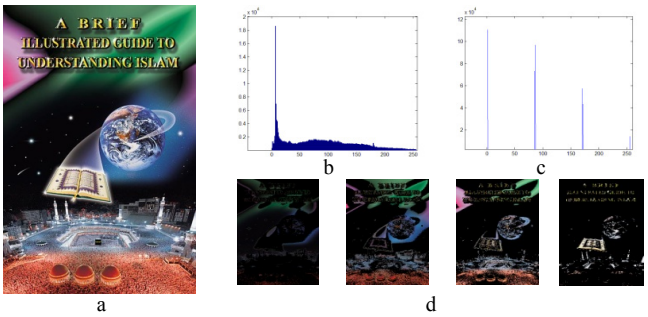


Figure 2. Wavelet Quantization: (a) Sample input image (b) Histogram of continuous intensity values (c) histogram of discretized intensity values (d) The sample input image in four channels separately.

3.1.2 Triangulation

The regions obtained from GMM segmentation are piecewise and contain both text and non-text regions. Hence, we group the neighboring regions together by using the Delaunay triangulation [7]. In theory, the extrema points of a region and the smallest distances between these extrema points of two regions are the best way to represent the relationship of two regions. However, in real practice, this only complicates the procedure but cannot gain better performance. The reason is that this kind of representation is sensitive to region size and shape. Thus, in the implementation, we use centroid to represent a region that is clustered into two sets based on area. Specifically, we assign the regions with area less than 20 pixels to the small area region set. Otherwise, they are assigned to the big area region set. Two Delaunay triangulation graph are built on these two area region sets respectively (Fig. 4b). In one triangulation graph, each node represents a centroid and two adjacent nodes are connected by an edge. The length of an edge is the Euclidean distance value of the two connected nodes. Then in the graph formed in the small area region set, we remove the edges with lengths longer than 25 if the distance of two connected nodes in x-axis is less than 5, otherwise, we remove the edges with lengths longer than 10. Similarly, in the graph formed in the big area region set, we remove the edge with a larger threshold of 70 if the distance of two connected nodes in x-axis is less than 15, or the edges are removed with the length longer than 20. After removing the long edges, the two graphs are segmented into many subgraphs and we consider each subgraph as a text candidate region for future probabilistic learning.

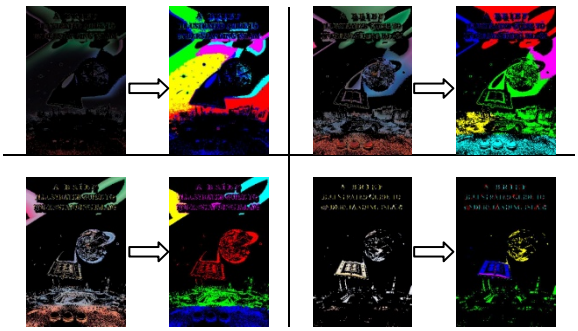
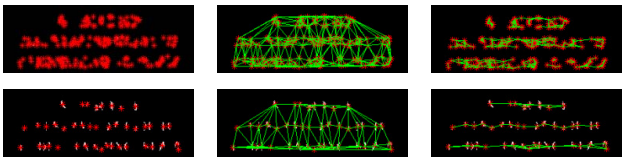


Figure 3. GMM segmentation results for four channels in Fig. 1d.



a b c

Figure 4. Triangulation on two area region sets. The first row is the small area region set; the second row is the big area region set: (a) Red dots represent the centroids of the regions. (b) Delaunay triangulation graph formed by the centroids. (c) Sub-graphs are built after removing the long edges.

3.2 Probabilty Learning

For each candidate region obtained from section 3.1, its likelihood of being a text region is learnt based on the features extracted from these regions. Based on the observation that text is usually geometrically constrained and has regular oriented contours, we select two features to represent the pattern of text, namely, the Histogram of Oriented Gradient (HOG) [8] and Local Binary Pattern Histogram Fourier Feature (LBP-HF) [9].

HOG captures the local shape of an image region by distributing edge orientations into K quantized bins within the image region. The contribution of each edge is weighted according to its magnitude. HOG has been widely accepted as one of the best feature to capture the edge or local shape information. In our implementation, we compute a HOG vector with 8 bins in each image region. However, shape features alone are not sufficient to distinguish all text regions from other text-shape-like graphics in web images, such as synthetic logos, leaves and ladder. Thus, we need another complementary feature to remove these noise patterns.

On the other hand, we observe that text normally appears in groups, i.e. in words or sentences. Local Binary Patterns (LBP) [10] is a powerful texture descriptor and is able to capture this distinguish characteristic of text. In implementation, we adopt LBP-HF [9] as the complementary feature. It is a rotation invariant image descriptor based on uniform Local Binary Patterns (LBP). The LBP feature that takes n sample points with radius r with center value n_c and the corresponding n neighbor points n_i is defined in (1).

$$LBP_{n,r} = \sum_i^{n-1} s(n_i - n_c)2^i \quad (1)$$

where $s(x)$ is 1 if $x \geq 0$ and 0 otherwise. A local binary pattern is called uniform, denoted by LBP^u, if it contains at most "u" 0-1 transitions as predefined. For example, the pattern 0010010 is a non-uniform pattern for LBP^2 but is a uniform pattern for LBP^4. Then the LBP-HF descriptor is form by first computing a non-invariant LBP histogram over the whole region and then constructing rotationally invariant features from the histogram. Specifically, we denote a specific uniform LBP by $U_P(n,r)$ and then LBP-HF is defined as

$$LBP^{u2} - HF(n_1, n_2, u) = H(n_1, u)\overline{H(n_2, u)} \quad (2)$$

where $H(n,\cdot)$ is the Discrete Fourier Transform of nth row of the histogram $h_I(U_P(n,r))$, i.e.

$$H(n,u) = \sum_{r=0}^{P-1} h_I(U_P(n,r))e^{-2\pi ur/P} \quad (3)$$

and $\overline{H}(n_2,u)$ denotes the complex conjugate of $H(n_2,u)$.

These two features are extracted from each of the candidate region respectively and then concatenated linearly with equal weights into a single feature vector. The principal component analysis is then applied to reduce the dimensions of the feature vector. Finally, the integrated HOG and LBP-HF feature comparison of text region and non-text region is illustrated in Fig. 5.

The extracted feature vector is fed into the naïve Bayes. The probability of the candidate region being text is then learnt from the model.

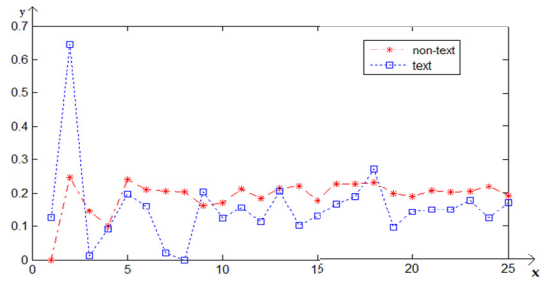


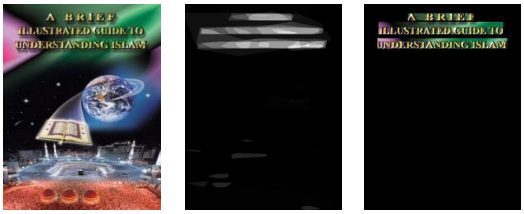
Figure 5. The integrated HOG and LBP-HF feature comparison of text and non-text. The x-axis represents the dimensions of the integrated feature vector; the y-axis represents the value of feature vector in each dimension.

3.3 Probabilty Integration

From the probability learning in section 3.2, we have obtained each candidate region with a likelihood of being text. Then each candidate region is broken into pixels. Normally, each pixel should have the same probability of being text within the same region. However, as the candidate regions are grouped together in different channels, the position of the candidate regions in the original image may overlay. Thus, a pixel may belong to more than one candidate region. Therefore, we have to integrate the probability of being text for all pixels in all candidate regions from different channels.

Let p be the pixel in image, $R = \{r_1, r_2, \ldots r_n\}$be the set of the candidate regions r_i that p belongs to; we define the probability of being text for p in (4).

$$P(p) = \begin{cases} 0, & R = \emptyset \\ \sum_i^n P(r_i), & r_i \in R \end{cases} \quad (4)$$



a b c

Figure 6. Probability Integration result: (a) the input image (b) Probability integration result of input image (c) The final extraction result.

The probability integration result is shown in Fig. 6b. From Fig. 6b, we can observe that the probability integration provides fuzzy value for each candidate region being text.

IV. EXPERIMENTS

4.1 Evaluation

The evaluation method follows the evaluation criteria of ICDAR 2003 robust reading competitions [2]. We denote $\boldsymbol{E}$ as the set of the estimate text rectangles, $\boldsymbol{T}$ as the set of text rectangles from ground truth. Then we define the area match m_a between two rectangles r_1 and r_2 as twice the area of intersection divided by the sum of the areas of each rectangle i.e.:

$$m_a(r_1 + r_2) = \frac{2a(r_1 \cap r_2)}{a(r_1) + a(r_2)}$$

Where $a(r)$is the area of rectangle r. m_a has the value one for identical rectangles and zero for rectangles that have no intersection. For each rectangle in the set of estimates we find the closest match in the set of ground truth, and vice versa. Hence, the best match $m(r, R)$for a rectangle r in a set of rectangles $\boldsymbol{R}$ is defined as:

$$m(r, R) = \max(m_a(r, r')), \qquad \forall\, r' \in R$$

Then the precision p and the recall r are defined as follows respectively:

$$p = \frac{\sum_{r_e \in E} m(r_e, T)}{|E|}$$

$$r = \frac{\sum_{r_t \in T} m(r_t, E)}{|T|}$$

Finally, we adopt the standard f measure to combine the precision and recall figures into a single measure of quality. The relative weights of p and r are controlled by a parameter α, which we set to 0.5 to give equal weight to precision and recall:

$$f = \frac{1}{\alpha/p + (1-\alpha)/r}$$

4.2 Experiments

The training and test datasets consist of web images crawled from Internet, including headers, banners, book covers, album covers and etc. All images are full-color and vary in size from 105×105 to 1005×994 pixels with 96 dpi on average. All texts are contained in non-homogenous background. The texts vary greatly in font styles, sizes, colors and appearance. 262 text images are used as training data. Specifically, the text bounding boxes of these 262 train images are extracted manually to train the Bayesian network model. Then another 155 images are used to evaluate the performance of our algorithm. The text regions of the ground truth are manually tagged in advance. However, the output of our algorithm is fuzzy values of regions being text. In order to meet the requirement of evaluation method in section 4.1, we learn the threshold of being text empirically from the developing data set to extract the bounding boxes of the text regions in the original image. Finally, the experimental result of our algorithm is compared with the ground truth with respect to f-measure discussed in section 4.1 (Table. 1).

Some sample results are shown in Fig. 7 below. Note that the original input images are shown at the first row and the text regions located at the second row.

Table 1. Evaluation of the proposed algorithm

Algorithm	Precision	Recall	f
Our algorithm	0.56	0.61	0.58
Algorithm in [11]	0.48	0.55	0.51

Figure 7. Sample results of our algorithm and the algorithm proposed in [11]. The first row is the original images and the second row is the extracted results of our algorithm. The third row is the extracted results using algorithm in [11].

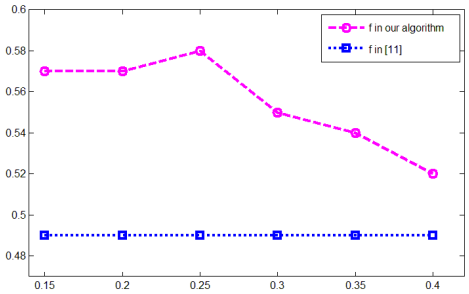


Figure 8. f-measure comparison between our algorithm with different probability thresholds and the algorithm in [11].

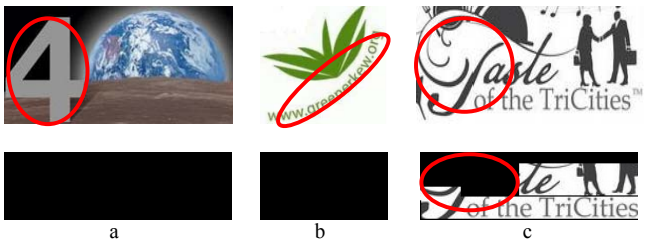
4.3 Discussion

Fig. 7 shows that our algorithm can achieve acceptable performance on text extraction of web images with text in multi-color and complex background. We are even able to extract very small size fonts and exclude the text-like graphics.

As few existing algorithm on text extraction for web images aims to address the problem of text localization on multi-color and complex web images, we choose a similar algorithm that detects text in video frames for comparison [11]. From Table 1, we can observe that our algorithm achieves competitive performance with the algorithm proposed in [11]. From Fig. 7, we show that our algorithm (Row 2 of Fig. 7) outperforms the algorithm in [11] (Row 3 of Fig. 7) for distinguishing text patterns and non-text patterns. Furthermore, our algorithm returns a probability of being text for each candidate region. This fuzzy classification can provide more information for final text region integration and future extension, while algorithm in [11] only achieves a simple binary classification (Fig. 8). Another similar context for text localization is done in natural scene images. The latest published algorithm [12] estimates the local width of stroke features in the image to identify text regions on the basis of local homogeneity of the stroke width. It is important to note that the local stroke width is only well defined when the text is resolved clearly enough. However, web images usually contain small size fonts and low resolution text regions. This inherent character makes the algorithm in [12] incompatible to address the problem of text localization in web images. As our algorithm is evaluated on a different data set from that in [12], the evaluation results cannot be compared directly.

In Fig. 9 we present typical cases where text was not detected. For example, a single character is hard to identify because little text pattern information can be captured in this region (Fig. 9a). If the text is aligned curly (Fig. 9b) or with an excessive fancy style (Fig. 9c), the detection rate is low because these text pattern information is limited in our training data.

The experimental results show that our algorithm can achieve competitive performance on text localization with high complex web images. The comparison with other contexts on text localization illustrates that our algorithm captures the essence challenge of web images and present an effective approach to address this problem.



a b c

Figure 9. Examples of failure cases. The first row is the input images; the second row is extracted results. These include: single character appearance (a), text with curvature beyond our range (b) and text with excessive fancy style (c).

V. CONCLUSION

In this work we propose a probability candidate selection model to locate the text regions in web images. Unlike the existing approaches that only aims to extract the text regions with homogeneous color and high contrast, our proposed algorithm is able to handle more complex situation. In this situation, text is non-uniform color and imposed in complex background. First, we use the wavelet and GMM to segment the input color image into regions coarsely, and then apply triangulation to produce text candidate regions. Then HOG and LBP-HF features computed in each candidate text region are fed to a naïve Bayes model. Each candidate region is assigned the likelihood of being text in probability learning procedure. Finally, we select best candidate regions to be text regions based on probability. Our algorithm is evaluated with a standard evaluation criteria and the experimental result shows that our algorithm is able to locate the text regions in non-homogenous web images effectively.

There are several possible extensions for this work. The grouping of neighbor regions should consider the curly text alignment in order to improve the final text detection rate. Furthermore, we may integrate a more powerful feature set to improve the performance of text pattern recognition. On the other hand, although we can locate text regions in web images correctly, the located text regions may be too blurred to extract the characters effectively. Thus we may explore a super-resolution approach to enhance the text regions. We intend to explore these directions in future.

REFERENCES

[1] H. Petrie, C. Harrison, S. Dev, "Describing images on the Web: a survey of current practice and prospects for the future," In Proceedings of Human Computer Interaction International (HCII) 2005, July 2005.

[2] S. M. Lucas and et al., "ICDAR 2003 robust reading competitions: entries, results, and future directions", International Journal on Document Analysis and Recognition (IJDAR), 7:105-122, 2005, doi: 10.10-7/s10032-004-0134-3.

[3] D. Karatzas, A. Antonacopoulos, "Colour text segmentation in web images based on human perception," Image and Vision Computing, 25(5), pp. 564-577, 2007, doi: 10.1016/j.imavis.2006.05.003.

[4] D. Lopresti, J. Zhou, " locating and recognizing text in www images," Inf. Retrieval 2 (2000), pp 177-206.

[5] A. Antonacopoulos and F. Delporte, "Automated interpretation of visual representations: extracting textual information from www images," in: R. Paton, I. Neilson(Eds.), Visual Representations and Interpretations, Springer, London, 1999.

[6] A. K. Jain and B. Yu, "Automatic text location in images and video frames," Pattern Recognition. 31 (12) (1998) 2055-2076.

[7] M. De Berg, M. van Kreveld, M. Overmars, O. Schwarzkopf,: Computational Geometry. Springer, Heidelberg (2000).

[8] N. Dalal and B. Triggs. Histogram of oriented gradients for human detection. In CVPR 2005,volume 1, pages 886-893,2005.

[9] T. Ahonen, J. Matas, C. He & M. Pietikäinen, "Rotation invariant image description with local binary pattern histogram fourier features," Proc. 16th Scandinavian Conference on Image Analysis (SCIA 2009), Oslo, Norway.

[10] T. Ojala, M. Pietikäinen, T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with Local Binary Patterns," IEEE Transactions on Pattern Analysis and Machine Intelligence 24(7):971-987, 2002.

[11] Shivakumara P, T Q Phan and C. L. Tan, "New Fourier-statistical features in RGB space for video text detection, " IEEE Transactions on Circuits and Systems for Video Technology, Vol.20, pp.1520-1532, November 2010.

[12] Boris Epshtein, Eyal Ofek, Yonatan Wexler, "Detecting text in natural scenes with stroke width transform," 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp.2963-2970 , 2010.