TEXT DETECTION IN BORN-DIGITAL IMAGES USING MULTIPLE LAYER IMAGES

Chao Zeng, Wenjing Jia and Xiangjian He

Research Centre for Innovation in IT Services and Applications (iNEXT)
University of Technology, Sydney, Australia

ABSTRACT

In this paper, a new framework for detecting text from webpage and email images is presented. The original image is split into multiple layer images based on the maximum gradient difference (MGD) values to detect text with both strong and weak contrasts. Connected component processing and text detection are performed in each layer image. A novel texture descriptor named T-LBP, is proposed to further filter out non-text candidates with a trained SVM classifier. The ICDAR 2011 born-digital image dataset is used to evaluate and demonstrate the performance of the proposed method. Following the same performance evaluation criteria, the proposed method outperforms the winner algorithm of the ICDAR 2011 Robust Reading Competition Challenge 1.

Index Terms— Multiple layer image, T-LBP, maximum gradient difference, text detection

1. INTRODUCTION

A large amount of text information has appeared in the form of digital images on web pages and emails. Texts in images on web pages usually have the function of advertising, decoration and highlighting, while those images containing texts embedded in emails often convey the content of product promotion. Especially, the text content embedded in images of spam emails can not be recognised by text-based methods and those spam emails can not be filtered out by anti-spam softwares. So, text detection of webpage and email images plays an important role in retrieving textual information on web pages and detecting spam emails. Because texts need to be detected and segmented from images before being recognised, the research on text detection has attracted much attention and many text detection algorithms have been reported.

Due to the fact that text regions often contain highly dense edges, edge information has been widely used for detecting text from images. In [1], binary Canny edge information of an image was computed to obtain the stroke width information of characters. In [2], binary edge maps of gray scale images were combined with colour information for text detection. Chen et al. [3] and Anthimopoulos et al. [4] performed morphological operations on binary edge maps to generate connected components of text lines. Binary edge detection can generate binarised single-pixel-width edges of a text line. However, the edges of non-text background regions can also be detected. When such background is complex, binary edge-based approaches may produce a large connected component that mixes up the text line and the background. This will bring difficulties in the later process of an accurate text detection algorithm.

Instead of using binary edges, the edge strength information is considered in our work. The novelty of the proposed framework is that a multiple layer image scheme is used for detecting text lines with strong and weak edge strength. Based on the idea in [5], we further develop the maximum gradient difference (MGD) to better describe the edge strength of an image for finding the possible text lines. In [6], it was pointed out that text regions usually have larger MGD values than non-text regions. Hence, in our method, the edge information of text regions is extracted based on clustering MGD values. Multiple layer images are generated for detecting text lines with strong and weak MGD responses. In order to distinguish text from non-text regions in fine detection, we propose a variant of local binary pattern (LBP), named T-LBP, to depict the characteristics of text in the viewpoint of treating text as a kind of texture and use a supervised learning scheme to remove the false alarms generated in the coarse detection step. The flow chart of the proposed method is illustrated in Figure 1.

The remaining of the paper is organised as follows. We first present multiple layer image generation in Section 2. Then, in Section 3, the details of text detection in each layer image using the proposed T-LBP and the training procedure are presented. Section 4 provides the criterion of integration of bounding boxes. Experimental results are shown in Section 5. Finally, the paper is concluded in Section 6.

2. MULTIPLE LAYER IMAGE GENERATION

In this step, an input colour image is firstly converted into a gray scale image for computing MGD values for every pixel to generate the MGD map. The most critical procedure in this step is to obtain the connected component (CC) clusters that include the possible CCs of text lines. In order to keep the text lines with both strong and weak MGD responses, instead of simply separate the MGD map into two CC cluster maps, we

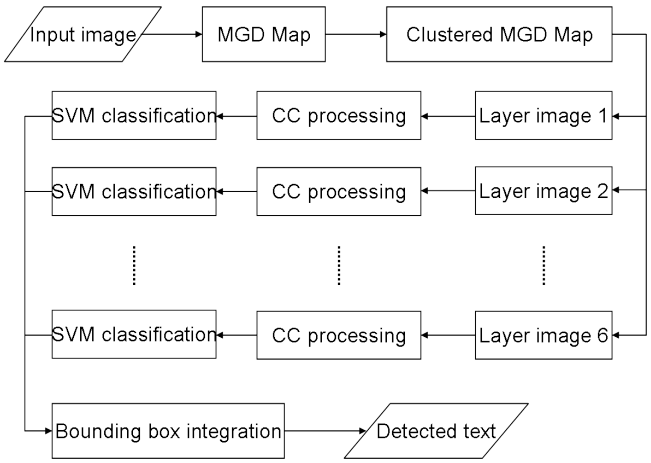


Fig. 1. The flow chart of the proposed method.

split the MGD map into several CC cluster maps. Different combinations of these CC cluster maps produce multiple layer images for further process.

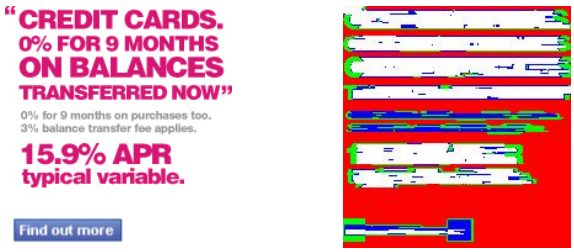
Firstly, the horizontal gradient map g of the gray scale image is obtained. The selection of horizontal gradient map is due to the richness of vertical strokes of texts. Then, the MGD values for pixel (i, j) is computed as the difference between the largest and the lowest gradient values in a 1×21 horizontal neighbourhood window, which is shown as below.

$$MGD(i,j) = max(g(i,j-t)) - min(g(i,j-t)), \quad (1)$$

where $t \in [-10, 10]$.

This horizontal window can result in stronger MGD response at text regions than at background. Then, the k-means clustering is conducted on pixels' MGD values to classify each pixel into either a text cluster (marked as '1') or a non-text cluster (marked as '0'). Due to the fact that different text lines may have strong or weak MGD responses, if k for k-means clustering is set to be 2, some text lines with weak MGD responses may be classified into non-text cluster due to the low contrast between text and background. In our method, the pixels of the MGD map are classified into four clusters, which are called CC cluster maps. Those four CC cluster maps are denoted as $CCMAP_i (i = 1, 2, 3, 4)$. The order is sorted according to the means of the four clusters from smallest to greatest. One example is shown in Figure 2. In Figure 2(b), the four colours represent the four clusters obtained by k-means. It can be clearly seen that the pixels with different MGD responses belong to different clusters. The four clusters are represented by four colours ($CCMAP_1$ is red, $CCMAP_2$ is green, $CCMAP_3$ is blue and $CCMAP_4$ is white).

Since the cluster with the smallest mean usually belongs to background, $CCMAP_1$ is not considered in later processes. The remaining three CC cluster maps are used for generating six layer images, denoted as $LayerImg_i (i = 1, \cdots, 6)$, in the



(a) (b)

Fig. 2. An example of MGD map clustering. (a) An original image. (b) the four clusters of the MGD map of (a).

following way:

$$\begin{aligned} LayerImg_i &= CCMAP_{i+1} (i = 1, 2, 3), \\ LayerImg_4 &= CCMAP_2 + CCMAP_3, \\ LayerImg_5 &= CCMAP_2 + CCMAP_4, \\ LayerImg_6 &= CCMAP_3 + CCMAP_4. \end{aligned} \quad (2)$$

The next step is to detect text by making use of these six layer images. The layer images are selected empirically.

3. TEXT DETECTION IN LAYER IMAGES

Each layer image is composed of CCs that may be formed by text or non-text. All CCs should be processed until every CC is a single text line candidate which will be fed to SVM classifier for text/non-text classification.

3.1. Connected Component Processing

In order to remove tiny CCs, morphological opening operation is performed with a cross-shape structure element. The vertical closeness of text lines may cause the CCs of text lines connect together vertically. Horizontal profile projection with CC pixels is used to split complex CCs into individual text line CCs. Vertical profile projection using edge information is also implemented to separate the horizontally connected background from text. Then, the remaining CCs are enclosed by bounding boxes. Each region enclosed by a bounding box will be sent to a trained classifier for classification which will be discussed in next section.

3.2. T-LBP Based SVM Classification

The task of fine text region detection is to decide whether the region enclosed by each bounding box contains a text line or not. Heuristic rule-based methods [7, 8] and learning-based methods [9, 10] are used for this purpose. Support vector machine (SVM) supervised learning methods have been widely used in text detection algorithms [4, 11, 12]. In this work, we propose a variant of Local Binary Pattern (LBP), called T-LBP, to better depict the textual characteristic of text line.

3.2.1. The Proposed T-LBP Descriptor

The original LBP, introduced in [13], considers a 3×3 neighbourhood. The intensity of the central pixel is compared with those of its eight neighbour pixels. If a neighbour pixel has a greater or equal intensity value than that of the central pixel, this neighbour pixel is marked as '1'; otherwise, marked as '0'. Then, the LBP value of the central pixel is calculated as:

$$LBP(P_c) = \sum_{n=0}^{7} s(i_n - i_c)2^n, s(x) = \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases}, \quad (3)$$

where P_c denotes the central pixel, i_n and i_c denote the intensities of the n-th neighbour pixel and the central pixel.

Variants of LBP have been invented and proved to be a strong texture descriptor for object detection [14, 15] and image categorisation [16, 17]. According to [4], that text on brighter background and lighter background should produce similar histograms of LBP values, eLBP was created as:

$$eLBP(P_c) = \sum_{n=0}^{7} s_e(i_n - i_c)2^n, s_e(x) = \begin{cases} 1, |x| \geq e \\ 0, |x| < e \end{cases}, \quad (4)$$

following the above denotations in (3), where e is chosen to be 20.

According to our observation, text lines usually consist of horizontal and vertical strokes. This has motivated us to better make use of the abundant vertical edges of text lines. Also, there are usually gradual changes of intensities along the horizontal direction at the edge of vertical strokes. We propose a T-LBP to describe these characteristics of texture of text lines, which is defined as:

$$T\text{–}LBP_h(P_c) = \sum_{n=1}^{2} s_1(i_n - i_c)2^{n-1} + \sum_{n=3}^{4} s_2(i_n - i_c)2^{n-1},$$

$$T\text{–}LBP_v(P_c) = \sum_{n=5}^{10} s_2(i_n - i_c)2^{n-5},$$

$$s_1(x) = \begin{cases} 1, |x| \geq e_1 \\ 0, |x| < e_1 \end{cases} \quad and \quad s_2(x) = \begin{cases} 1, |x| \geq e_2 \\ 0, |x| < e_2 \end{cases} \quad (5)$$

following the above denotations in (3), where e_1=10, and e_2=20. T-LBP$_h$ is the T-LBP value in the horizontal direction and T-LBP$_v$ is the T-LBP value in the vertical direction. The neighbour assignment for T-LBP computation is shown in Figure 3.

The occurrence frequencies of T-LBP$_h$ values and T-LBP$_v$ values form two histograms. The dimension number of T-LBP$_h$ is $2^4 = 16$ and that of T-LBP$_v$ is $2^6 = 64$. So, there are one 16-dimension feature vector and one 64-dimension feature vector. Catenated by these two feature vectors, the final feature vector is used to train an SVM classifier.

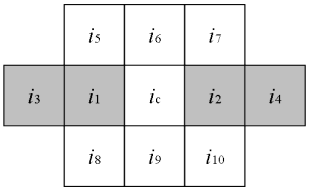


Fig. 3. Neighbour assignment for T-LBP computation. The shadowed pixels represent horizontal neighbourhood pixels of the central pixel P_c.

3.2.2. Text Candidate Verification

In order to determine whether a candidate text block contains text or not, it is firstly normalised to 20 pixels in height keeping its aspect ratio unchanged. For each text block candidate, the T-LBP values are calculated and an 80-dimension feature vector is formed. Then, the feature vector is fed into a trained SVM classifier to verify whether the text block candidate contains text or not. The verification of each normalised bounding box proceeds using a 20×20 sliding window with a 4-pixel step. We used 3000 positive samples and 6000+ negative samples to train the RBF-kernel SVM classifier. We used 10-fold cross-validation to find the optimal parameters for the kernel function. The trained classifier is then used to classify whether each scanning window is text or not. The SVM decision value $G(z)$ of each sliding window is accumulated when the sliding window moves along a candidate text block. The confidence $Conf(R)$ of a candidate text line R is computed by the definition in [3]:

$$Conf(R) = \sum_{z \subseteq R} G(z) \cdot \frac{1}{\sqrt{2\pi}\sigma_0} exp\left(\frac{d_z^2}{2\sigma_0^2}\right), \quad (6)$$

where d_z is the distance between the center of window z and the center of the text region R, and $\sigma_0 = 10$. A candidate text line R is identified as a text line if $Conf(R) \geq 0$.

4. BOUNDING BOX INTEGRATION

In each layer image, the verified text lines are enclosed by bounding boxes. All of the bounding boxes from all layer images will be integrated together to get the final bounding boxes. Since a single text line may appear in more than one layer image, the purpose of bounding box integration is to combine the overlapping bounding boxes. Let $B1$ and $B2$ stand for two bounding boxes. If the following criterion is met, $B1$ and $B2$ are integrated into one bounding box:

$$\frac{AREA(B1 \cap B2)}{min(AREA(B1), AREA(B2))} > 0.6, \quad (7)$$

where $AREA(B1 \cap B2)$ stands for the overlapping area of $B1$ and $B2$. $AREA(B1)$ and $AREA(B2)$ are the area of $B1$ and $B2$ respectively.

Method	Recall	Precision	Harmonic Mean
Ours	**74.88**	**85.35**	**79.78**
Textorter	69.62	85.83	76.88
TH-TextLoc	73.08	80.51	76.62
TDM_IACAS	69.16	84.64	76.12
OTCYMIST	75.91	64.05	69.48
SASA	65.62	67.82	66.70
Text Hunter	57.76	75.52	65.46

Table 1. Comparisons between our method and the algorithms in ICDAR2011 Robust Reading Competition Challenge 1 [18].

5. EXPERIMENTAL RESULTS

Our algorithm is tested on the ICDAR2011 born-digital image dataset, which is made and published for the ICDAR 2011 Robust Reading Competition Challenge 1: Reading Text in Born-Digital Images [18]. In order to compare with other algorithms under the same condition, we use all of the 102 test images and the same performance evaluation system [19] in the competition. The comparisons between our method and the algorithms in the competition (see [18] for more details) are illustrated in Table 1.

Some detection results are shown in Figure 4. High contrast of text is caused by opposite shades of text and background, such as the pink texts on white background in Figure 4(a) and the white texts on black background in Figure 4(f). Low contrast of text is due to similar shades of text and background. For example, the gray texts on white background in Figure 4(a) and the purple texts on black background in Figure 4(e). The examples of detection results in Figure 4 illustrate that the proposed method can detect text lines with high contrast and low contrast. Also, it can be seen that the classifier trained by T-LBP can greatly remove the non-text areas caused by complex background, which can be observed in Figure 4(f) and (g).

6. CONCLUSIONS

In this paper, a new framework of text detection for webpage and email images is proposed. Six layer images are generated for detecting texts with various contrasts. In order to distinguish text and non-text blocks, we propose a variant of LBP descriptor, named T-LBP, by considering the horizontal and vertical strokes of texts and use that as a feature to train an SVM classifier. The trained classifier is used to remove the non-text. The experimental results with comparison with the state-of-the-art demonstrated the effectiveness of our proposed method.

Fig. 4. Some text detection results by the proposed method (the detection results are shown in green bounding boxes).

7. REFERENCES

[1] B. Epshtein, E. Ofek, and Y. Wexker, "Detecting text in natural scenes with stroke width transform," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2963–2970, 2010.

[2] T. Dinh, J. Park, and G. Lee, "Text localization using image cues and text line information," in *IEEE International Conference on Image Processing (ICIP)*, pp. 2261–2264, 2010.

[3] D. Chen, H. Bourlard, and J. Thiran, "Text identification in complex background using svm," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. II–621 – II–626, 2001.

[4] M. Anthimopoulos, B. Gatos, and I. Pratikakis, "A two-stage scheme for text detection in video images," *Image and Vision Computing*, vol. 28, no. 9, pp. 1413–1426, 2010.

[5] E. Wong and M. Chen, "A new robust algorithm for video text extraction," *Pattern Recognition*, vol. 36, no. 6, pp. 1397–1406, 2003.

[6] T. Phan, P. Shivakumara, and C. Tan, "A laplacian method for video text detection," in *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 66–70, 2009.

[7] C. Liu, C. Wang, and R. Dai, "Text detection in images based on unsupervised classification of edge-based features," in *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 610–614, 2005.

[8] P. Shivakumara, T. Pham, and C. Tan, "New fourier-statistical features in rgb space for video text detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 11, pp. 1520–1532, 2010.

[9] D.-Q. Zhang and S.-F. Chang, "Learning to detect scene text using a higher-order mrf with belief propagation," in *IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, p. 101, 2004.

[10] X. Chen and A. L. Yuille, "Detecting and reading text in natural scenes," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. II–366–II–373, 2004.

[11] D. Chen, J.-M. Odobez, and H. Bourlard, "Text detection and recognition in images and video frames," *Pattern Recognition*, vol. 37, no. 3, pp. 595–608, 2004.

[12] Q. Ye, Q. Huang, W. Gao, and D. Zhao, "Fast and robust text detection in images and video frames," *Image and Vision Computing*, vol. 23, no. 6, pp. 565–576, 2005.

[13] T. Ojala, M. Pietikainen, and D. Harwood, "A comprehensive study of texture measures with classification based on feature distributions," *Pattern Recognition*, vol. 29, no. 11, pp. 51–59, 1996.

[14] M. Heikkila and M. Pietikainen, "A texture-based method for modeling the background and detecting moving objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 657–662, 2006.

[15] N. Armanfard, M. Komeili, and E. Kabir, "Ted: a texture-edge descriptor for pedestrian detection in video sequences," *Pattern Recognition*, vol. 45, no. 3, pp. 983–992, 2012.

[16] M. Heikkila, M. Pietikainen, and C. Schmid, "Description of interest regions with local binary patterns," *Pattern Recognition*, vol. 42, no. 3, pp. 425–436, 2009.

[17] X. Qian, X.-S. Hua, P. Chen, and L. Ke, "Plbp: An effective local binary patterns texture descriptor with pyramid representation," *Pattern Recognition*, vol. 44, no. 10-11, pp. 2502–2515, 2011.

[18] D. Karatzas, S. Mestre, J. Mas, F. Nourbakhsh, and P. Roy, "Icdar 2011 robust reading competition challenge 1: Reading text in born-digital images (web and email)," in *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1485 – 1490, 2011.

[19] C. Wolf and J. Jolion, "Object count/area graphs for the evaluation of object detection and segmentation algorithms," *International Journal on Document Analysis and Recognition*, vol. 8, no. 4, pp. 280–296, 2006.