ROBUST SCENE TEXT DETECTION USING INTEGRATED FEATURE DISCRIMINATION

Qixiang Ye, David S. Doermann

University of Maryland, Institute of Advanced Computer Studies, College Park, MD, 20742.
{Qxye, Doermann}@umiacs.umd.edu

ABSTRACT

Scene text detection in images of cluttered backgrounds and/or multilingual context is very challenging. In this paper, we propose a discriminative approach that integrates appearance and consensus features for robust scene text detection. We propose an integrated discrimination model to perform text classification as well as control component grouping. We design shape, stroke and structural features to describe text component appearance and the consensus among them. Experimental results on three public datasets show that the proposed approach is robust to cluttered backgrounds, and is applicable in multilingual environments.

Index Terms— Text detection, Discriminative model, Feature integration

1. INTRODUCTION

The text detection and recognition in natural scene images have received significant attention for demand for applications in scene understanding and visual input [1, 2, 3]. Text detection refers to a task of localizing and grouping text components, as well as discriminating them from the background.

There are two commonly used approaches in scene text detection: those based on connected components and those which use a sliding window classifier on a pixel or a small patch. Component based methods often use color [2], point [3], edge/gradient [4], stroke [5, 6], region [7, 8, 9] features and a combination of them [10, 11, 2, 12] to localize characters or character parts, which are then grouped for further classification. Sliding window methods usually train discriminative models to detect text at multiple scales [10, 4]. Image patches can be classified with texture, shape or appearance models and then are grouped into text regions.

Stroke analysis is a preferred component based method for text localization, and the SWT is competitive for localizing high resolution text, in particular, when it is combined with a learning method or enhanced with other cues such as opposite edge pairs (OEPs) or the Bandlet-based edge detector [6]. Typical stroke based text detection approaches [13] uses regions of strokes as text candidates. However, in clutter background, they often require additional cues for text/non-text discrimination.

Maximally Stable Extremal Regions (MSER) based text detection has attracted attentions in recent years. The main advantage of this representation over other component based approaches is rooted in that the MSER algorithm can adaptively detects stable color regions as text components. After components are localized and grouped, a classification procedure on component shape [14] structure [8, 9] or appearance features [15] could be used to filter out false detections. Despite of the effectiveness of MSER method in text localization, it also require additional cues for text/non-text discrimination.

On the other hand, sliding window approaches [16, 17, 18, 19] usually use discriminative methods to localize text patches and group the patches into regions. These approaches use sliding window classification to localize text, and are, therefore, less sensitive to low resolution text. However, it is validated that in the clutter backgrounds and multilingual environments, direct patch discrimination is often difficult because a small image patch often does not contains sufficient discriminative information.

This paper proposes an approach that fully leverages the appearance and consensus of components for robust text detection. For detection, MSER [20] is used to extract text components, which are then grouped with an agglomerative clustering algorithm. In each clustering iteration, the appearance and consensus features are extracted and are input into the IDM for text discrimination. The agglomerative component grouping procedure stops when the output of the discriminative model is negative. As an extension of previous work [14], this paper contains a more general formulation of a discriminative text detection framework, and a training procedure on partially marked training samples (bounding boxes of text components are not available). It also explores a language independent text representation using the shape filters and stroke filters to capture the characteristics of text appearance. Instead of using syntactic methods, adopted stroke filters are based on unsupervised learning and therefore is robust cluttered backgrounds and multilingual text. A block diagram of the propose approach is shown in Fig.1.

The remainder of the paper is organized as follows: The text detection approach is described in Section 2. Experiments are presented in Section 3 and we conclude the paper in Section 4.

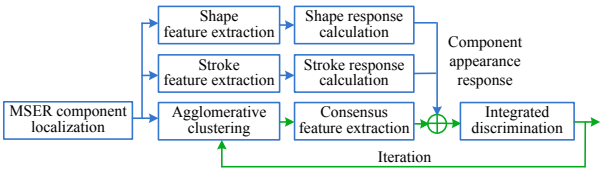

Fig. 1. Block diagram of our text detection approach.

2. TEXT DETECTION APPROACH

2.1. Integrated discrimination model

Formulation: The IDM is a linear discriminative model for a sequence of components, defined as:

$$F(X) = \alpha^T \cdot \psi(X) + \beta^T \cdot \phi(X) + \gamma, \quad (1)$$

where $\psi(X)$ are appearance responses and $\phi(X)$ are consensus features of text components, $X = \{x_z\}, z = 1, ..., Z$. The appearance responses are based on the outputs of two low level classifiers, $f(x_z)$ and $g(x_z)$ (called filters), which correspond to shape features and stroke features, respectively. $\psi(X)$ is the average response from $f(x_z)$ and $g(x_z)$. $\phi(X)$ is component consensus features. α and β are weight vectors of the model and γ is a threshold for text/non-text discrimination. If Eq.1 is positive, X is text, and non-text, otherwise.

The number of characters in text objects varies a lot, so the component number Z should be determined in detection. This is an agglomerative component grouping procedure where

$$\begin{aligned} &\tilde{X} \leftarrow \tilde{X} \cup X_i \equiv \\ &\{x_z\}_{z=1,...\tilde{Z}} \leftarrow \{x_z\}_{z=1,...,\tilde{Z}} \cup \{x_i\}_{i=1,...,I}, \end{aligned} \quad (2)$$

and $\{x_i\}_{i=1,...,I}$ is the nearest component group to $\tilde{X} = \{x_z\}_{z=1,...,\tilde{Z}}$. For detection, we maximize the component number in each text candidate under the discrimination constraint, as:

$$\max_{\tilde{Z}} \{x_z\}_{z=1,...,\tilde{Z}}, \ \ s.t.\ F(\tilde{X} = \{x_z\}_{z=1,...,\tilde{Z}}) > 0, \quad (3)$$

where the objective function maximizes the component number in each text region by merging components into it, recursively. However, if there are non-text components merged, $F(\tilde{X})$ returns a negative value, so the merging procedure is stopped. In addition, when more components merged, the consensus among the components decreases, which can make $F(\tilde{X})$ return a negative value. Eq.3 integrated the text localization and the text/non-text discrimination procedures.

2.2. Appearance feature extraction

The first appearance representation is based on HOG filters. The second is based on stroke filters, which are learned with an unsupervised method, and are applied on sub-blocks of components to capture the stroke characteristics. When extracting appearance features, components will be normalized so that features are extracted at the proper scale. This is illustrated in Fig.2, and detailed as follows.

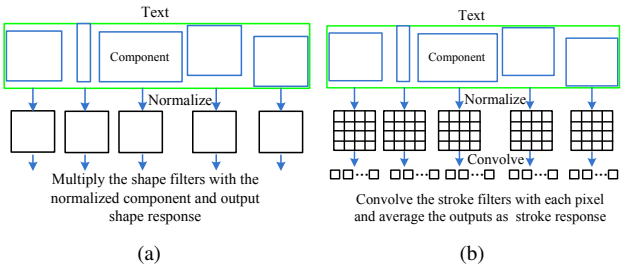

Fig. 2. Illustration of the component appearance response calculation. (a) Shape response by a multiplication operation. (b) Stroke response by a convolution operation.

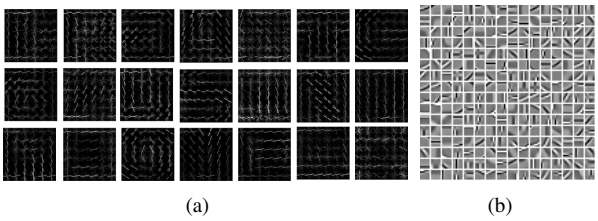

Fig. 3. Visualization of the shape (a) and stroke filters (b).

Shape features: HOG features are employed as low level shape features. When extracting HOG features, a normalized component is divided into cells of 4×4 pixels, and each group of 2×2 cells is integrated into a block with overlapping windows. Each component is represented by 36 blocks, on which 1296 dimensional HOG features are extracted.

Component samples are partitioned into K groups with a K-means clustering algorithm. A multi-class linear SVM training algorithm with a one-against-all strategy is used to train K linear SVMs $f_k(x) = {w_k}^T \cdot x + b_k$, each of which corresponds to a component cluster. The trained the weight vectors of the classifiers are shape filters, as shown in Fig.3(a). When calculating shape appearance response classification, a component will be normalized to a 28×28 patch and multiplied with all of the shape filters. The maximum of the outputs is used as the shape response of the component, as

$$f(x) = max\{f_k(x)\}, k = 1, ..., K. \quad (4)$$

p

Stroke features: The stroke filters, which attempt to capture the stroke response, are learned using a K-means based unsupervised learning method [19] on 7×7 text image patches. The filters refer to the learned K centroids from the input gray value component sub-patches, as shown in Fig.3.

Given the learned stroke filters, we use convolution operations between a normalized component image and the stroke

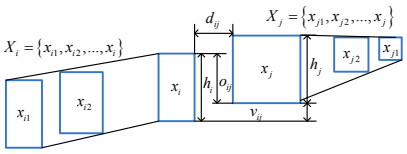

Fig. 4. Spatial relationship of component group X_i and X_j.

filters to compute the pixel level stroke response, as illustrated in Fig.2(b). For each pixel we obtain a K dimensional response, and we then use a non-linear feature mapping [19] to reduce the feature dimension from each component x. After the stroke response of each pixel is calculated, a "pooling" operation is used to convert the response vectors $g_k(x^p), k = 1, ..., K, x^p \in x$, of all the pixels in a component to a K dimensional feature vector. The elements of this vector are further summarized to the stroke response, as

$$g(x) = \sum_{k=1}^{K} \sum_{x^p \in x} g_k(x_p). \tag{5}$$

2.3. Consensus feature extraction

Component consensus that includes the pairwise relationships between components and the holistic variance of grouped components refers to the spatial alignment and color similarity of components in a text region (Fig.4). Assuming i and j denote the indexes component group X_i and X_j, the component consensus is represented using the following features:

Color difference:

$$\phi_1(X_i, X_j) = \phi_1(x_i, x_j) = |c_i - c_j|_2, \tag{6}$$

where c_i and c_j are color means of component i and j.

Spatial distance:(symbols are defined in Fig.4).

$$\phi_2(X_i, X_j) = \phi_2(x_i, x_j) = \frac{v_{ij}}{min(h_i, h_j)}. \tag{7}$$

$$\phi_3(X_i, X_j) = \phi_3(x_i, x_j) = \frac{d_{ij}}{min(h_i, h_j)}. \tag{8}$$

$\phi_4 = \phi_2^2$ and $\phi_5 = \phi_3^2$ are quadratic distances, which can prevent the components of large distances from being grouped.

Alignment:(symbols are defined in Fig.4)

$$\phi_6(X_i, X_j) = \phi_6(x_i, x_j) = \frac{o_{ij}}{min(h_i, h_j)}. \tag{9}$$

$$\phi_7(X_i, X_j) = \phi_7(x_i, x_j) = \frac{|h_i - h_j|}{min(h_i, h_j)}. \tag{10}$$

Color variance: Assuming that X_i is merged with X_j and forms a new text candidate $\tilde{X}$, the color variance of the new text candidate is calculated as follows:

$$\phi_8(X_i, X_j) = variance(c_{i1}, ..., c_i, c_{j1}, ..., c_j), \tag{11}$$

where $c_{i1}, ..., c_i$ and $c_{j1}, ..., c_j$ are component color means.

2.4. Text detection

Model training: The training of appearance models and the calculation of consensus features require the component locations. However, the ground-truth only has the bounding boxes of text regions, but has no component locations. To mark all of the component samples by hand is expensive and time consuming. We use automatic component marking to solve this problem. This procedure trains initial models and use these models to boost the performance. In the procedure, a text sample can be divided into multiple text samples, each of which contains more than three components. At each training iteration, given the locations of text and component samples, the weight vector $W^T = \{\alpha^T, \beta^T\}$ in Eq.1 could be trained with a Support Vector Machine.

Algorithm 1 Text detection algorithm

1. **Initialization**
   Initialize the components $x_m, m = 1, ..., M$ as groups $X_m, m = 1, ..., M$, each of which has one component.
2. **Recursion**
   - 2.1. **Search:** Find the nearest groups (X_i, X_j) by $\text{argmin}_{i,j}\{\phi_1(X_i, X_j) \cdot \phi_2(X_i, X_j) \cdot \phi_3(X_i, X_j)\}$.
   - 2.2. **Coarse discrimination:** If (X_i, X_j) fulfill the following conditions, go to 2.3.
     1) The color distance of (X_i, X_j) in Eq.6 is smaller than a threshold value;
     2) The spatial distances of (X_i, X_j) in Eq.7 and Eq.8 are less than 1.0, and the vertical overlap of (X_i, X_j), Eq.9 is larger than 0.
   - 2.3. **Discrimination:** Group and classify the text candidate $\tilde{X} = X_i \cup X_j$ with $F(\tilde{X})$. If $F(\tilde{X})$ returns positive, $X_i \leftarrow \tilde{X}$, remove X_j.
   - 2.4. If there is no component group pair that passes the coarse discrimination, stop the recursion.
3. **Merging overlapped text regions**

Text detection: We first use the MSER algorithm for text component localization. MSERs from the luminance and chrominance channels are extracted and pooled. A Gamma correction on the image is used as a preprocessing step. Following Eqs.2 and 3, an agglomerative clustering algorithm is used to group the components into text candidates, as well as discriminate the grouped components at each step. The text detection procedure is described in algorithm 1.

3. EXPERIMENTS

Three datasets are used for evaluation: the ICDAR'11 dataset [21], the SVT dataset [16] and a multilingual text dataset [11].

On the ICDAR'11 dataset, our evaluation protocol is consistent with ICDAR'11 [21]. Precision, recall and a harmonic mean (f-measure) are used as metrics. On the SVT and multilingual datasets, the bounding boxes in groundtruth are not precise. So, precision is defined as the ratio between the area of intersection regions and that of detected regions, and recall is obtained as the ratio between area of detected groundtruth regions and that of groundtruth regions. By adjusting the threshold in Eq.(1), we can obtain pprecision and recall rates.

In Table 1, we compare our approach (IDM) with other recent approaches on the ICDAR'11 dataset. It can be seen that our approach has improvement in precision. In particular, it can produce a much higher precision without significant recall drop. Table 2 compares our approach with two representative approaches [16, 8] on the SVT dataset (performance of above approaches is not available on this dataset). It can be seen that our approach shows significant improvement in the f-measure (more than 12%). Table 3 compares the proposed approach and with Pan's approach [11], which is designed for multilingual text. It can be seen that our approach also has significant improvement on the recall rate and f-measure.

Table 1. Performance (%) comparison on ICDAR'11 dataset.

Approach	Precision	Recall	f
IDM(Shape filters)	**85.26**	63.99	**73.11**
IDM(Shape&stroke filters)	72.61	61.09	66.30
IDM(stroke filters)	79.26	54.28	64.43
Neumann and Matas[13] (ICCV2013)	79.303	**66.40**	72.30
Koo (ICDAR'11 winner)[9]	82.98	62.47	71.28
Neumann and Matas [8]	73.10	64.71	68.70
Epshtein et al.[5]	73.00	60.00	66.00
Yi et al.[12]	67.22	58.09	62.32
TH-TextLoc System [21]	66.97	57.68	61.98

Table 2. Performance (%) comparison on SVT dataset.

Approach	Precision	Recall	f
IDM(Shape filters)	**67.52**	**43.89**	**53.20**
IDM(Shape&stroke filters)	64.47	42.10	50.94
IDM(stroke filters)	56.22	41.31	47.63
Wang et al. [16]	67.008	19.00	40.48
Neumann and Matas [8]	32.90	19.10	24.17

Table 3. Performance (%) comparison on multilingual text.

Approach	Precision	Recall	f
IDM(Shape filters)	74.86	60.13	66.70
IDM(Shape&stroke filters)	**75.56**	**65.57**	**70.21**
IDM(stroke filters)	65.43	54.41	59.42
Pan and Liu [11]	65.90	64.90	65.20

It should be noted that on the English text datasets (ICDAR'11 and SVT), the best performance is from the IDM with shape filters, while on the Multilingual text dataset, the best performance is from the IDM with the shape and stroke filters. This shows that for text with a relative small number of character classes, the shape appearance representation is effective, while for text with a large number of character classes, the combination of shape and stroke representation is more effective. Fig.5 shows some detection examples.

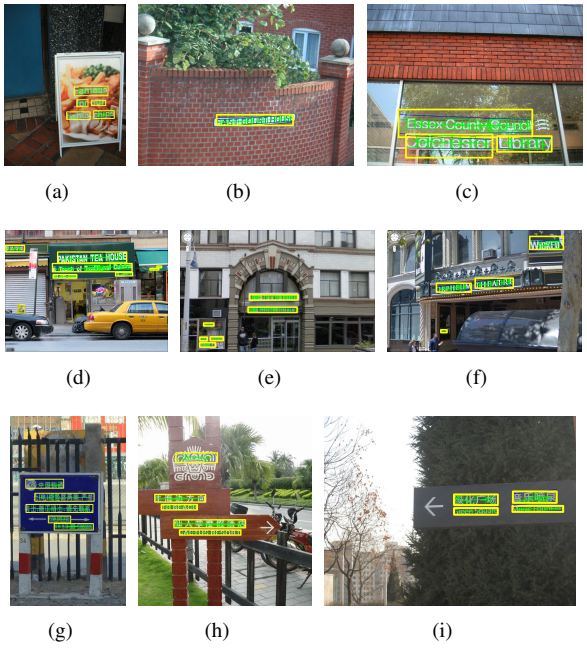

Fig. 5. Text detection examples from the ICDAR'11, SVT and multilingual datasetsp.

With only shape filters, our approach runs at a speed of about 1.6 images per second (for images of 720×576 pixels) on a PC with an Intel CORE i5 CPU. With both shape and stroke filters, however, our approach require tens of seconds to process one image, on average. The most computational cost is about the convolution operations.

4. CONCLUSION

We described an approach that leverages both the shape, stroke and consensus of components for text detection. The reported 85% precision rate on ICDAR'11 dataset is the highest one among competing approaches. On the SVT dataset with cluttered backgrounds and the multilingual dataset, our approach has significant performance improvement, showing robustness of our approach. In addition, the detection framework is simplified by integrating text/non-text classification and component grouping with one discriminative model.

Acknowledgement

The partial support of this research by BBN/DARPA Award HR0011-08-C-0004 under subcontract 9500009235, the China NSF with No.61271433 is gratefully acknowledged.

5. REFERENCES

[1] Jerod J. Weinman, Zachary Butler, Dugan Knoll, and Jacqueline Feild, "Toward integrated scene text reading," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 99, no. 1, pp. 234–778, 2013.

[2] Chucai Yi and Yingli Tian, "Text string detection from natural scenes by structure-based partition and grouping," *IEEE Trans. Image Processing*, vol. 20, no. 9, pp. 2594–2605, 2011.

[3] Xu Zhao, Kai-Hsiang Lin, Yun Fu, Yuxiao Hu, Yuncai Liu, and Thomas S. Huang, "Text from corners: A novel approach to detect text and caption in videos.," *IEEE Trans. Image Processing*, vol. 20, no. 3, pp. 790–799, 2011.

[4] Trung Quy Phan, Palaiahnakote Shivakumara, and Chew Lim Tan, "Text detection in natural scenes using gradient vector flow-guided symmetry," in *ICPR*, 2012, pp. 3296–3299.

[5] Boris Epshtein, Eyal Ofek, and Yonatan Wexler, "Detecting text in natural scenes with stroke width transform," in *CVPR*, 2010, pp. 2963–2970.

[6] Ali Mosleh, Nizar Bouguila, and A. Ben Hamza, "Image text detection using a bandlet-based edge detector and stroke width transform," in *BMVC*, 2012, pp. 1–12.

[7] Lukas Neumann and Jiri Matas, "Text localization in real-world images using efficiently pruned exhaustive search," in *ICDAR*, 2011, pp. 687–691.

[8] Lukas Neumann and Jiri Matas, "Real-time scene text localization and recognition," in *CVPR*, 2012, pp. 3538–3545.

[9] Hyung Il Koo and Duck Hoon Kim, "Scene text detection via connected component clustering and non-text filtering," *IEEE Trans. Image Processing*, vol. 22, no. 6, pp. 2296–2305, 2013.

[10] Sam S. Tsai, Huizhong Chen, David M. Chen, Georg Schroth, Radek Grzeszczuk, and Bernd Girod, "Mobile visual search on printed documents using text and low bit-rate features," in *ICIP*, 2011, pp. 2601–2604.

[11] Yi-Feng Pan, Xinwen Hou, and Cheng-Lin Liu, "A hybrid approach to detect and localize texts in natural scene images," *IEEE Trans. Image Processing*, vol. 20, no. 3, pp. 800–813, 2011.

[12] Chucai Yi and Yingli Tian, "Localizing text in scene images by boundary clustering, stroke segmentation, and string fragment classification," *IEEE Transactions on Image Processing*, vol. 21, no. 9, pp. 4256–4268, 2012.

[13] L. Neumann and J. Matas, "Scene text localization and recognition with oriented stroke detection," *ICCV 2013*.

[14] Qixiang Ye and David S. Doermann, "Scene text detection via integrated discrimination of component appearance and consensus," in *CBDAR*, 2013, pp. 13–18.

[15] Chucai Yi and Yingli Tian, "Text extraction from scene images by character appearance and structure modeling," *Computer Vision and Image Understanding*, vol. 117, no. 2, pp. 182–194, 2013.

[16] Kai Wang, Boris Babenko, and Serge Belongie, "End-to-end scene text recognition," in *ICCV*, 2011, pp. 1457–1464.

[17] Jung-Jin Lee, Pyoung-Hean Lee, Seong-Whan Lee, Alan L. Yuille, and Christof Koch, "Adaboost for text detection in natural scene," in *ICDAR*, 2011, pp. 429–434.

[18] Tao Wang, David J. Wu, Adam Coates, and Andrew Y. Ng, "End-to-end text recognition with convolutional neural networks," in *ICPR*, 2012, pp. 3304–3308.

[19] Adam Coates, Blake Carpenter, Carl Case, Sanjeev Satheesh, Bipin Suresh, Tao Wang, David J. Wu, and Andrew Y. Ng, "Text detection and character recognition in scene images with unsupervised feature learning," in *ICDAR*, 2011, pp. 440–445.

[20] David Nistér and Henrik Stewénius, "Linear time maximally stable extremal regions," in *ECCV (2)*, 2008, pp. 183–196.

[21] Asif Shahab, Faisal Shafait, and Andreas Dengel, "Icdar 2011 robust reading competition challenge 2: Reading text in scene images," in *ICDAR*, 2011, pp. 1491–1496.