Strokelets: A Learned Multi-Scale Representation for Scene Text Recognition

Cong Yao Xiang Bai Baoguang Shi Wenyu Liu
Huazhong University of Science and Technology
yaocong2010@gmail.com, xbai@hust.edu.cn, chn.edward@gmail.com, liuwy@hust.edu.cn

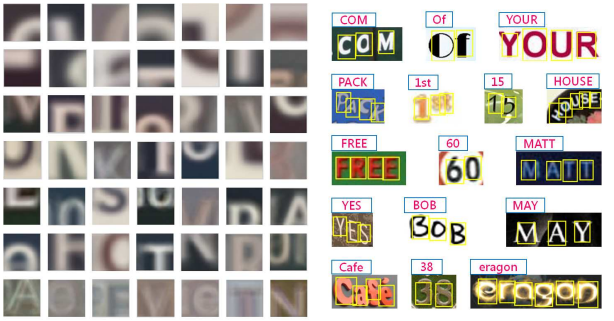

Figure 1. Illustration of strokelets and character recognition. (a) Strokelets learned on IIIT 5K-Word [2]. Strokelets capture the structural characteristics of characters at multiple scales, ranging from local primitives, like bar, arc and corner (top), to whole characters (bottom). (b) Character recognition examples. Strokelets produce accurate character identification and recognition.

1. Introduction

Text in natural images is an important source of information for scene understanding. Though considerable progress has been achieved in recent years [8, 3, 2], detecting and recognizing text in uncontrolled environments are still open problems in computer vision. Excellent representations should be able to effectively describe the characteristics of characters in natural images and meanwhile be robust to interference factors.

In this work, we are concerned with the problem of text recognition in natural scenes and propose a novel multi-scale representation (Fig. 1). This representation consists of a set of multi-scale mid-level primitives, termed as *strokelets*, each of which under ideal conditions represents a stroke shape. Strokelets possess four distinctive advantages:

- ***Usability***: automatically learned from bounding box labels, not requiring detailed annotations.
- ***Robustness***: insensitive to interference factors, endowing the system with the ability to deal with real-world complexity.
- ***Generality***: applicable to variant languages, as long as sufficient training examples are available.
- ***Expressivity***: effective at describing characters in natural scenes, bringing high recognition accuracy.

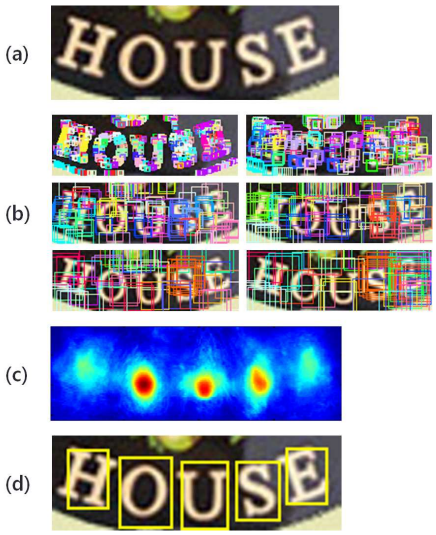

Figure 2. Character identification procedure.

2. Methodology

2.1. Strokelet Generation

Given a set of training images containing scene text $S = \{(I_i, B_i)\}_{i=1}^{n}$, where I_i is an image and B_i is a set of bounding boxes specifying the location and extent of the characters in the image I_i, the goal of strokelet generation is to learn a set of universal part prototypes Ω from S. As S only provides bounding box level annotations for each character, the part prototypes should be automatically discovered. In this paper, we adopt the discriminative clustering algorithm proposed by Singh *et al.* [5] to learn the strokelet set Ω from S.

2.2. Recognition Algorithm

2.2.1 Character Identification

Character identification is a key stage in scene text recognition. In this paper, we propose a voting scheme (Fig. 2) to seek characters, based on multi-scale strokelet detection. The centers of the character candidates are found by seeking maxima in the Hough map using Mean Shift and the extents of these candidates are determined by computing the weighted average of the attributes of the clusters.

Lexicon	Small	Medium	Large
Proposed	**80.2**	**69.3**	38.3
Higher Order [2](with edit distance)	68.25	55.50	28
Higher Order [2](without edit distance)	64.10	53.16	**44.30**
ABBYY9.0	24.33	-	-

Table 1. Performances of different algorithms evaluated on the IIIT 5K-Word dataset.

Dataset	ICDAR 2003(FULL)	SVT
Proposed	80.33	75.89
CNN [7]	**84**	70
Whole [1]	-	**77.28**
TSM+CRF [4]	79.30	73.51
TSM+PLEX [4]	70.47	69.51
Large-Lexicon Attribute-Consistent [3]	82.8	72.9
SYNTH+PLEX [6]	62	57
ICDAR+PLEX [6]	57	56
ABBYY9.0	55	35

Table 2. Performances of different algorithms evaluated on the ICDAR 2003 and SVT dataset.

2.2.2 Character Description

Bag of Strokelets. For each character, all the strokelets that have voted for it are sought via back-projection. A histogram is formed by binning the strokelets.

HOG. Following [3], we also adopt the HOG descriptor to describe characters. A template with 5×7 cells is constructed for each character candidate.

2.2.3 Character Classification

We consider English letters (52 classes) and Arabic numbers (10 classes). To handle invalid characters, we also introduce a special class, so there are 63 classes in total. We train 63 character recognizers, one for each character class, in a one-vs-all manner.

3. Experiments

The performances of the proposed algorithm and other recently published works are illustrated in Tab. 1. In general, the proposed algorithm outperforms all the competing methods. With small lexicon, the proposed algorithm achieves a recognition accuracy of 80.2%, which is 12% higher than that of the closest competitor Higher Order [2] without edit distance (68.25%); with medium lexicon, the improvement (13.8%) is even more notable. The comparison between the proposed approach and Higher Order with edit distance is much fairer, where the improvement (from 28% to 38.3%) is also very significant.

The performances of the proposed algorithm as well as other competing methods on the ICDAR 2003 and SVT dataset are depicted in Tab. 2. The proposed algorithm achieves recognition accuracy of 80.33% and 75.89% on ICDAR 2003(FULL) and SVT respectively, outperforming the competing methods of [6, 4], but still behind those

Figure 3. Learned strokelets on different languages. (a) Chinese. (b) Korean. (c) Russian.

in [7, 1]. Note that the amount of training data used in [7] is far more than that of our algorithm, while [1] cannot handle words out of the given dictionary. Compared to these methods, the proposed algorithm requires less training examples and has a broader scope of application.

We demonstrate three sets of strokelets learned on different languages in Fig. 3. The learned strokelets faithfully reflect the characteristics of the corresponding languages. For example, the strokelets learned on Chinese capture the rich horizontal and vertical structures, while those on Korean additionally highlight the arc structures. In order to cope with multilingual scenarios, we could learn a hybrid set of strokelets on multiple languages.

Please refer to our full paper [9] for more details.

References

[1] V. Goel, A. Mishra, K. Alahari, and C. V. Jawahar. Whole is greater than sum of parts: Recognizing scene text words. In *Proc. of ICDAR*, 2013. 2

[2] A. Mishra, K. Alahari, and C. V. Jawahar. Scene text recognition using higher order language priors. In *Proc. of BMVC*, 2012. 1, 2

[3] T. Novikova, , O. Barinova, P. Kohli, and V. Lempitsky. Large-lexicon attribute-consistent text recognition in natural images. In *Proc. of ECCV*, 2012. 1, 2

[4] C. Shi, C. Wang, B. Xiao, Y. Zhang, S. Gao, and Z. Zhang. Scene text recognition using part-based tree-structured character detection. In *Proc. of CVPR*, 2013. 2

[5] S. Singh, A. Gupta, and A. A. Efros. Unsupervised discovery of mid-level discriminative patches. In *Proc. ECCV*, 2012. 1

[6] K. Wang, B. Babenko, and S. Belongie. End-to-end scene text recognition. In *Proc. of ICCV*, 2011. 2

[7] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng. End-to-end text recognition with convolutional neural networks. In *Proc. of ICPR*, 2012. 2

[8] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu. Detecting texts of arbitrary orientations in natural images. In *Proc. of CVPR*, 2012. 1

[9] C. Yao, X. Bai, B. Shi, and W. Liu. Strokelets: A learned multi-scale representation for scene text recognition. In *Proc. of CVPR*, 2014. 2