Neural Network Based Over-Segmentation for Scene Text Recognition

Xin He, Yi-Chao Wu, Kai Chen, Fei Yin, Cheng-Lin Liu
National Laboratory of Pattern Recognition
Institute of Automation of Chinese Academy of Sciences,
Beijing 100190, China
{xin.he, yichao.wu, kchen, fyin, liucl}@nlpr.ia.ac.cn

Abstract

Over-segmentation is often used in text recognition to generate candidate characters. In this paper, we propose a neural network-based over-segmentation method for cropped scene text recognition. On binarized text line image, a segmentation window slides over each connected component, and a neural network is used to classify whether the window locates a segmentation point or not. We evaluate several feature representations for window classification and combine sliding window-based segmentation with shape-based splitting. Experimental results on two benchmark datasets demonstrate the superiority and effectiveness of our method in respect of segmentation point detection and word recognition.

1. Introduction

Understanding text in scene images is important for many applications, such as street view translation, content based image classification, video retrieval and so on. Therefore, it has gained an increasing interest of the pattern recognition and computer vision community in the last few years. However, scene text recognition suffers much from complex background and appearance variance. Either text location or text line recognition in scene images is still a challenging problem.

Thanks to the advances of computer vision and deep learning techniques, many novel methods that are quite different from conventional OCR methods have been proposed [5, 6, 17]. For example, Yao et al. [17] use a mid-level representation of strokes to describe characters, and then perform classification using random forests. Jaderberg et al. [6] train a word classifier of up to 90k word classes using deep convolutional neural networks. They directly classify words in the image and gives impressive results. However, training a word classifier of as many as 90k classes is quite time consuming. Gordo et al. [5] treat the recognition task as a word retrieval problem. They use a Fisher vector to represent the query word, and then identify the most similar word in the lexicon as the answer. Such lexicon-based recognition is constrained to in-vocabulary words only.

On the other hand, conventional OCR methods, such as the over-segmentation based recognition framework is widely used in printed and handwritten text recognition because of its efficiency and robustness [4, 9, 11]. Compared with methods mentioned above, over-segmentation based recognition is faster in the training and testing phase, and is applicable to either lexicon-driven or lexicon-free case. We present an over-segmentation method which combines sliding window classification and shape-based splitting. Sliding window classification-based segmentation utilizes machine learning technique to adequately consider the spatial context around segmentation point, and has shown success in OCR [1, 2]. We evaluate different feature representations in sliding window classification and compare our method with two exiting methods. One is contour analysis based over-segmentation in [11] (referred to as ContourSeg in the rest of this paper) , which is very successful in handwritten Chinese recognition. The other one, referred to as GraySeg [3], combines the output of a sliding window classifier and boundaries of connected components (CCs) as the over-segmentation result. The recognition performance was evaluated on two benchmark datasets, IIIT-5K-Word and SVT. The experimental results show that our proposed method performs superiorly in both segmentation point detection and word recognition. Fig. 1 shows some examples of correct and incorrect word recognition by our system.

The remainder of this paper is organized as follows. Section 2 describes the framework of our recognition system. Section 3 describes the neural network based over-segmentation method in detail. Section 4 gives experimental results and Section 5 concludes the paper.

2. System design

In order to evaluate our over-segmentation method on scene text recognition and to compare with other methods, we design a scene text recognition system briefly described

978-1-4799-6100-9/15/$31.00 ©2015 IEEE

(a) Correctly read images.

(b) Incorrectly read images.

Figure 1. Examples of images correctly and incorrectly read by our system. Images are from the IIIT 5K-Word test set.

below.

The system diagram is shown in Fig. 2. The input text line image first undergoes a binarization step to generate text connected components (CCs). Then the CCs are over-segmented into primitives each consisting of no more than one character. Candidate characters formed by concatenating consecutive primitives are classified by an modified quadratic discriminant function (MQDF) classifier [7, 16]. Then the candidate segmentation-recognition path with the lowest cost is found by a refined beam search algorithm [16]. The path search is either lexicon-driven [11] if a lexicon is available, or lexicon-free combined with a language prior (statistical model).

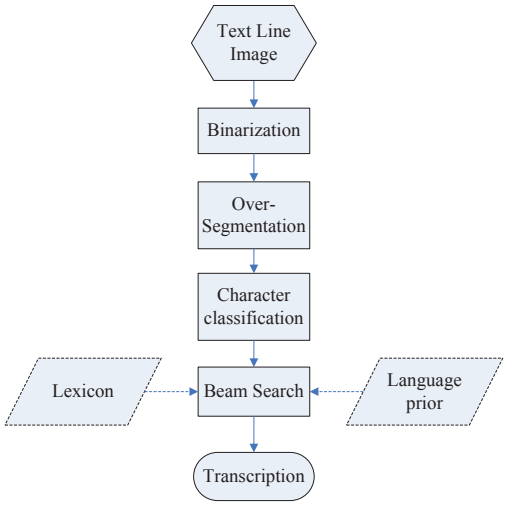

Figure 2. Flow chart of the recognition system

2.1. Binarization

Binarization of scene text image is done by extracting CCs in the YIQ and RGB color space. First we binarize the image of each channel into high-intensity and low-intensity CCs using the Otsu's algorithm. From the two sets of CCs, one set is selected to be candidate text CCs considering text geometrics. After filtering the candidate CCs of each channel using a text/non-text classifier, the channel with most remaining text CCs is selected as the binarization result.

The binarized image is further improved by removing three types of noise CCs: those with few foreground pixels, those of a long line, and those located close to a corner of the text line bounding box.

2.2. Character classifier

We train a MQDF classifier of 73 classes (letters, numbers and some frequent symbols) with about 8 million synthetic characters of different fonts. Some samples were contaminated by Gaussian blurring, motion blurring or morphological operation. From each character sample, normalization-cooperated gradient feature (NCGF) [10] is extracted as the input of the MQDF classifier. On a test set of about 2 million synthetic characters, the classification accuracy is 96.75%.

2.3. Over-segmentation

In text line recognition, character classification applies to candidate characters generated by over-segmentation, which tries to find all the real character boundaries with as few extraneous as possible. The image part between two neighboring segmentation points (hypothesized character boundary) is a primitive, and one primitive or the combination of multiple consecutive primitives can be a candidate character. The objective of over-segmentation is to improve the recall of between-character boundary while reduce extraneous segmentation points.

3. Over-segmentation Method

After the binarization step, foreground text pixels are grouped into CCs. A CC may contain a single character, part of a character or a group of touching characters. Our purpose is to segment CCs that contain more than one characters. To achieve this, we train a binary output sliding window classifier to detect segmentation points on CCs. The classifier is a neural network with one hidden layer. After sliding window segmentation, we adopt a forced splitting technique [11] to improve the recall.

3.1. Sliding Window Segmentation

In the image of a CC, the segmenting window slides from left to right with a stride of 0.1 times the CC height. The window has the same height as the CC, and the width as

0.5 times the CC height. Our experiments revealed that the stride coefficient has little influence on the segmentation and recognition performance when it ranges from 0.04 to 0.1. The width coefficient has greater influence. When it is too large, CCs that have a relatively small aspect ratio will not be segmented. And a small window has insufficient image context to locate the segmentation point. Empirically, we chose 0.5 as the width coefficient and only CCs having width-to-height ratio larger than 0.5 are evaluated by the window classifier.

For feature extraction form the sliding window, we use a combination of NCGF and Weighted Direction Code Histogram(WDCH) features [8]. In the extraction of NCGF, we view both the original image and the normalized image as functional in continuous 2D space and associate them by coordinate mapping. Then we replace the direction of normalized gradient with that in the original image while maintaining the magnitude of the normalized gradient. Thus the resulting gradient feature records the undistorted stroke directions and keeps stroke-width invariance. The WDCH feature vector is computed using the method in [8]. We concatenate the 256 dimensional NCGF and the 392 dimensional WDCH features to form the final feature vector. The configuration of the neural network is 648-200-1.

The center position of a sliding window which gets a positive response is identified as a potential segmentation point (PSP). Adjacent PSPs with a distance less than the stroke width (estimate from foreground area and contour length) are merged by comparing the foreground pixel numbers on each PSP. This merging strategy generates a better result than choosing the middle PSP or the one with the highest classification response. Fig. 3 shows some representative segmentation results.

Figure 3. Representative segmentation results of different methods. Left: ContourSeg, Middle: GraySeg, Right: Our method.

3.2. Forced splitting

CCs are further split if the width is large than 1.5 times the average width of other CCs in the text line image after sliding window segmentation. This is to find segmentation points failed by sliding window classification. For forced splitting, a characteristic function is designed as vertical projection plus distance from center position of the CC:

$$f(x) = \sum_{y} b(x, y) + |x - x_c|, \tag{1}$$

where $b(x, y)$ denotes the binary image of the CC and x_c denotes the center position. The position with minimum value of the function above is taken as the cut point.

4. Experiments

We evaluated the segmentation method with different features on cropped scene text images in respect of segmentation point detection and word recognition.

4.1. Generation of segmentation samples

We manually labeled 6,473 word images collected from the training sets of several benchmark datasets. Those images are divided into a training set and a test set with a ratio of $4:1$. Characters in a labeled image are separated by vertical lines in different horizontal positions. When generating training samples for segmentation, window slides on CCs in the labeled image. We compute the distance D between the center position of the window and each labeled separating line. If D is smaller than a threshold of T_1, the window is received as an positive sample. If D is larger than T_2 ($T_2 > T_1$), the window is received as an negative sample. Otherwise, the window is ambiguous and not used for training. T_1 and T_2 are set as 0.1 and 0.12 times the CC height, respectively. During the training phase, we repeat the positive samples by 3 since the training samples are biased to the negative class.

4.2. Feature selection

We evaluated three features HOG, NCGF, WDCH, and the combinations of them. Table. 1 shows the segmentation performance of using the six sets of features. Segmentation performance (recall rate and precision of segmentation points) was evaluated on our manually labeled word images. It is shown that the combined WDCH and NCGF features perform best in segmentation. As for single features, the NCGF performs best. Though the three types of features all consider stroke contour or gradient direction histograms, the NCGF takes advantage of the un-distorted gradient direction.

4.3. Comparison of segmentation methods

We compared the proposed segmentation method with previous methods ContourSeg and GraySeg. The segmentation results on the labeled images are shown in Table. 2. Note that we implemented the GraySeg method according to [3], but the classifier and training samples are the same as our proposed method.

Feature	Precision	Recall
HOG	83.80	92.71
WDCH	84.30	92.56
NCGF	85.31	92.90
HOG + WDCH	85.63	92.82
HOG + NCGF	85.59	92.85
WDCH + NCGF	**87.93**	**92.92**

Table 1. Segmentation performance of different features.

Segmentation method	Precision	Recall
ContourSeg	85.10	86.05
GraySeg	90.01	91.58
Proposed	87.93	92.92
Proposed + Forced	83.90	93.34

Table 2. Segmentation performance of different methods.

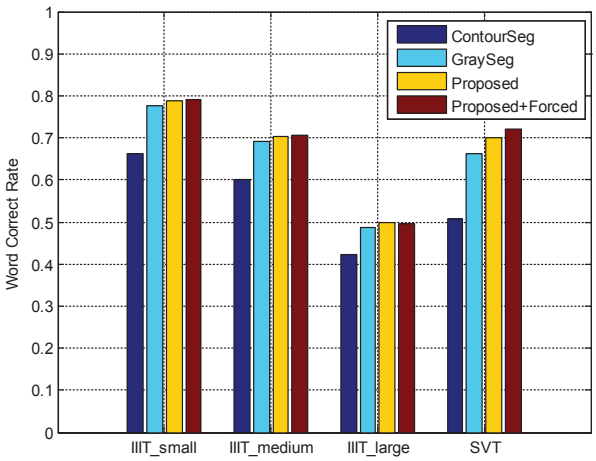

Figure 4. Recognition performance of different methods

We can see from Table 2 that our method outperforms the ContourSeg method in both precision and recall. The GraySeg method generates segmentation points with a high precision but the recall rate is lower. The balance between precision and recall of segmentation affects the final word recognition result, but empirically a higher segmentation recall can bring better potential of correct recognition.

4.4. Recognition Experiments on Benchmark Datasets

We conducted recognition experiments on the IIIT 5k-Word dataset and the Street View Text (SVT) dataset. The IIIT 5K-Word dataset [12] is the largest and most challenging benchmark in this field up to date. It contains 5,000 scene word images, among which 2,000 are used for training and 3,000 for testing. Three versions of lexicons (small, medium and large) are provided for each test image in this dataset. The SVT dataset [14, 15] was harvested from Google Street View. Each image in this dataset is provided with a lexicon of about 50 words. We adopt the SVT-Word subset in which cropped image localizations are annotated.

On one hand, we compare performance of different segmentation methods on the two datasets. The results of word recognition are displayed in Fig. 4. Our method shows superior performance on all four experiments (three different lexicon size on IIIT 5k-Word and one on SVT-Word). Forced splitting contributes to a further improvement of recognition except on the large lexicon of IIIT 5k-Word.

On the other hand, we compare our recognition results with published results on the two benchmark datasets, as shown in Tabel 3. It is observed that on the IIIT 5k-Word dataset, the advantage of our method increases when the lexicon size grows. The word recognition accuracy of our method on the IIIT_large dataset is 11 percent higher than the competitive method in [17]. Moreover, unlike other methods listed in Table 3, our method can work without a predefined lexicon. We tested our system without the lexicon provided, and the word accuracy rate is 45.07%, which is still the highest among other results on the IIIT_large dataset.

One advantage of using an over-segmentation method rather than a word matching method is that, even when a word is wrongly recognized, most characters in it may be correct. That is to say, when the word-level accuracies are comparable, our method may have a higher character accuracy. Unfortunately, the other methods in Table 3 have not published their character-level results. The character-level accuracies of our method are listed in Table 4.

Despite the superior performance, our method still has many segmentation failures, especially when the characters in the scene text image are tightly merged or in unusual fonts (see Fig. 1 (b)).

5. Conclusion

We presented an effective neural network based over-segmentation method for scene text recognition. Experiments on both character segmentation level and word recognition level demonstrate the superiority of our method. Word recognition experiments on two benchmark datasets show that our method perform superiorly compared to previous methods. In the future, our segmentation method can be improved in several ways, e.g., better feature extraction and classifier for sliding window classification, combination with better shape-based over-segmentation.

Acknowledgement

This work has been supported in part by the National Basic Research Program of China (973 Program) Grant

Dataset	IIIT_small	IIIT_medium	IIIT_large	SVT
Proposed without forced splitting	78.66	70.37	**49.90**	70.02
Proposed with forced splitting	79.10	**70.47**	49.47	72.20
Strokelets [17]	**80.2**	69.3	38.3	**75.89**
Higher Order(with edit distance) [12]	68.25	55.50	28	73.57
Higher Order(without edit distance) [12]	64.10	53.16	44.30	73.26
Pairwise CRF(with edit distance) [13]	66	57.5	24.25	68.00
Pairwise CRF(without edit distance) [13]	55.5	51.25	20.25	62.28
SYNTH+PLEX [14]	-	-	-	57
ICDAR+PLEX [14]	-	-	-	56
ABBYY	24.33	-	-	35

Table 3. Word recognition accuracies (%) on the IIIT 5K-Word and SVT-Word datasets. IIIT_small, IIIT_medium and IIIT_large refers to the IIIT 5K-Word dataset with small, medium and large lexicon.

Dataset	IIIT_small	IIIT_medium	IIIT_large	SVT
Proposed without forced splitting	80.42	75.02	62.05	73.63
Proposed with forced splitting	80.69	75.38	60.79	75.05

Table 4. Character-level accuracies on the IIIT 5K-Word and SVT-Word datasets.

2012CB316302, the National Natural Science Foundation of China (NSFC) Grant 61411136002 and the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant XDA06040102).

References

[1] J. H. Bae, K. C. Jung, J. W. Kim, and H. J. Kim. Segmentation of touching characters using an mlp. *Pattern Recognition Letters*, 19(8):701–709, 1998. 1

[2] T. Bayer, U.-G. Kressel, et al. Cut classification for segmentation. In *Proc. Int. Conf. on Document Analysis and Recognition*, pages 565–568. IEEE, 1993. 1

[3] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven. Photoocr: Reading text in uncontrolled conditions. In *Int. Conf. on Computer Vision (ICCV)*, pages 785–792. IEEE, 2013. 1, 3

[4] C. K. Cheng and M. Blumenstein. The neural-based segmentation of cursive words using enhanced heuristics. In *Proc. Int. Conf. on Document Analysis and Recognition*, pages 650–654. IEEE, 2005. 1

[5] A. Gordo. Supervised mid-level features for word image representation. *arXiv preprint arXiv:1410.5224*, 2014. 1

[6] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. *arXiv preprint arXiv:1406.2227*, 2014. 1

[7] F. Kimura, K. Takashina, S. Tsuruoka, and Y. Miyake. Modified quadratic discriminant functions and the application to chinese character recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(1):149–153, 1987. 2

[8] F. Kimura, T. Wakabayashi, S. Tsuruoka, and Y. Miyake. Improvement of handwritten japanese character recognition using weighted direction code histogram. *Pattern recognition*, 30(8):1329–1337, 1997. 3

[9] A. L. Koerich, R. Sabourin, and C. Y. Suen. Recognition and verification of unconstrained handwritten words. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(10):1509–1522, 2005. 1

[10] C.-L. Liu. Normalization-cooperated gradient feature extraction for handwritten character recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29(8):1465–1469, 2007. 2

[11] C.-L. Liu, M. Koga, and H. Fujisawa. Lexicon-driven segmentation and recognition of handwritten character strings for japanese address reading. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(11):1425–1437, 2002. 1, 2

[12] A. Mishra, K. Alahari, and C. Jawahar. Scene text recognition using higher order language priors. In *British Machine Vision Conference (BMVC)*. BMVA, 2012. 4, 5

[13] A. Mishra, K. Alahari, and C. Jawahar. Top-down and bottom-up cues for scene text recognition. In *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2687–2694. IEEE, 2012. 5

[14] K. Wang, B. Babenko, and S. Belongie. End-to-end scene text recognition. In *Int. Conf. on Computer Vision (ICCV)*, pages 1457–1464. IEEE, 2011. 4, 5

[15] K. Wang and S. Belongie. Word spotting in the wild. In *European Conference on Computer Vision (ECCV)*. Springer, 2010. 4

[16] Q.-F. Wang, F. Yin, and C.-L. Liu. Handwritten chinese text recognition by integrating multiple contexts. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 34(8):1469–1481, 2012. 2

[17] C. Yao, X. Bai, B. Shi, and W. Liu. Strokelets: A learned multi-scale representation for scene text recognition. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 4042–4049. IEEE, 2014. 1, 4, 5