eCS: Enhanced Character Segmentation – A Structural Approach for Handwritten Kannada Scripts

C. Naveena[1], V.N. Manjunath Aradhya[2], and S.K. Niranjan[2]

[1] Dept. of CSE, HKBK College of Engineering, Bangalore, India
[2] Dept. of Master of Computer Applications,
Sri Jayachamarajendra College of Engineering,
Mysore - 570 006, India
naveena.cse@gmail.com, aradhya1980@yahoo.co.in
sriniranjan@yahoo.com

Abstract. To build an efficient OCR system, preprocessing task of segmentation process should be in accurate way. In segmentation process, character segmentation plays an important role to obtain clear isolated characters. Character segmentation in Kannada word is a crucial task due to the presence of bottom extension characters (called as Vatthus in Kannada and as extra modifiers in English) and Modifiers. Due to the presence of modifiers and few cursive form of characters the script becomes semi-cursive while writing. With this nature some of the letters are touching each other and also bottom extension characters may get touch to main characters. In this regard, an enhanced Character Segmentation (eCS) approach is proposed for an unconstrained handwritten Kannada scripts. The method is based on thinning, branch point and mixture models. The Expectation-Maximization (EM) algorithm is used to learn the mixture of Gaussians. A cluster mean points are used to estimate the direction and branch point as a reference point for segmenting characters. We experimentally evaluated the proposed method on Kannada words and shown encouraging results.

Keywords: Character Segmentation, Thinning, Branch Points, Mixture Models, Kannada Script.

1 Introduction

Character segmentation has long been a critical area of the OCR process. It is important because incorrectly segmented characters are less likely to be recognized correctly. An OCR system may be designed to work for either on-line or on-line purposes. On-line OCR systems collect input data by recording the order of strokes written on an electronic bit-pad. On-line OCR systems do the same by recording pixel by pixel digital image of the entire writing with a digital scanner. OCR has a wide field of applications covering handwritten document transcription, automatic mail address recognition, machine processing of bank checks, faxes etc [1].

Segmenting characters from an unconstrained handwritten text is a difficult task because: (i) two consecutive characters of a word may touch (ii) two side-by-side

R. Prasath and T. Kathirvalavakumar (Eds.): MIKE 2013, LNAI 8284, pp. 289–298, 2013.
© Springer International Publishing Switzerland 2013

non-touching characters are rarely vertically separable (iii) varies from individual to individual . In [2] basic segmentation algorithms are classified into three main categories: region, contour and recognition based methods. Zhao et al. [17], proposed an improved algorithm for segmenting and recognizing connected handwritten characters. The method gradient descent mechanism is used to weight the distance measure in applying KNN for segmenting/recognizing connected characters in the left to right direction. Tan et al. [13], presented handwritten character segmentation method based on nonlinear clustering. Two stage segmentation of unconstrained handwritten Chinese characters is reported in [16]. Maragoudakis et al [6], describes improved handwritten character segmentation by incorporating Bayesian knowledge with supprot vector machines. Zheng et al. [18], presented character segmentation system based on C# design and implementation. Sari et al. [11], presented on-line handwritten Arabic character segmentation algorithm based on morphological rules. Lee and Verma [5], presented binary segmentation algorithm for English cursive handwriting recognition. The binary segmentation algorithm is a hybrid segmentation technique and consists of over-segmentation and validation modules. The main advantage of binary segmentation technique is that, it adopts an unordered segmentation strategy.

Basu et al [1], presented segmentation of on-line handwritten Bengali script. In this method an isolated words are subdivided into four horizontal imaginary regions to segment a character. Selection of these regions is based on the basic characteristics of Bengali script. Pal et al [9], proposed touching numeral segmentation using water reservoir concept. A water reservoir concept is illustrated to find the touching regions in the numerals. A Reservoir is obtained by considering accumulation of water poured from the top or from the bottom of the numerals. Based on the analysis of reservoir boundary, touching position is detected. Sharma and Singh [12], proposed segmentation of handwritten text in Gurumukhi script. The segmentation of half characters in handwritten Hindi text is presented in [4].

From the above literature, many methods on handwritten character segmentation have been reported in English, Chinese and Arabic scripts. Also some works are carried out in Indian scripts such as Bengali, Gurumukhi. Recently, work on character segmentation for online Kannada text can be seen in [7, 10]. To the best of our knowledge, it is of first kind in the literature for an online handwritten Kannada character segmentation.

The outline of the paper is as follows: In section 2, properties of Kannada script are explained. In section 3, the proposed methodology is detailed. Experimental result is presented in section 4. Finally, conclusion is drawn.

2 Properties of Kannada Script

Kannada script is written horizontally from left to right and an absence of lower and upper case like in English language. Moreover, the Kannada characters are formed by combination of basic symbols, segmentation of the Kannada character is complex and challenging task & increased character set, it contains Vowels, Consonants & Compound characters. Some of the character may get overlap together. Kannada text is difficult when compared with Latin based languages because of its structured complexity. Moreover, Kannada language uses 49 phonemic letters and it is divided into 3-groups,Vowels (Swaragalu- Anusvara (o), & Visarga (:)15), Consonants (Vyanjanagalu-34) and modifier glyphs (Half-letter) from the 15 vowels are used, to alter the 34 base consonants, creating a total of (34*15) +34=544 characters, sample of modifier glyphs additionally a consonants emphasis glyph called Consonant conjuncts in Kannada (vattakshar/ also called extra modifiers in English), exists for each of the 34 consonants. This gives total of (544*34) +15=18511 distinct characters [14].

3 Proposed Method

This section presents the enhanced Character Segmentation approach to the earlier work presented in [8]. Here in this work, segmentation-then-recognition approach is used to enhance the segmentation process. Initially, all components in a word image are detected by connected component analysis (CCA) algorithm which is as shown in the Figure 1. For a component c_i, its height and width are represented by h_i and w_i respectively. From this process the components those having average and below average height and width components are segmented, the remaining components are considered as touching components. To segment these touching components, we follow three steps namely: Thinning, Branch Points and Mixture Models. The steps are explained in following subsections.

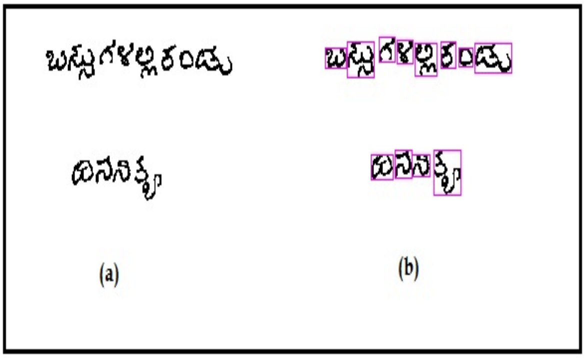

Fig. 1. (a)Input images (b) Results obtained after applying CCA algorithm

3.1 Thinning and Branch Points

In Kannada, some of the letters are cursive in nature and the script become semi-cursive while writing. Also from the statistical analysis most of the touching portion can be find within the half height from the bottom of character. These touching causes mainly because of modifiers and extra modifiers. To find this touching portion in the components, we have applied morphological thinning operation to touching components for further process, which is as shown in the Figure 2. Then, the thinned image is used to find the touching portion using branch points from templates of branch point identifiers. A branch point is a junction point, it connects three branches like a capital T rotated by different angles. In this work, we have used 16 different branches of templates and each one has three branches, Figure 3 shows the several occurrences of 16 types. Figure 4 shows the branch points present in the thinned image and selection of best branch point from many points for accurate segmentation is the main task. For this purpose, we statistically analyzed that most of the touching portions are present at the right half of the average width of a component. Hence, we choose right most branch point as a segmentation point of the touching components. After getting the segmentation point, it is not straight forward way to cut touching portion like horizontal or vertical direction. This is due to the circular nature of Kannada script. To resolve this issue, we applied Mixture Models to thinned image to find the angle of direction to obtain an accurate segment of touching characters.

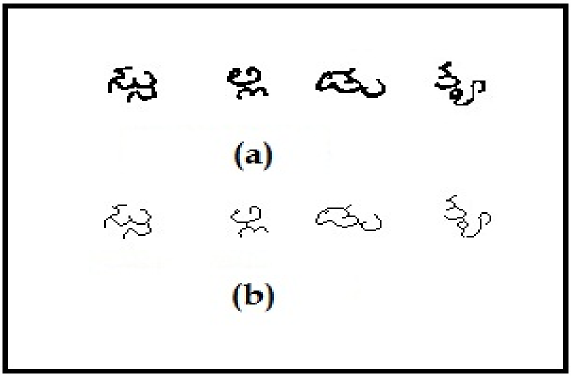

Fig. 2. Results of morphological thinning operation: (a) input image (b) thinned image

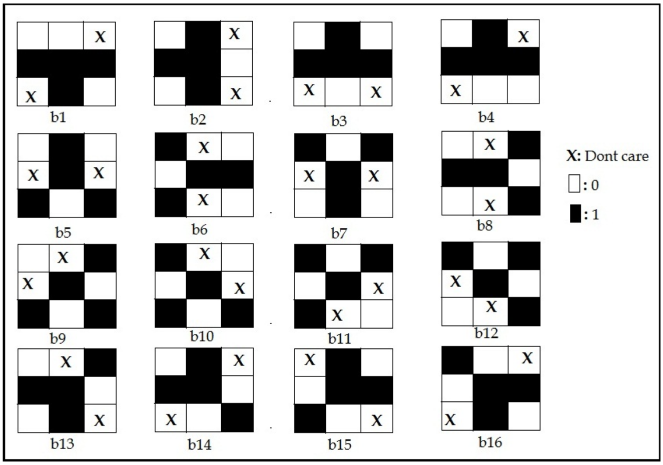

Fig. 3. Branch points extracting templates

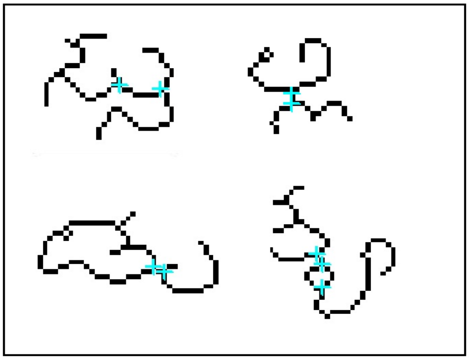

Fig. 4. Results obtained after applying branch point templates

3.2 Mixture Models

Mixture models use a set of data points as a input for clustering input data. Finding a clusters in a set of data points is a considerable problem. To obtain marginalization from joint distribution over observed and latent variables is relatively complex. To solve this problem, use of latent variable in a mixture distributions in which the discrete latent variables can be interpreted as defining assignment of data points to specific components of the mixture. Expectation-Maximization (EM) algorithm is one of the technique for finding maximum likelihood estimation in latent variable. More detailed description regarding EM algorithm can be seen in [3]. The advantage of using EM algorithm is that, it generalizes more complex models with combinations of discrete and real valued hidden variables. Also, it provides grouped or clustered observation even though for incomplete data and missing information.

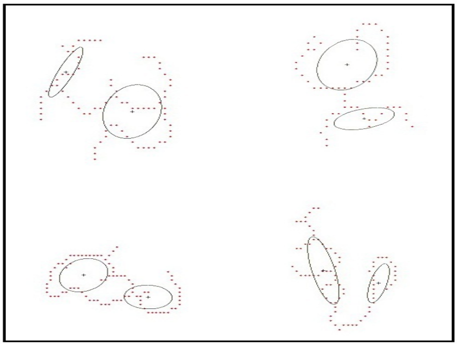

Fig. 5. Results obtained after EM algorithm

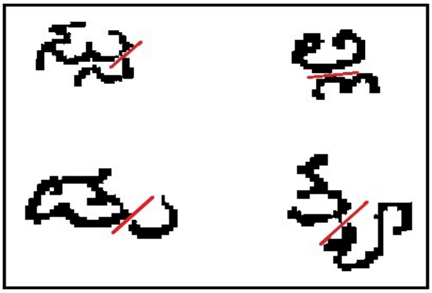

Fig. 6. Segment the touching character based on the skew angle orientation (solid line represents the skew angle orientation)

Means of two clusters are used for estimating the skew of a segmented touching component, because touching characters normally contain two components. Therefore in this work, two clusters are enough to find a skew angle. Figure 5 depicts the mean points obtained for the input skewed word using mixture-of-gaussians. In this work, we have used skew estimators as Linear Regression Analysis (LRA) to estimate the skew angle of the touching character. Finally, with the reference to skew angle direction and best branch point, we segment the touching character which is as shown in the Figure 6.

4 Experimental Results

This section presents the results obtained in the conduct of experiment to study the performance of the proposed method. The method has been implemented in MATLAB 10.0 on a Core2duo processor with 1GB RAM. For our experiment we have collected a data set comprising 400 handwritten Kannada words. Most of the words are presented with one or two touching components. Figure 7 shows the samples of word images of our own collected dataset. To find the segmentation accuracy there are two fundamental analytical segmentation strategies, which are segmentation-then-recognition and segmentation-base strategy.

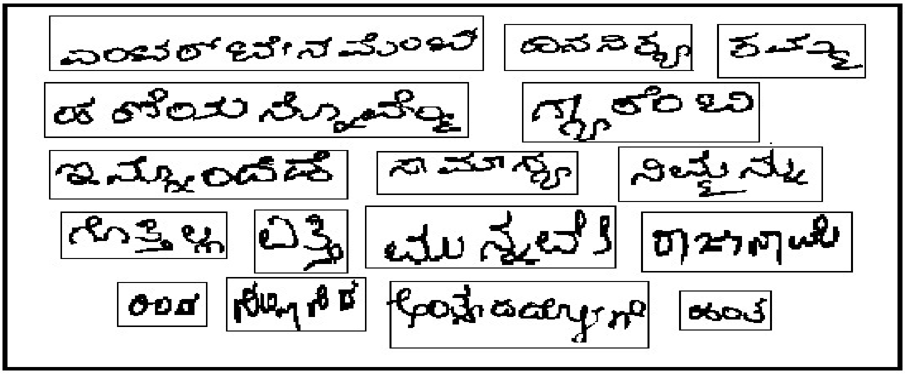

Fig. 7. Sample images of Kannada words

The segmentation accuracy measure as follows:

$$Segmentation\ Accuracy = (SC/N) * 100 \qquad (1)$$

where *SC* is a complete segmented characters in the dataset. *N* is a total number of characters presented in the dataset.

Initially, we adopted a segmentation-base strategy, in that explicit segmentation is used to segment a word into an ideal part of isolated characters. In this strategy, the proposed method achieves a segmentation accuracy of 85.5%. In this, we choose right most branch point as a segmentation point. But in few cases, this assumption leads to an incorrect segmentation, which is shown in Figure 8. To resolve this, we used segmentation-then-recognition strategy. In this, we first trained priory segmented characters of modifiers and extra modifiers using PCA algorithm [15]. Initially, touching characters are segmented by right most branch point and these segmented characters are subjected to recognition process with trained characters. A Euclidean distance measure is used as a classifier in recognition process.

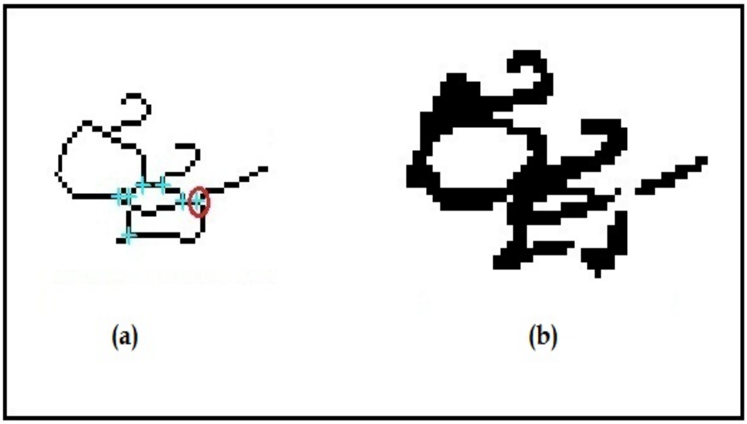

Fig. 8. Incorrect segmentation by selecting right most branch point as segmentation point: (a) Selecting a right most branch point (b) Incorrect segmentation.

Suppose, if the segmented characters are not recognized correctly again segmentation needs to be performed like a feedback process, which is based on a another branch point nearer to right most branch point. Then after recognition process is performed. This process is repeated until the best branch point segments character correctly. Figure 9, shows correctly recognized segmented characters. By adapting segmentation-then-recognition strategy, the proposed method segmentation accuracy increases around 4%. The successful segmentation results are shown in Figure 10. The proposed method fails in two issues. First, if the branch points are not encountered in touching portion, which is shown in Figure 11. Second, if more than two components get touched/overlapped to each other, which is shown in Figure 12.

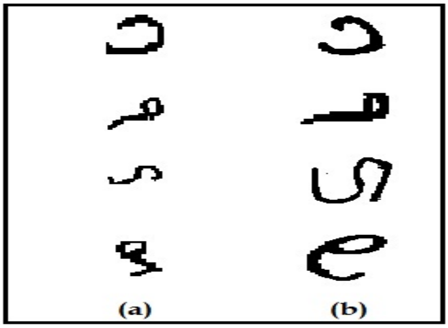

Fig. 9. Recognized the characters after segmentation: (a) segmented characters (b) correctly recognized characters

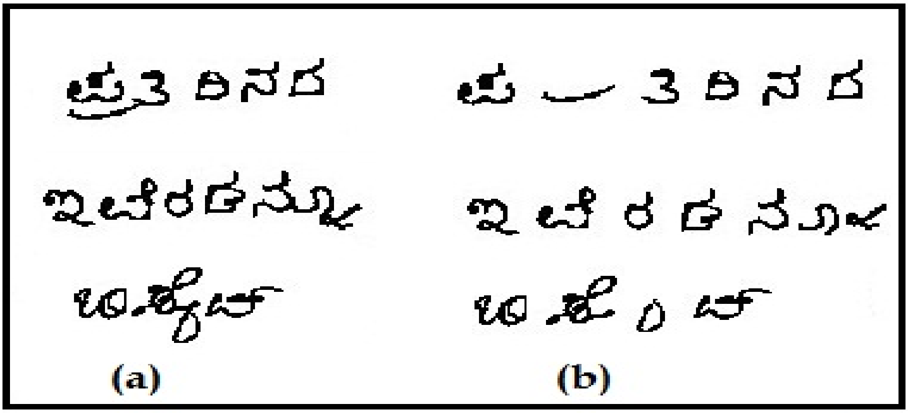

Fig. 10. Successful results obtained for the proposed method: (a) input images (b) segmentation results

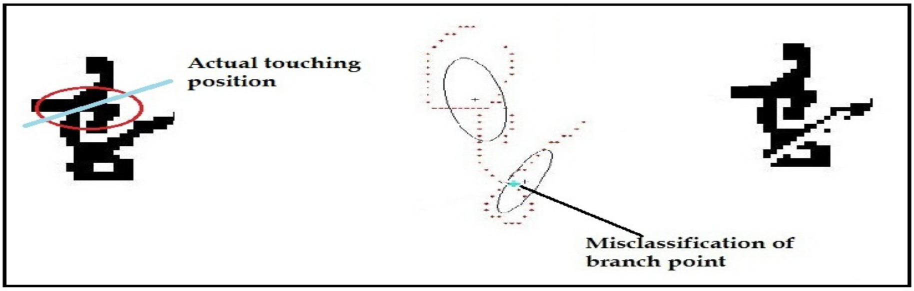

Fig. 11. Failure case of the proposed method: branch points are not present in the touching portion of the component

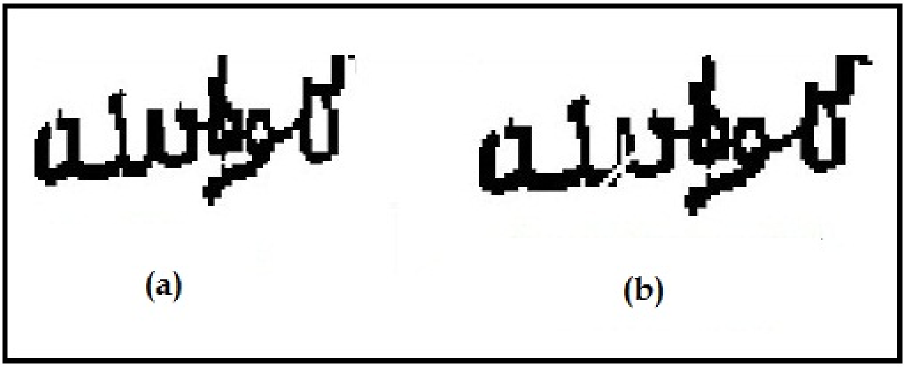

Fig. 12. Failure case of the proposed method: more than two components touching each other, (a) input image (b) incorrect segmentation

5 Conclusion

In this paper, enhanced character segmentation of Kannada script is presented. The proposed method is based on the thinning, branch point and mixture models. The thinning is the most commonly adopted technique to skeletonize an input image and is used to find the branch points present in an image. From the reference of the best branch point and skew angle obtained from the mixture models, we segment the character from touching character. The proposed method is tested on handwritten Kannada words and experimentation is performed based on two strategies: segmentation-base strategy and segmentation-then-recognition strategy. In segmentation-based experiment the selected best branch point may leads to incorrect segmentation in some touching characters and it will reduces the segmentation accuracy. To resolve this issue, we use segmentation-then-recognition strategy. With this strategy, we selected the best branch point and the system's segmentation accuracy increased around 4%. In future, we plan to solve failure issues to increase the segmentation accuracy.

References

1. Basu, S., Chaudhuri, C., Kundu, M., Nasipuri, M., Basu, D.K.: Segmentation of online handwritten Bengali script. In: Proceedings of 28th IEEE ACE, pp. 171–174 (2002)
2. Bhowmik, T.K., Roy, A., Roy, U.: Character segmentation for handwritten Bangla words using artificial neural network. In: Proceedings of 1st IAPR TC3NNLDAR (2005)
3. Bishop, C.: Pattern Recognition and Machine Learning. Springer (2006)
4. Garg, N.K., Kaur, L., Jindal, M.K.: The Segmentation of Half Characters in Handwritten Hindi Text. In: Singh, C., Singh Lehal, G., Sengupta, J., Sharma, D.V., Goyal, V. (eds.) ICISIL 2011. Communications in Computer and Information Science, vol. 139, pp. 48–53. Springer, Heidelberg (2011)
5. Lee, H., Verma, B.: Binary segmentation algorithm for english cursive handwriting recognition. Pattern Recognition 45(4), 1306–1317 (2012)
6. Maragoudakis, M., Kavallieratou, E., Fakotakis, N.: Improving handwritten character segmentation by incorporating Bayesian knowledge with support vector machines. In: Proceedings of ICASSP 2002, vol. 4, pp. IV-4174 (2002)
7. Mohan, P., Shashikiran, K., Ramakrishnan, A.G.: Unrestricted kannada online handwritten akshara recognition using sdtw. In: ACM - Proceedings of International Workshop on Multilingual OCR (2009)
8. Naveena, C., Manjunath Aradhya, V.N.: Handwritten character segmentation for kannada scripts. In: Proceedings of 2nd World Congress on Information and Communication Technologies (WICT 2012), pp. 144–149 (2012)
9. Pal, U., Belaid, A., Choisy, C.: Touching numeral segmentation using water reservoir concept. Pattern Recognition Letters 24(3), 261–272 (2003)
10. Prasad, M.M., Sukumar, M., Ramakrishnan, A.G.: Divide and conquer technique in online handwritten kannada character recognition. In: Proceedings of International Conference on Signal Processing and Communications (SPCOM), pp. 1–5 (2010)
11. Sari, T., Souibi, L., Sellami, M.: Off-line handwritten Arabic character segmentation algorithm: ACSA. In: Proceedings of 8th International Workshop on Frontiers in Handwriting Recognition (IWFHR 2002), pp. 452–457 (2002)

12. Sharama, R.K., Singh, A.: Segmentation of handwritten text in Gurumukhi script. International Journal of Image Processing 2(3), 12–17 (2004)
13. Tan, J., Lai, J.H., Wang, C.D., Wang, W.X., Zuo, X.X.: A new handwritten character segmentation method based on nonlinear clustering. Neurocomputing 89(15), 213–219 (2012)
14. Thungamani, M., Kumar, P.R.: A survey of methods and strategies in handwritten kannada character segmentation. International Journal of Science Research 1(1), 18–23 (2012)
15. Turk, M., Pentland, A.: Eigenfaces for recognition. Journal of Cognitive Neuroscience 3, 71–86 (1991)
16. Zhao, S., Chi, Z., Shi, P., Yan, H.: Two-stage segmentation of unconstrained handwritten Chinese characters. Pattern Recognition 36(1), 145–156 (2003)
17. Zhao, X., Chi, Z., Feng, D.: An improved algorithm for segmenting and recognizing connected handwritten characters. In: Proceedings of 11th International Conference on Control, Automation, Robotics and Vision, pp. 1611–1615 (2010)
18. Zheng, Z., Zhao, J., Guo, H., Yang, L., Yu, X., Fang, W.: Character segmentation system based on C# design and implementation. In: Proceedings of 2012 International Workshop on Information and Electronics Engineering (IWIEE). Procedia Engineering, pp. 4073–4078. Elsevier (2012)