ISSN (Online) : 2319 – 8753
ISSN (Print) : 2347 - 6710

IJIRSET

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)

On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

Radiant Vector Flow Method for Arbitrarily Oriented Scene Text Detection

B.Nishanthi[1], S. Shahul Hammed[2]

PG Scholar, Computer Science and Engineering, Karpagam University, Coimbatore, Tamil Nadu, India[1]

Assistant Professor Dept. Computer Science and Engineering, Karpagam University, Coimbatore, Tamil Nadu, India[2]

ABSTRACT

Text detection and recognition is a hot topic for researchers in the field of image processing. It gives attention to Content based Image Retrieval community in order to fill the semantic gap between low level and high level features. Several methods have been developed for text detection and extraction that achieve reasonable accuracy for natural scene text as well as multi-oriented text. However, it is noted that most of the methods use classifier and large number of training samples to improve the text detection accuracy. The multi-orientation problem can be solved using the connected component analysis method. Since the images are high contrast images, the classifier with connected component analysis based feature training work well for achieving better accuracy. It cannot be used directly for text detection in video because of low contrast and complex background which causes problem such as disconnections and loss of shapes. The deciding classifier and geometrical features of the components is not that much easy in this case. To overcome from this problem our proposed research uses radiant Vector Flow and Grouping based Method for Arbitrarily Oriented Scene text Detection method. The GVF of edge pixels in the Sobel edge map of the input frame is explored to identify the dominant edge pixels which represent the text components. These extracts edge components method corresponding to dominant pixels in the Sobel edge map, which we call Text Candidates (TC) of the text lines. Experimental results on different datasets including arbitrarily oriented text data, horizontal and non-horizontal text data, Hua's data and ICDAR-03 data.

Index Terms— Text detection, Content based Image Retrieval, Connected component based approach, CC clustering, machine learning classifier, non text filtering, scene text detection.

I.INTRODUCTION

SINCE mobile devices equipped with high-resolution digital cameras are widely available, research activities using these devices in the field of human computer interaction (HCI) have received much attention for the last decades. Among them, text detection and recognition in camera captured images have been considered as very important problems in computer vision community [1]–[3]. It is because text information is easily recognized by machines and can be used in a variety of applications. Some examples are aids for visually impaired people, translators for tourists, information retrieval systems in indoor and outdoor environments, and automatic robot navigation. Although there exist a lot of research activities in this field, scene text detection is still remained as a challenging problem. This is because scene text images usually suffer from photometric degradations as well as geometrical distortions so that many algorithms faced the accuracy and/or speed issues [4]–[6]. Text detection algorithm based on two machine learning classifiers: one allows us to generate candidate word regions and the other filters out non text ones.

A .Related Work

The scene text detection algorithms in the literature can be classified into region-based and connected component (CC)-based approaches [1]–[3]. Region-based methods [7], [8] adopted a sliding window scheme, which is basically a bruteforce approach which requires a lot of local decisions. Therefore, the region-based methods have focused on an efficient binary classification text versus non text of a small image patch. In other words, they have focused on the following problem: 1) Problem (A): to determine whether a given patch is a part of a text region.

In their approaches, simple features such as horizontal and vertical derivatives were used at the early stages of the cascade and complex features were incrementally employed [7], [8]. Even though this structure enables efficient text detection, Problem-(A) is still challenging. It is not straightforward even for human to determine the class of a small image patch when we do not have knowledge of text properties such as scale, skew, and color. Multi-scale scheme using different window sizes can alleviate the scale issues, however, it makes boxes in different scales overlap. Experimental results on ICDAR 2005 dataset have shown that this region based approach is efficient, however, it yields worse performance compared with CC-based approaches [5], [9], [10]. CC-based methods begin with CC extraction and localize text regions by processing only CC-level information. Therefore, they have focused on the following problems: 2) Problem (B) : to extract text-like CCs,3) Problem (C) : to filter out non text CCs,4) Problem (D): to infer text blocks from CCs.

In the literature, many CC extraction methods were developed to address Problem (B). For example, some methods assumed that the boundaries of text components should show strong discontinuities and they extracted CCs from edge maps. Others were inspired by the observation that text is written in the same color and they applied color segmentation (or reduction) techniques [11]. On the other hand, some researchers developed their own CC extraction methods from the scratch: the curvilinearity of text was exploited in [10], [12] and local binarization by using estimated scales was adopted in [9].

B. Our Approach

This method consists of three steps: candidate Generation, candidate normalization, and non text filtering but reliable text/non text classifier. In our approach, both problems ((D) and (A)) are addressed based on machine learning techniques, so that our method is largely free from heuristics. We have trained a classifier that determines adjacency relationship between CCs for Problem-(D) and we generate candidates by identifying adjacent pairs. In training, we have selected efficient features and trained the classifier with the AdaBoost algorithm [17]. Apply the above CC clustering method to raw CC sets (that usually contains many nontext CCs) and some candidates may correspond to "non text clusters." Therefore, non text rejection scheme should be followed, which is the main problem of region-based methods. Our CC extraction algorithm is the maximally stable extremal region (MSER) algorithm that is invariant to scales and affine intensity changes, and other blocks in our method are also designed to be invariant to these changes. These invariance allows us to exploit multi-channel information: we can apply our method to multiple channels at the same time and treat their outputs as if they are from a single source.

II. CANDIDATE GENERATION

For the generation of candidates, we extract CCs in images and partition the extracted CCs into clusters, where our clustering algorithm is based on an adjacency relation classifier. In this section, we first explain our CC extraction method. Then, we will explain our approaches (i) to build training samples, (ii) to train the classifier, and (iii) to use that classifier in our CC clustering method.

A. CC Extraction

Among a number of CC extraction methods, we have adopted the MSER algorithm because it shows good performance with a small computation cost [16], [20]. This algorithm can be considered as a process to find local binarization results that are stable over a range of thresholds, and this property allows us to find most of the text components [14], [15]. The MSER algorithm yields CCs that are either darker or brighter than their surroundings. That many CCs are overlapping due to the properties of stable regions [16].

B.Building Training Sets

Our classifier is based on pairwise relations between CCs and CC pair. However, it is not straightforward to train such a classifier. It is not straightforward to determine whether they are in the same word without considering other characters. Therefore, rather than focusing on this difficult problem, we address a relatively simple problem by adopting an idea in region-based approaches. If we have c, ic it will yield a candidate consisting of non text CCs and this candidate will be rejected at the non text rejection step. Also, we will perform word segmentation as a post processing step and the case (2) does not mean negative samples. Based on these observations, we build training sets. Specifically, we first obtain sets of CCs by applying the MSER algorithm to a training set released. Then, for every pair $(ci , c j) \in C \times C$ $(i = j)$, we identify its category among 5 cases. Our classifier is based on pairwise relations between CCs, and consider cases that can happen for a CC pair

$(ci , c j) \in C \times C$ $(i = j)$:

1)$ci \in T , c j \in T , ci \sim c j$

2) $ci \in T , c j \in T , ci \_ c j , t (ci) = t (c j)$

3) $ci \in T , c j \in T , ci \_ c j , t (ci) \quad = t (c j)$

4) $ci \in T , c j \in N$

5) $ci \in N, c j \in N$.

A positive set is built by gathering samples corresponding to the case (1) and a negative set by gathering samples corresponding to the case (3) or (4). Samples from other cases were discarded.

C. AdaBoost Learning

With the collected samples, we train an AdaBoost classifier that tells us whether it is adjacent or not. For the presentation of our method, let us define some local properties of CCs. Given a pair, the horizontal distance, horizontal overlap, and vertical overlap between two boxes are denoted. We have used 6-dimensional feature vectors consisting of five geometrical features and one color- based feature. All of geometric features are designed to be invariant to the scale of an input image and the color feature is given by the color distance between two CCs in RGB space: Features in (8) reflect the relative scales between two CCs and features in (9) encode their relative locations. The scalar in (10) is the product of normalized height difference and normalized distance, which will be large when ci and c j are not adjacent. All of these features are informative and we consider each feature as a weak classifier. For example, the heights of adjacent English characters are similar means that it is likely that ci and c j are adjacent. From these weak classifiers, we build a strong classifier with the Gentle AdaBoost learning algorithm [17], [21]. The Gentle AdaBoost is a variant of AdaBoost learning, and it is easy to implement and known to show good performance in many applications [17], [22].

D. CC Clustering

The AdaBoost algorithm yields a function

$\varphi : C \times C \rightarrow \mathbb{R}$

and we use this function in binary decisions:

$\varphi(ci , c j) > \tau 1 \Longleftrightarrow ci \sim c j$

with a threshold τ1. Given φ(·, ·) and τ1, we can find all adjacent pairs by evaluating that function for all possible pairs in C. Based on these adjacency relations, C is partitioned into a set of clusters

$W = \{wk\}$

where wk ⊂ C. Formally speaking, ci , c j ∈ wk (i.e., ci and c j are in the same cluster) means that there exists $\{ei\}^{m}_{i=1} \subset C$ such that

$ci \sim e1 \sim e2 \sim \cdots em \sim c j$

We build W by using the union-find algorithm . After clustering, we have discarded clusters having only one CC.

E. Comparison to Other MSER-Based Methods

The MSER algorithm has desirable properties for text detection: (i) detection results are invariant to affine transformation of image intensities and (ii) no smoothing operation is involved so that both very fine and very large structures can be detected at the same time [16]. Therefore, the algorithm has been adopted in many methods [13]–[15]. However, unlike our approach, they focused on the retrieval of CCs corresponding to individual characters: the authors in [13] developed a variant of MSER in order to prevent the merging of individual characters, and a Support Vector Machine (SVM) based classifier was developed for the character and non-character classification in [15]. That is, they tried to develop MSER-based CC extractors yielding individual characters (i.e., high precision and high recall). On the other hand, we mainly focus on retrieving the text components as much as possible. As a result, redundant and noisy CCs could be involved in finding clusters.. Moreover, some of them do not correspond to individual characters. The advantages of our approach are its efficiency and robustness. Our method can be efficiently implemented because CC-level feature extraction and classification are not involved. We can also deal with the variations of characters (caused by the font variations and blurs) because we do not exploit the features of individual characters. This approach has drawbacks that text regions could be overlapping and non text regions are sometimes detected, which will be addressed in the following sections.

III. CANDIDATE NORMALIZATION

After CC clustering, we have a set of clusters. In this section, we normalize corresponding regions for the reliable text/non text classification.

A. Geometric Normalization

Given wk ∈ W, we first localize its corresponding region. Even though text boxes can experience perspective distortions, we approximate the shape of text boxes with parallelograms whose left and right sides are parallel to y axis. This approximation alleviates difficulties in estimating text boxes having a high degree of freedom (DOF): we only have to find a skew and four boundary supporting points. To estimate the skew of a given word candidate wk , we build two sets:

$Tk = \{t (ci)|ci \in wk \}$

$Bk = \{b(ci)|ci \in wk \}$

where t (ci) and b(ci) are the top-center point and the bottom center point of a bounding box of ci , respectively.

B. Binarization

Given geometrically normalized images, we build binary images. In many cases, MSER results can be considered as binarization results. However, we perform the binarization separately by estimating text and background colors. It is because (i) the MSER results may miss some character components and/or yield noisy regions (mainly due to the blur) and (ii) we have to store the point information of all CCs for the MSER-based binarization. We consider the average color of CCs as the text color: The average color of an entire block as the background color. Then, we obtain a binary value of each pixel by comparing the distances to the estimated text color and the estimated background color. We have used l2 norm in RGB space.

IV. TEXT/NONTEXT CLASSIFICATION

We develop a text/non text classifier that rejects non text blocks among normalized images. In our classification, we do not adopt sophisticated techniques such as cascade structures, since the number of samples to be classified is usually small. However, one challenge for our approach is the variable aspect ratio as shown. One possible approach to this problem is to split the normalized images into patches covering one of the letters and develop a character/non-character classifier and there are examples. Rather, we split a normalized block into overlapping squares as illustrated and develop a classifier that assigns a textness value to each square block. Finally, decision results for all square blocks are integrated so that the original block (in the left hand side) is classified.our training method that allows us to have a textness value for each square. Then, we explain our text/nontext classification method for normalized images.

A. Feature Extraction from a Square Block

Our feature vector is based on mesh and gradient features as adopted in [25]. We divide each square into 4 horizontal (a) (b) we divide a square block into four horizontal and four vertical blocks. and vertical ones as shown in Fig. 9(b) and extract features. For a horizontal block Hi (i = 1, 2, 3, 4), we consider

1) the number of white pixels,

2) the number of vertical white-black transitions,

3) the number of vertical black-white transitions as features, and features for a vertical block is similarly defined.

B.Multilayer Perceptron Learning

For the training, we need normalized images. For this goal, we applied our algorithm presented in the previous sections (i.e., candidate generation and normalization algorithms) to the training images in [6]. Then, we manually classified them into text and non text. We discarded some images showing poor binarization results, and collected 676 text block images and 863 non text block images. However, we have found that more negative samples are needed for the reliable rejection of non text components and collected more negative samples by applying the same procedure to images that do not contain any text. Finally, we have 3, 568 non text images. These text/non text images are divided into squares.

C. Integration of Decision Results

For the integration of square classification results, we accumulate the outputs of the classifier: where P is the square patch set and Fi is the continuous output of the classifier for the i -th square block in P. We consider ψ(wk) as a textness measure and classify wk as a text region when 1Textness measure (19) is also useful for imposing a non-overlap constraint, i.e., two text blocks should not be overlapping. When two localized regions are significantly overlapping, we simply choose a block showing a higher ψ(·) value.

D. Discussion

We tried to build a text/non text classifier based on normalized gray-scale images (without binarization), because gray scale images seem to be more informative. To be precise, we adopted the AdaBoost learning method and gradient features [26]. However, experiments have shown that this

approach yielded almost the same performance, with a considerable amount of overhead in training. We also tried to use both classifiers in a row (neural network with binary images and Adaboost with gray images), however, we could not find noticeable gains.

V. EXPERIMENTAL RESULTS

A. Word Segmentation Heuristics

Although word segmentation is not the main issue of the scene text detection problem, it is essential in the evaluation. Hence, we have developed a heuristic rule that partitions detected text boxes into words. Given a cluster wk, we sort distances between adjacent CCs in a normalized image in descending order and estimate a (minimum) word spacing distance T .

B. Exploitation of Multichannel Information

The MSER algorithm extracts CCs in a single-channel(scalar valued) image. Therefore, our method sometimes suffers from the loss of information during color reduction. However, such a text may become clear in another channel , and we can alleviate this problem by applying our method to multiple channels. Note that our method is invariant to affine transform of intensity and we can apply our method to the chrominance channel without any modification. In case that similar (significantly overlapping) text blocks are detected in more than one channel, we have selected one text block having the largest textness value.

One may think that CC extraction methods in multi-channel images can alleviate the same problem, even if they are much slower than single channel methods (for example, the maximally stable color region algorithm is several times slower than the original MSER algorithm [27]). However, they may yield over-segmented results. Note that characters in color (vector valued) images are not homogeneous but they consist of several clusters in color spaces.

C. Experimental Results on ICDAR 2011 Dataset

We have also conducted experiments showing the effects of multi-channel processing. Five points on the green dotted line are operating points when we add a new channel image sequentially. Starting from a luminance channel image, we have added Cb, Cr, R, and B channels sequentially. The performance of our method is improved when we add two chrominance channels, however, its precision drops when other channels (such as R and B channels) are added. The effect of our multi-channel scheme. Our method takes about 50 ms for 640 × 480 luminance inputs for a standard PC with 2.8 GHz Intel processor. Also, it takes about 120 ms when three channels are used. Specifically, the complexity of our method is not linear to the number of channels since chrominance channel images usually have less CCs than luminance one.

D. Experimental Results on ICDAR 2005 Dataset

We have evaluated the performance on ICDAR 2005 dataset with the traditional measure [4].Color spaces, such as L–a–b, H–S–V, and Y–Cb–Cr, were evaluated and the Y–Cb–Cr space also shows the best performance. Note that we have also illustrated operating points of conventional methods [4], [7], [9], [10]. Although our recall rate is worse than the method in [9], our method shows better precision and f -measure. Especially, our method is (at least) several times faster than that method.

E. Effects of Parameters

Our method has two trained classifiers, i.e., adjacency relation classifier with a parameter τ1 in (13) and text/non text classifier with a parameter τ2 in (20). Then, τ1 and τ2 can be utilized in controlling the overall precision-recall. Specifically, it shows that the overall recall could drop in the case of a small τ1 which definitely increases the recall of adjacency relation classifier. The increased recall in the adjacency relation classifier is likely to decrease its precision so it could be classified as positive samples. This may result in the failure in localizing text boxes and drop the overall recall. On the other hand, a large τ2 increases the overall precision and vice versa, so we think τ1 should be fixed while τ2 can be changed according to the purpose of the application.

F. Future Work

Our system is designed to address the scene text detection problem in natural images, where English alphabets are placed horizontally [4]–[6]. We have exploited these properties in our algorithm (especially in feature selection), and our method should be changed in order to detect Asian scripts and/or texts of arbitrary orientations [9], [12]. We think our general framework can be extended by developing new features and merging rules, and this is our future research direction.

VI. CONCLUSION

In this paper, we have presented a novel scene text detection algorithm based on machine learning techniques. To be precise, we developed two classifiers: one classifier was designed to generate candidates and the other classifier was for the filtering of non text candidates. We have also presented a novel method to exploit multi-channel information. We have conducted experiments on ICDAR 2005 and 2011 datasets which showed that our method yielded the state-of-the-art performance in both new and traditional evaluation protocols.

REFERENCES

[1] K. Jung, "Text information extraction in images and video: A survey," Pattern Recognit., vol. 37, no. 5, pp. 977–997, May 2004.

[2] J. Liang, D. Doermann, and H. Li, "Camera-based analysis of text and documents: A survey," Int. J. Document Anal. Recognit., vol. 7, nos. 2–3, pp. 84–104, 2005.

[3] J. Zhang and R. Kasturi, "Extraction of text objects in video documents: Recent progress," in Proc. 8th IAPR Int. Workshop Document Anal. Syst., Sep. 2008, pp. 5–17.

[4] S. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young, "ICDAR 2003 robust reading competitions," in Proc. Int. Conf. Document Anal. Recognit., 2003, pp. 682–687.

[5] S. Lucas, "Icdar 2005 text locating competition results," in Proc. Int. Conf. Document Anal. Recognit., 2005, pp. 80–84.

[6] A. Shahab, F. Shafait, and A. Dengel, "ICDAR 2011 robust reading competition challenge 2: Reading text in scene images," in Proc. Int. Conf. Document Anal. Recognit., 2011, pp. 1491–1496.

[7] X. Chen and A. Yuille, "Detecting and reading text in natural scenes," in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., 2004, pp. 366–373.

[8] X. Chen and A. Yuille, "A time-efficient cascade for real-time object detection: With applications for the visually impaired," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Workshops, Jun. 2005, pp. 1–8.

[9] Y.-F. Pan, X. Hou, and C.-L. Liu, "A hybrid approach to detect and localize texts in natural scene images," IEEE Trans. Image Process., vol. 20, no. 3, pp. 800–813, Mar. 2011.

[10] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2010, pp. 2963–2970.

[11] C. Yi and Y. Tian, "Text string detection from natural scenes by structure-based partition and grouping," IEEE Trans. Image Process., vol. 20, no. 9, pp. 2594–2605, Sep. 2011.

[12] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2012, pp. 1083–1090.

[13] H. Chen, S. Tsai, G. Schroth, D. Chen, R. Grzeszczuk, and B. Girod, "Robust text detection in natural images with edge-enhanced maximally stable extremal regions," in Proc. IEEE Int. Conf. Image Process., Sep. 2011, pp. 2609–2612.

[14] J. Matas and K. Zimmermann, "A new class of learnable detectors for categorisation," in Proc. Scandinavian Conf. Image Anal., Jun. 2005, pp. 541–550.

[15] L. Neumann and J. Matas, "A method for text localization and recognition in real-world images," in Proc. Asian Conf. Comput. Vis., 2010, pp. 770–783.

[16] J. Matas, O. Chum, U. Martin, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in Proc. Brit. Mach. Vis. Conf., 2002, pp. 384–393.

[17] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting," Ann. Stat., vol. 28, no. 2, pp. 337–407, 1998.

[18] S. Haykin, Neural Networks: A Comprehensive Foundation. Englewood Cliffs, NJ, USA: Prentice-Hall, 1998.

[19] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[20] D. Nistér and H. Stewénius, "Linear time maximally stable extremal regions," in Proc. Eur. Conf. Comput. Vis., 2008.

[21] G. Bradski and A. Kaehler, Learning OpenCV: Computer Vision with the OpenCV Library. Cambridge, MA, USA: O'Reilly, 2008.

[22] A. Torralba, K. P. Murphy, and W. T. Freeman, "Sharing features: Efficient boosting procedures for multiclass object detection," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun.–Jul. 2004, pp. 762–769.

[23] M. A. Weiss, Data Structures and Algorithm Analysis in C++, 2nd ed. Boston, MA, USA: Addison-Wesley, 1998.

[24] R. Casey and E. Lecolinet, "A survey of methods and strategies in character segmentation," IEEE Trans. Pattern Anal. Mach. Intell., vol. 18, no. 7, pp. 690–706, Jul. 1996.

[25] I.-S. Oh and C. Y. Suen, "Distance features for neural network-based recognition of handwritten characters," Int. J. Document Anal. Recognit., vol. 1, no. 2, pp. 73–88, 1998.

[26] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2001, pp. 511–518.

[27] Hyung Il Koo, and Duck Hoon Kim, "Scene Text Detection via Connected Component Clustering and Nontext Filtering", VOL. 22, NO. 6, JUNE 2013