Natural Scene Text Detection Post-Print Based on Stroke Angle Transformation and Width Features

Authors: Chen Shuo, Zheng Jianbin, Zhan Enqi, Wang Yang

Date: 2018-05-02T00:00:00+00:00

Abstract

To address the missed detection and false detection phenomena of text in natural scene text detection caused by non-uniform illumination and background complexity, we propose a natural scene text detection method based on stroke angle transformation and width features. Analysis reveals that compared with non-text regions, text possesses more stable stroke angle transformation counts and stroke widths. Based on these two characteristics, we propose two features: stroke outer boundary convex-concave angle transformation count and enhanced stroke support pixel area ratio. The former statistically counts the stroke outer contour angle transformation times in segments; the latter calculates the proportion of stroke width stable regions in the total stroke area, which are used to reflect the stability characteristics of stroke angle and width changes respectively. To reduce the text missed detection rate, multi-channel Maximally Stable Extremal Regions (MSER) detection is employed, all candidate regions are merged, stroke features and texture features of the candidate regions are extracted, and a Support Vector Machine is used to complete text and non-text region classification. On the ICDAR2015 database, the algorithm achieves precision and recall rates of 79.3% and 72.8% respectively, and to a certain extent solves the problems of non-uniform illumination and complex background.

Full Text

Natural Scene Text Detection Based on Stroke Angle Transformation and Width Features

Chen Shuo[1],[2], **Zheng Jianbin**[1],[2], **Zhan Enqi**[1],[2]†, **Wang Yang**[1],[2] 1. College of Information Engineering, Wuhan University of Technology, Wuhan 430070, China 2. Key Laboratory of Fiber Optic Sensing Technology and Information Processing of Ministry of Education, Wuhan 430070, China

Abstract: To address the issues of missed detection and false detection of text in natural scene text detection caused by uneven illumination and complex backgrounds, this paper proposes a natural scene text detection method based on stroke angle transformation and width features. Analysis reveals that compared to non-text regions, text exhibits more stable stroke angle transformation counts and stroke width. Based on these two characteristics, we propose two novel features: the number of transformations of the outer corner of the stroke and the enhanced stroke support pixel area ratio. The former statistically analyzes the number of angle transformations on the stroke outer contour through segmentation, while the latter calculates the proportion of stroke width-stable regions within the total stroke area, reflecting the stability characteristics of stroke angle and width variations respectively. To reduce the text miss detection rate, we employ multi-channel Maximally Stable Extremal Regions (MSER) detection, merge all candidate regions, extract stroke features and texture features from candidate regions, and utilize Support Vector Machines to classify text and non-text regions. On the ICDAR2015 database, the algorithm achieves precision and recall rates of 79.3% and 72.8% respectively, and to some extent resolves problems related to uneven illumination and complex backgrounds.

Keywords: natural scene; text detection; stroke feature

0 Introduction

Text detection serves as the foundation for text information extraction and holds significant research value. Natural scene images, which are captured from real-life environments such as product packaging, road signs, and promotional slogans, contain abundant textual information that conveys explicit semantic meaning and better expresses visual scenes. Statistics show that approximately 70% of daily information acquisition comes from visual observation and perception. Therefore, extracting textual information from images can effectively improve the speed of information exchange. However, natural scene text detection faces substantial challenges for three main reasons: (a) text across different scenes is often uncontrollable, exhibiting variations in color, font, scale, and orientation; (b) backgrounds in natural scenes are more complex, with objects similar to text features (such as railings and bricks) interfering with text detection; and (c) issues arising during the capture process, including image blur, partial text occlusion, and uneven illumination, further increase detection difficulty.

Current natural scene text detection methods are primarily divided into two categories based on how text candidate regions are extracted: sliding-window based methods and connected-component based methods. Sliding-window methods mainly leverage the fact that text regions and non-text regions have different texture features, such as local intensity, filter responses, and wavelet coefficients. While these methods are robust to noise and illumination variations, they cannot completely detect individual text components, and the excessive number of sliding windows results in low computational efficiency, making them suitable

only for simple scene images.

Currently, the most commonly used text detection methods are connected-component based approaches, which utilize the characteristic that text has uniform color and texture features to extract text regions through a series of connected component analyses. Common text region extraction methods include the Extremal Regions (ER) algorithm, Maximally Stable Extremal Regions (MSER) algorithm, and Stroke Width Transform (SWT) algorithm. Although these methods are relatively sensitive to illumination and noise, they offer significant improvements in detection speed. By adjusting detection parameters, they can basically detect all text regions, with most text candidate regions being individual characters. Yin et al. constructed a feature space comprising spatial distance, width and height differences, top and bottom connection line tilt angles, color differences, and stroke width differences, used a character classifier to estimate the posterior probability of non-text regions, eliminated non-text regions with high probabilities, and finally employed a Bayesian classifier for text classification. This algorithm achieved high accuracy but missed a large number of text regions in images with complex backgrounds.

1 Multi-channel MSER Text Candidate Region Extraction and Deduplication

Given that traditional MSER algorithms are sensitive to illumination and noise, we first apply bilateral filtering to the original natural image to eliminate noise effects, then extract MSERs from multiple color channels as text candidate regions to reduce interference from uneven illumination. The filtered RGB image is converted to the Perception-Based Illumination Invariant (PII) color space. The advantages of the PII color space are: (a) under different illumination conditions, the difference vector between any two colors remains essentially unchanged; and (b) the Euclidean distance between any two colors is basically consistent with human visual perception distance. To evaluate the effectiveness of different channel combinations, we randomly selected 50 images from the ICDAR2015 dataset for testing, comparing them using two metrics: effective region ratio and text coverage rate. The results are shown in .

shows that combining the R, G, B channels with the H' , S' channels yields the maximum text coverage rate, meaning it can detect the maximum number of text regions in images, while maintaining a relatively high effective region ratio.

Multi-layer MSER detection causes color-stable regions to be detected multiple times, generating a large number of duplicate text candidate regions. To reduce unnecessary computation, we perform deduplication processing. Here, we determine whether two text candidate regions overlap using area overlap rate and region color change rate, calculated as follows:

In equation (1), $\psi(R_i, R_j)$ represents the area overlap rate between text candidate regions R_i and R_j; $A(R_i)$ and $A(R_j)$ represent the areas of regions R_i and R_j respectively. In equation (2), $\mathrm{var}(R_i, R_j)$ represents the color change rate

between regions R_i and R_j, where R_i^R, R_i^G, R_i^B and R_j^R, R_j^G, R_j^B represent the values of the three component channels in RGB color space for regions R_i and R_j respectively; std(x) represents the standard deviation of vector x, and norm(y) represents the L2 norm of vector y. If the area overlap rate $\psi(R_i, R_j) \approx 1$ and the color change rate var(R_i, R_j) is less than threshold s, the two regions are considered duplicates, and only the text candidate region with the smallest color change rate is retained. The deduplication effect is shown in [Figure 2: see original paper].

2 Stroke Feature Based Text Detection

The extracted text candidate regions exhibit large size variations and different aspect ratios. Regions with excessively large or small aspect ratios are mostly non-text regions, though they may also represent multiple connected characters. For candidate regions with aspect ratios outside $[0.3, 1.5]$, we introduce a weight coefficient α when calculating stroke features, where $\alpha = \frac{1}{\max\{s, \lfloor A \rceil\}}$ and $\lfloor \cdot \rceil$ represents rounding to the nearest integer. The physical meaning is the number of regions into which the candidate region can be divided to achieve an aspect ratio of 1.

2.1 Stroke Feature Calculation

The extracted text candidate regions include not only text regions but also a large number of non-text regions. The primary problem addressed in this section is how to accurately filter out non-text regions. Compared to non-text, text exhibits distinct stroke features such as clear stroke contours and stable stroke width. We select two features—the number of transformations of the outer corner of the stroke and the enhanced stroke support pixel area ratio—as stroke features for classification.

The 26 English letters are composed of a limited number of strokes, which form a limited number of stable outer boundary corner points. Outer boundary corner points are divided into convex corners (优角) and concave corners (劣角), as shown in [Figure 3: see original paper]. The red points in the figure represent convex corners, and the green points represent concave corners (see electronic version). The transformation count between convex and concave corners on the outer boundary is 6.

Since binary images are either 0 or 1 with large edge gradients and non-smooth contours, using Harris corner detection would mistakenly detect excessive corners and be insufficiently sensitive to corners approaching straight angles. Therefore, we propose a new corner detection method specifically for binary images. First, we extract edge pixels from the binary image and classify them into 8 categories based on the distribution of pixels in their four-neighborhood, numbered sequentially from 1 to 8, as shown in [Figure 4: see original paper]. In [Figure 4: see original paper], 0/1 indicates that the two positions in an edge pixel type must be both 0 or both 1. Starting from the edge pixel at the upper right corner

of the outer contour, we mark the type of each edge pixel in clockwise order, with the mark values forming a one-dimensional vector. We then segment this one-dimensional vector so that each segmentation point corresponds to a corner on the outer edge.

The segmentation process is as follows: a) If the number of pixels between two transition points exceeds threshold m, convert the two endpoints to candidate segmentation points, as shown in Figure 5: see original paper. b) For n consecutive transition points (including candidate segmentation points) where $n > \alpha$, if the mark values before and after these points are i and j respectively, merge the mark values between these transition points into one segment vector, denoted as segment vector p_{ij}, with the intermediate mark values also converted to p_{ij} (since in this case edge pixels 1, 3, 5, 7 cannot be adjacent, and edge pixels 2, 4, 6, 8 cannot be adjacent, i and j must be one odd and one even; here we let i be odd and j be even), as shown in Figure 5: see original paper. c) Repeat processes a) and b) until no further merging is possible. d) If the interval pixels between two segment vectors p_{ij} and p_{kl} $(i, j, k, l$ 取值 0-8) are less than or equal to threshold β, merge these two segment vectors into one segment vector p_{ij}. Other transition points are directly converted to candidate segmentation points, as shown in Figure 5: see original paper.

In [Figure 5: see original paper], blue triangles represent mark value transition points, and red triangles represent candidate segmentation points (see electronic version). At this point, the one-dimensional integer vector composed of mark values becomes a one-dimensional decimal vector composed of segment vectors. The numerical values of segment vectors and the contour directions they represent are shown in [Figure 6: see original paper].

If the values before and after a segmentation point change in clockwise direction (ensuring the included angle a is less than 180°), the segmentation point corresponds to a convex corner; if they change in counterclockwise direction, it corresponds to a concave corner. The transformation count is denoted as k. Let the current segmentation point be P, with the mark value before the segmentation point being p_i and after being p_j. The calculation formula is:

In equation (3), the case $\{i, j\} \in \{0, 4\}$ does not exist; k represents the number of positive-to-negative value transitions. For candidate regions with $s_A \notin [0.3, 1.5]$, the influence of thresholds α and β on k is shown in .

shows that selecting $\alpha = 3, \beta = 2$ yields the best results for text candidate regions.

2.1.2 Enhanced Stroke Support Pixel Area Ratio (ESSP)

We use the Stroke Support Pixel (SSP) method to calculate stroke skeletons. The stroke skeleton distance to edge pixels is defined as d_i, stroke width as w_i, and stroke skeleton length as L. For ideal strokes, the stroke width w_i remains basically stable. As shown in [Figure 7: see original paper], the number of stroke

support pixels essentially equals the stroke area.

The stroke skeleton calculation requires first computing the distance transform map, where each pixel value in the binary image represents its distance to the nearest background point. Pixels with distance values greater than or equal to their eight-neighborhood values are selected as stroke skeleton pixels. Under normal circumstances, issues such as discontinuous stroke skeletons and multiple stroke skeleton pixels in a row cause inaccurate results. Therefore, we introduce a weight coefficient ξ_i for each skeleton pixel to normalize it to a single stroke length. The new support pixel count calculation formula is:

In equation (5), N_3 represents the number of skeleton pixels in the 3$×$3 neighborhood, and the stroke area ratio is $\frac{S_A}{A}$, where A is the binary region area. The calculation process is shown in [Figure 8: see original paper].

Even with the weight coefficient, for text with varying stroke width, the S_A value remains small. The fundamental cause is the loss of too many skeleton pixels. When stroke width changes from small to large or when multiple strokes connect, skeleton pixel distance values are no longer local maxima and thus cannot be detected, as shown in Figure 9: see original paper.

To solve this problem, we propose the Enhanced Stroke Support Pixel (ESSP) method. For candidate regions with $s_A \notin [0.3, 1.5]$, we first scale the larger edge of the candidate region to ensure the region can be evenly divided, with a scaling ratio of ratio $= \max\{s, 1/A\}$. The candidate region is then divided into N sub-candidate regions along the larger edge, and the above calculation is performed on each sub-region to obtain skeleton pixels for each sub-region, as shown in Figure 9: see original paper.

The ESSP method calculation process is as follows: a) Calculate the average value of skeleton pixel distances: $\bar{d} = \frac{1}{n}\sum_{i=1}^{n} d_i$ b) Select endpoint pixels on the stroke skeleton with pixel distances close to $\bar{d}$ (including points with only one skeleton pixel in their neighborhood) as origin points. Find the two nearest skeleton pixels to each origin point and connect the origin point to these two pixels with straight lines. c) If background pixels exist on the line, abandon this line and proceed to the next origin point, repeating step b). If no background pixels exist and the difference between the distance values of the two pixels is less than $0.3\bar{d}$, select the maximum value in the 3$×$3 neighborhood of pixels on the line (if the maximum is a skeleton pixel or lies on the line, select the second maximum) and add it to the skeleton pixels; otherwise, abandon this line and repeat step b).

Calculate the pixel area ratio. The results show significant improvement, with the calculation process shown in [Figure 10: see original paper].

We randomly selected 500 text sample images and 500 non-text sample images for algorithm testing. The comparison results between SSP and ESSP are shown in [Figure 11: see original paper].

In [Figure 11: see original paper], 0.1-0.2 indicates the stroke area ratio is in the interval $(0.1, 0.2]$, with other values following similarly. The figure shows that for text samples, ESSP effectively increases the stroke area ratio; for non-text samples, there is also enhancement but with smaller magnitude. The ESSP algorithm increases the gap between text and non-text regions.

2.2 Text Detection

Text and non-text regions differ not only in stroke features but also in that text has more regular contours and stable edges and texture features. We select Histogram of Oriented Gradients (HOG) features and Mean Local Binary Pattern (MLBP) features combined with Support Vector Machine (SVM) to filter out remaining non-text regions. HOG constructs feature vectors by calculating and statistically analyzing gradient direction histograms of local image regions, describing image gradient direction and edge information. MLBP is an improvement over Local Binary Pattern (LBP) that enhances robustness to noise.

The above steps remove most complex background elements and yield relatively complete individual character regions. Next, we utilize the Breadth-First Search (BFS) concept to aggregate these individual character regions into text lines as the final natural scene text localization result. BFS is a traversal method for connected graphs that starts from a vertex v_0, sequentially visits its adjacent points $v_1, v_2, ..., v_k$, and if features are similar, merges them into one vertex before visiting adjacent points of $v_1, v_2, ..., v_k$ until all vertices in the graph are traversed. The selected features include CIE94 color distance, character gap distance, center connection line tilt angle, and stroke width as constraints for text region aggregation. Text regions with similar color distance and stroke width are merged into a group. Character gap distances and center connection line tilt angles between text regions in each group are statistically analyzed, and text regions are merged according to similarity to form text lines.

3.1 Experimental Database and Evaluation Metrics

We selected the publicly available ICDAR2015 dataset to evaluate our algorithm. The ICDAR2015 database contains text images with different colors, fonts, and sizes, some of which suffer from issues such as uneven illumination, low contrast, blur, and occlusion. The dataset includes 229 training images and 233 test images.

This paper adopts the evaluation metrics provided by ICDAR to assess the proposed method, consisting of Precision Rate P, Recall Rate R, and comprehensive evaluation value F. The definitions are as follows:

In the formulas, the set of ground truth text region bounding boxes in images is T; the set of algorithm output text region bounding boxes is E; $m(r, r')$ is the area overlap rate, calculated as in equation (1), except $A(r)$ represents the area

of the text region bounding box. Combining precision P and recall R yields the comprehensive evaluation metric F, generally taking $\alpha = 0.5$:

The ICDAR public database is a commonly used dataset for text detection. Partial result displays are shown in [Figure 12: see original paper].

From [Figure 12: see original paper], we can see that our method can adapt to different backgrounds, text sizes, and text colors, effectively suppressing background elements similar to text such as leaves, arrows, and windows. It accurately locates text positions and achieves good detection results.

To further verify algorithm effectiveness, we compare the precision P, recall R, and comprehensive evaluation metric F of our algorithm with advanced domestic and international algorithms. The comparison results are shown in .

shows that although the precision P of our algorithm tested on the ICDAR2015 dataset is relatively low, only surpassing Bais' s algorithm, the F value is the same as Yi' s method and higher than other algorithms, with the highest recall rate R. To preserve as many complete true text regions as possible (i.e., improve recall), our algorithm employs multi-channel MSER for candidate region extraction, and the extracted stroke features target the essential differences between text and non-text, eliminating a large number of non-text regions and maximizing the screening of true text. However, this also retains artificial graphics with stroke characteristics. Further distinction between text and non-text using edge and texture features leads to some non-text regions being misclassified as text, resulting in lower precision. Considering all evaluation criteria, our algorithm demonstrates good performance.

4 Conclusion

This paper proposes a natural scene text detection method based on stroke features. Deduplication processing of multi-channel MSER algorithm overcomes the illumination sensitivity of traditional MSER algorithms while suppressing the explosive growth of candidate regions. Extracting stroke features of candidate regions, including both contour and skeleton features, can essentially describe the differences between text and non-text. Combined with edge and texture features, this approach can effectively detect text regions in natural scenes. However, for natural scene images with low contrast, high blur, or text occluded by railings, our algorithm cannot accurately locate text and requires further research to improve algorithm performance.

References

[1] Xu Xiaoli. Research on image segmentation algorithm based on cluster analysis [D]. Harbin: Harbin Engineering University, 2012.

[2] Jaderberg M, Simonyan K, Vedaldi A, et al. Reading text in the wild with convolutional neural networks [J]. International Journal of Computer Vision,

2016, 116(1): 1-20.

[3] Huang Weilin, Qiao Yu, Tang Xiaoou. Robust scene text detection with convolutional neural network induced MSER trees [C]//Proc of European Conference on Computer Vision. 2014: 497-511.

[4] Zhang Zheng, Shen Wei, Yao Cong, et al. Symmetry-based text line detection in natural scenes [C]//Proc of IEEE Conference on Computer Vision & Pattern Recognition. 2015: 2558-2567.

[5] Zhu Yingying, Yao Cong, Bai Xiang. Scene text detection and recognition: recent advances and future trends [J]. Frontiers of Computer Science, 2016, 10(1): 19-36.

[6] Neumann L, Matas J. Real-time scene text localization and recognition [C]//Proc of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2012: 3538-3545.

[7] Neumann L, Matas J. Efficient scene text localization and recognition with local character refinement [C]//Proc of International Conference on Document Analysis and Recognition. 2015: 746-750.

[8] Xu Hailiang, Xue Like, Su Feng. Scene text detection based on robust stroke width transform and deep belief network [C]//Proc of Asian Conference on Computer Vision. 2014: 195-209.

[9] Yin Xucheng, Yin Xuwang, Huang Kaizhu, et al. Robust text detection in natural scene images [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2014, 36(5): 970-983.

[10] Chong H Y, Gortler S J, Zickler T. A perception-based color space for illumination-invariant image processing [J]. ACM Trans on Graphics, 2008, 27(3): 1-7.

[11] Sun Lei. Text detection in natural scene images [D]. Hefei: University of Science and Technology of China, 2014.

[12] Cho H, Sung M, Jun B. Canny text detector: fast and robust scene text localization algorithm [C]//Proc of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2016: 3566-3573.

[13] Buta M, Neumann L, Matas J. FASText: efficient unconstrained scene text detector [C]//Proc of IEEE International Conference on Computer Vision. 2015: 1206-1214.

[14] LUCAS S M. ICDAR 2005 text locating competition results [C]//Proc of International Conference on Document Analysis and Recognition. 2005: 80-84.

[15] Shi Cunzhao, Wang Chunheng, Xiao Baihua, et al. Scene text recognition using part-based tree-structured character detection [C]//Proc of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2013: 2961-2968.

[16] Bai Bo, Yin Fei, Liu Chenglin. Scene text localization using gradient local correlation [C]//Proc of International Conference on Document Analysis and Recognition. 2013: 1380-1384.

Note: Figure translations are in progress. See original paper for figures.