FAST SCENE TEXT LOCALIZATION BY LEARNING-BASED FILTERING AND VERIFICATION

Yi-Feng Pan, Cheng-Lin Liu, Xinwen Hou

National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences
95 Zhongguancun East Road, Beijing 100190, P. R. China
{yfpan,liucl,xwhou}@nlpr.ia.ac.cn

ABSTRACT

This paper proposes a new method for fast text localization in natural scene images by combining learning-based region filtering and verification in a coarse-to-fine strategy. In each pyramid layer, a boosted region filter is used to extract candidate text regions, which are segmented into candidate text lines by multi-orientation projection analysis. A polynomial classifier with combined features is used to verify patches of candidate text lines for removing non-texts. The remaining text patches over all pyramid layers are grouped into text lines based on their spatial relationships. The text lines are further refined and partitioned into words by connected component analysis. Experimental results show that the proposed method provides competitive localization performance at high speed.

Index Terms— Text Detection, Coarse-to-Fine, Feature Extraction, Classification

1. INTRODUCTION

Scene text detection and localization, as a key part of content-based image analysis, has received intensive attention in the last decade. This problem is challenging due to the cluttered background of image, the arbitrary font and size of text, and the variation of lighting condition. Many methods have been proposed and some have reported promising results [1], but this problem still remains unsolved in the sense that the accuracy and speed are not satisfactory.

The existing methods can be roughly categorized into region-based and connected component (CC)-based ones [2, 3]. Region-based methods classify text regions from non-text ones using textural features. By region classification at multiple scales and locations, the classified text regions are then grouped into text blocks for localization. On the other hand, CC-based methods detect candidate text blocks by fast analysis of image edge or color properties. The candidate blocks then undergo a verification stage for discriminating text blocks from non-text ones. The fast but coarse candidate detection stage may complicate the verification process and deteriorate the overall performance.

In this paper, we propose a new coarse-to-fine method for fast scent text localization by learning-based region filtering and verification, unlike the previous methods that use learning-based classification for only filtering or verification. We guarantee localization accuracy and speed by selecting discriminative features and training efficient classifiers. A boosted classifier and a polynomial classifier are used for coarse region filtering and fine verification, respectively.

For the verification stage, we also evaluate five widely used features: HOG (histogram of oriented gradients), LBP (local binary pattern), DCT (discrete cosine transform), Gabor, and wavelets. We show that combining the HOG and wavelets gives the best verification performance. Experimental results on the ICDAR 2003 competition dataset show that the proposed method provides competitive localization performance with existing methods at high speed.

2. SYSTEM OVERVIEW

Our system (Fig. 1) comprises two stages: coarse text detection and fine localization. After building the image pyramid, the coarse stage uses a boosted region filter for quickly finding candidate text regions in each layer of pyramid. The candidate text regions are then segmented into candidate text lines by multi-orientation projection analysis.

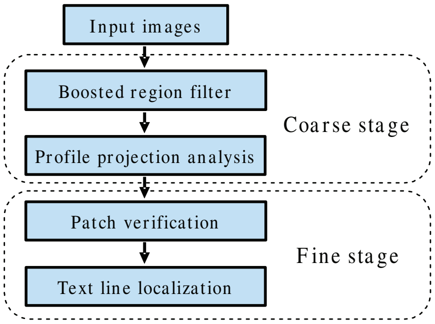


Fig. 1. Framework of the proposed system.

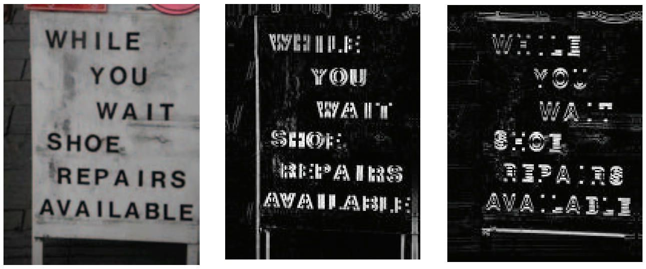
At the fine stage, candidate text lines are partitioned into local patches, which are verified by a polynomial classifier with combined features. The passed text patches over all pyramids are grouped into text lines according to their spatial relationships, and the text lines are further refined and partitioned into words by connected component analysis.

3. COARSE TEXT DETECTION

For detecting texts of various sizes, the original gray-level image is first converted into an image pyramid with scaling factor 2 by nearest interpolation. Coarse text detection and fine verification are performed in each layer of pyramid.

3.1. Boosted Region Filter

Generally, regions between texts and image background have strong contrast of gradient responses density, as shown in Fig. 2. For quickly filtering our non-text regions, we design a boosted classifier for selecting gradient and edge features. The horizontal and vertical gradients of a pixel $f(i, j)$ (f is the gray-level) are calculated by $gx(i, j) = abs[f(i-1, j) - f(i+1, j)]$ and $gy(i, j) = abs[f(i, j-1) - f(i, j+1)]$ and normalized by dividing the gray-level STD in a window. Assuming that text heights in one pyramid layer range from 8 to 30 pixels, a 8×8 window around the pixel is chosen for calculating the STD. The horizontal and vertical gradients are also thresholded by the gray-level STD to give corresponding binary edge maps.



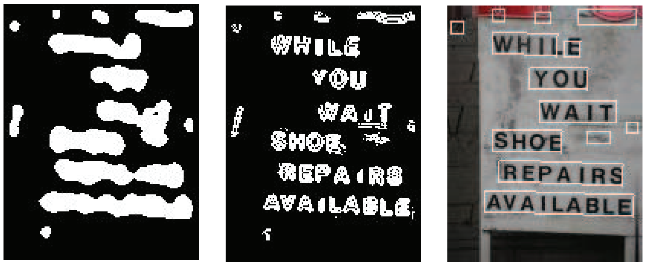
(a) (b) (c)

Fig. 2. Example of text gradient responses. (a) original image. (b) horizontal gradient response. (c) vertical gradient response.

On calculating the abstract and binarized horizontal/vertical gradients (four values) of each pixel, feature values are extracted from 16×16 windows for region filtering. For each window (region), four gradient values are calculated from the whole window and as well from three vertically partitioned strips, thus give 16 features in total. A cascade boosting classifier: Waldboost [4] using decision stumps as weak learners, is adopted. In the training phase, Waldboost recursively selects the best weak learner. In the testing phase, only the regions passing through all previous weak learners are estimated by the next one. This mechanism makes most regions rejected by the first few weak learners. After filtering by the boosted classifier, the text and non-text pixels are labeled as 255 and 0, respectively, followed by a smoothing step with a 8×8 mask to remove small noises which have less than 35% text pixels in the window.

3.2. Projection Analysis

The edge pixels in the candidate text regions are segmented into text lines by projection analysis which recursively cut the pixels horizontally and vertically. First, the edge pixels are partitioned into rows according to the horizontal projection profile. Then, each row is partitioned into columns according to the vertical projection profile. This partitioning is performed recursively until the projection profile has no gap. At last, each row region is a candidate text line. To deal with skewed text lines, we try recursive cut at different orientations (from -10^o to 10^o with step 2^o), and select the best segmentation result of line separability measured by the criterion of [5]. Fig. 3 gives an example of coarse text detection.



(a) (b) (c)

Fig. 3. Example of coarse text detection. (a) candidate text regions. (b) edge map. (c) segmented text lines.

4. FINE TEXT LOCALIZATION

For fine localization, each candidate text line is normalized to 16 pixels height and partitioned horizontally into overlapping patches with 4 pixels step.

4.1. Patch Verification

For patch verification, we use a polynomial classifier with five types of features extracted from a patch: histogram of oriented gradient (HOG), local binary pattern (LBP), discrete cosine transform (DCT), Gabor filter and wavelets. These features have been widely used in text detection and object recognition. We evaluate these features in text patch verification and select the best combination of features.

4.1.1. Feature Extraction

In HOG implementation, the gradients computed by Sobel operator are decomposed into 4 orientations by the parallelogram law. Considering the high dimensionality of the original LBP feature, we adopt an improved variant: center-symmetric

LBP (CS-LBP) [6], which has similar discrimination power with the original LBP but reduces the code space from 255 to 32.

For the DCT feature, we adopt the method of [7] partitioning the frequency coefficients into low, middle and high bands, which has shown better performance than other partition ways. The method of [8] is used for Gabor implementation, where 4-orientation (0, $\pi/4$, $\pi/2$, $3\pi/4$) and 4-frequency (1,$\sqrt{2}$, 2, 2$\sqrt{2}$) Gabor filters are constructed to extract the local space-frequency characteristics of texts. The method of [9] is adopted for wavelet implementation. The Daubechies-4 base wavelet is chosen for horizontal, vertical and diagonal decomposition and the summation of the abstract coefficient values are used as features.

Except the DCT feature, all the features are implemented in a local manner by dividing the patch into small blocks. We empirically divide into 3×3 blocks for good tradeoff between effectiveness and efficiency.

4.1.2. *Polynomial Classifier*

For patch classification, we select the one-class polynomial classifier (PC), which has good generalization and relatively low computational complexity [10]. The PC uses first- and second-order polynomials of subspace features after dimensionality reduction by positive class PCA as well as the projection residual as the inputs of a single-layer network. The vectors of features described in 4.1.1 are reduced to 2/3 of original dimensionality except the DCT feature which has a low dimensionality 3. The weight parameters are estimated on training samples by minimizing the mean square error (MSE).

4.2. Text Line Localization

After patch verification in each layer of pyramid and removing the non-text patches, the left text patches in all layers are grouped into text lines, which are further refined and partitioned into words by connected component analysis.

4.2.1. *Patch Grouping*

The text patches over all pyramid layers are grouped into text lines according to there spatial relationships on the original image. Briefly, any two patches can be grouped together if their vertical overlapping rate exceeds 60% of their heights and they overlap horizontally with each other. This grouping repeats until no more patches can be grouped. Each set of grouped patches form a text region.

To prune incorrect text regions which contain more than one correct text lines or embed in another correct one, we define a rule that for two lines vertically overlapping rate exceeds 30% of their minimal height, the one with lower confidence is removed. Therein, the line confidence is the average of classifier outputs on its grouped patches.

4.2.2. *Connected Component Analysis*

For fine localization, each text line is segmented into three kinds of components using Niblack's local binarization method:

$$b(x) = \begin{cases} 0, & \text{if} \quad gray(x) < \mu_r(x) - k \cdot \sigma_r(x); \\ 255, & \text{if} \quad gray(x) > \mu_r(x) + k \cdot \sigma_r(x); \\ 100, & \text{otherwise,} \end{cases}$$

where the window size r is chosen as the half of the text line height. The components with 0 or 255 values are extracted as candidate text components based on the fact that scene texts could be lighter or darker than the background.

Based on the observations that text components generally have distinct contours, relatively stable vertical positions in the text line and similar gray-levels, we define several rules for non-text component pruning. If any component satisfy one of the following conditions, it will be removed: 1) the gradient magnitude of the component contour is below the text line gray-level STD; 2) the component centroid is above or below of the text middle line over 25% of the line height; 3) the component has gray-level difference greater than 0.9 times the line STD with majority (over half) of the components in the same line.

Finally, the text components in a text line are grouped into words according to the gaps between horizontally adjacent components. The edge linking two adjacent components is cut off if the minimum distance between the bounding boxes of them exceeds 3 times of the average between-component distance. The remaining edges link the components into words. Thus the texts are finally localized. Fig. 4 gives an example of fine text localization.

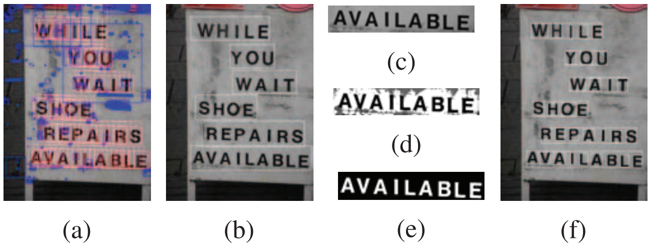


Fig. 4. Example of fine text localization. (a) patch verification (red: text, blue: non-text). (b) patch grouping. (c) text line image. (d) local binarization. (e) text components. (f) text localization.

5. EXPERIMENTS

We have evaluated the text localization performance on the *Trail* set of ICDAR 2003 Competition dataset, which contains a *TrialTrain* set with 258 images for training and a *TrialTest* set with 251 images for testing.

For an image of typical size 640×480, a 5-layer pyramid is built with scaling factor 2. For training the boosted filter, about 100K text region samples and more non-text samples generated during training were used. By setting the false negative rate as 5%, three features shown in Fig. 5 were sequentially selected as the weak learners by Waldboost learning.

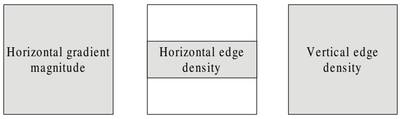
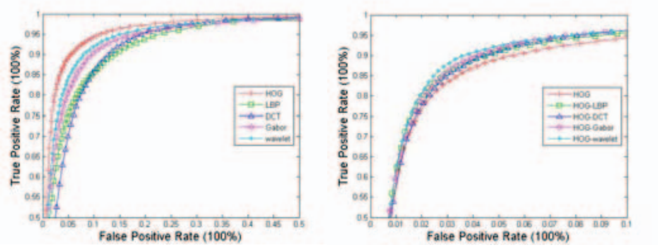


Fig. 5. Region filter features selected by Waldboost.

For patch verification, about 27K text samples and 261K non-text samples were collected from the ground-truths for training the polynomial classifier with different features. As shown in Fig. 6, the HOG feature gives the best performance and it is also computationally fast by using the integral image technique. We further compared the combinations of HOG with each of the other features. As shown in Fig. 6, the combination of HOG and wavelet has better tradeoff between effectiveness and efficiency. This combination was selected for patch verification in the following experiments.



(a) single features (b) combined features

Fig. 6. Patch verification using different features.

We measure text localization performance using the same criterion of [11] in word level. The rates of precision p and recall r are defined as

$$p = C/D, r = C/G, \quad (1)$$

where C is the intersection area between detected text region and ground-truthed region, D and G are the area of detected region and ground-truthed region, respectively. The harmonic mean $f = \frac{2pr}{p+r}$ is used for evaluating the overall performance.

We compare the performance of the proposed method with that of [11], which reported results on the same dataset with us. The results in Table 1 show that our method gives better performance. Meanwhile, our method is fast enough. On a 640 × 480 image, our method takes about 370 ms on a PIV 3.4GHz desktop, including 180ms for the coarse stage and 190ms for the fine stage. Considering the multi-scale nature of the proposed method, the acceleration with the coarse-to-fine strategy is significant.

Table 1. Text word localization results.

	Precision	Recall	*f*
Method [11]	0.56	0.70	0.63
Proposed method	0.66	0.70	0.68

6. CONCLUSION

In this paper, a new coarse-to-fine method is proposed for fast text localization in natural scene images. Both the boosted classifier for region filtering and the polynomial classifier for patch verification are fast and powerful. Proper techniques are used to group text regions into text lines and words. Experimental results on the ICDAR 2003 Competition dataset show that the proposed method provides competitive localization performance with existing methods and the processing speed is high enough for practical application.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (NSFC) under grants no.60775004 and no.60825301.

References

[1] J. Liang, D. Doermann, and H.-P. Li, "Camera-based analysis of text and documents: a survey," *Int. J. Document Analysis and Recognition*, vol. 7, no. 2-3, pp. 84–104, 2005.

[2] K. Jung, K. Kim, and A. Jain, "Text information extraction in images and video: A survey," *Pattern Recogntion*, vol. 37, no. 5, pp. 977–997, 2004.

[3] J. Zhang and R. Kasturi, "Extraction of text objects in video documents: Recent progress," in *Proc. 8th IAPR Workshop on DAS*, 2008, pp. 1–13.

[4] J. Sochman and J. Matas, "Waldboost - learning for time constrained sequential detection," in *Proc. IEEE Conf. CVPR*, 2005, pp. 150–156.

[5] T.-H. Su, T.-W. Zhang, H.-J. Huang, and Y. Zhou, "Skew detection for chinese handwriting by horizontal stroke histogram," in *Proc. 9th ICDAR*, 2007, pp. 899–903.

[6] M. Heikkilä, M. Pietikäinen, and C. Schmid, "Description of interest regions with local binary patterns," *Pattern Recognition*, vol. 42, no. 3, pp. 425–436, 2009.

[7] H. Goto, "Redefining the DCT-based feature for scene text detection: Analysis and comparison of spatial frequency-based features," *Int. J. Document Analysis and Recognition*, vol. 11, no. 1, pp. 1–8, 2008.

[8] C.-L. Liu, M. Koga, and H. Fujisawa, "Gabor feature extraction for character recognition: Comparison with gradient feature," in *Proc. 8th ICDAR*, 2005, pp. 121–125.

[9] J. Gllavata, R. Ewerth, and B. Freisleben, "Text detection in images based on unsupervised classification of high-frequency wavelet coefficients," in *Proc. 17th ICPR*, 2004, pp. 425–428.

[10] L.-L. Huang, A. Shimizu, Y. Hagihara, and H. Kobatake, "Face detection from cluttered images using a polynomial neural network," *Neurocomputing*, vol. 51, pp. 197–211, 2003.

[11] Nobuo Ezaki, Marius Bulacu, and Lambert Schomaker, "Text detection from natural scene images: Towards a system for visually impaired persons," in *Proc. 14th ICPR*, 2004, pp. 683–686.